# Acesso a Dados Baseado em Ontologias com NoSQL

Bárbara Tieko Agena

DISSERTAÇÃO APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DO TÍTULO
DE
MESTRE EM CIÊNCIAS

Programa: Mestrado em Ciência da Computação Orientadora: Prof<sup>a</sup>. Dr<sup>a</sup>. Renata Wassermann

São Paulo, agosto de 2017

### Acesso a Dados Baseado em Ontologias com NoSQL

Esta versão da dissertação contém as correções e alterações sugeridas pela Comissão Julgadora durante a defesa da versão original do trabalho, realizada em 27/11/2017. Uma cópia da versão original está disponível no Instituto de Matemática e Estatística da Universidade de São Paulo.

#### Comissão Julgadora:

- Prof<sup>a</sup>. Dr<sup>a</sup>. Renata Wassermann (orientadora) IME-USP
- Prof<sup>a</sup>. Dr<sup>a</sup>. Kelly Rosa Braghetto IME-USP
- Prof<sup>a</sup>. Dr<sup>a</sup>. Laís do Nascimento Salvador UFBA

# Agradecimentos

À professora Renata Wassermann e à professora Kelly Rosa Braghetto, pela orientação, confiança, paciência e por todo tempo dedicado a este trabalho.

A todos os professores que depositaram em mim sua confiança.

À minha família por me apoiarem em todos os momentos.

Ao meu companheiro Márcio, por estar ao meu lado e me acompanhar nesta jornada.

Aos meus amigos, por estarem próximos e compartilharem os momentos de dificuldade e alegria.

Por último, a Deus, por ter iluminado meu caminho e tornado tudo isso possível.

## Resumo

AGENA, B. T. Acesso a Dados Baseado em Ontologias com NoSQL. 2017. Dissertação (Mestrado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2017.

O acesso a dados baseado em ontologia (OBDA, de Ontology-Based Data Access) propõe facilitar ao usuário acesso a dados sem o conhecimento específico de como eles estão armazenados em suas fontes. Para isso, faz-se uso de uma ontologia como camada conceitual de alto nível, explorando sua capacidade de descrever o domínio e lidar com a incompletude dos dados. Atualmente, os sistemas NoSQL (Not Only SQL) estão se tornando populares, oferecendo recursos que os sistemas de bancos de dados relacionais não suportam. Desta forma, surgiu a necessidade dos sistemas OBDA se moldarem a estes novos tipos de bancos de dados. O objetivo desta pesquisa é propor uma arquitetura nova para sistemas OBDA possibilitando o acesso a dados em bancos de dados relacionais e bancos de dados NoSQL. Para tal, foi proposta a utilização de um mapeamento mais simples responsável pela comunicação entre ontologia e bancos de dados. Foram construídos dois protótipos de sistemas OBDA para sistemas NoSQL e sistemas de bancos de dados relacional para uma validação empírica da arquitetura proposta neste trabalho.

Palavras-chave: Acesso a dados baseado em ontologias, NoSQL, MongoDB, OBDA, ontologia.

## Abstract

AGENA, B. T. Ontology-Based Data Access with NoSQL. 2017. Dissertação (Mestrado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2017.

Ontology-based data access (OBDA) proposes to facilitate user access to data without specific knowledge of how they are stored in their sources. For this, an ontology is used as a high level conceptual layer, exploring its capacity to describe the domain and deal with the incompleteness of the data. Currently, NoSQL (Not Only SQL) systems are becoming popular, offering features that relational database systems do not support. In this way, the need arose for shaping OBDA systems to deal with these new types of databases. The objective of this research is to propose a new architecture for OBDA systems allowing access to data in relational databases and NoSQL databases. For this, we propose the use of a simpler mapping responsible for the communication between ontology and databases. Two OBDA systems prototypes were constructed: one for NoSQL systems and one for relational database systems for an empirical validation.

Keywords: Ontology-Based Data Access, NoSQL, MongoDB, OBDA, ontology.

# Sumário

Li	sta d		
Li	sta d	de Figuras	xi
1	Inti	rodução	1
	1.1	Objetivos	1
	1.2	Contribuições	2
	1.3	Organização do texto	2
2	Ont	tologia	3
	2.1	Construindo uma ontologia	3
		2.1.1 Definindo conceitos	3
		2.1.2 Organizando a hierarquia dos conceitos	4
		2.1.3 Definindo propriedades	5
	2.2	Consulta SPARQL	6
3	Ace	esso a dados baseado em ontologias	7
	3.1	Sistemas OBDA	8
	3.2	Ontop	8
		3.2.1 Camada conceitual	8
		3.2.2 Camada de dados	8
		3.2.3 Mapeamento	9
		3.2.4 Quest	9
4	ОВ	DA com sistemas de banco de dados NoSQL	13
	4.1	MongoDB	13
		4.1.1 Mecanismos de consulta	14
	4.2	Extensão do sistema Ontop para NoSQL	15
	4.3	OBDA com NoSQL utilizando xR2RML	17
5	Fra	mework genérico de acesso a dados baseado em ontologias	19
	5.1	Método	19
		5.1.1 Camada intermediária	20
		5.1.2 Mapeamento e comunicação com o SGBD	21
		5.1.3 Tradutor	21
		5.1.4 Exportação para RDF	22

	5.2	Arquit	tetura genérica para sistemas OBDA	22		
		5.2.1	Categorias de desenvolvimento	23		
6	Ont	oMong	go	25		
	6.1	Estudo	o de caso $\dots$	25		
		6.1.1	Definição do domínio	25		
		6.1.2	Base de dados	26		
		6.1.3	Mapeamento	26		
		6.1.4	Tradutor	28		
		6.1.5	Resultado	32		
7	Ont	oSQL		33		
	7.1	Base of	le dados	33		
	7.2	Const	ruindo consultas SQL	34		
	7.3	Utiliza	ando o Ontop	35		
		7.3.1	Mapeamento Ontop	35		
		7.3.2	Execução da Consulta SPARQL pelo Ontop	37		
8	Conclusões					
	8.1	Consid	derações finais	43		
	8.2	Sugest	ões para pesquisas futuras	44		
A	Arq	juivo C	OWL para ontologia de câncer de pulmão	45		
В	Maj	peame	nto do OntoSQL	49		
$\mathbf{C}$	Maj	peame	nto Ontop	51		
D	Esq	uema (	de banco de dados	55		
			SQL gerada pelo sistema Ontop	59		
F	Est	rutura	do arquivo JSON armazenado no MongoDB	61		
$\mathbf{G}$	Est	rutura	do arquivo XML extraído pelo $ScriptLattes$	63		
Н	Arq	juivo C	${ m DWL}$ da ontologia ${\it basic-lattes}$	67		
Ι	Cla	sses do	o modelo de objetos do OntoMongo	79		
Re	eferê	ncias I	Bibliográficas	81		

# Lista de abreviaturas

ACID Atomicidade, Consistência, Isolamento e Durabilidade

CSV Comma-Separeted Values

DL Lógica de Descrição (Description Logics)

ETL Extração, Transformação, Carregamento (Extract, Transform and Load)

JSON JavaScript Object Notation

NoSQL Not only SQL

NSCLC Câncer de Pulmão de Células Não Pequenas Non-Small Cell Lung Cancer OBDA Acesso a Dados Baseado em Ontologias (Ontology-Based Data Access)

ODM Mapeamento Objeto-Documento (Object Document Mapping)
ORM Mapeamento Objeto-Relacional (Object-Relational Mapping)

OWL Web Ontology Language

R2RML RDB to RDF Mapping Language

RDB Banco de Dados Relacional (Relational Database)

RDF Resource Description Framework

SCLC Câncer de Pulmão de Células Pequenas Small Cell Lung Cancer

SGBD Sistema Gerenciador de Banco de Dados

SPARQL SPARQL Protocol and RDF Query Language

 ${\bf SQL} \qquad \quad {\bf Linguagem \ de \ Consulta \ Estruturada \ } ({\it Structured \ Query \ Language})$ 

URI Uniform Resource Identifier
XML Extensible Markup Language

# Lista de Figuras

2.1	Organização hierárquica das classes	5
2.2	Domínio e imagem das propriedades $temNeoplasia$ e $temEstagio$	5
2.3	Grafo RDF com dois nós (sujeito e objeto) e um predicado conector feito por Klyne e Car	rrol
	(2006)	6
3.1	Estrutura conceitual de OBDA por Calvanese et al. (2013)	7
3.2	Camada de dados: tabela tbl_patient	9
3.3	Árvore construída a partir da consulta SPARQL 3.3	10
4.1	Ontologia que descreve o domínio para a coleção bios	15
5.1	Estrutura conceitual do modelo OBDA	20
5.2	Incompatibilidade de impedância entre a representação de um objeto e os dados	
	armazenados em um banco relacional (Sadalage e Fowler, 2012)	20
5.3	Etapa de processamento da consulta SPARQL no modelo OBDA proposto	21
5.4	Arquitetura genérica para sistemas OBDA	23
6.1	Hierarquia das classes definidas na ontologia	25
6.2	Ontologia basic-lattes por Araujo et al. (2017)	26
6.3	Diagrama das classes de modelo: Researcher e Publication	27
6.4	Equivalência entre classes da ontologia e classes do modelo de objeto	27
6.5	Grafo construído a partir da consulta SPARQL	29
6.6	Nó representando <b>?pesquisador</b>	29
6.7	Exemplo do funcionamento do comando $project$	30
6.8	Nó representando ?publicacao1	30
6.9	Simulação da execução do comando descrito no Arquivo 6.5	30
6.10	Nó representando ?publicacao2	31
7.1	Diagrama do modelo de objeto	33
7.2	Grafo representando a consulta SPARQL	35
7.3	Árvore representando a consulta SPARQL (Arquivo 7.8)	38
7.4	Trecho da Ontologia descrevendo relação entre classes $Pesquisador$ e $Artigo$	38
7.5	Encontrando mapeamentos que casam com a tripla $T1$	39
7.6	Combinação através do comando $JOIN$ entre as consultas SQL descritas para $Arti$ -	
	ao Publicado	40

### ${ m xii}$ LISTA DE FIGURAS

7.7	Combinação através do comando JOIN entre as consultas SQL descritas para Tra-	
	balhoEmEvento	40
7.8	Grafo para consulta SPARQL do Arquivo 7.8	41
7.9	Mapeamento para classe $Pesquisador$	42
7.10	Mapeamento para classe Artigo	42

## Capítulo 1

# Introdução

As áreas de pesquisa envolvendo representação de conhecimento e banco de dados evoluíram de forma que elas convergiram para a integração de suas tecnologias. A representação de conhecimento descreve um domínio utilizando o formalismo lógico tornando-o mais preciso, permitindo que se façam inferências sobre o domínio modelado, normalmente utilizando bases de conhecimento consideradas pequenas. Com o avanço da área, houve a necessidade de se manipular dados do mundo real que normalmente são disponíveis em grandes quantidades e em diversos formatos.

A área de pesquisa de banco de dados se concentra em armazenar e recuperar dados de forma eficiente. No entanto, as informações armazenadas em bancos de dados são hoje muito complexas, tornando-as pouco intuitivas e transparentes.

Destas necessidades surgiu o interesse em se utilizar o poder de expressividade oferecido pela área de representação do conhecimento e a capacidade de armazenamento e recuperação dos bancos de dados, dando origem ao paradigma de acesso a dados baseado em ontologia.

Para se extrair informações de valor, um usuário precisa, além de conhecimento técnico, de conhecimento do domínio para formular consultas relevantes. O paradigma OBDA (Calvanese et al., 2007) (Do inglês Ontology-Based Data Access, Acesso a Dados Baseado em Ontologias) propõe a utilização de ontologia como camada conceitual para uma descrição mais expressiva da base de dados, propiciando uma forma mais simples e eficaz de consulta através do acesso por ontologias.

Muitos cenários atuais apresentam grandes quantidades de dados disponíveis em diversos formatos. Em alguns destes cenários o uso de bancos de dados relacionais não é a opção mais adequada para o armazenamento e recuperação dos dados. Desta forma, os sistemas NoSQL (Not Only SQL) surgiram para satisfazer as necessidades destas aplicações modernas.

Com o surgimento e a popularização dos sistemas NoSQL, surgiu a questão: Dado um sistema de banco de dados NoSQL, é possível utilizá-lo com sistemas OBDA?

## 1.1 Objetivos

O objetivo principal deste trabalho é propor um mapeamento OBDA que possa ser aplicado tanto para sistemas de bancos de dados NoSQL quanto para sistemas de bancos de dados relacionais. A proposta nasce da importância das aplicações OBDA suportarem o NoSQL para que seja possível aplicá-lo a domínios dotados de grandes volumes de dados. Temos ainda como objetivo facilitar o mapeamento e o processo de conversão de consultas SPARQL em consultas para diferentes tipos de bancos de dados, pois é necessário que esse processo seja flexível, dada a grande quantidade e variedade desses sistemas. Para tal, propomos uma camada conceitual intermediária entre a ontologia e o banco de dados que provê uma estrutura mais próxima da ontologia para os dados armazenados no banco de dados. Além disso, descrevemos um método de tradução das consultas para os SGBDs (Sistemas Gerenciadores de Bancos de Dados) de destino. Esta ideia foi aplicada inicialmente a um sistema que utiliza NoSQL orientado a documentos (MongoDB) e em seguida a um SGBD relacional para avaliar a viabilidade do modelo.

2 INTRODUÇÃO 1.3

### 1.2 Contribuições

A principal contribuição deste trabalho é a construção de uma arquitetura para sistemas OBDA capaz de utilizar bancos de dados relacionais e bancos de dados NoSQL. Parte deste trabalho foi realizada em conjunto com Thiago Araújo e publicada em Araujo et al. (2017). A generalização da arquitetura resultou em uma nova abordagem para a construção de um mapeamento mais simples em sistemas OBDA e para a conversão de consultas SPARQL em consultas para SGBDs.

### 1.3 Organização do texto

O capítulo 2 apresenta a descrição de conceitos básicos das ontologias. O capítulo 3 trata do paradigma de Acesso a Dados Baseado em Ontologias (OBDA), descrevendo um estudo que utiliza esse paradigma com sistemas de bancos de dados relacionais. No capítulo 4 discutimos os estudos recentes que tratam de OBDA com NoSQL. O capítulo 5 apresenta o framework genérico para acesso a dados baseado em ontologias. Nos capítulos 6 e 7 são apresentados os protótipos que utilizam o framework genérico para os sistemas de bancos de dados MongoDB e PostgreSQL, fazendo uma breve comparação com soluções mais recentes na literatura. Finalmente, o capítulo 8 resume as contribuições desta dissertação e sugere trabalhos futuros.

## Capítulo 2

# Ontologia

Como definido por Gruber (2009), na computação, uma ontologia modela um domínio do conhecimento. Para tal, são definidos classes (conceitos), atributos (propriedades), restrições das classes e relações entre classes. As ontologias podem ser vistas como uma abstração dos modelos de dados no contexto de bancos de dados. Porém, as ontologias permitem abstrair o conhecimento a respeito de indivíduos, seus atributos, e relações possibilitando modelar o domínio independentemente da estrutura dos dados. Desta forma, pode-se dizer que as ontologias descrevem um domínio em nível "semântico" e os esquemas de bancos de dados descrevem o domínio em nível "físico". Possuindo uma descrição no nível semântico, as ontologias possibilitam a comunicação entre fontes de dados distintas.

As ontologias são utilizadas para diversos propósitos, alguns deles são:

- Definir um vocabulário comum sobre o domínio de interesse para compartilhar conhecimento entre pesquisadores da mesma area, ou mesmo, entre humanos e máquinas;
- Possibilitar o reuso do domínio do conhecimento. Uma vez definida uma ontologia para um determinado domínio, essa ontologia pode ser compartilhada para ser reutilizada por outras aplicações do mesmo domínio;
- Utilizar as especificações explícitas sobre o domínio para proporcionar às pessoas um fácil
  entendimento, possibilitando validar o conhecimento descrito buscando por contradições e
  inconsistências na representação do conhecimento.

### 2.1 Construindo uma ontologia

Nesta seção, usaremos a ontologia para o domínio de câncer de pulmão desenvolvida por Calvanese et al. (2017) para ilustrar o processo de construção de uma ontologia.

#### 2.1.1 Definindo conceitos

O primeiro passo para se construir uma ontologia é definir quais conceitos a ontologia vai descrever. Esses conceitos fazem parte do vocabulário utilizado pelos especialistas do domínio. Neste cenário, queremos o vocabulário utilizado pelos especialistas em câncer de pulmão. Na Tabela 2.1 são definidas as classes da ontologia.

Existe uma variedade de linguagens para se descrever uma ontologia. Neste exemplo, a linguagem utilizada é a linguagem Web Ontology Language<sup>1</sup> (OWL) que permite a descrição de propriedades e classes, relações entre classes (por exemplo disjunção), cardinalidade (por exemplo, "exatamente um"), igualdade, características das propriedades (por exemplo, simetria) e classes enumeradas.

A descrição da classe Paciente em OWL pode ser verificada no Arquivo 2.1.

<sup>&</sup>lt;sup>1</sup>https://www.w3.org/OWL/

4 ONTOLOGIA 2.1

Paciente
Pessoa
NeoplasiaBenigna
Cancer
CancerDePulmao
NeoplasiaMaligna
NSCLC
Neoplasia
SCLC
Estagio

Tabela 2.1: Classes para ontologia sobre câncer de pulmão.

```
Arquivo 2.1: Descrição da classe Paciente em OWL

<Declaration>

<Class IRI="#Paciente"/>

</Declaration>
```

#### 2.1.2 Organizando a hierarquia dos conceitos

As classes definidas na Tabela 2.1 são organizadas em hierarquia. Na Figura 2.1, é possível verificar as superclasses e subclasses. A neoplasia é definida como o aumento celular não controlado pelo organismo, possuindo duas categorias, a neoplasia benigna e maligna. Como podemos acompanhar pela Figura 2.1, a classe Neoplasia é superclasse das classes NeoplasiaBenigna e NeoplasiaMaligna. A neoplasia maligna é popularmente conhecida como câncer, desta forma, a classe Cancer é subclasse da classe Neoplasia. E como a ontologia tem foco somente no câncer de pulmão, temos que a classe CancerDePulmao é subclasse de Cancer. E por fim, o câncer de pulmão possui duas subclasses NSCLC (Do inglês Non-Small Cell Lung Cancer, Câncer de Pulmão de Células Não Pequenas) e SCLC (Do inglês Small Cell Lung Cancer, Câncer de Pulmão de Células Pequenas).

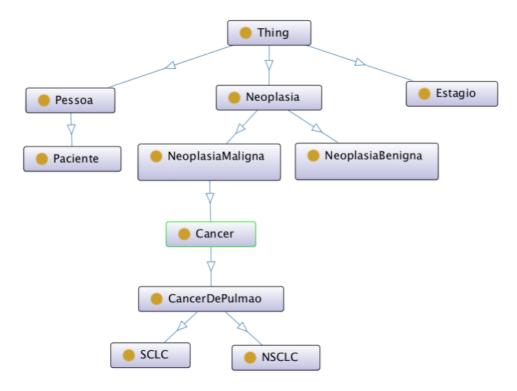


Figura 2.1: Organização hierárquica das classes.

Podemos ver no Arquivo 2.2 como é representado na linguagem OWL a relação que a classe Paciente é subclasse de Pessoa.

Arquivo 2.2: Descrição de hierarquia entre as classes Paciente e Pessoa em OWL

#### 2.1.3 Definindo propriedades

Com as classes e hierarquia definidas a próxima etapa consiste em definir as propriedades das classes. A ontologia sobre câncer de pulmão possui três propriedades chamadas de: temNome, tem-Neoplasia e temEstagio. A propriedade temNome tem como domínio (domain) a classe Pessoa e a imagem (range) uma string simples. Como ilustrada na Figura 2.2, as propriedades temNeoplasia e temEstagio possuem como imagem instâncias de outras classes. Desta forma, a imagem para estas propriedades são vinculadas a outras classes.

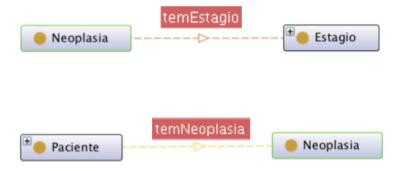


Figura 2.2: Domínio e imagem das propriedades temNeoplasia e temEstagio.

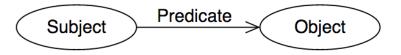
As propriedades são herdadas pelas subclasses, então a classe Paciente herda a propriedade

6 ONTOLOGIA 2.2

temNome de sua superclasse Pessoa.

E por fim, as propriedades são preenchidas criando as instâncias das classes para formar a base de conhecimento da ontologia. Esta base de conhecimento pode ser representada através de um grafo RDF. O Resource Description Framework<sup>2</sup> é uma estrutura baseada em XML criada para representar as informações na Web.

A estrutura base da sintaxe do RDF é composta por um conjunto de triplas, cada uma consistindo de um sujeito, um predicado e um objeto. Um conjunto de triplas é chamado de um "grafo RDF". Uma tripla RDF pode ser visualizada como um conjunto de dois nós e um arco dirigido, em que cada tripla é representada como um link do tipo nó-arco-nó como pode ser visto na Figura 2.3. Uma tripla RDF estabelece uma relação entre os sujeito e objeto definidos pelos nós.



**Figura 2.3:** Grafo RDF com dois nós (sujeito e objeto) e um predicado conector feito por Klyne e Carroll (2006).

### 2.2 Consulta SPARQL

A linguagem de consulta SPARQL Protocol and RDF Query  $Language^3$  (SPARQL) é utilizada para se extrair informações de grafos RDF.

Dada uma fonte de dados D (que pode ser a união de várias fontes), uma consulta consiste em um padrão que é comparado com D para que os valores obtidos a partir desta correspondência sejam processados para dar uma resposta. Podemos verificar no Arquivo 2.3 um exemplo de consulta SPARQL.

Arquivo 2.3: Exemplo de uma consulta SPARQL

Dentro do comando WHERE da consulta é definido um padrão de grafo (Graph Pattern ou Basic Graph Pattern) utilizado para percorrer dados de um grafo RDF e retornar os dados que correspondem ao padrão de grafo. Considere o grafo RDF do Arquivo 2.4. Na consulta SPARQL, ?pessoa, ?email e ?sobrenome são variáveis (por começarem com o símbolo ?). A consulta SPARQL percorre os dados do grafo (Arquivo 2.4) verificando a correspondência dos dados. As triplas com os predicados sobrenome e email coincidem com o padrão da consulta e as variáveis são vinculadas com os dados do grafo RDF.

Arquivo 2.4: Grafo RDF de exemplo

```
1 : barbara : sobrenome "Agena"
2 : barbara : email "bagena@ime.usp.br"
```

O comando *SELECT* define as informações a serem apresentadas. Desta forma, os dados vinculados às variáveis *?email* e *?sobrenome* são recuperadas pela consulta SPARQL do Arquivo 2.3.

<sup>&</sup>lt;sup>2</sup>https://www.w3.org/RDF/

<sup>&</sup>lt;sup>3</sup>https://www.w3.org/TR/rdf-sparql-query/

# Capítulo 3

# Acesso a dados baseado em ontologias

O acesso a dados baseado em ontologias (OBDA), como descrito por Calvanese et al. (2007), tem como objetivo facilitar o acesso do usuário aos dados de uma aplicação utilizando uma camada conceitual. A ontologia é a melhor candidata para representar a camada conceitual, disponibilizando ao usuário o vocabulário do domínio que ele está familiarizado e abstraindo todos os detalhes sobre o armazenamento dos dados. Além disso, as ontologias dispõem de mecanismos de inferência para encontrar possíveis inconsistências nos dados ou para recuperar informações implícitas na base de dados.

Como descrito por Calvanese *et al.* (2013), a estrutura OBDA é composta pela ontologia que descreve o domínio, a base de dados que armazena os dados da aplicação e o mapeamento responsável pela comunicação entre a camada conceitual e a camada de dados, como pode ser visto na Figura 3.1.

Uma aplicação que faça uso de OBDA recebe como entrada uma consulta SPARQL, que é convertida em uma consulta SQL capaz de recuperar os dados necessários para responder a consulta SPARQL. Esses dados são convertidos em instâncias da ontologia para que a consulta SPARQL possa ser executada sobre eles.

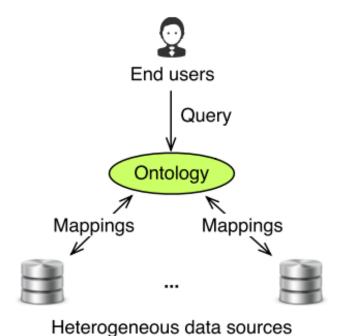


Figura 3.1: Estrutura conceitual de OBDA por Calvanese et al. (2013).

Sem o conceito de OBDA, as aplicações semânticas (aplicações utilizando ontologias) passam por um processo conhecido como ETL (*Extract, Transform and Load*). Esse processo consiste em extrair os dados da base de dados, transformá-los de forma adequada para futuras consultas e

análises e armazenar esses dados em um repositório de triplas.

Esse processo apresenta algumas desvantagens, tais como o consumo extra de memória pela necessidade de se duplicar os dados da base e a desatualização dos dados sempre que ocorrer uma alteração na base original. Então, na maior parte das vezes é vantajoso ter acesso direto aos dados na base original, podendo evitar tais problemas e também podendo explorar as vantagens que os gerenciadores de banco de dados (SGBD) disponibilizam.

#### 3.1 Sistemas OBDA

Alguns sistemas OBDA já foram criados e apresentados na literatura. Entre eles o D2RQ (Bizer e Seaborne, 2004) que é um dos primeiros sistemas OBDA desenvolvidos. Possuindo sua própria linguagem de mapeamento e utilizando a biblioteca Jena ontology API¹ para construção de aplicações semânticas. Rodriguez-Muro et al. (2013b) faz uma análise do sistema D2RQ e constata que o sistema não é capaz de executar algumas consultas. Isso se dá pelo método de tradução de SPARQL-para-SQL, onde a conversão de uma consulta SPARQL em alguns casos resulta em várias consultas SQL sobrecarregando o SGBD.

O Ultrawrap (Sequeda et al., 2014) utiliza consultas pré-definidas (views) para construir o mapeamento entre a ontologia e o banco de dados. Para otimizar o processo de acessar os dados este sistema combina o processo de reescrever a consulta SPARQL-para-SQL com a materialização parcial dos dados.

### 3.2 Ontop

Dentre os sistemas OBDA desenvolvidos, se destaca o sistema Ontop por ser o mais recente e o que aprimorou questões de desempenho que os outros sistemas não cobrem. Ontop² é um sistema OBDA de código aberto desenvolvido por pesquisadores da Free University of Bozen-Bolzano que permite a consulta sobre fontes de dados relacionais através de uma ontologia que faça a representação do domínio de interesse. Ontop se destacou por sua capacidade de responder a consultas sem a necessidade de uma conversão prévia do banco de dados. Um exemplo dado por Calvanese et al. (2017) que explica em mais detalhes o funcionamento do sistema Ontop é apresentado nas seções a seguir. Neste exemplo, o domínio da aplicação é sobre pacientes com câncer de pulmão.

#### 3.2.1 Camada conceitual

Ontop utiliza uma ontologia como camada conceitual de forma a definir um vocabulário, modelar o domínio de aplicação e esconder a estrutura das fontes de dados. Ilustramos o funcionamento do Ontop com um exemplo apresentado por Calvanese et al. (2017). A ontologia utilizada para este exemplo é a ontologia referente ao domínio de câncer de pulmão descrita no capítulo 2. O arquivo OWL completo da ontologia pode ser verificado no Anexo A.

#### 3.2.2 Camada de dados

Neste exemplo, a camada de dados possui uma tabela em um banco relacional chamada  $tbl\_patient$  ilustrada na Figura 3.2. Esta tabela armazena dados como identificador do registro (pid), nome do paciente (name), estágio do câncer (stage) e tipo do câncer (type) com valores true para SCLC e false para NSCLC.

<sup>&</sup>lt;sup>1</sup>https://jena.apache.org/index.html

<sup>&</sup>lt;sup>2</sup>http://ontop.inf.unibz.it/

3.2 Ontop 9

<u>pid</u>	name	type	stage
1	'Mary'	false	4
2	'John'	true	7

Figura 3.2: Camada de dados: tabela tbl patient.

#### 3.2.3 Mapeamento

O mapeamento é feito na linguagem de mapeamento R2RML (RDB - relational database - to RDF Mapping Language)<sup>3</sup> que foi definida pela comunidade internacional dos padrões web conhecida como World Wide Web Consortium (W3C)<sup>4</sup>. Esta linguagem foi criada para descrever mapeamentos de uma base de dados relacional para dados RDF.

Como explicado por Das et al. (2012), para mapear uma base de dados, cada tabela desta base dever estar mapeada para uma regra que gera triplas RDF. No Arquivo 3.1 podemos visualizar o mapeamento (versão simplificada) responsável por popular a ontologia da Figura 2.1 com os dados da tabela da Figura 3.2.

**Arquivo 3.1:** Mapeamento criado para conectar a ontologia de câncer de pulmão (Anexo A) e base de dados (Figura 3.2)

```
source :http://exemplo/hospital/{pid} rdf:type :Paciente .
2
   target SELECT pid FROM tbl_patient
3
4
   source : http://exemplo/hospital/neoplasia/{pid} rdf:type : NSCLC .
   target SELECT pid, stage FROM tbl_patient WHERE type = false
5
6
7
   source : http://exemplo/hospital/neoplasia/{pid} rdf:type : SCLC .
8
   target SELECT pid, stage FROM tbl patient WHERE type = true
   source : http://exemplo/hospital/{pid} :temNome {name} .
10
   target SELECT pid, name FROM tbl patient
11
12
   source : http://exemplo/hospital/{pid} :temNeoplasia : http://exemplo/
13
       \hookrightarrow hospital/neoplasm/{pid}.
14
   target SELECT pid FROM tbl_patient
15
   source : http://exemplo/hospital/neoplasia/{pid} : temEstagio : estagio -IIIa .
16
   target SELECT pid FROM tbl patient WHERE stage = 4
```

Se esta não fosse uma aplicação OBDA, seria necessário converter a base de dados para triplas utilizando o mapeamento. Neste caso, é possível comparar o mapeamento como um método para migração dos dados da tabela  $tbl\_patient$  para uma tabela de triplas (triplestore), onde o campo source do mapeamento representa a tripla RDF que será gerada e o campo target representa a consulta SQL responsável por recuperar os dados do banco de dados para gerar a tripla RDF. Porém, no sistema OBDA são geradas apenas as triplas RDF necessárias para a responder a consulta SPARQL.

#### 3.2.4 Quest

Rodriguez-Muro e Calvanese (2012) desenvolveram Quest que é o núcleo do sistema Ontop responsável por converter as consultas SPARQL para SQL e gerar um grafo RDF em memória. Existem algumas ferramentas que fazem esse processo (D2RQ, Triplify, Spyder, etc), porém, essas ferramentas, como mencionado por Rodriguez-Muro et al. (2012), perdem eficiência na geração das consultas SQL, seja gerando consultas excessivamente complexas, ou enviando múltiplas consultas ao sistema gerenciador de banco de dados (SGBD). Enquanto isso, Quest otimiza a geração da consulta SQL produzindo uma única consulta otimizada para ser executada de forma eficiente pelo SGBD.

 $<sup>^3 \</sup>mathrm{http://www.w3.org/TR/r2rml/}$ 

<sup>&</sup>lt;sup>4</sup>https://www.w3.org/

Rodriguez-Muro e Calvanese (2011) explicam que o sistema Quest gera um segundo mapeamento denominado T-Mapping, responsável por complementar o mapeamento original. Como no exemplo, se observarmos o mapeamento do Arquivo 3.1 podemos verificar que não existe mapeamento definido para o conceito :Neoplasm, mas sabemos que os conceitos :NSCLC e :SCLC são subclasses de :Neoplasm. Desta forma, o mapeamento para :Neoplasm é gerado conforme podemos ver no Arquivo 3.2.

#### Arquivo 3.2: T-mapping gerado para :Neoplasm

```
1 source :http://exemplo/hospital/neoplasia/{pid} rdf:type :NSCLC .
2 taget SELECT pid, stage FROM tbl_patient WHERE type = false
3
4 source :http://exemplo/hospital/neoplasia/{pid} rdf:type :SCLC .
5 taget SELECT pid, stage FROM tbl_patient WHERE type = true
6
7 source :http://exemplo/hospital/neoplasia/{pid} rdf:type :Neoplasia .
8 taget SELECT pid, stage FROM tbl_patient WHERE type = false OR type = true
```

Com o mapeamento finalizado, o sistema Quest é capaz de transformar a consulta SPARQL em uma consulta SQL. Otimizações com técnicas de Semantic Query Optimization são feitas sobre a consulta SQL para se obter consultas mais eficientes, como podemos observar através do exemplo a seguir. O Arquivo 3.3 apresenta a consulta SPARQL para recuperar todos os tumores com estágio 3.

Arquivo 3.3: Consulta SPARQL para recuperar todos os tumores com estágio 3

```
1 SELECT ?tumor WHERE {
2          ?tumor rdf:type :Neoplasm ;
3          ?tumor :hasStage :stage-IIIa . }
```

A partir desta consulta SPARQL é gerada uma árvore, onde cada nó da árvore é uma tripla do comando **WHERE** da consulta do Arquivo 3.3. Neste exemplo existem duas triplas, essas triplas se encontram nas linhas 2 e 3. Podemos ver a árvore resultante na Figura 3.3.



Figura 3.3: Árvore construída a partir da consulta SPARQL 3.3.

Pelo mapeamento, cada nó da árvore na Figura 3.3 é transformado para a consulta SQL correspondente como podemos ver no Arquivo 3.4.

```
Arquivo 3.4: Convertendo nós T1 eT2 para consulta SQL
```

```
T1 = ?x rdf:type :Neoplasm <- SELECT pid, stage FROM tbl_patient

→ WHERE type = false OR type = true

T2 = ?x :hasStage :stage-IIIa <- SELECT pid FROM tbl_patient WHERE

→ stage = 4
```

A partir das sub-consultas geradas em 3.4, a consulta no Arquivo 3.5 é gerada.

Arquivo 3.5: Consulta SQL gerada a partir de T1 eT2

Rodriguez-Muro et al. (2013a) afirma que esta consulta pode ser executada pelo Sistema Gerenciador de Banco de Dados (SGBD), porém, dependendo da consulta SPARQL de entrada, a consulta

3.2 ONTOP 11

SQL correspondente gerada desta forma pode conter muitas sub-consultas e joins (self-joins redundantes) desnecessários, tornando-a bastante ineficiente. São utilizadas técnicas de Semantic Query Optimization com o intuito de gerar consultas SQL mais eficientes. Podemos ver no Arquivo 3.6 a consulta 3.5 após sofrer otimização.

```
Arquivo 3.6: Consulta 3.5 otimizada
```

```
1 SELECT concat(":db1/neoplasm/", pid) AS x FROM tbl_patient WHERE (type = \hookrightarrow false OR type = true) AND stage = 4
```

Essa consulta otimizada é executada pelo SGBD e os dados recuperados são convertidos para um grafo RDF onde a consulta SPARQL pode ser executada.

## Capítulo 4

# OBDA com sistemas de banco de dados NoSQL

Com o avanço da *Internet*, da Computação em Nuvem e das aplicações *Web*, surgiram novas necessidades para o armazenamento de dados. Como descrito por Han *et al.* (2011), algumas aplicações com alta concorrência necessitam de capacidade de leitura e escrita com baixa latência, para que sejam capazes de lidar com uma alta quantidade de requisições concorrentes sem prejudicar o funcionamento do sistema para o usuário final. Outra necessidade comum nas aplicações é a escalabilidade e disponibilidade. Com o crescimento dos dados e das requisições, os sistemas de bancos de dados precisam ser capazes de suportar fácil expansão e atualização, garantindo o bom funcionamento do sistema. Os sistemas de bancos de dados relacionais dominam o mercado, porém, quando deparados com problemas como os listados acima, apresentam limitações. Algumas aplicações priorizam os itens acima, tendo como consequência a flexibilização das propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade) na execução das transações.

Com o surgimento dos sistemas NoSQL, estudos recentes começam a estender a utilização de sistemas OBDA em conjunto com NoSQL. Estes estudos utilizam como experimento para suas soluções o sistema de banco de dados MongoDB, que será melhor explicado na seção 4.1. Nas seções 4.2 e 4.3 serão descritos os estudos realizados por Botoeva et al. (2016) e Michel et al. (2016).

## 4.1 MongoDB

MongoDB é um dos bancos de dados orientados a documentos mais populares atualmente. A família de banco de dados orientados a documentos faz o armazenamento dos dados em documentos que podem ser em formato XML, JSON, entre outros. Uma coleção em bancos de dados orientados a documentos é equivalente a uma tabela em um banco de dados relacional, assim como cada documento equivale a um registro. No Arquivo 4.1 é possível ver um exemplo de um documento armazenado no MongoDB. Este documento possui informações de nome (name), data de nascimento (birth), contribuições (contribs) e premiações (awards). Alguns campos do documento podem conter estruturas complexas, como listas e outros objetos. Esta característica é conhecida como orientação a agregados.

Arquivo 4.1: Exemplo de um documento armazenado no Mongo, retirado da coleção bios<sup>1</sup>

```
1
        " id" : 5,
2
        "name" : {
3
            "first": "Ole-Johan",
4
            "last" : "Dahl"
5
6
        "birth": ISODate("1931-10-12T04:00:00Z"),
7
8
        "death": ISODate("2002-06-29T04:00:00Z"),
        "contribs" : [
9
10
            "OOP",
```

```
11
             "Simula"
        ],
"awards" : [
12
13
14
             {
                 "award" : "Rosing Prize",
15
                 "year": 1999,
16
                 "by" : "Norwegian Data Association"
17
18
19
                 "award" : "Turing Award",
20
21
                 "year" : 2001,
22
                 "by" : "ACM"
23
24
                 "award" : "IEEE John von Neumann Medal",
25
                  "year" : 2001,
26
                 "by" : "IEEE"
27
28
29
30
```

Os documentos armazenados em uma coleção são, de forma usual, semanticamente relacionados, mas não precisam ser estruturalmente iguais, ou seja, podem ter atributos diferentes. Podemos verificar isto considerando os documentos 4.1 e 4.2. Eles podem pertencer à mesma coleção.

**Arquivo 4.2:** Exemplo modificado de um documento armazenado no MongoDB, retirado da coleção bios (estrutura alterada para exemplificar diferença de estrutra)

```
1
2
        " id" : 1,
        "name": {
3
            "first" : "John"
4
            "last" : "Backus"
5
6
        "birth": ISODate("1924-12-03T05:00:00Z"),
7
8
        "death": ISODate("2007-03-17T04:00:00Z"),
9
        "adresses" :
10
            {
                     "state" : "PA",
11
                     "city": "Philadelphia"
12
            }
13
14
   }
15
```

#### 4.1.1 Mecanismos de consulta

Os sistemas de bancos de dados de documentos oferecem diferentes recursos de consultas. Cada produto possui propriedades que podem não ser encontradas em outros sistemas de bancos de dados orientados a documentos.

É possível consultar documentos no MongoDB usando comandos *JavaScript* através do terminal utilizando *mongo Shell Methods*. Um exemplo para este tipo de recurso são consultas com o comando "db.collection.find(query, projection)", que recebem como parâmetro filtros para a consulta (query) e especificam quais os campos de retorno da consulta (projection).

Outra forma disponível é o Aggregation Pipeline, que é uma das possíveis formas de agregação oferecidas pelo framework de agregação<sup>2</sup>. As consultas neste padrão são realizadas através de uma sequência de operações que são executadas em ordem, transformando o conjunto de documentos a cada comando executado. Alguns dos comandos disponíveis são:

• macth: Filtragem dos documentos que casam com o campo determinado na operação;

<sup>&</sup>lt;sup>1</sup>Coleção de exemplo disponibilizada em: https://docs.mongodb.com/manual/reference/bios-example-collection/

 $<sup>^2</sup> Aggregation\ Framework: https://docs.mongodb.com/manual/aggregation/$ 

- unwind: Fragmentação do campo lista de um documento e produção de uma saída com N documentos onde N é o tamanho da lista. Assim, cada documento contém um elemento do vetor;
- project: Configuração dos documentos, podendo alterar o valor de algum campo e adicionar ou remover campos;
- group: Agrupar documentos a partir de uma expressão.

### 4.2 Extensão do sistema Ontop para NoSQL

Botoeva et al. (2016) propõe um framework OBDA para sistemas de banco de dados NoSQL que propõe estender o sistema Ontop para possibilitar a utilização de sistemas bancos de dados NoSQL. Como experimento, a solução foi testada sobre o sistema de banco de dados MongoDB. Botoeva et al. (2016) utiliza a base de dados bios disponibilizada pelos criadores do MongoDB. Os documentos da coleção bios possuem informações de nomes, premiações, contribuições e data de nascimento de pessoas. No trabalho de Botoeva et al. (2016) esta base foi reduzida a um documento contendo as informações sobre "Kristen Nygaard" como podemos ver no Arquivo 4.3.

Arquivo 4.3: Documento da coleção bios

```
{ "\_id ": 4, }
1
    "awards": [ {"award": "Rosing Prize", "year": 1999, "by": "Norwegian Data
2

→ Association " },
                 {"award": "Turing Award", "year": 2001, "by": "ACM" },
3
                 {"award": "IEEE John von Neumann Medal", "year": 2001, "by": "
4
                     \hookrightarrow IEEE"} ],
5
     "birth": "1926-08-27".
     "contribs": ["OOP", "Simula"],
6
     "death": "2002-08-10",
     "name": {"first": "Kristen", "last": "Nygaard"}
  }
9
```

Para acessar os dados na coleção do Arquivo 4.3 existe uma ontologia na Figura 4.1 que a descreve.



Figura 4.1: Ontologia que descreve o domínio para a coleção bios.

O mapeamento é construído a partir de funções que representam o resultado de uma consulta nativa do sistema de banco de dados. Neste exemplo, as funções definidas são:

- $f_f = \{name.first\}$ : função que recupera todos os valores de first armazenados dentro de name dos documentos da coleção;
- $f_l = \{name.last\}$ : função que recupera todos os valores de *last* armazenados dentro de *name* dos documentos da coleção;
- $f_x =: /\{\_id\}$ : função que recupera o id do documento; a forma :/id denota a URI(Uniform Resource Identifier) utilizada na ontologia para representar o objeto pesquisador;
- $f_a =: /\{ id\}/Award/\{awards.\#\}$ : função que recupera os id's de premiações (award);

- $f_y = \{awards. \#.year\}$ : função que recupera os valores de ano de premiação (year);
- $f_n = \{awards.\#.award\}$ : função que recupera os valores para nome da premiação (award-Name);

O mapeamento é definido da seguinte forma:

```
    {?X → f<sub>x</sub>}(?X a : Scientist)
    {?X → f<sub>x</sub>, ?F → f<sub>f</sub>}(?X : firstName ?F)
    {?X → f<sub>x</sub>, ?L → f<sub>l</sub>}(?X : lastName ?L)
    {?X → f<sub>x</sub>, ?A → f<sub>a</sub>}(?X : gotAward ?A)
    {?A → f<sub>a</sub>, ?Y → f<sub>y</sub>}(?A : awardedInYear ?Y)
    {?A → f<sub>a</sub>, ?N → f<sub>n</sub>}(?A : awardName ?N)
```

Este mapeamento descreve o método de acesso aos dados a partir das triplas de uma consulta SPARQL. Caso alguma tripla da cláusula *where* na consulta case com a tripla definida no mapeamento, a consulta mapeada é executada.

Desta forma, a consulta SPARQL descrita no Arquivo 4.4 é utilizada para consultar os pesquisadores que receberam duas premiações no mesmo ano.

**Arquivo 4.4:** Consulta SPARQL para recuperar os pesquisadores que receberam duas premiações no mesmo ano.

Na consulta acima, a primeira tripla da cláusula where é ?scientist : firstName? firstName. Essa tripla casa com a tripla (?X : firstName ?F) do mapeamento. Com isso, o sistema sabe que deve executar as funções  $\{?X \mapsto f_x, ?F \mapsto f_f\}$ , ou seja, as funções que retornam o id e o valor de first armazenado dentro de name, utilizados para gerar a tripla (: /4 : firstName "Kristen").

Utilizando o mapeamento, o grafo RDF resultante da consulta no Arquivo 4.4 é gerado e pode ser visto no Arquivo 4.5.

Arquivo 4.5: Triplas RDF gerada

```
1  (:/4 a : Scientist)
2  (:/4/Award/0 : awardedInYear 1999)
3  (:/4 : firstName "Kristen")
4  (:/4/Award/1 : awardedInYear 2001)
5  (:/4 : lastName "Nygaard")
6  (:/4/Award/2 : awardedInYear 2001)
7  (:/4 : gotAward :/4/Award/0)
8  (:/4/Award/0 : awardName "Rosing Prize")
9  (:/4 : gotAward :/4/Award/1)
10  (:/4/Award/1 : awardName "Turing Award")
11  (:/4 : gotAward :/4/Award/2)
12  (:/4/Award/2 : awardName "von Neumann Medal")
```

A solução apresentada por Botoeva et al. (2016) propõe a partir da consulta SPARQL gerar várias sub-consultas na linguagem de consulta correspondente ao sistema de banco de dados NoSQL

utilizado. E com os dados recuperados dessas sub-consultas é possível gerar o grafo RDF. Podemos verificar que os filtros são executados pela consulta SPARQL. Mesmo com o experimento apresentado, Botoeva et al. (2016) afirma que não é possível saber se o modelo apresentado será capaz de suportar qualquer tipo de banco de dados.

### 4.3 OBDA com NoSQL utilizando xR2RML

Michel et al. (2016) propõe uma solução para sistema OBDA capaz de utilizar um banco de dados MongoDB arbitrário utilizando para o mapeamento a linguagem xR2RML que é a extensão da linguagem de mapeamento R2RML.

Como evolução da linguagem R2RML que suportava somente consultas SQL, xR2RML foi desenvolvida para dar suporte a outros tipos de sistemas de bancos de dados. Como mencionado por Michel et al. (2015), até o momento, xR2RML suporta o acesso a sistemas de bancos de dados NoSQL em formato CSV (Comma-Separeted Values), JSON<sup>3</sup> (JavaScript Object Notation) e XML<sup>4</sup> (Extensible Markup Language).

Em 4.7 é apresentado um exemplo de mapeamento para uma base de dados em MongoDB produzido por Michel et~al.~(2016). A base de dados no MongoDB consiste de uma coleção chamada "project" (Arquivo 4.6) contendo dois arquivos. Cada projeto ("project") possui um nome (name), código (code) e uma lista de times (teams). Cada time é representado por uma lista de pessoas, cuja última pessoa da lista é o líder do time (<#TmLeader>).

#### Arquivo 4.6: Coleção MongoDB "projects"

O mapeamento xR2RML para a consulta do <#TmLeader> nos dados da coleção do Arquivo 4.6 está representado no Arquivo 4.7.

Arquivo 4.7: Mapeamento xR2RML para a coleção MongoDB "projects"

Na linha 2 do arquivo de mapeamento 4.7 a propriedade xrr:logicalSource define a fonte de onde os dados serão recuperados (armazenamento lógico dos dados). Neste caso, o mapeamento acessa os dados através da consulta "db.projects.find()" que recupera todos os documentos da coleção. Para cada documento recuperado, o mapeamento construirá o sujeito RDF conforme descrito na linha 3 pela propriedade rr:subjectMap e as relações predicado-objeto descrito nas linhas 4 e 5 pelas propriedades rr:predicateObjectMap e rr:objectMap. Neste mapeamento, a única relação definida é a relação que conecta o projeto com o líder do time. O mapeamento do objeto líder do time é definido pela regra s.teams [0,1][(@.length - 1)]. name"] que recupera o último nome ([(@.length - 1)]. name"] das listas de índice 0 e 1(s.teams [0,1]).

Com o mapeamento definido, é possível descrever o método utilizado no trabalho de Michel et al. (2016) para acessar os dados de uma coleção de arquivos no MongoDB. Utilizando o mesmo exemplo, a coleção "projects" será acessada através da consulta SPARQL no Arquivo 4.8 que recupera todos os projetos que possuam "H. Dunbar" como líder do time.

<sup>&</sup>lt;sup>3</sup>http://www.json.org/

<sup>&</sup>lt;sup>4</sup>https://www.w3.org/XML/

Arquivo 4.8: Consulta SPARQL que recupera os projetos que possuam "H. Dunbar" como líder do time

1 SELECT ?proj WHERE {?proj ex:teamLeader "H. Dunbar".}

Esta consulta SPARQL (Arquivo 4.8) será transformada em uma consulta abstrata conforme as regras da gramática no Arquivo 4.9.

#### Arquivo 4.9: Gramática da consulta abstrata.

Cada tripla na cláusula **WHERE** da consulta SPARQL é traduzida através da união das consultas abstratas (regra da linha 5 do Arquivo 4.9). E cada componente da regra **AtomicQuery** é extraída do mapeamento da seguinte forma:

- From: representa a fonte em que os dados serão recuperados (xrr:logicalSource);
- Project: os dados de retorno da consulta;
- Where: representam as condições a serem respeitadas na consulta. Essas condições podem ser filtros da consulta SPARQL ou variáveis da consulta SPARQL que devam possuir algum valor definido pelo mapeamento;

Utilizando a gramática definida no Arquivo 4.9 é gerada a consulta abstrata descrita no Arquivo 4.10 para a consulta SPARQL no Arquivo 4.8.

```
Arquivo 4.10: Consulta abstrata gerada a partir da gramática 4.9 para a consulta SPARQL 4.8
```

Em seguida, o próximo passo consiste em converter a consulta abstrata definida no Arquivo 4.10 para a consulta no padrão find() do MongoDB.

Duas condições surgem a partir do componente *WHERE*:

- C1: isNotNull(\$.code)
- C2: equals(\$.teams[0,1][(@.length 1)].name, "H. Dunbar").

Através destas condições, são gerados trechos de consultas do MongoDB, que passam por um processo de otimização e resultam na consulta final descrita abaixo.

#### Arquivo 4.11: Consulta final gerada.

```
 \begin{array}{lll} 1 & \{"code": \{\$exists:true, \$ne:null\}, \ "teams.0": \{\$exists:true\}, \$where: ?this. \\ & \hookrightarrow teams [0][this.teams [0].length-1)].name == "H. Dunbar"? \} \\ 2 & \{"code": \{\$exists:true, \$ne:null\}, \ "teams.1": \{\$exists:true\}, \$where: ?this. \\ & \hookrightarrow teams [1][this.teams [1].length-1)].name == "H. Dunbar"? \} \\ \end{array}
```

A consulta descrita no Arquivo 4.11 retorna o resultado apresentado no Arquivo 4.12:

```
Arquivo 4.12: Retorno da consulta do Arquivo 4.11
```

```
1 { "project": "Customer Relation", "code": "crm", "teams": [ [ {"name": "R. \hookrightarrow Posner"}, {"name": "H. Dunbar"}]]}
```

Finalmente, a partir do resultado da consulta, são geradas as triplas RDF (Arquivo 4.13).

```
Arquivo 4.13: Tripla RDF para o retorno da consulta do Arquivo 4.11
```

```
1 <http://example.org/project/crm> ex:teamLeader "H. Dunbar".
```

# Capítulo 5

# Framework genérico de acesso a dados baseado em ontologias

Para atingir os objetivos desta dissertação, propomos um novo modelo OBDA capaz de acessar sistemas de bancos de dados relacional e NoSQL.

#### 5.1 Método

A abordagem propõe um modelo OBDA que permita o acesso a dados a qualquer repositório seja ele um sistema de bancos de dados relacional ou NoSQL. Neste trabalho foi criado dois protótipo deste modelo OBDA para validação, um protótipo utilizou o sistema gerenciador de bancos de dados relacional PostgreSQL $^1$  e o outro utilizou o sistema de bancos de dados MongoDB. Respectivamente, os protótipos foram nomeados de OntoSQL e OntoMongo.

O foco principal deste trabalho estava em possibilitar o acesso aos dados armazenados em bancos de dados orientado a documentos através do paradigma OBDA. Isso porque, os sistemas OBDA para acesso em bancos de dados relacionais já estão muito robustos atualmente. Possuímos também dois casos de uso onde os dados estão em formato de documentos, um caso de uso abordado por de Oliveira e Wassermann (2017) envolve documentos do Supremo Tribunal Federal e o outro caso de uso que é utilizado nesta dissertação envolve documentes referentes a currículos acadêmicos. Além disso, utilizar o modelo OBDA com o sistema MongoDB possibilita a comparação com o trabalho feito por Botoeva et al. (2016).

A primeira etapa deste trabalho resultou a construção do protótipo OntoMongo que foi realizado em conjunto com Araujo et al. (2017). A partir da construção deste protótipo foi possível identificar quais etapas do modelo poderiam ser generalizadas para a utilização com um sistema de bancos de dados qualquer, como por exemplo um sistema de bancos de dados relacional.

Nas próximas seções deste capítulo será explicado o modelo OBDA genérico e nos capítulos 6 e 7 serão explicados a construção dos protótipos para os casos de uso utilizando o SGBD MongoDB e PostgreSQL.

A Figura 5.1 ilustra a estrutura conceitual do modelo OBDA proposto nesta dissertação. A estrutura é similar a estrutura definida pela literatura, ilustrada pela Figura 3.1. Onde temos a ontologia como camada conceitual, o banco de dados para armazenamento de dados e o mapeamento que liga a ontologia ao banco de dados. Pensando em viabilizar a utilização de qualquer sistema de bancos de dados foi proposto a adição de uma camada intermediária com o modelo de objeto representando o banco de dados. A utilização desta camada intermediário tem como objetivo facilitar a generalização da construção do mapeamento. Observando os trabalhos existentes, os mapeamentos para bancos relacionais são feitos utilizando consultas SQL, porém, como não existe uma linguagem de consulta universal para os sistemas NoSQL, o mapeamento precisaria ser modificado para ser compatível com cada linguagem de consulta nova.

<sup>&</sup>lt;sup>1</sup>https://www.postgresql.org/

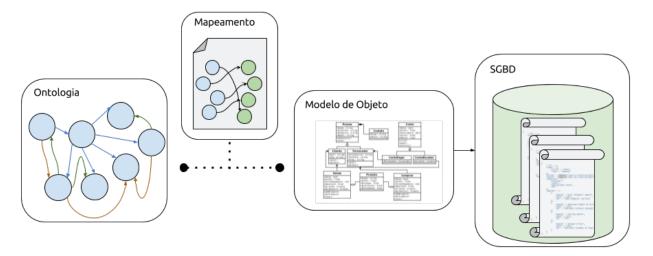


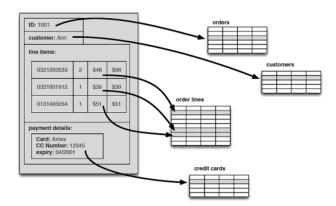
Figura 5.1: Estrutura conceitual do modelo OBDA

Com o objetivo de construir um mapeamento que seja independente da linguagem de consulta utilizada pelo SGBD, definimos uma camada conceitual intermediária através de um conjunto de classes em um modelo orientado a objetos. A utilização do modelo de objetos para representar o esquema da base de dados é útil pois se aproxima da representação do domínio da ontologia, uma vez que tanto na orientação a objetos quanto na representação do conhecimento da ontologia podemos descrever classes com suas relações, hierarquias e heranças.

#### 5.1.1 Camada intermediária

O modelo proposto possui uma camada intermediária responsável por traduzir os dados armazenados na base para o modelo de objetos (object model), como podemos verificar na Figura 5.1. Em aplicações que armazenam dados em sistemas de bancos de dados relacionais e que são desenvolvidas em linguagens de programação orientadas a objetos, esse mecanismo de tradução de objetos para o formato relacional (e vice-versa) é bastante comum. A diferença na representação dos dados relacionais e as estruturas dos dados em memória é chamado por Sadalage e Fowler (2012) de **incompatibilidade de impedância**. Como pode ser visto na Figura 5.2 um único objeto pode ser dividido em muitos registros de muitas tabelas.

Como explicado por Sadalage e Fowler (2012), se tornou mais fácil tratar a incompatibilidade de impedância pela ampla disponibilidade de frameworks de mapeamento objeto-relacional também conhecidos por ORM (Object Relational Mapping). No caso de sistemas de bancos de dados orientados a documentos, o mesmo problema é solucionado por ODMs (Object Document Mapping).



**Figura 5.2:** Incompatibilidade de impedância entre a representação de um objeto e os dados armazenados em um banco relacional (Sadalage e Fowler, 2012)

5.1 Método 21

Por serem conceitos bastante semelhantes, a construção de um mapeamento entre a ontologia e as classes do modelo de objetos fica mais simples, uma vez que ORM e ODM possuem a tarefa de transformar os dados armazenados nas bases de dados em coleções de objetos. A utilização de ORM/ODM traz grandes vantagens, como a capacidade de fazer a representação dos dados em um modelo de objetos e disponibilizar operações do banco de dados, como inserção, consulta, alteração e remoção.

#### 5.1.2 Mapeamento e comunicação com o SGBD

O mapeamento é responsável por conectar os conceitos descritos pela ontologia aos conceitos definidos pelo modelo de objetos. Alguns métodos foram criados para ligar o nome de cada classe e atributos descritos na ontologia com sua respectiva classe e atributos nas classes do modelo de objetos.

Para cada classe descrita pelo modelo de objeto será necessário criar um mapeamento conectando este a um conceito da ontologia. Foram criados alguns métodos para definir o mapeamento. Arquivo 5.1 demonstra a sintaxe do mapeamento. O mapeamento para um conceito é inicializado pelo comando **OntoMap.mapping** recebendo como parâmetro o nome da classe da ontologia que será mapeada. O comando **model** recebe como parâmetro o nome da classe do modelo de objeto que equivale a classe da ontologia, estabelecendo o mapeamento entre os dois conceitos. E para definir o mapeamento entre os atributos são utilizados os comandos **maps from** que recebe como parâmetro o atributo descrito pela ontologia e o comando **to** que recebe como parâmetro o atributo descrito pelo modelo de objeto.

#### Arquivo 5.1: Sintaxe do mapeamento

```
1 OntoMap.mapping <classe_da_ontologia> do
2    model <classe_do_modelo_objeto>
3    maps from: '<atributo_ontologia1>',to: :<atributo_modelo_de_objeto1>
4    maps from: '<atributo_ontologia2>',to: :<atributo_modelo_de_objeto2>
5    end
```

Com o mapeamento configurado para fazer a comunicação entre a ontologia e a base de dados, o sistema OBDA é capaz de receber uma consulta SPARQL, como pode ser visto na Figura 5.3, e transforma-la em uma consulta para o SGBD para fazer o acesso aos dados. Através da consulta SPARQL é gerado um grafo contendo as informações que a consulta SPARQL vai acessar na base. Cada nó do grafo representa os diferentes objetos e possui os atributos que estão sendo acessados pela consulta SPARQL. Cada aresta indica a relação entre os dois objetos.

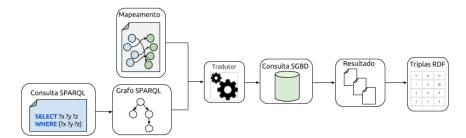


Figura 5.3: Etapa de processamento da consulta SPARQL no modelo OBDA proposto

A partir deste grafo e do mapeamento, é possível saber quais dados a consulta SPARQL precisa acessar da base de dados. Assim, os dados podem ser recuperados através das operações disponibilizadas pelas ORM/ODM ou de mecanismos internos que geram consultas no SGBD utilizado.

#### 5.1.3 Tradutor

O tradutor transforma consultas SPARQL em consultas ao SGBD, utilizando para isso o mapeamento. A consulta SPARQL é primeiro transformada em uma SPARQL Syntax Expression

(S-Expression) através da biblioteca  $sxp^2$ , pois, como podemos ver no Arquivo 6.4, a biblioteca sxp reorganiza o corpo da consulta SPARQL facilitando o processamento da mesma. A consulta é então transformada em um grafo que representa as triplas RDF, filtros e atributos que devem ser retornados para esta consulta. O método de construção deste grafo está representado no Arquivo 5.2.

Arquivo 5.2: Algoritmo que constrói grafo que representa a consulta SPARQL

```
Entrada: Consulta SPARQL Q
   Saída: Grafo caracterizando a consulta SPARQL
   L = lista de triplas do corpo da consulta Q (BGP)
5
   G = grafo vazio
7
   paracada tripla t \in L faça
8
      defina t. sujeito como o sujeito da tripla
9
     defina t.predicado como o predicado da tripla
     defina t.objeto como o objeto da tripla
10
11
12
     se não existe um nó em G para t.sujeito então
13
        cria nó em G para t. sujeito
14
     fim se
15
     se t. predicado é um DatatypeProperty então
16
        adiciona a informação ao nó para t.sujeito
17
     senão se t. predicado é ObjectProperty então
18
19
        cria nó em G para t.objeto adjacente a t.sujeito
20
     fim se
21
   fim paracada
22
23
   retorne G
```

O tradutor utiliza então os métodos das classes de modelo e o grafo gerado para produzir uma consulta para o SGBD.

#### 5.1.4 Exportação para RDF

Os dados extraídos do banco de dados são finalmente transformados em triplas RDF. As triplas RDF são geradas de acordo com a resposta obtida pela consulta ao banco de dados. Para cada valor recuperado, é gerada uma tripla utilizando como sujeito o id do registro, como predicado o nome equivalente ao atributo/relação na ontologia dado pelo mapeamento e como objeto o valor recuperado.

Com isso, é então populada a ontologia, onde é realizada a consulta SPARQL.

### 5.2 Arquitetura genérica para sistemas OBDA

A Figura 5.4 caracteriza a arquitetura genérica proposta para sistemas OBDA.

Como é possível verificar na Figura 5.4, os itens ontologia, base de dados e mapeamento devem ser fornecidos pelo usuário que construirá o sistema OBDA. Desta forma, assumimos a existência de uma base de dados e uma ontologia que descreve o domínio para a base de dados. O mapeamento, como representado no item (a), deverá ser construído pelo usuário especialista com conhecimentos do domínio (ontologia) e conhecimento da estrutura em que os dados estão armazenados. O item (b) ilustra a representação da base de dados através do modelo de objeto.

O modelo de objeto é um facilitador para a construção do mapeamento entre a ontologia e a base de dados. Nos casos em que os bancos de dados são NoSQL, a utilização do modelo de objeto provê uma estrutura para os dados que possuem características semi-estruturadas.

<sup>&</sup>lt;sup>2</sup>Disponível em: https://github.com/dryruby/sxp.rb

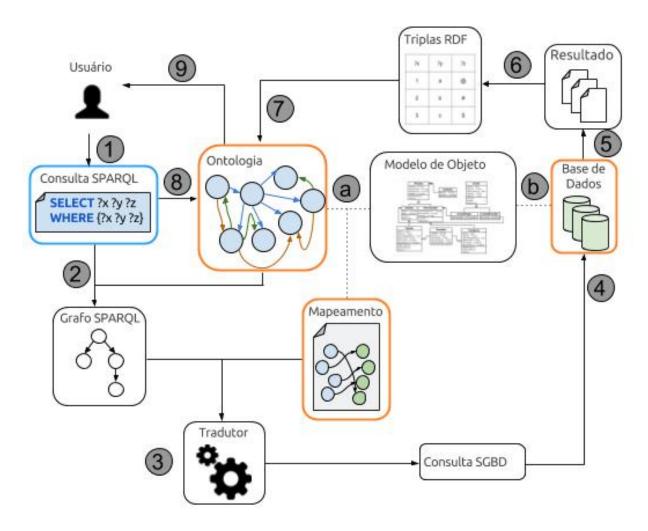


Figura 5.4: Arquitetura genérica para sistemas OBDA

A entrada que o sistema OBDA recebe está representada no passo (1), onde o usuário cria uma consulta SPARQL com o vocabulário oferecido pela ontologia.

No passo (2), o sistema fica responsável por construir um grafo que caracteriza as informações que a consulta SPARQL deseja recuperar.

Através do grafo que representa a consulta SPARQL e o mapeamento, o passo (3) da arquitetura fica responsável por construir uma consulta para o SGBD referente ao banco de dados utilizado. Esta consulta para SGBD recupera as informações necessárias para responder a consulta SPARQL passada pelo usuário.

Assim, no passo (4) a consulta é executada pelo SGBD e seu resultado retornado no passo (5). No passo (6) os dados são convertidos em triplas RDF. Então, as triplas RDF geradas populam a ontologia (passo (7)), para que a consulta SPARQL possa ser executada sobre a ontologia (passo (8)). Por fim, o resultado é retornado ao usuário em (9).

#### 5.2.1 Categorias de desenvolvimento

Com esse modelo, podemos separar as etapas de processamento em quatro categorias, de acordo com a necessidade de modificação de cada uma.

A primeira consiste das etapas que são fixas dentro da arquitetura. São elas os passos (2), (6), (7) e (8). Esses passos não necessitam de alterações pela parte do usuário especialista que instancia o sistema, uma vez que são independentes do domínio e do SGBD utilizados.

A segunda categoria contém as etapas que são dependentes do SGBD. Nessa categoria, se encontram os passos 4 e 5. Esses passos devem ser implementados especificamente para a linguagem de consulta do SGBD a ser utilizado. A vantagem deste modelo é que essa tarefa, por mais custosa que seja, só deve ser realizada uma única vez para cada linguagem. Como neste trabalho são apresentadas, por exemplo, traduções para a linguagem de consulta do MongoDB e para SQL, qualquer usuário especialista que queira instanciar o modelo para um SGBD que utilize alguma dessas linguagens não precisará realizar essa tarefa.

A terceira categoria abarca as etapas que devem ser criadas sempre que se for instanciar o modelo para utilização. A ontologia, o mapeamento (a), o modelo de objetos (b) e a base de dados devem ser providos por um usuário especialista no domínio e na estrutura em que os dados estão armazenados. É possível, no entanto, através de processos automáticos ou semi-automáticos gerar o modelo de objetos de forma não manual, restando assim somente a ontologia e o mapeamento para serem construídos.

A última categoria apresenta as etapas que são dependentes do usuário que for utilizar a instância criada pelo especialista. Nas etapas 1 e 9 o usuário deve fornecer uma consulta SPARQL e receber a resposta para sua consulta com os dados encontrados na base de dados.

Desta forma, podemos delimitar as tarefas de cada um dos envolvidos no processo de utilização deste modelo para tornar claro o esforço individual dos participantes.

## Capítulo 6

# OntoMongo

O OntoMongo<sup>1</sup> é um protótipo funcional que utiliza a arquitetura descrita no Capítulo 5 e foi desenvolvido em conjunto com outros pesquisadores e o conteúdo deste capítulo foi apresentado por Araujo *et al.* (2017).

Onto Mongo é capaz de receber como entrada uma consulta em linguagem SPARQL e responder a essa consulta com um conjunto de informações em forma de triplas RDF. Este protótipo foi desenvolvida na linguagem de programação Ruby e utiliza o framework Ruby on  $Rails^2$  criado para facilitar a criação de aplicações e serviços web.

## 6.1 Estudo de caso

Foi criado um estudo de caso aplicando a abordagem descrita para responder a uma consulta parecida à respondida por Botoeva et al. (2016), que é a de descobrir quais são os pesquisadores com duas publicações no mesmo ano. Para isto, aplicamos apenas uma parte da linguagem de consulta do MongoDB que trata do método de agregação.

## 6.1.1 Definição do domínio

Criamos uma ontologia chamada *basic-lattes* a respeito de pesquisadores e sua produção bibliográfica, a hierarquia das classes pode ser vista na Figura 6.1.



Figura 6.1: Hierarquia das classes definidas na ontologia

A ontologia completa pode ser encontrada no Anexo H. A Figura 6.2 ilustra as classes de pesquisador e artigo e a relação entre os dois que serão de maior forco nos próximos exemplos.

<sup>&</sup>lt;sup>1</sup>Projeto de código-aberto: http://github.com/thdaraujo/onto-mongo

<sup>&</sup>lt;sup>2</sup>Framework para Ruby http://rubyonrails.org/

26 ONTOMONGO 6.1

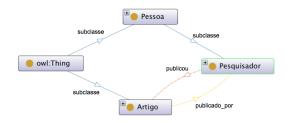


Figura 6.2: Ontologia basic-lattes por Araujo et al. (2017)

#### 6.1.2 Base de dados

O banco de dados *MongoDB* utilizado contém 39.833 de currículos com informações sobre a produção científica de pesquisadores, que foram extraídos da Plataforma Lattes do CNPq³ através do projeto *scriptLattes* desenvolvido por Mena-Chalco e Junior (2009). Essa ferramenta exporta currículos em formato XML contendo informações detalhadas sobre as publicações dos pesquisadores. A estrutura dos arquivos pode ser encontrada no apêndice G. Os arquivos XML foram transformados em documentos JSON e alguns campos de interesse foram extraídos, tais como o nome do pesquisador, sua lista de artigos publicados e coautores. A estrutura dos documentos armazenados no banco de dados foi construída de forma similar ao arquivo utilizado no trabalho de Botoeva *et al.* (2016) e pode ser encontrada no apêndice F.

A recuperação dos dados é feita utilizando uma biblioteca  $Ruby^4$  chamada  $mongoid^5$ , desenvolvida pelos criadores do MongoDB como um ODM (Object-DocumentMapper) que transforma os objetos da linguagem de programação em documentos persistidos no banco de dados MongoDB. Essa biblioteca também possui métodos para inserção e atualização desses documentos.

A Figura 6.3 ilustra a construção de duas classes que representam os documentos armazenados no banco de dados. As classes *Researcher* e *Publication* foram construídas manualmente e seu código pode se encontrado no Anexo I.

Para fazer uma comparação com o artigo citado, também adicionamos o pesquisador "Kristen Nygaard" e duas de suas publicações com o mesmo ano, que foram salvas na coleção researcher no MongoDB. Esse pesquisador será utilizado para ilustrar o processo de recuperação dos dados.

## 6.1.3 Mapeamento

O proximo passo é construir o mapeamento conectando os conceitos definidos pela ontologia aos conceitos definido pelo modelo de objeto. A Figura 6.4 ilustra as classes da ontologia basic-lattes e suas classes equivalente no modelo de objeto.

O mapeamento da classe *Pesquisador* é apresentado no Arquivo 6.1:

Arquivo 6.1: Mapeamento da classe Pesquisador com a model Researcher.

```
OntoMap.mapping 'Pesquisador' do
model Researcher
maps from: 'nome', to: :name
maps from: 'pais', to: :country
maps from: 'nome_em_citacoes', to: :name_in_citations
maps relation: 'publicou', to: :publications
end
```

A linha 1 do Arquivo 6.1 indica o começo do mapeamento para o conceito *Pesquisador* da ontologia e que está mapeado para a classe *Researcher* do modelo de objeto. Nas linhas 3 a 5,

<sup>&</sup>lt;sup>3</sup>http://lattes.cnpq.br/

<sup>&</sup>lt;sup>4</sup>Linguagem de Programação Orientada a Objetos https://www.ruby-lang.org/en/

 $<sup>^5 \</sup>mathrm{https://docs.mongodb.com/mongoid/master/}$ 

6.1 ESTUDO DE CASO 27

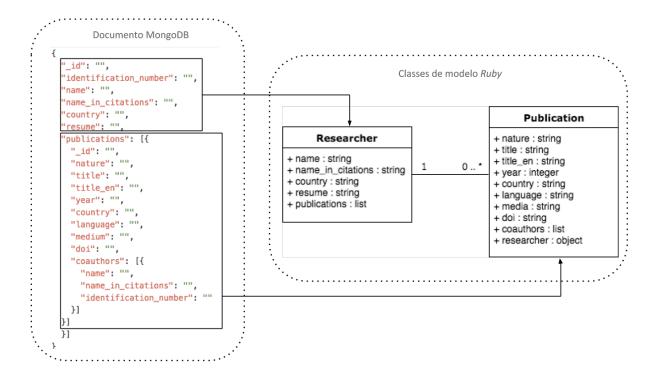


Figura 6.3: Diagrama das classes de modelo: Researcher e Publication

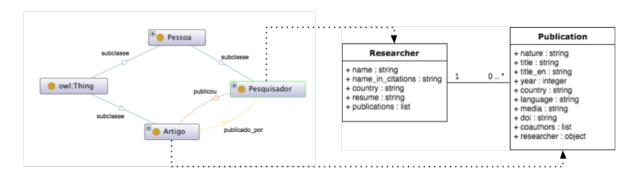


Figura 6.4: Equivalência entre classes da ontologia e classes do modelo de objeto.

cada atributo definido no modelo de objeto é mapeado para o atributo correspondente descrito pela ontologia através do método maps.

A linha 6 descreve a relação *publicou* entre as classes *Pesquisador* e *Artigo* mapeada para a relação *publications*.

De forma similar o Arquivo 6.2 apresenta o mapeamento entre a classe *Artigo* e o modelo **Publication**.

Arquivo 6.2: Script de mapeamento da classe Artigo com a model Publication.

```
OntoMap.mapping 'Artigo' do
 1
 2
      model Publication
3
      maps from:
                    'natureza', to: :nature
                    "titulo", to: :title
 4
      maps from:
                    'titulo_em_ingles', to: :title_en
 5
      maps from:
                    'ano', to: :year
'pais', to: :country
      maps from:
7
      maps from:
                    'idioma', to: language
'veiculo', to: media
9
      maps from:
      maps from: 'doi', to: doi
10
```

28 ONTOMONGO 6.1

11 end

#### 6.1.4 Tradutor

Utilizando a ontologia e o mapeamento definidos, desejamos descobrir quais são as publicações distintas publicadas no mesmo ano e seu autor. Isso pode ser alcançado pela consulta SPARQL descrita em 6.3:

**Arquivo 6.3:** Consulta SPARQL para consultar todos os pesquisadores que possuem duas publicações no mesmo ano

```
1
   PREFIX : <http://onto-mongo/basic-lattes/#>
2
   SELECT
       ?nomePesquisador ?tituloPublicacao1 ?tituloPublicacao2 ?ano
3
   WHERE
5
   {
6
       ?pesquisador :nome ?nomePesquisador .
7
       ?pesquisador :publicou ?publicacao1
       ?pesquisador :publicou ?publicacao2
8
       ?publicacao1 :ano_publicacao ?ano .
9
10
       ?publicacao2 : ano publicacao ? ano .
       ?publicacao1 :titulo ?tituloPublicacao1 .
11
12
       ?publicacao2 : titulo ?tituloPublicacao2 .
13
       FILTER
14
            (?publicacao1 != ?publicacao2)
15 }
```

Primeiro, transformamos essa consulta em uma  $SPARQL\ Syntax\ Expression$ , usando a biblioteca sxp:

Arquivo 6.4: Consulta SPARQL transformada em S-Expression

```
(project
2
     (?nomePesquisador ?tituloPublicacao1 ?tituloPublicacao2 ?ano)
3
4
       (!= ?publicacao1 ?publicacao2)
5
       (bgp
6
          (triple ?pesquisador :nome ?nomePesquisador)
          (triple ?pesquisador :publicou ?publicacao1)
          (triple ?pesquisador :publicou ?publicacao2)
          (triple ?publicacao1 :ano publicacao ?ano)
10
          (triple ?publicacao2 :ano publicacao ?ano)
11
          (triple ?publicacao1 : titulo ?tituloPublicacao1)
12
          (triple ?publicacao2 :titulo ?tituloPublicacao2))))
```

Essa expressão é desmembrada em suas componentes, para que possam ser transformadas na consulta ao sistema de banco de dados. O algoritmo desmembra a consulta nas seguintes partes:

- Variáveis de saída (project);
- Filtros que devem ser aplicados (filter);
- Corpo da consulta (bgp basic graph pattern) com as triplas (triple);

A partir da consulta SPARQL do Arquivo 6.3, é construído o grafo da Figura 6.5. Este grafo é construído analisando cada tripla da cláusula WHERE da consulta SPARQL.

Na linha 6 da consulta do código 6.4, a primeira tripla contém como sujeito a variável **?pesquisador** representando uma instância para a classe **Pesquisador** na ontologia. E como não há nenhum nó no grafo, criamos o primeiro nó representando a variável **?pesquisador**. Como o predicado desta tripla é um atributo (**nome**) e o objeto é uma variável, inserimos o atributo **nome** na lista de atributos do nó **?pesquisador**.

Nas linhas 7 e 8, o sujeito também se refere à variável **?pesquisador**, porém, o predicado é uma relação. Essa relação aponta para as variáveis **?publicacao1** e **?publicacao2** que representam

6.1 ESTUDO DE CASO 29

instâncias da classe **Publicacao** na ontologia. Desta forma, são criados dois nós para cada uma destas variáveis. Esses dois nós são conectados ao nó **?pesquisador** por causa da relação **publicou**.

Por fim, as triplas nas linhas 9 a 12 que têm como predicado os atributos **ano\_publicacao** e **titulo**, são adicionadas à lista de atributos de seus respectivos nós.

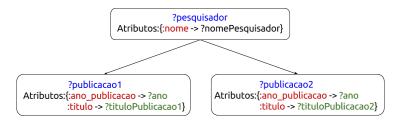


Figura 6.5: Grafo construído a partir da consulta SPARQL

O grafo (Figura 6.5) construído será processado para gerar uma consulta no formato do framework de agregação MongoDB. O processo de geração da consulta é específico para cada SGBD. Se o modelo utiliza um banco de dados relacional, é necessário processar o grafo a fim de gerar uma consulta SQL. Se o sistema de banco de dados utilizado for MongoDB é necessário ser capaz de processar o grafo e gerar uma consulta no formato para o framework de agregação ou em outro formato disponível. Diferente dos SGBD relacionais que utilizam como padrão a linguagem de consulta SQL, os sistemas NoSQL não possuem formato padrão de consultas. Uma vez modelada a solução OBDA para SGBDs relacionais, a solução será a mesma para qualquer sistema de banco de dados relacional. Porém, no cenário dos sistemas de bancos de dados NoSQL, modelar um sistema OBDA que seja capaz de suportar novos sistemas NoSQL pode se tornar uma tarefa custosa. Em trabalhos como o de Botoeva et al. (2016) e Michel et al. (2016), quando se deseja modelar OBDA para um novo SGBD é necessário aumentar a abrangência da linguagem de mapeamento e criar otimizações para construir consultas eficientes.

Desta forma, o grafo com a estrutura da consulta SPARQL é um recurso facilitador a ser utilizado em sistemas OBDA, pois independentemente do SGBD, com o seu uso, o único processo que é específico para cada SGBD é a etapa da construção da consulta.

Criado o grafo de estrutura, queremos então construir a consulta que será executada no MongoDB a partir da consulta SPARQL descrita no Arquivo 6.3. Neste protótipo, a consulta objetivo é gerada no padrão Aggregation Pipeline do MongoDB.

O nó raiz, presente na Figura 6.6 é o primeiro a ser processado.



Figura 6.6: Nó representando ?pesquisador

A variável **?pesquisador** representa instâncias da classe **Pesquisador** mapeada para a classe *Ruby "Researcher"*. O nó da Figura 6.6 informa que somente o nome do pesquisador deve ser recuperado na busca que será executada no *MongoDB*. Desta forma, o primeiro comando para a consulta será o comando "*\$project: "name": true"*. Neste comando, informamos que os documentos que serão retornados como resultado da consulta devem conter o campo *name*, como pode ser verificado na Figura 6.7.

Em seguida, os dois nós adjacentes ao nó raiz da Figura 6.6 são processados.

O nó da Figura 6.8 para a variável **?publicacao1** representa instâncias da classe **Artigo** mapeada para a classe *Ruby "Publication"*. As publicações são representas dentro do documento armazenado no MongoDB como uma lista. Para consultar este tipo de estrutura dentro dos documentos, faz-se uso do comando *unwind*. A lista de publicações de cada documento é desmembrada produzindo novos documentos para cada elemento da lista. Antes disso, no entanto, é necessário informar

30 ONTOMONGO 6.1

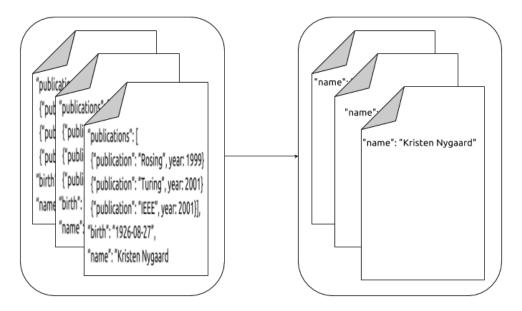


Figura 6.7: Exemplo do funcionamento do comando project

```
?publicacao1
Atributos:{:ano_publicacao -> ?ano
:titulo -> ?tituloPublicacao1}
```

Figura 6.8: Nó representando ?publicacao1

que o campo *publications* também deve ser projetado, para depois aplicar o comando *unwind* sobre a lista. Assim, o comando é atualizado para o comando do Arquivo 6.5

```
Arquivo 6.5: Adicionando comando unwind para o campo publications

{$project : {"nomePesquisador": "name", "publicacao1": "

→ $publications"}},

{$unwind : "$publicacao1"}
```

A Figura 6.9 simula a execução do comando descrito no Arquivo 6.5.

{

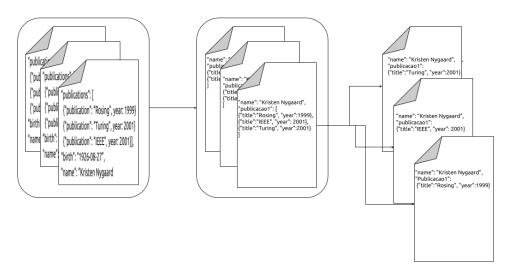


Figura 6.9: Simulação da execução do comando descrito no Arquivo 6.5

6.1 ESTUDO DE CASO 31

Os atributos ano\_publicacao e titulo são adicionados em outro comando project após o comando unwind, como será mostrado mais adiante.

Em seguida, o nó da Figura 6.10 é processado. Este nó é similar ao nó que representa **?publicacao1**. Assim, o mesmo processamento é realizado. O processo resulta na consulta do Arquivo 6.6.

```
?publicacao2
Atributos:{:ano_publicacao -> ?ano
:titulo -> ?tituloPublicacao2}
```

Figura 6.10: Nó representando ?publicacao2

Por fim, os filtros são adicionados à consulta. A linha 4 do Arquivo 6.4 apresenta filtros que definem que **?publicacao1** deve ser diferente de **?publicacao2**. Mas também levamos em conta a informação presente nas linhas 9 e 10: a consulta exige que o ano das publicações seja o mesmo, pois a variável **?ano** é igual para **?publicacao1** e **?publicacao2**. Portanto, o filtro deve garantir que cada um dos pares de publicações sejam diferentes entre si, mas sejam do mesmo ano. Em resumo, a consulta deve trazer todas as publicações distintas que foram publicadas no mesmo ano. Podemos verificar a consulta final no Arquivo 6.7.

```
Arquivo 6.7: Consulta Final
{
         \{\$project : \{"pesquisador.nomePesquisador": "name", "publicacao1": \\
           → "$publications", "publicacao2": "$publications"}},
        {$unwind : "$publicacao1" }
        {$unwind : "$publicacao2" }
        {\tt \{\$project : \{"pesquisador.nomePesquisador": true\,,}
                                "publicacao1": true,
                        "publicacao2": true,
                        "publicacao1.ano": "$publicacao1.year",
                        "publicacao1.titulo": "$publicacao1.title",
                        "publicacao2.ano": "\$publicacao2.year",\\
                        "publicacao2.titulo": "$publicacao2.title",
                        \hookrightarrow $publicacao2.year"] },{ $ne: ["$publicacao1.

    title ", "$publicacao2.title "] }] }
}
        { :\smatch => \{ ?filtro ? => true \}\}
}
```

32 Ontomongo 6.1

## 6.1.5 Resultado

Utilizamos o mesmo documento utilizado no trabalho de Botoeva *et al.* (2016) para comparar o retorno. O resultado da consulta é o documento JSON do Arquivo 6.8, contendo as publicações distintas que foram publicadas no mesmo ano:

Arquivo 6.8: Documentos retornados pela consulta

```
[{ '_id' => BSON:: ObjectId('5835 effcc048bd0001cdc239'), 'pesquisador.nomePesquisador'=> "Kristen Nygaard", 'publicacao1.id'=> '5835 effcc048bd0001cdc23b', 'publicacao2.id'=> '5835 effcc048bd0001cdc23c', 'publicacao1.ano'=> 2001, 'publicacao1.titulo'=> 'Turing Award', 'publicacao2.ano'=> 2001, 'publicacao2.titulo'=> 'von Neumann Medal'}]
```

Com esses dados, as triplas RDF são geradas para serem inseridas na ontologia (Arquivo 6.9). As triplas RDF são geradas de acordo com as triplas do comando WHERE da consulta SPARQL. Por exemplo, a tripla da linha 6 na consulta do Arquivo 6.4 gera triplas para o predicado :name. A variável ?pesquisador é substituído pelo id do registro e a variável ?nomePesquisador é substituída pelo valor do campo "pesquisador.nomePesquisador" retornado pelos documentos de retorno.

Arquivo 6.9: Grafo RDF construído a partir dos dados de retorno

```
:/5835effcc048bd0001cdc239 :nome "Kristen Nygaard"
:/5835effcc048bd0001cdc23b :titulo "Turing Award"
:/5835effcc048bd0001cdc23b :ano_publicacao 2001
:/5835effcc048bd0001cdc23c :titulo "von Neumann Medal"
:/5835effcc048bd0001cdc23c :ano_publicacao 2001
:/5835effcc048bd0001cdc239 :publicou :/5835effcc048bd0001cdc23b
:/5835effcc048bd0001cdc239 :publicou :/5835effcc048bd0001cdc23b
```

Com as triplas inseridas, a consulta SPARQL pode ser executada sobre o grafo RDF.

No trabalho de Botoeva et al. (2016), a consulta MongoDB gerada utiliza-se de várias subconsultas para a geração das triplas RDF, deixando os filtros por conta da consulta SPARQL. Assim, são geradas múltiplas triplas desnecessárias que serão ignoradas após a aplicação do filtro. No trabalho de Michel et al. (2016) também são geradas sub-consultas, porém, as consultas geradas passam por um processo de otimização, o que muitas vezes é custoso para a aplicação.

O modelo descrito até aqui mostrou-se então capaz de gerar uma única consulta MongoDB capaz de recuperar os mesmos dados que seriam recuperados pela consulta SPARQL equivalente, sem a necessidade de otimizações sobre a consulta gerada. É importante salientar neste momento que o OntoMongo foi desenvolvido em conjunto com Araujo et al. (2017).

## Capítulo 7

# OntoSQL

OntoSQL é uma segunda instância do framework genérico para a aplicação do modelo proposto nesta dissertação utilizando um banco de dados relacional. Para este estudo de caso a ontologia é a mesma utilizada pelo OntoMongo (Anexo A).

## 7.1 Base de dados

Foi utilizado o SGBD *PostgreSQL* para armazenar os dados extraídos dos currículos da Plataforma Lattes. Na Figura 7.1 temos um trecho do diagrama da modelagem dos dados extraídos dos currículos Lattes. O esquema de banco de dados completo está descrito no Anexo D.

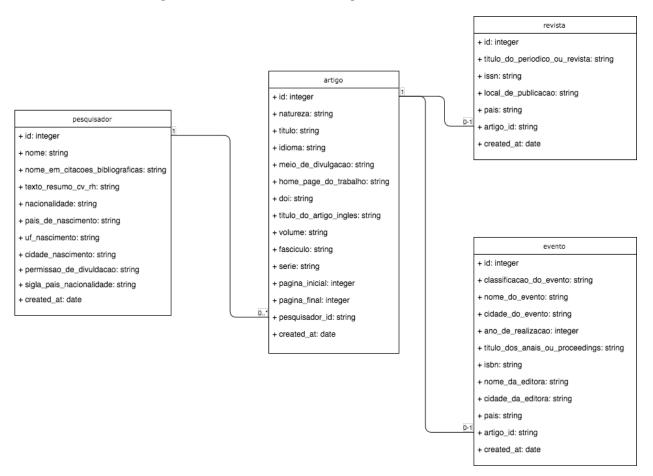


Figura 7.1: Diagrama do modelo de objeto

A construção das classes do modelo na linguagem Ruby para esta base foi feita de forma semelhante às do OntoMongo, seguindo agora a estrutura do banco relacional. O mapeamento para o

34 Ontosql 7.2

OntoSQL pode ser encontrado no Apêndice B.

## 7.2 Construindo consultas SQL

Como mencionado anteriormente, a única etapa do método que muda de acordo com o tipo de sistema gerenciador de banco de dados utilizado é a etapa de geração da consulta a ser executada no SGBD. É possível verificar a ideia da construção da consulta SQL pela descrição dada no Arquivo 7.1.

Arquivo 7.1: Algoritmo responsável por construir a consulta SQL

```
Entrada: Grafo caracterizando a consulta SPARQL G e lista de filtros F,
       → Mapeamento M
   Saída: Consulta SQL
3
   Q = Consulta SQL
   paracada nó n ∈ G então
5
6
     se n possui um filho então
        Cria join entre o pai e o filho e adiciona na consulta Q
7
8
9
        Verifica em M qual tabela SQL é mapeada para n e adiciona na consulta Q
10
     fim se
11
12
      Verifica em M a correspondência dos atributos descritos em n e adiciona
13
      Verifica em M a correspondência dos filtros descritos em n e adiciona em
         \hookrightarrow Q
14
   fim paracada
15
   Verifica em M a correspondência dos filtros descritos em F e adiciona em Q
16
17
18
   retorne Q
```

Utilizaremos a consulta SPARQL do código 7.2 para exemplificar a construção de uma consulta SQL pelo OntoSQL. Esta consulta SPARQL recupera informações de nome do pesquisador, título do artigo e nome da revista de todos os pesquisadores que possuem artigos publicados em revista.

#### Arquivo 7.2: Consulta SPARQL

O grafo da Figura 7.2 descreve a consulta SPARQL e através deste grafo é construída a consulta SQL.

Para a construção das consultas SQL, o Onto SQL utiliza as chamadas de métodos disponíveis no framework ORM do Ruby chamado  $Active\ Record^1$ .

A primeira etapa da construção da consulta SQL consiste em percorrer o grafo (Figura 7.2) e criar os  $JOINs^2$  necessários. Para associações simples entre duas entidades, a sintaxe utilizada seria "Model1.joins(:model2)". Desta forma, se analisarmos a primeira relação no grafo, é possível verificar que existe uma relação entre os nós ?pesquisador e ?artigo. Pelo mapeamento, sabemos que a relação no banco de dados está entre as entidades Pesquisador e Artigo. Então, definimos

<sup>&</sup>lt;sup>1</sup>https://github.com/rails/rails/tree/master/activerecord

<sup>&</sup>lt;sup>2</sup>Comando SQL utilizado para combinar registros de duas ou mais tabelas

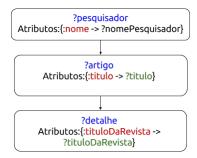


Figura 7.2: Grafo representando a consulta SPARQL

um JOIN entre as duas entidades da seguinte forma, "Pesquisador.joins(:artigos)". Este método produz a consulta SQL descrita no Arquivo 7.3.

```
Arquivo 7.3: Consulta SQL gerada a partir do comando "Pesquisador.joins(:artigos)"

SELECT pesquisador.∗, artigo.∗ FROM pesquisador INNER JOIN artigo ON

→ artigo.pesquisador_id = pesquisador.id
```

Se o nó ?pesquisador possuísse mais nós adjacentes, bastaria adicionar as entidades como parâmetro para o método joins.

A próxima relação descrita pelo grafo conecta os nós ?artigo e ?detalhe. Neste caso, será necessário adicionar um JOIN para as entidades **Artigo** e **Revista**. A chamada de método será modificada para "Pesquisador.joins(:artigos=> [:revista])". Este comando gera a consulta SQL descrita no Arquivo 7.4.

Por fim, os atributos descritos pelo grafo são adicionados à consulta através do método select ("Pesquisador.joins(:artigos=> [:revista])'.select(pesquisador.id, pesquisador.nome, artigos.id, artigos.titulo, revista.id, revista.titulo do periodico ou revista)').

A consulta resultante pode ser vista no Arquivo 7.5.

```
Arquivo 7.5: Consulta SQL final
```

O passo final consiste em gerar as triplas RDF da mesma maneira que foi descrito na seção 6.1.5.

## 7.3 Utilizando o Ontop

Nesta seção descrevemos a utilização do sistema OBDA Ontop com o domínio da Plataforma curricular *Lattes* para que possamos verificar as similaridades e diferenças entres os dois métodos. Utilizamos a ontologia descrita na seção 6.1.1 e a base de dados descrita em 7.1.

#### 7.3.1 Mapeamento Ontop

O mapeamento para o Ontop pode ser construído tanto na linguagem de mapeamento R2RML desenvolvida pela W3C, ou pela linguagem nativa do sistema. Para a construção do mapeamento

36 Ontosql 7.3

para este exemplo, foi utilizada a linguagem de mapeamento nativa oferecida pelo sistema Ontop por se mostrar mais simples para a compreensão.

No Arquivo 7.6 está descrito o trecho do mapeamento para o conceito *Pesquisador* e seus atributos descritos na ontologia.

Arquivo 7.6: Mapeamento Ontop o conceito Pesquisador definida na ontologia

```
mappingId mapping-Pesquisador
2 target : Pesquisador / {id} rdf:type : Pesquisador .
   source select id from pesquisador
3
4
                      {\it mapping-Pesquisador-nomeCompleto}
5
   mappingId
   {\bf target} \ : Pesquisador/\{id\} \ : nomeCompleto \ \{nome\}^^xsd: string \ .
6
   source select id, nome from pesquisador;
7
8
                     mapping-Pesquisador-nacionalidade
9
10
   target : Pesquisador / {id} : nacionalidade { nacionalidade } ^ xsd: string .
11
   source select id, nacionalidade from pesquisador;
12
                      mapping-Pesquisador-paisDeNascimento
13
   mappingId
   target : Pesquisador / {id} : pais De Nascimento { pais de nascimento}^^xsd: string
14
   source select id, pais de nascimento from pesquisador;
15
16
                      mapping-Pesquisador-ufNascimento
17
   mappingId
   target : Pesquisador / {id} : ufNascimento {uf nascimento}^^xsd: string .
18
   source select id, uf nascimento from pesquisador;
20
   mappingId
                      mapping-Pesquisador-siglaPaisNacionalidade
21
22
   target : Pesquisador / {id} : sigla Pais Nacionalidade { sigla pais nacionalidade
       \hookrightarrow }^^xsd:string
   source select id, sigla_pais_nacionalidade from pesquisador;
23
24
                      {\it mapping-Pesquisador-pais} De Nacional idade
25
   mappingId
26
   target : Pesquisador/{id} : paisDeNacionalidade { pais de nacionalidade}^^xsd:

→ string .

27
   source select id, pais de nacionalidade from pesquisador;
28
29
                      mapping-Pesquisador-nome Em Citacoes Bibliograficas
   mappingId
30
   target : Pesquisador / {id} : nomeEmCitacoesBibliograficas {
       → nome em citacoes_bibliograficas}^^xsd:string .
   source select id, nome em citacoes bibliograficas from pesquisador;
31
32
                      mapping-Pesquisador-textoResumoCvRh
33
   mappingId
   target : Pesquisador/{id} :textoResumoCvRh {texto resumo cv rh}^^xsd:string
34
35
   source select id, texto resumo cv rh from pesquisador;
36
                      mapping-Pesquisador-textoResumoRvRhEn
37
   target : Pesquisador/{id} :textoResumoRvRhEn {texto resumo cv rh en}^^xsd:
38
       \hookrightarrow string.
   source select id, texto resumo cv rh en from pesquisador;
39
40
                      mapping-Pesquisador-permissao De Divulgacao\\
41
   mappingId
   target : Pesquisador/{id} : permissaoDeDivulgacao {permissao de divulgacao}^^
42
       \hookrightarrow xsd:string.
43
   source select id, permissao de divulgacao from pesquisador;
44
                      {\tt mapping-Pesquisador-numeroIdentificador}
45
   mappingId
   target : Pesquisador/{id} : numeroIdentificador {numero identificador}^^xsd:
46
   source select id, numero identificador from pesquisador;
47
48
                      mapping-Pesquisador-publicou-ArtigoPublicado\\
49
   mappingId
   target : Pesquisador/{id pesquisador} : publicou : ArtigoPublicado/{id artigo}
```

7.3 Utilizando o ontop 37

```
    → .
    51 source select pesquisador.id as id_pesquisador, revista.id as id_artigo
    → from pesquisador join artigo on pesquisador.id =artigo.pesquisador_id
    → join revista on artigo.id = revista.artigo_id
    52
    53 mappingId mapping-Pesquisador-publicou-TrabalhoEmEvento
    54 target : Pesquisador/{id_pesquisador} : publicou : TrabalhoEmEvento/{id_artigo → }.
    55 source select pesquisador.id as id_pesquisador, evento.id as id_artigo from → pesquisador join artigo on pesquisador.id =artigo.pesquisador_id → join evento on artigo.id = evento.artigo_id
```

O mapeamento é composto por *mappingId*, *source* e *target*. O *mappingId* representa um identificador para o trecho do mapeamento, *source* é a consulta SQL e *target* é a tripla com variáveis que serão preenchidas com os valores retornados pela consulta SQL. Este mapeamento é construído para cada tabela e coluna da base de dados. O mapeamento completo construído para conectar a ontologia com a base de dados pode ser encontrado no Apêndice C.

No Arquivo 7.7 podemos comparar o mapeamento do OntoSQL para os mesmo conceitos.

Arquivo 7.7: Mapeamento Onto SQL para o conceito Pesquisador definida na ontologia

```
OntoMap.mapping 'Pesquisador' do
1
2
      model Pesquisador
      maps from: 'nomeCompleto', to: :nome
3
      maps from: 'nacionalidade', to: :nacionalidade
4
      maps from: 'paisDeNascimento', to: :pais_de_nascimento
5
      maps from: 'ufNascimento', to: :uf_nascimento
6
7
      maps from: 'siglaPaisNacionalidade', to: :sigla_pais_nacionalidade
8
      maps from: 'paisDeNacionalidade', to: :pais_de_nacionalidade
      maps from: 'nomeEmCitacoesBibliograficas', to: :
9
          → nome em citacoes bibliograficas
      maps from: 'textoResumoCvRh', to: :texto_resumo_cv_rh
10
      maps from: 'textoResumoRvRhEn', to: :texto_resumo_cv_rh_en
11
      maps from: 'permissaoDeDivulgacao', to: :permissao_de_divulgacao
12
      maps \hspace{0.2cm} \textbf{from} \colon \text{ 'numeroIdentificador'}, \hspace{0.2cm} \textbf{to} \colon : numero\_identificador
13
14
   end
```

O mapeamento utilizado pelo OntoSQL está interessado somente em mapear o vocabulário utilizado pela ontologia com o vocabulário utilizado pela base de dados, transferindo ao sistema a responsabilidade de obter as informações como o *source* e *target* descritos no mapeamento do Ontop.

## 7.3.2 Execução da Consulta SPARQL pelo Ontop

Com o mapeamento construído é possível fazer consultas sobre a ontologia para acessarmos os dados armazenados na base de dados.

Vamos utilizar a consulta SPARQL definida no Arquivo 7.8 para detalhar a etapa de conversão da consulta SPARQL para uma consulta SQL do Ontop. A consulta do Arquivo 7.8 recupera a lista dos pesquisadores que possuem publicações e a lista destas publicações.

Arquivo 7.8: Consulta SPARQL que lista os pesquisadores e suas respectivas publicações de artigos

```
1 PREFIX : <http://www.onto.org/obda/basic-lattes-final#>
2
3 SELECT ?pesquisador ?titulo
4 WHERE {
5     ?pesquisador :publicou ?artigo .
6     ?artigo :titulo ?titulo
7  }
```

A consulta SPARQL (Arquivo 7.8) está representada pela árvore da Figura 7.3.

Através da árvore representada na Figura 7.3 são identificados os trechos do mapeamento que serão utilizados para gerar a consulta SQL. Esta identificação é feita analisando cada nó folha da

38 Ontosul 7.3

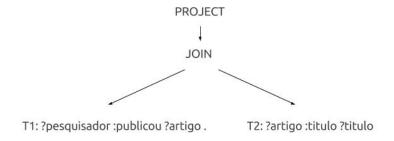


Figura 7.3: Árvore representando a consulta SPARQL (Arquivo 7.8)

árvore e procurando todos os mapeamentos em que a descrição do *target* case com a tripla descrita pelo nó folha.

Desta forma, para o nó folha T1 temos a tripla ?pesquisador :publicou ?artigo. Pelo trecho da ontologia descrita na Figura 7.4 sabemos que a variável ?pesquisador representa instâncias pertencentes à classe Pesquisador da ontologia e a variável ?artigo representa instâncias pertencentes à classe Artigo.

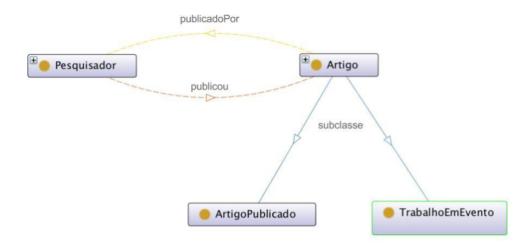


Figura 7.4: Trecho da Ontologia descrevendo relação entre classes Pesquisador e Artigo

A Figura 7.5 ilustra os trechos target do mapeamento que casam com a tripla T1. No Arquivo 7.9 podemos verificar o código descrevendo os mapeamento que casam com T1 (?pesquisador: publicou ?artigo).

Arquivo 7.9: Mapeamentos que casam com a tripla T1

Da mesma forma, podemos verificar no Arquivo 7.10 o código dos mapeamentos que casam com

7.3 UTILIZANDO O ONTOP 39



Figura 7.5: Encontrando mapeamentos que casam com a tripla T1

T2 (?artigo :titulo ?titulo).

Arquivo 7.10: Mapeamentos que casam com a tripla T2

As consultas SQL descritas pelo mapeamento dos Arquivos 7.9 e 7.10 são utilizadas para gerar uma consulta SQL única.

Na Figura 7.6 podemos ver a combinação das consultas SQL descritas nos mapeamentos referentes a *ArtigoPublicado*.

O mesmo ocorre para as consultas SQL descritas nos mapeamentos referentes a *TrabalhoEmE-vento*, ilustrado na Figura 7.7.

 $40 \quad \text{ONTOSQL}$  7.3

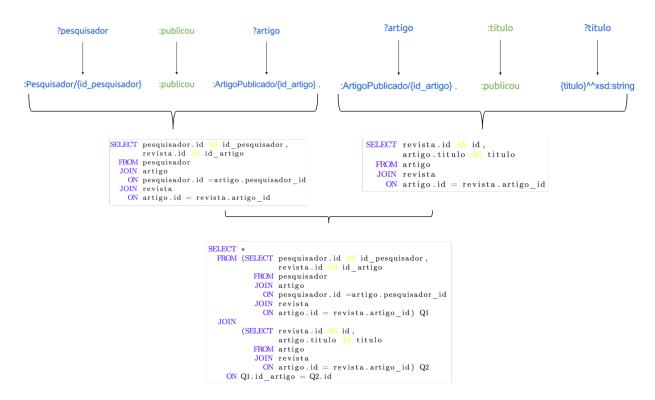


Figura 7.6: Combinação através do comando JOIN entre as consultas SQL descritas para ArtigoPublicado

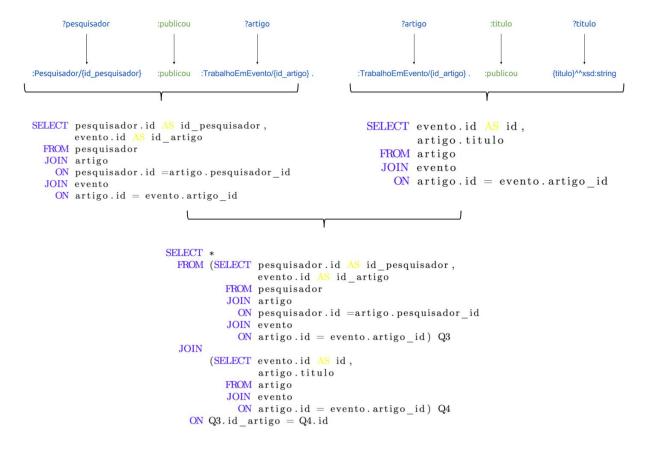


Figura 7.7: Combinação através do comando JOIN entre as consultas SQL descritas para TrabalhoEmEvento

As duas consultas SQL são unidas pelo comando SQL UNION para combinar o resultado das

7.3 Utilizando o ontop 41

consultas. Por fim, a consulta gerada passa por alguns processos de otimização. A consulta final gerada pelo sistema Ontop está descrita no Anexo E e uma versão simplificada é apresentada no Arquivo 7.11.

Arquivo 7.11: Consulta SQL gerada a partir da consulta SPARQL do Arquivo 7.8

```
1
2
   SELECT *
     FROM (
3
              SELECT
4
                      (': Pesquisador /' || QVIEW1." pesquisador id") AS "researcher
5
                      (':TrabalhoEmEvento/' || QVIEW1."id") AS "artigo",
 6
                      QVIEW1. "titulo " AS "titulo "
 7
                FROM artigo QVIEW1,
8
9
                      evento QVIEW2,
                      pesquisador QVIEW3,
10
11
                      evento QVIEW4
               WHERE (QVIEW1. "pesquisador_id" = QVIEW3. "id") AND
12
                      (QVIEW1." id" = QVIEW4." artigo_id") AND
13
                      (QVIEW1."id" = QVIEW2."artigo_id")
14
               UNION
15
              SELECT (': Pesquisador/' || QVIEW1." pesquisador_id") AS "
16

→ researcher ",

                      (':ArtigoPublicado/' || QVIEW1."id") AS "artigo",
17
                      QVIEW1. "titulo" AS "titulo"
18
                FROM artigo QVIEW1,
19
20
                      revista QVIEW2,
21
                      pesquisador QVIEW3,
22
                      revista QVIEW4
               WHERE (QVIEW1." id" = QVIEW2." artigo id") AND
23
                      (QVIEW1. "pesquisador id" = QVIEW3. "id") AND
24
25
                      (QVIEW1." id" = QVIEW4." artigo id")
26
            ) SUB QVIEW
```

## 7.3.2.1 Execução da Consulta SPARQL pelo OntoSQL

Nesta seção, é descrito o processo de construção da consulta SQL para a consulta SPARQL descrita no Arquivo 7.8 realizado pelo OntoSQL.

A partir da consulta SPARQL (Arquivo 7.8), o grafo da Figura 7.8 é construído.



Figura 7.8: Grafo para consulta SPARQL do Arquivo 7.8

Percorrendo o grafo da Figura 7.8, a consulta SQL é construída. Avaliando o nó referente à variável ?pesquisador é possível saber pela relação publicou definida na ontologia que a variável receberá valores de instâncias referentes à classe Pesquisador. O sistema OntoSQL procura então pelo mapeamento referente à classe Pesquisador. Como mostrado na Figura 7.9, o mapeamento indica que Pesquisador está mapeada para a classe Pesquisador da linguagem Ruby.

Seguindo para o próximo nó representando a variável ?artigo, sabemos que esta variável refere-se à instâncias da classe Artigo descrita pela ontologia. O atributo título também pertence à classe

42 Ontosql 7.3

Figura 7.9: Mapeamento para classe Pesquisador

Artigo. A Figura 7.10 mostra o mapeamento entre a classe Artigo e seus atributos com a classe Artigo modelada na linguagem Ruby.

```
OntoMap mapping 'Artigo' do
model Artigo
maps from: 'natureza', to: :natureza
maps from: 'titulo', to: :titulo
maps from: 'anoDoArtigo', to: :ano_do_artigo
maps from: 'idioma', to: :idioma
maps from: 'meioDeDivulgacao', to: :meio_de_divulgacao
maps from: 'homePageDoTrabalho', to: :home_page_do_trabalho
maps from: 'flagRelevancia', to: :flag_relevancia
maps from: 'doi', to: :doi
maps from: 'tituloDoArtigoIngles', to: :titulo_do_artigo_ingles
maps from: 'flagDivulgacaoCientifica', to: :flag_divulgacao_cientifica
maps from: 'volume', to: :volume
```

Figura 7.10: Mapeamento para classe Artigo

Com as informações obtidas pelo mapeamento, sabemos que a consulta SQL acessará as tabelas Pesquisador e Artigo. Utilizando as chamadas de métodos disponíveis no Framework Active Record do Ruby é construído uma consulta SQL combinando as tabelas Pesquisador e Artigo através da operação JOIN do SQL. No Arquivo 7.12 estão descritas as chamada de método construída para o Framework Active Record acompanhada da consulta SQL resultante.

Arquivo 7.12: Chamada de método e consulta SQL geradas.

```
Pesquisador.joins(:artigo).select(pesquisador.id, artigo.id, artigo.titulo)

SELECT pesquisador.id, artigo.id, artigo.titulo

FROM pesquisador INNER JOIN artigo

ON artigo.pesquisador_id = pesquisador.id
```

É possível ver então através deste exemplo, que existem casos onde o OntoSQL produz consultas sem os JOINS e UNIONS desnecessários apresentados pela consulta gerada pelo Ontop.

## Capítulo 8

## Conclusões

## 8.1 Considerações finais

Este trabalho abordou o problema de acessar dados armazenados em sistemas de bancos de dados NoSQL por meio de ontologias. Muitos trabalhos foram feitos para tornar possível o acesso aos dados armazenados em sistemas de bancos de dados relacionais utilizando ontologias. Com o surgimento de sistemas de bancos de dados NoSQL, um novo cenário surgiu para o acesso a dados baseado em ontologias.

Ong et al. (2014) afirma que diferentemente dos bancos de dados relacionais que possuem uma linguagem de consulta universal (SQL), os sistemas de bancos de dados NoSQL dispõem de inúmeras linguagens de consulta. Consequentemente, são necessárias soluções menos dependentes da linguagem de consulta.

Esta dissertação propôs um novo método para a construção de sistemas OBDA. Ao invés de mapeamentos entre triplas RDF para sub-consultas do SGBD, o mapeamento proposto conecta conceitos da ontologia com entidades de um modelo de objetos. Com este novo mapeamento, não é necessário fazer otimizações nas consultas do SGBD. Este ponto é tido como uma grande vantagem, pois, como mencionado por Calvanese et al. (2017), essas otimizações de consultas SQL são processos custosos. Além disso, permite a criação de um mapeamento menos verboso, pois não há a necessidade de definir comandos de consulta para cada um dos atributos da ontologia.

Para avaliar o funcionamento deste método, foi utilizado como estudo de caso o domínio voltado aos dados curriculares de pesquisadores. No experimento foram utilizadas ontologias descritas em RDF e RDFS, que são os formatos atualmente suportados pelo protótipo. Essas ontologias descrevem classes, subclasses, propriedades que descrevem relações entre classes e atributos de classes. Com este estudo de caso, foi possível utilizar o método para construir duas instâncias do sistema OBDA, uma utilizando um sistema de bancos de dados orientado a documentos (MongoDB) e outra utilizando um sistema de bancos de dados relacional (PostgreSQL) que são capazes de traduzir as consultas SPARQL e produzir um grafo RDF com os dados presentes no banco de dados. E diferente do trabalho de Botoeva et al. (2016), onde a arquitetura do sistema OBDA relacional é diferente da arquitetura para OBDA NoSQL, a arquitetura construída nesta dissertação é a mesma para os sistemas de bancos de dados relacional e NoSQL.

O protótipo Onto<br/>Mongo faz consultas somente para documentos da mesma coleção implementando somente os comandos de <br/> project, match, unwind e filter do framework de consulta aggregation pipeline. Onto<br/>Mongo e Onto<br/>SQL ainda não dão suporte para mapeamentos que necessitem de filtros, como por exemplo no caso do mapeamento do Arquivo 3.1 para o conceito de estágio 4 onde o mapeamento utiliza filtro na consulta SQL. A implementação desta funcionalidade pode ser disponíveis em versões futuras dos protótipos.

Com a construção do OntoMongo é possível verificar que ainda existe muitos desafios para se utilizar sistemas OBDA com bancos NoSQL. É importante ressaltar os sistemas NoSQL possuem diversas características que não foram tratadas nesta dissertação. Da mesma forma, existem outras características do sistema MongoDB que não foram tratados neste trabalho.

## 44 CONCLUSÕES

Este trabalho conseguiu disponibilizar dois protótipos de sistema OBDA OntoMongo e OntoSQL. Apresentando mapeamento mais simples do que o mapeamento utilizado na literatura.

## 8.2 Sugestões para pesquisas futuras

Como próximos passos, alguns pontos deste trabalho merecem pesquisas mais aprofundadas:

- Criação de instâncias utilizando o método proposto neste trabalho para outros sistemas de banco de dados NoSQL pertencentes a outras famílias de bancos de dados, que contenham características diferentes do SGBD MongoDB;
- Ampliar o suporte às consultas SPARQL, incluindo os comandos Optional e Order by;
- Viabilizar a utilização de ontologias mais expressivas no modelo proposto, como OWL;
- Disponibilizar o método proposto como uma biblioteca para desenvolvimento, para que esta solução possa ser utilizada em outros projetos;
- Automatizar a construção da camada intermediária (modelo de objetos) para bancos relacionais;
- Através do trabalho feito por Curé et al. (2013), automatizar a construção do modelo de objetos para bancos de dados NoSQL;
- Analisar em detalhe a diferença de desempenho entre as consultas SQL geradas pelo OntoSQL e pelo Ontop, verificando em quais caso uma das ferramentas produz consultas melhores do que a outra.

## Apêndice A

# Arquivo OWL para ontologia de câncer de pulmão

**Arquivo A.1:** Ontologia de câncer de pulmão.

```
<?xml version="1.0"?>
<!DOCTYPE Ontology [
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
<Ontology xmlns="http://www.w3.org/2002/07/owl#"</pre>
     xml: base="http://exemplo/hospital"
     xmlns: rdf = "http://www.w3.org/1999/02/22 - rdf - syntax - ns \#"
     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\#"
     ontologyIRI="http://exemplo/hospital">
    <Prefix name="" IRI="http://exemplo/hospital#"/>
    <Pre>refix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>
   < Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
   <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>
   <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/>
    <Declaration>
        < Class IRI="#Cancer"/>
    </Declaration>
    <Declaration>
        < Class IRI="#CancerDePulmao"/>
    </Declaration>
    <Declaration>
        < Class IRI="#Estagio"/>
    </Declaration>
    <Declaration>
        <Class IRI="#NSCLC"/>
    </Declaration>
    <Declaration>
        < Class IRI="#Neoplasia"/>
    </Declaration>
    <Declaration>
        < Class IRI="#NeoplasiaBenigna"/>
    </Declaration>
    <Declaration>
        < Class IRI="#NeoplasiaMaligna"/>
    </Declaration>
    <Declaration>
        < Class IRI="#Paciente"/>
    </Declaration>
    <Declaration>
```

```
< Class IRI="#Pessoa"/>
</Declaration>
<Declaration>
    < Class IRI="#SCLC"/>
</Declaration>
<Declaration>
    <ObjectProperty IRI="#temEstagio"/>
</Declaration>
<Declaration>
    <ObjectProperty IRI="#temNeoplasia"/>
</Declaration>
<Declaration>
    <DataProperty IRI="#temNome"/>
</Declaration>
<SubClassOf>
    < Class IRI="#Cancer"/>
    <Class IRI="#NeoplasiaMaligna"/>
</SubClassOf>
<SubClassOf>
    < Class IRI="#CancerDePulmao"/>
    < Class IRI="#Cancer"/>
</SubClassOf>
<SubClassOf>
    < Class IRI="#NSCLC"/>
    < Class IRI="#CancerDePulmao"/>
</SubClassOf>
<SubClassOf>
    < Class IRI="#NeoplasiaBenigna"/>
    < Class IRI="#Neoplasia"/>
</SubClassOf>
<SubClassOf>
    < Class IRI="#NeoplasiaMaligna"/>
    < Class IRI="#Neoplasia"/>
</SubClassOf>
<SubClassOf>
    < Class IRI="#Paciente"/>
    <Class IRI="#Pessoa"/>
</SubClassOf>
<SubClassOf>
    < Class IRI="#SCLC"/>
    < Class IRI="#CancerDePulmao"/>
</SubClassOf>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#temEstagio"/>
    < Class IRI="#Neoplasia"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#temNeoplasia"/>
    < Class IRI="#Paciente"/>
</ObjectPropertyDomain>
<ObjectPropertyRange>
    <ObjectProperty IRI="#temEstagio"/>
    < Class IRI="#Estagio"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
    <ObjectProperty IRI="#temNeoplasia"/>
    < Class IRI="#Neoplasia"/>
</ObjectPropertyRange>
<DataPropertyDomain>
    < Data Property IRI = "#temNome"/>
    < Class IRI="#Pessoa"/>
</DataPropertyDomain>
<DataPropertyRange>
    < DataProperty IRI="#temNome"/>
    <Datatype abbreviatedIRI="xsd:string"/>
```

## Apêndice B

# Mapeamento do OntoSQL

#### Arquivo B.1: Mapeamento Onto SQL

```
OntoMap.mapping 'Pesquisador' do
      model Pesquisador
      maps from: 'nomeCompleto', to: :nome
      maps from: 'nacionalidade', to: :nacionalidade
      maps from: 'paisDeNascimento', to: :pais de nascimento
      maps from: 'ufNascimento', to: :uf_nascimento
 6
      maps from: 'siglaPaisNacionalidade', to: :sigla_pais_nacionalidade
 7
      maps from: 'paisDeNacionalidade', to: :pais_de_nacionalidade
 8
      maps from: 'nomeEmCitacoesBibliograficas', to::
           → nome_em_citacoes_bibliograficas
10
      maps from: 'textoResumoCvRh', to: :texto resumo cv rh
11
      maps \hspace{0.2cm} \textbf{from}: \hspace{0.2cm} `textoResumoRvRhEn' \hspace{0.2cm} , \hspace{0.2cm} \textbf{to}: \hspace{0.2cm} :texto\_resumo\_cv\_rh\_en \\
12
      maps from: 'permissaoDeDivulgacao', to: :permissao_de_divulgacao
13
      maps from: 'numeroIdentificador', to: :numero identificador
14
15
    OntoMap. mapping 'Artigo' do
16
       model Artigo
      maps from: 'natureza', to: :natureza
maps from: 'titulo', to: :titulo
17
18
      maps from: 'anoDoArtigo', to: :ano_do_artigo
19
      maps from: 'idioma', to: :idioma
20
      maps \hspace{0.2cm} \textbf{from} : \hspace{0.2cm} \verb"meioDeDivulgacao", \hspace{0.2cm} \textbf{to} : \hspace{0.2cm} \verb"meio_de_divulgacao"
21
      maps from: 'homePageDoTrabalho', to: :home page do trabalho
22
23
      maps from: 'flagRelevancia', to: :flag relevancia
      maps from: 'doi', to: :doi
24
25
      maps from: 'tituloDoArtigoIngles', to: :titulo do artigo ingles
26
      maps from: 'flag Divulgacao Cientifica', to: :flag divulgacao cientifica
      maps from: 'volume', to: :volume
27
      maps from: 'fasciculo', to: :fasciculo
28
29
      maps from: 'serie', to: :serie
30
      maps from: 'paginaInicial', to: :pagina_inicial
31
      maps from: 'paginaFinal', to: :pagina_final
32
      maps from: 'tipo', to: :tipo
33
    OntoMap. mapping 'ArtigoEmRevista' do
34
35
       model Revista
36
      maps from: 'tituloDoPeriodicoOuRevista', to: :
      \begin{array}{c} \hookrightarrow \  \, titulo\_do\_periodico\_ou\_revista \\ maps \  \, \begin{array}{c} from: \  \, 'issn\ ', \  \, to: :issn \end{array} \end{array}
37
      maps from: 'localDePublicacao', to: :local_de_publicacao
38
      maps from: 'pais', to: :pais
39
40
    OntoMap.mapping 'ArtigoEmEvento' do
41
42
      model Evento
      maps from: 'classificacao Do Evento', to: :classificacao do evento
43
      maps from: 'nomeDoEvento', to: :nome do evento
```

```
45
       maps from: 'cidadeDoEvento', to: :cidade_do_evento
       maps from: 'anoDeRealizacao', to: :ano de realizacao
46
       maps from: 'tituloDosAnaisOuProceedings', to: :
47
            \hookrightarrow \ titulo\_dos\_anais\_ou\_proceedings
       maps from: 'isbn', to: :isbn
48
       maps from: 'nomeDaEditora', to: :nome_da_editora
49
       maps from: 'cidadeDaEditora', to: :cidade_da_editora
50
51
       maps from: 'pais', to: :pais
52
    end
53
    OntoMap. mapping 'Coautor' do
54
       model Coautor
       maps from: 'nomeCompletoDoAutor', to::nome_completo_do_autor
maps from: 'nomeParaCitacao', to::nome_para_citacao
maps from: 'ordemDeAutoria', to::ordem_de_autoria
maps from: 'nroIdCnpq', to::nro_id_cnpq
55
56
57
58
59
    Onto Map.\, mapping \quad 'Area Do Conhecimento' \quad do
60
       model \ Area Do Conhecimento
61
       maps \hspace{0.2cm} \textbf{from}: \hspace{0.2cm} "nomeGrandeAreaDoConhecimento", \hspace{0.2cm} \textbf{to}: \hspace{0.2cm} :
62
           → nome grande area do conhecimento
       maps from: 'nomeDaAreaDConhecimento', to: :nome_da_area_do_conhecimento
63
       maps from: 'nomeDaSubAreaDoConhecimento', to: :
64
           \hookrightarrow nome da sub area do conhecimento
65
       maps from: 'nomeDaEspecialidade', to: :nome_da_especialidade
66
```

## Apêndice C

# Mapeamento Ontop

#### Arquivo C.1: Mapeamento Ontop

```
[PrefixDeclaration]
      : http://www.onto.org/obda/basic-lattes-final#
 3 owl: http://www.w3.org/2002/07/owl#
 4 rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
     xsd: http://www.w3.org/2001/XMLSchema#
 6
       rdfs: http://www.w3.org/2000/01/rdf-schema\#
 7
       [SourceDeclaration]
 8
 9
       \mathbf{sourceUri}
                                             ontosal
                                             jdbc:postgresql://localhost/myapp development
10 connectionUrl
      username
                                             Barbara
11
12
       password
                                             org.postgresql.Driver
13
      driverClass
14
15
        [MappingDeclaration] @collection [[
16
       mappingId mapping-Pesquisador
17
        target : Pesquisador / {id} rdf: type : Pesquisador .
        source select id from pesquisador
18
19
                                               {\it mapping-Pesquisador-nomeCompleto}
20
       target : Pesquisador/{id} : nomeCompleto {nome}^^xsd: string .
21
22
       source select id, nome from pesquisador;
                                               mapping-Pesquisador-nacionalidade
       target : Pesquisador/{id} : nacionalidade {nacionalidade}^^xsd: string .
26
       source select id, nacionalidade from pesquisador;
27
28
       mappingId
                                               mapping-Pesquisador-paisDeNascimento
       {\bf target} : Pesquisador/\{id\} : paisDeNascimento \ \{pais\_de\_nascimento\} ^ \hat xsd: string
29
        source select id, pais_de_nascimento from pesquisador;
30
31
32
       mappingId
                                               mapping-Pesquisador-ufNascimento
33
        target : Pesquisador/{id} : ufNascimento {uf nascimento}^^xsd:string .
34
        source select id, uf nascimento from pesquisador;
35
36
       mappingId
                                               mapping-Pesquisador-siglaPaisNacionalidade
37
        target : Pesquisador / {id} : sigla Pais Nacionalidade { sigla pais nacionalidade
                → }^^xsd:string
38
        source select id, sigla_pais_nacionalidade from pesquisador;
39
40
       mappingId
                                               mapping-Pesquisador-paisDeNacionalidade
        {\bf target} \ : Pesquisador/\{id\} \ : paisDeNacionalidade \ \{pais\_de\_nacionalidade\} ^ ?xsd : Pesquisador/\{id\} \ : PaisDeNacionalidade \ \{pais\_de\_nacionalidade\} ^ ?xsd : Pesquisador/\{id\} \ : PaisDeNacionalidade \ \{pais\_de\_nacionalidade\} ^ ?xsd : Pesquisador/\{id\} \ : PaisDeNacionalidade \ \{pais\_de\_nacionalidade\} ^ ?xsd : Pesquisador/\{id\} \ : PaisDeNacionalidade \ \{pais\_de\_nacionalidade\} ^ ?xsd : Pesquisador/\{id\} \ : PaisDeNacionalidade \ \{pais\_de\_nacionalidade\} ^ ?xsd : Pesquisador/\{id\} \ : PaisDeNacionalidade \ \{pais\_de\_nacionalidade\} ^ ?xsd : Pesquisador/\{id\} \ : PaisDeNacionalidade \ \{pais\_de\_nacionalidade\} ^ ?xsd : Pesquisador/\{id\} \ : PaisDeNacionalidade \ \{pais\_de\_nacionalidade\} ^ ?xsd : Pesquisador/(id) \ : PaisDeNacionalidade \ (pais\_de\_nacionalidade) ^ ?xsd : Pesquisador/(id) \ : PaisDeNacionalidade \ (pais\_de\_nacionalidade) ^ ?xsd : Pesquisador/(id) \ : PaisDeNacionalidade \ (pais\_de\_nacionalidade) ^ ?xsd : Pesquisador/(id) \ : PaisDeNacionalidade \ (pais\_de\_nacionalidade) ^ ?xsd : Pesquisador/(id) \ : PaisDeNacionalidade \ (pais\_de\_nacionalidade) ^ ?xsd : Pesquisador/(id) \ : PaisDeNacionalidade \ (pais\_de\_nacionalidade) ^ ?xsd : Pesquisador/(id) \ : PaisDeNacionalidade \ (pais\_de\_nacionalidade) ^ ?xsd : Pesquisador/(id) \ : PaisDeNacionalidade \ (pais\_de\_nacionalidade) ^ ?xsd : Pesquisador/(id) \ : PaisDeNacionalidade \ (pais\_de\_nacionalidade) ^ ?xsd : Pesquisador/(id) \ : PaisDeNacionalidade \ (pais\_de\_nacionalidade) ^ ?xsd : Pesquisador/(id) \ : PaisDeNacionalidade \ (pais\_de\_nacionalidade) ^ ?xsd : Pesquisador/(id) \ : PaisDeNacionalidade \ (pais\_de\_nacionalidade) ^ ?xsd : Pesquisador/(id) \ : PaisDeNacionalidade \ (pais\_de\_nacionalidade) ^ ?xsd : Pesquisador/(id) \ : PaisDeNacionalidade \ (pais\_de\_nacionalidade) ^ ?xsd : Pesquisador/(id) \ : PaisDeNacionalidade \ (pais\_de\_nacionalidade) ^ ?xsd : Pesquisador/(id) \ : PaisDeNacionalidade \ (pais\_de\_nacionalidade) ^ ?xsd : Pesquisador/(id) \ : Pes
41

→ string

        source select id, pais_de_nacionalidade from pesquisador;
43
```

```
44
   mappingId
                     mapping-Pesquisador-nomeEmCitacoesBibliograficas
   target : Pesquisador / { id } : nomeEmCitacoesBibliograficas {
       → nome em citacoes bibliograficas}^^xsd:string .
46
   source select id, nome em citacoes bibliograficas from pesquisador;
47
                     {\it mapping-Pesquisador-textoResumoCvRh}
48
   mappingId
   target : Pesquisador/{id} :textoResumoCvRh {texto resumo cv rh}^^xsd:string
49
   source select id, texto resumo cv rh from pesquisador;
50
51
52
                     mapping-Pesquisador-textoResumoRvRhEn
   mappingId
   target : Pesquisador / {id} : textoResumoRvRhEn {texto resumo cv rh en}^^xsd:
       \hookrightarrow string.
54
   source select id, texto resumo cv rh en from pesquisador;
55
                     mapping-Pesquisador-permissao De Divulgacao\\
56
   mappingId
   target : Pesquisador/{id} : permissaoDeDivulgacao {permissao_de_divulgacao}^^
57
       \hookrightarrow xsd:string.
   source select id, permissao_de_divulgacao from pesquisador;
58
59
                     {\tt mapping-Pesquisador-numeroIdentificador}
60
   mappingId
   target : Pesquisador/{id} : numeroIdentificador {numero identificador}^^xsd:
61
       \hookrightarrow string.
   source select id, numero identificador from pesquisador;
62
63
64
   mappingId
                     mapping-Pesquisador-publicou-ArtigoPublicado
65
   target : Pesquisador/{id pesquisador} : publicou : ArtigoPublicado/{id artigo}
   source select pesquisador.id as id_pesquisador, revista.id as id_artigo
66
       → from pesquisador join artigo on pesquisador.id =artigo.pesquisador id

→ join revista on artigo.id = revista.artigo id
67
68
   mappingId
                     mapping-Coautor
   target :Coautor/{id} rdf:type :Coautor .
69
   source select "id" from coautor;
70
71
72
   mappingId
                     mapping-Coautor-nomeCompleto
   target :Coautor/{id} :nomeCompleto {nome completo do autor}^^xsd:string .
73
   source select id, nome completo do autor from coautor;
74
75
76
   mappingId
                     mapping-Coautor-nome Para Citacao\\
   target :Coautor/{id} :nomeParaCitacao {nome_para_citacao}^^xsd:string .
77
   source select id, nome para citacao from coautor;
                     mapping-Coautor-ordemDeAutoria
80 mappingId
   target :Coautor/{id} :ordemDeAutoria {ordem de autoria}^^xsd:int .
82
   source select id, ordem de autoria from coautor;
83
                     mapping-Coautor-nroIdCnpq
84
   mappingId
   target : Coautor/{id} :nroIdCnpq {nro id cnpq}^^xsd:string .
85
86
   source select id, nro id cnpq from coautor;
87
88
   mappingId
                     {\it mapping-AreaDoConhecimento}
   target : AreaDeConhecimento / { id } a : AreaDeConhecimento .
89
   source select id from area do conhecimento;
90
91
92
                     mapping-Area Do Conhecimento-nome Grande Area Do Conhecimento\\
   mappingId
93
   target : AreaDeConhecimento/{id} : nomeGrandeAreaDoConhecimento {
       \hookrightarrow nome grande area do conhecimento}^^xsd:string .
   source select id, nome grande area do conhecimento from
94

→ area do conhecimento;

95
                     mapping-AreaDoConhecimento-nomeDaAreaDConhecimento
96
   mappingId
97
   target : AreaDeConhecimento / { id } : nomeDaAreaDConhecimento {
       → nome da area do conhecimento}^^xsd:string .
```

```
source select id, nome_da_area_do_conhecimento from area_do_conhecimento;
100
                                          mapping-Area Do Conhecimento-nome Da Sub Area Do Conhecimento\\
      mappingId
        target : AreaDeConhecimento / { id } : nomeDaSubAreaDoConhecimento {
               → nome da sub area do conhecimento}^^xsd:string .
        source select id, nome_da_sub_area_do_conhecimento from
102

→ area do conhecimento;

103
104
       mappingId
                                          mapping-Area Do Conhecimento-nome Da Especial idade\\
        target :AreaDeConhecimento/{id} :nomeDaEspecialidade {nome da especialidade
105
               \hookrightarrow }^^xsd:string .
106
                       select id, nome da especialidade from area do conhecimento;
        source
107
108
                                         mapping-ArtigoPublicado
109
        target :ArtigoPublicado/{id} rdf:type :ArtigoPublicado .
        source select id from revista
110
111
112
                                         mapping-Pesquisador-publicou-Trabalho Em Evento\\
        mappingId
        {\bf target} : Pesquisador/\{id\_pesquisador\} : publicou : TrabalhoEmEvento/\{id\_artigorum entre of the control of
113
               → } .
114
        source select pesquisador.id as id_pesquisador, evento.id as id_artigo from
               → pesquisador join artigo on pesquisador.id =artigo.pesquisador id
               → join evento on artigo.id = evento.artigo id
115
116 mappingId
                                          mapping-Trabalho Em Evento\\
117
        target :TrabalhoEmEvento/{id} rdf:type :TrabalhoEmEvento .
118
        source select id from evento
119
                                          {\tt mapping-ArtigoPublicado-titulo}
120 mappingId
        {\bf target} : {\rm ArtigoPublicado}/\{{\rm id}\} : {\rm titulo} \ \{{\rm titulo}\} \hat{\ \ } xsd : {\rm string} \ .
121
        source select revista.id as id, artigo.titulo as titulo from artigo join

→ revista on artigo.id = revista.artigo id
123
124
                                          mapping-TrabalhoEmEvento-titulo
        target :TrabalhoEmEvento/{id} :titulo {titulo}^^xsd:string .
126
        source select evento.id as id, artigo.titulo from artigo join evento on
               → artigo.id = evento.artigo id
127
128
                                          mapping-Coautors-publicou-ArtigoPublicado
        mappingId
        target :Coautors/{coautor id} :publicou :ArtigoPublicado/{revista id} .
129
        source select coautor.id as coautor id, revista.id as revista id from
               ⇔ coautor join artigo on coautor.artigo id = artigo.id join revista on
               → artigo.id = revista.artigo id
131
                                          mapping-Coautor-publicou-Trabalho Em Evento\\
132 mappingId
        target : Coautor / {coautor id } : publicou : Trabalho Em Evento / {evento id } .
        source select coautor.id as coautor id, evento.id as evento id from coautor

→ join artigo on coautor.artigo id = artigo.id join evento on artigo.

    → id = evento.artigo id

135
      - 11
```

## Apêndice D

# Esquema de banco de dados

Arquivo D.1: Esquema de banco de dados

```
ActiveRecord::Schema.define(version: 20170629010350) do
3
      enable extension "plpgsql"
4
5
      create table "area de atuacaos", force: :cascade do |t|
6
        t.string
                    "sequencia area de atuacao"
7
                    "nome\_grande\_area\_do\_conhecimento"
        t.string
8
                    "nome_da_area_do_conhecimento"
        t.string
9
                    "nome_da_sub_area_do_conhecimento"
        t.string
10
                    "nome_da_especialidade"
        t.string
                    "pesquisador_id"
11
        t.integer
12
        t.datetime "created_at"
                                                           null: false
13
        t.datetime "updated_at"
                                                           null: false
        t.index ["pesquisador id"], name: "
14
            → index area de atuacaos on pesquisador id", using: :btree
15
16
      create_table "area_do_conhecimentos", force: :cascade do |t|
17
                    "nome\_grande\_area\_do\_conhecimento"
18
        t.string
                    "nome\_da\_area\_do\_conhecimento"
19
        t.string
                    "nome\_da\_sub\_area\_do\_conhecimento"
20
        t.string
                    "nome\_da\_especialidade"
21
        t.string
        t.datetime "created_at"
                                                           null: false
        t.datetime "updated_at",
23
                                                           null: false
                    "artigo id"
        t.integer
25
        t.index ["artigo id"], name: "index area do conhecimentos on artigo id

→ ", using: :btree

26
     end
27
      create_table "artigos", force: :cascade do |t|
28
29
                    "natureza"
        t.string
30
                    "titulo"
        t.string
31
        t.integer
                    "ano_do_artigo"
32
                    "idioma"
        t.string
                    "meio_de_divulgacao"
33
        t.string
                    "home\_page\_do\_trabalho"
34
        t.string
                    "flag_relevancia"
35
        t.boolean
36
        t.string
37
                    "titulo_do_artigo_ingles"
        t.string
                    "flag_divulgacao_cientifica"
38
        t.boolean
                    "volume"
39
        t.string
                    "fasciculo"
40
        t.string
                    " \operatorname{serie} "
41
        t.string
                    "pagina_inicial"
42
        t.integer
                    "pagina_final"
43
        t.integer
        t.string
```

```
t.integer "pesquisador id"
45
         t.datetime "created_at",
                                                     null: false
46
         t.datetime "updated at",
47
                                                     null: false
         t.index ["pesquisador id"], name: "index artigos on pesquisador id",
48
             → using: :btree
49
      end
50
51
       create_table "atuacao_profissionals", force: :cascade do |t|
52
                     "codigo_instituicao"
         t.string
53
                     "nome instituicao"
         t.string
54
         t.integer
                     "\,sequencia\_\,ativid\,ad\,e\,"
55
         t.integer
                     "sequencia\_importancia"
         t.integer
56
                     "pesquisador_id"
57
         t.datetime "created_at",
                                                null: false
         t.datetime "updated at",
58
                                                null: false
         t.index ["pesquisador_id"], name: "
59
             → index_atuacao_profissionals_on_pesquisador_id", using: :btree
60
      end
61
       create table "coautors", force: :cascade do |t|
62
                     "nome\_completo\_do\_autor"
63
         t.string
                     "nome_para_citacao"
64
         t.string
                     "ordem de autoria"
65
         t.integer
66
                     "nro id cnpq"
         t.string
67
         t.datetime "created at",
                                                 null: false
         t.datetime "updated_at",
68
                                                 null: false
         t.integer "artigo_id"
69
70
         t.index ["artigo id"], name: "index coautors on artigo id", using: :
             → btree
71
      end
72
73
      create_table "enderecos", force: :cascade do |t|
                     "flag de preferencia"
74
         t.string
75
                     "codigo instituicao empresa"
         t.string
76
                     "nome instituicao empresa"
         t.string
77
                     "codigo unidade"
         t.string
                     "nome unidade"
78
         t.string
                     "codigo_orgao"
79
         t.string
                     "nome\_orgao"
80
         t.string
                     "pais"
81
         t.string
                     " u f "
82
         t.string
                     "logradouro\_complemento"
83
         t.string
                     "bairro"
84
         t.string
                     "cidade"
85
         t.string
                    "\,caixa\_postal"
86
         t.string
                     "cep"
87
         t.string
                     "ddd"
88
         t.string
                     "telefone"
89
         t.string
                     "ramal"
90
         t.string
91
                     " fax "
         t.string
92
         t.string
                     "home\_page"
93
         t.integer
                    "pesquisador_id"
94
         t.datetime "created at",
                                                     null: false
         t.datetime "updated at".
95
                                                     null: false
         t.index ["pesquisador id"], name: "index enderecos on pesquisador id",
96
             → using: :btree
97
      end
98
       create_table "eventos", force: :cascade do |t|
99
                     "\,classificacao\,\_\,do\,\_\,evento\,"
100
         t.string
101
                     "nome\_do\_evento"
         t.string
102
                     "cidade do evento"
         t.string
103
                     "ano de realização"
         t.integer
                     "titulo_dos_anais_ou_proceedings"
104
         t.string
105
         t.string
                     "isbn"
```

```
106
         t.string
                     "nome da editora"
                     "cidade da editora"
107
         t.string
                     "pais"
108
         t.string
                    "artigo_id"
109
         t.integer
         t.datetime "created at",
110
                                                           null: false
         t.datetime "updated_at",
111
                                                           null: false
         t.index ["artigo_id"], name: "index_eventos_on_artigo_id", using: :
112
            → btree
113
      end
114
115
       create_table "formacao_academicas", force: :cascade do |t|
116
         t.string
                     "tipo"
117
         t.string
                     "titulo_do_trabalho"
118
         t.string
                     "nome_do_orientador"
119
         t.string
                     "nome instituicao"
                     "nome_curso"
120
         t.string
                     "status_do_curso"
121
         t.string
                     "\,ano\_de\_inicio"
122
         t.integer
                     "ano_de_conclusao"
123
         t.integer
                    "flag bolsa"
124
         t . boolean
                     "nome_agencia"
125
         t.string
                     "codigo_curso_capes"
126
         t.string
         t.integer "ano de obtencao do titulo"
127
                    "nome completo do_orientador"
128
         t.string
         t.integer "pesquisador id"
129
         t.datetime "created_at",
130
                                                      null: false
131
         t.datetime "updated at",
                                                      null: false
         132

→ index_formacao_academicas_on_pesquisador_id", using: :btree

133
      end
134
135
       create table "idiomas", force: :cascade do |t|
                     "idioma"
136
         t.string
                     "descricao_do_idioma"
137
         t.string
138
                     "proficiencia_de_leitura"
         t.string
                    "proficiencia_de_fala"
"proficiencia_de_escrita"
"proficiencia_de_compreensao"
139
         t.string
140
         t.string
141
         t.string
                     "pesquisador id"
142
         t.integer
         t.datetime "created at"
143
                                                      null: false
         t.datetime "updated at",
                                                      null: false
144
         t.index ["pesquisador_id"], name: "index_idiomas_on_pesquisador_id",
145
            → using: :btree
146
147
148
       create table "keywords", force: :cascade do |t|
                    "descricao"
149
         t.string
         t.datetime "created at", null: false
150
         {\tt t.datetime} \ "updated\_at", \ null: \ false
151
152
         t.integer "artigo id"
153
         t.index ["artigo id"], name: "index keywords on artigo id", using: :
            → btree
154
      end
155
       create table "pesquisadors", force: :cascade do |t|
156
157
         t.string
                     "nome"
                     "nome_em_citacoes_bibliograficas"
158
         t.string
                     "texto\_resumo\_cv\_rh"
159
         t.string
         t.datetime "created_at",
                                                           null: false
160
         t.datetime "updated at",
                                                           null: false
161
                     "texto_resumo_cv_rh_en"
162
         t.string
                     "nacionalidade"
163
         t.string
164
                     "pais_de_nascimento"
         t.string
165
                    "uf nascimento"
         t.string
166
                     "cidade nascimento"
         t.string
```

```
167
         t.string
                      "permissao de divuldacao"
168
                      "sigla pais nacionalidade"
         t.string
                      "pais de nacionalidade"
169
         t.string
                      "permissao_de_divulgacao"
170
         t.string
                      "numero identificador"
171
         t.string
172
       end
173
       create_table "revista", force: :cascade do |t|
174
175
                      "titulo_do_periodico_ou_revista"
         t.string
176
                      "issn"
         t.string
177
         t.string
                      "local_de_publicacao"
                      "pais"
178
         t.string
                      "artigo_id"
179
         t.integer
180
         t.datetime "created_at",
                                                             null: false
         t.datetime "updated at",
181
                                                             null: false
         t.index ["artigo_id"], name: "index_revista_on_artigo_id", using: :
182
             → btree
183
       end
184
185
       create table "vinculos", force: :cascade do |t|
                      "sequencia historico"
186
         t.integer
                      "tipo de vinculo"
187
         t.string
                      "enquadramento funcional"
188
         t.string
189
         t.integer
                      "carga horaria semanal"
190
         t.boolean
                      "flag_dedicacao_exclusiva"
                      "\,mes\_inicio"
191
         t.integer
                      "ano_inicio"
192
         t.integer
193
                      "mes fim"
         t.integer
194
                      "ano_fim"
         t.integer
195
                      "outras informacoes"
         t.string
196
                      "flag\_vinculo\_empregaticio"
         t.boolean
                      "outro_vinculo_informado"
197
         t.string
198
                      "outro\_enquadramento\_funcional\_informado"
         t.string
199
                      "outro enquadramento funcional informado ingles"
         t.string
                      "outras informacoes_ingles"
200
         t.string
                      "\,atu\,ac\,ao\,\_\,profission\,al\,\_\,id\,"
201
         t.integer
         t.datetime "created at",
202
                                                                                null:

    false

         t.datetime "updated at",
203
                                                                                null:

→ false

204
         t.index ["atuacao profissional id"], name: "
             → index vinculos on atuacao profissional id", using: :btree
205
       end
206
207
       add foreign key "area de atuacaos", "pesquisadors"
       add foreign key "area do conhecimentos", "artigos"
208
       add foreign key "artigos", "pesquisadors"
209
       add foreign key "atuacao profissionals", "pesquisadors"
210
       add_foreign_key "coautors", "artigos" add_foreign_key "enderecos", "pesquisadors"
211
212
       \verb|add_foreign_key| "eventos", "artigos"|
213
214
       \verb| add_foreign_key "formacao_academicas", "pesquisadors"|
       add_foreign_key "idiomas", "pesquisadors" add_foreign_key "keywords", "artigos" add_foreign_key "revista", "artigos"
215
216
217
       add foreign key "vinculos", "atuacao profissionals"
218
219
```

#### Apêndice E

#### Consulta SQL gerada pelo sistema Ontop

Arquivo E.1: Consulta SQL gerada pelo Sistema Ontop

```
1 SELECT *
       2 FROM (
      3 SELECT
                                                               1 AS "researcherQuestType", NULL AS "researcherLang", ('http://www.onto.
                                                                                                  → org/obda/basic-lattes-final#Pesquisador/' || REPLACE(REPLACE(
                                                                                                 → REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE)
                                                                                                 → REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REP
                                                                                                 \hookrightarrow REPLACE(CAST(QVIEW1." pesquisador_id" AS VARCHAR(10485760)),''

→ researcher ",

                                                             1 AS "artigoQuestType", NULL AS "artigoLang", ('http://www.onto.org/obda \hookrightarrow /basic-lattes-final#TrabalhoEmEvento/' || REPLACE(REPLACE(REPLACE(REPLACE)
                                                                                                 \hookrightarrow \ \text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REP
                                                                                                 \rightarrow \ \text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REP
                                                                                              → CAST(QVIEW1." id" AS VARCHAR(10485760)), '', '\%20'), '!', '\%21'), '

⊕ @', '\%40'), '#', '\%23'), '$', '\%24'), '&', '\%26'), '*', '\%42'),

→ '(', '\%28'), ')', '\%29'), '[', '\%5B'), ']', '\%5D'), ',', '\%2C

→ '), ';', '\%3B'), ':', '\%3A'), '?', '\%3F'), '=', '\%3D'), '+',

→ '\%2B'), '''', '\%22'), '/', '\%2F')) AS "artigo",
                                                               7 AS "tituloQuestType", NULL AS "tituloLang", QVIEW1." titulo" AS "titulo
       6
                                        FROM
                               artigo QVIEW1,
                               evento QVIEW2,
10
                                  pesquisador QVIEW3,
                                  evento QVIEW4
11
12
                                  (QVIEW1." id" = QVIEW2." artigo id") AND
13
                               QVIEW1." id" IS NOT NULL AND
14
                               QVIEW1." titulo" IS NOT NULL AND
15
                                  (QVIEW1. "pesquisador_id" = QVIEW3. "id") AND
                                   (QVIEW1." id" = QVIEW4." artigo id") AND
17
                               QVIEW1. "pesquisador id" IS NOT NULL
18
                              UNION
19
20
                               SELECT
                                                                1 AS "researcherQuestType", NULL AS "researcherLang", ('http://www.onto.
21

→ org/obda/basic-lattes-final#Pesquisador/' || REPLACE(REPLACE)
                                                                                                 \rightarrow \ \text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REP
                                                                                                 \rightarrow \ \text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REP
                                                                                                 \hookrightarrow \texttt{REPLACE}(\underbrace{\texttt{CAST}(\texttt{QVIEW1."pesquisador\_id"}} \texttt{AS VARCHAR}(10485760)) \;, ` \; ',
```

```
\hookrightarrow '), '=', '\%3D'), '+', '\%2B'), '''', '\%22'), '/', '\%2F')) AS "

→ researcher ",

                                                           22
                                                                                            \hookrightarrow \ \text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REP
                                                                                          \hookrightarrow \ \text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REPLACE}(\text{REP
                                                         → CAST(QVIEW1."id" AS VARCHAR(10485760)), '', '\%20'), '!', '\%21'), '

→ @', '\%40'), '#', '\%23'), '$', '\%24'), '&', '\%26'), '*', '\%42'),

→ '(', '\%28'), ')', '\%29'), '[', '\%5B'), ']', '\%5D'), ',', '\%2C

→ '), ';', '\%3B'), ':', '\%3A'), '?', '\%3F'), '=', '\%3D'), '+',

→ '\%2B'), '''', '\%22'), '/', '\%2F')) AS "artigo",

7 AS "tituloQuestType", NULL AS "tituloLang", QVIEW1." titulo" AS "titulo
 23
 24
                             FROM
 25 artigo QVIEW1,
 26 revista QVIEW2,
 27 pesquisador QVIEW3,
 28
                           revista QVIEW4
 29 WHERE
30 (QVIEW1." id" = QVIEW2." artigo id") AND
 31 QVIEW1." id" IS NOT NULL AND
 32 QVIEW1." titulo" IS NOT NULL AND
 33 (QVIEW1." pesquisador id" = QVIEW3." id") AND
 34 (QVIEW1." id" = QVIEW4." artigo id") AND
 35 QVIEW1." pesquisador id" IS NOT NULL
36 ) SUB_QVIEW
```

#### Apêndice F

# Estrutura do arquivo JSON armazenado no MongoDB

#### Apêndice G

## Estrutura do arquivo XML extraído pelo ScriptLattes

```
<?xml version="" encoding="" standalone=""?>
<CURRICULO-VITAE SISTEMA-ORIGEM-XML="" DATA-ATUALIZACAO="" HORA-ATUALIZACAO
   → ="" NUMERO-IDENTIFICADOR="">
    <DADOS-GERAIS
        NOME-COMPLETO=""
        \hbox{NOME-EM-CITACOES-BIBLIOGRAFICAS=""}\\
        NACIONALIDADE=""
        PAIS-\!DE\!-\!NASCIMENTO\!=""
        UF-NASCIMENTO=""
        CIDADE-NASCIMENTO=""
        PERMISSAO-DE-DIVULGACAO=""
        DATA-FALECIMENTO=""
        SIGLA-PAIS-NACIONALIDADE=""
        PAIS-DE-NACIONALIDADE="">
        <RESUMO-CV
          TEXTO-RESUMO-CV-RH=""
          TEXTO-RESUMO-CV-RH-EN=""/>
          <OUTRAS-INFORMACOES-RELEVANTES OUTRAS-INFORMACOES-RELEVANTES=""/>
        <ENDERECO FLAG-DE-PREFERENCIA="">
          <ENDERECO-PROFISSIONAL
            CODIGO-INSTITUICAO-EMPRESA=""
            NOME-INSTITUICAO-EMPRESA=""
            CODIGO-UNIDADE=""
            NOME-UNIDADE=""
            CODIGO-ORGAO=""
            NOME-ORGAO=""
            PAIS=""
            UF=""
            LOGRADOURO-COMPLEMENTO=""
            BAIRRO=""
            CIDADE=""
            CAIXA-POSTAL=""
            CEP=""
            DDD=""
            TELEFONE=""
            RAMAL=""
            \mathrm{FAX}{=}""
            HOME-PAGE=""/>
        </ENDERECO>
        <FORMACAO-ACADEMICA-TITULACAO>
          < GRADUACAO
            SEQUENCIA-FORMACAO=""
            TITULO-DO-TRABALHO-DE-CONCLUSAO-DE-CURSO=""
            NOME-DO-ORIENTADOR=""
```

```
CODIGO-INSTITUICAO=""
    NOME-INSTITUICAO=""
    CODIGO-CURSO=""
    NOME-CURSO=""
    CODIGO-AREA-CURSO=""
    STATUS-DO-CURSO=""
    ANO-DE-INICIO=""
    ANO-DE-CONCLUSAO=""
    FLAG-BOLSA=""
    TIPO-GRADUACAO=""
    CODIGO-INSTITUICAO-GRAD=""
    NOME-INSTITUICAO-GRAD=""
    CODIGO-INSTITUICAO-OUTRA-GRAD=""
    NOME-INSTITUICAO-OUTRA-GRAD=""
    NOME-ORIENTADOR-GRAD=""
    CODIGO-AGENCIA-FINANCIADORA=""
    NOME-AGENCIA=""
    NUMERO\!\!-\!ID\!-\!\!ORIENTADOR\!\!=\!""
    \hbox{CODIGO-CURSO-CAPES}="""
    TITULO-DO-TRABALHO-DE-CONCLUSAO-DE-CURSO-INGLES=""
    NOME-CURSO-INGLES=""/>
    <MESTRADO
        SEQUENCIA-FORMACAO=""
        NIVEL=""
        CODIGO-INSTITUICAO=""
        NOME-INSTITUICAO=""
        CODIGO-CURSO=""
        NOME-CURSO=""
        CODIGO-AREA-CURSO=""
        STATUS-DO-CURSO=""
        ANO-DE-INICIO=""
        ANO-DE-CONCLUSAO=""
        FLAG-BOLSA=""
        TIPO-MESTRADO=""
        CODIGO-INSTITUICAO-DOUT=""
        NOME-INSTITUICAO-DOUT=""
        CODIGO-INSTITUICAO-OUTRA-DOUT=""
        NOME-INSTITUICAO-OUTRA-DOUT=""
        NOME-ORIENTADOR-DOUT=""
        CODIGO-AGENCIA-FINANCIADORA=""
        NOME-AGENCIA=""
        ANO-DE-OBTENCAO-DO-TITULO=""
        TITULO-DA-DISSERTACAO-TESE=""
        NOME-COMPLETO-DO-ORIENTADOR=""
        NUMERO-ID-ORIENTADOR=""
        CODIGO-CURSO-CAPES=""
        TITULO-DA-DISSERTACAO-TESE-INGLES=""
        NOME-CURSO-INGLES=""
        NOME-DO-CO-ORIENTADOR="">
        <PALAVRAS-CHAVE PALAVRA-CHAVE-1="" PALAVRA-CHAVE-2=""
            → PALAVRA-CHAVE-3="" PALAVRA-CHAVE-4="" PALAVRA-CHAVE
            \hookrightarrow -5="" PALAVRA-CHAVE-6=""/>
        <AREAS-DO-CONHECIMENTO>
          <AREA-DO-CONHECIMENTO-1 NOME-GRANDE-AREA-DO-CONHECIMENTO</p>
              → ="" NOME-DA-AREA-DO-CONHECIMENTO="" NOME-DA-SUB-
              → AREA-DO-CONHECIMENTO="" NOME-DA-ESPECIALIDADE=""/><
              → AREA-DO-CONHECIMENTO-2 NOME-GRANDE-AREA-DO-
              → CONHECIMENTO="" NOME-DA-AREA-DO-CONHECIMENTO=""
              → NOME-DA-SUB-AREA-DO-CONHECIMENTO="" NOME-DA-
              → ESPECIALIDADE=""/></AREAS-DO-CONHECIMENTO><SETORES-
              \hookrightarrow DE-ATIVIDADE SETOR-DE-ATIVIDADE-1="" SETOR-DE-
              \begin{tabular}{ll} \hookrightarrow & $\operatorname{ATIVIDADE}-2=""" \text{ SETOR-DE-ATIVIDADE}-3="""/></\text{MESTRADO}> \\ \end{tabular}
</FORMACAO-ACADEMICA-TITULACAO>
<ATUACOES-PROFISSIONAIS>
```

<a href="mailto:<a href="mailto:<a href="mailto:<a href="mailto:">a reference - ""<a href="mailto:<a href="mailto:<a href="mailto:">a reference - ""<a href="mailto:<a h

```
NOME-INSTITUICAO=""
          SEQUENCIA-ATIVIDADE="" SEQUENCIA-IMPORTANCIA="">
          <VINCULOS
            {\tt SEQUENCIA\!-\!HISTORICO} {\tt = ""}
            TIPO-DE-VINCULO=""
            ENQUADRAMENTO-FUNCIONAL=""
            CARGA-HORARIA-SEMANAL=""
            FLAG-DEDICACAO-EXCLUSIVA=""
            MES-INICIO=""
            ANO-INICIO=""
            MES-FIM=""
            ANO-FIM=""
            OUTRAS-INFORMACOES=""
            FLAG\!\!-\!\!VINCULO\!\!-\!\!EMPREGATICIO\!\!=\!""
            OUTRO-VINCULO-INFORMADO=""
            OUTRO-ENQUADRAMENTO-FUNCIONAL-INFORMADO=""
            \hbox{OUTRO-ENQUADRAMENTO-FUNCIONAL-INFORMADO-INGLES=""}
            OUTRAS-INFORMACOES-INGLES=""/>
           </ATUACAO-PROFISSIONAL>
    </ATUACOES-PROFISSIONAIS>
    <AREAS-DE-ATUACAO>
      <AREA-DE-ATUACAO
        SEQUENCIA-AREA-DE-ATUACAO=""
        NOME-GRANDE-AREA-DO-CONHECIMENTO=""
        NOME-DA-AREA-DO-CONHECIMENTO=""
        NOME-DA-SUB-AREA-DO-CONHECIMENTO=""
        NOME-DA-ESPECIALIDADE=""/>
      <AREA-DE-ATUACAO SEQUENCIA-AREA-DE-ATUACAO="" NOME-GRANDE-AREA-DO
          → -CONHECIMENTO="" NOME-DA-AREA-DO-CONHECIMENTO="" NOME-DA-
          \begin{tabular}{ll} \hookrightarrow & {\tt SUB-AREA-DO-CONHECIMENTO=""" NOME-DA-ESPECIALIDADE="""/></"} \\ \end{tabular}
          → AREAS-DE-ATUACAO>
    <IDIOMAS>
      <IDIOMA IDIOMA="" DESCRICAO-DO-IDIOMA="" PROFICIENCIA-DE-LEITURA
          → ="" PROFICIENCIA-DE-FALA="" PROFICIENCIA-DE-ESCRITA=""
          → PROFICIENCIA-DE-COMPREENSAO=""/></IDIOMAS>
</DADOS-GERAIS>
<PRODUCAO-BIBLIOGRAFICA>
    <TRABALHOS-EM-EVENTOS>
        <TRABALHO-EM-EVENTOS SEQUENCIA-PRODUCAO="">
           <DADOS-BASICOS-DO-TRABALHO</p>
            NATUREZA=""
            TITULO-DO-TRABALHO=""
            ANO-DO-TRABALHO=""
             PAIS-DO-EVENTO=""
            IDIOMA=""
            MEIO-DE-DIVULGACAO=""
            HOME-PAGE-DO-TRABALHO=""
            FLAG-RELEVANCIA=""
            DOI=""
            TITULO-DO-TRABALHO-INGLES=""
            FLAG-DIVULGACAO-CIENTIFICA=""/>
            <DETALHAMENTO-DO-TRABALHO
             CLASSIFICACAO-DO-EVENTO=""
            NOME-DO-EVENTO=""
            CIDADE-DO-EVENTO=""
            ANO-DE-REALIZACAO=""
            TITULO-DOS-ANAIS-OU-PROCEEDINGS=""
            VOLUME=""
            FASCICULO=""
            \mathbf{SERIE} \mathbf{=} ""
            PAGINA-INICIAL=""
            PAGINA-FINAL=""
            ISBN=""
            NOME-DA-EDITORA=""
            CIDADE-DA-EDITORA=""/>
```

```
→ ORDEM-DE-AUTORIA="" NRO-ID-CNPQ=""/>
                <PALAVRAS-CHAVE PALAVRA-CHAVE-1="" PALAVRA-CHAVE-2=""
                    → PALAVRA-CHAVE-3="" PALAVRA-CHAVE-4="" PALAVRA-CHAVE
                    \hookrightarrow -5="" PALAVRA-CHAVE-6=""/>
                <AREAS-DO-CONHECIMENTO>
                  <AREA-DO-CONHECIMENTO-1 NOME-GRANDE-AREA-DO-CONHECIMENTO</p>
                      → AREA-DO-CONHECIMENTO="" NOME-DA-ESPECIALIDADE=""/>
                  < AREA-DO-CONHECIMENTO-2 NOME-GRANDE-AREA-DO-CONHECIMENTO
                      → ="" NOME-DA-AREA-DO-CONHECIMENTO="" NOME-DA-SUB-
                      → AREA-DO-CONHECIMENTO="" NOME-DA-ESPECIALIDADE=""/>
                  <AREA-DO-CONHECIMENTO-3 NOME-GRANDE-AREA-DO-CONHECIMENTO</p>
                      → ="" NOME-DA-AREA-DO-CONHECIMENTO="" NOME-DA-SUB-
                      → AREA-DO-CONHECIMENTO="" NOME-DA-ESPECIALIDADE=""/>
                </AREAS-DO-CONHECIMENTO>
                \hookrightarrow ATIVIDADE-2="" SETOR-DE-ATIVIDADE-3=""/>
                  <INFORMACOES—ADICIONAIS DESCRICAO—INFORMACOES—ADICIONAIS</p>
                      \hookrightarrow= "" DESCRICAO-INFORMACOES-ADICIONAIS-INGLES=""/>
              </TRABALHO-EM-EVENTOS>
        </TRABALHOS-EM-EVENTOS>
        <ARTIGOS-PUBLICADOS>
            <ARTIGO-PUBLICADO SEQUENCIA-PRODUCAO="" ORDEM-IMPORTANCIA="">
              <DADOS-BASICOS-DO-ARTIGO
                NATUREZA=""
                TITULO-DO-ARTIGO=""
                ANO-DO-ARTIGO=""
                PAIS-DE-PUBLICACAO=""
                IDIOMA=""
                MEIO-DE-DIVULGACAO=""
                HOME-PAGE-DO-TRABALHO=""
                FLAG-RELEVANCIA=""
                DOI=""
                TITULO-DO-ARTIGO-INGLES=""
                FLAG-DIVULGACAO-CIENTIFICA=""/>
              <DETALHAMENTO-DO-ARTIGO TITULO-DO-PERIODICO-OU-REVISTA=""</pre>
                  \hookrightarrow ISSN="" VOLUME="" FASCICULO="" SERIE="" PAGINA-INICIAL
                  → ="" PAGINA-FINAL="" LOCAL-DE-PUBLICACAO=""/>
              <autores nome-completo-do-autor="" nome-para-citacao="" ordem
                  → -DE-AUTORIA="" NRO-ID-CNPQ=""/>
              <PALAVRAS-CHAVE PALAVRA-CHAVE-1="" PALAVRA-CHAVE-2="" PALAVRA</pre>
                  → -CHAVE-3="" PALAVRA-CHAVE-4="" PALAVRA-CHAVE-5=""
                  → PALAVRA-CHAVE-6=""/>
                <AREAS-DO-CONHECIMENTO>
                  <AREA-DO-CONHECIMENTO-1 NOME-GRANDE-AREA-DO-CONHECIMENTO</p>
                      \hookrightarrow ="" NOME-DA-AREA-DO-CONHECIMENTO="" NOME-DA-SUB-
                      \begin{tabular}{ll} \hookrightarrow & AREA-DO-CONHECIMENTO="" NOME-DA-ESPECIALIDADE \\ \end{tabular}
                      → =""/></AREAS-DO-CONHECIMENTO>
                  <SETORES-DE-ATIVIDADE SETOR-DE-ATIVIDADE-1="" SETOR-DE-
                      → ATIVIDADE-2="" SETOR-DE-ATIVIDADE-3=""/>
                  <INFORMACOES-ADICIONAIS DESCRICAO-INFORMACOES-ADICIONAIS
                     \hookrightarrow ="" DESCRICAO-INFORMACOES-ADICIONAIS-INGLES=""/>
            </ARTIGO-PUBLICADO>
        </ARTIGOS-PUBLICADOS>
    </PRODUCAO-BIBLIOGRAFICA>
</CURRICULO-VITAE>
```

<autores nome-completo-do-autor="" nome-para-citacao=""

#### Apêndice H

#### Arquivo OWL da ontologia basic-lattes

Arquivo H.1: Arquivo OWL da ontologia basic-lattes

```
<?xml version="1.0"?>
<rdf:RDF xmlns="http://www.onto.org/obda/basic-lattes-final#"
    xml: base="http://www.onto.org/obda/basic-lattes-final"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema\#"
    xmlns: rdfs = "http://www.w3.org/2000/01/rdf-schema#">
   <owl:Ontology rdf:about="http://www.onto.org/obda/basic-lattes-final"/>
     // Object Properties
    .
.
   <!-- http://www.onto.org/obda/basic-lattes-final#publicadoPor -->
   <owl: ObjectProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#publicadoPor">
       <owl:inverseOf rdf:resource="http://www.onto.org/obda/basic-lattes-</pre>

→ final#publicou"/>

       < rdf: type rdf: resource = "http://www.w3.org/2002/07/owl#
           \hookrightarrow InverseFunctionalProperty"/>
       <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-
           <rdfs:range rdf:resource="http://www.onto.org/obda/basic-lattes-</pre>

→ final#Coautor"/>

       <rdfs:range rdf:resource="http://www.onto.org/obda/basic-lattes-</pre>

→ final#Pesquisador"/>

   </owl:ObjectProperty>
   <!-- http://www.onto.org/obda/basic-lattes-final#publicou --->
   <owl:ObjectProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#publicou">

       <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
           → InverseFunctionalProperty"/>
```

```
<rd>fs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-</rd>

→ final#Coautor"/>

         <rd>fs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-

→ final#Pesquisador"/>

        <rdfs:range rdf:resource="http://www.onto.org/obda/basic-lattes-
                </owl:ObjectProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#temAreaDeAtuacao -->
<owl: ObjectProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#temAreaDeAtuacao">

        <rd>fs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-</rd>

→ final#Pesquisador"/>

        <rdfs:range rdf:resource="http://www.onto.org/obda/basic-lattes-</pre>

→ final#AreaDeAtuacao"/>

</owl:ObjectProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#temAreaDeConhecimento
        \hookrightarrow --->
<owl:ObjectProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#temAreaDeConhecimento">

        <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-

→ final#ProducaoBibliografica"/>

        \hookrightarrow final#AreaDeConhecimento"/>
</owl:ObjectProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#

→ temDetalhamentoDoArtigo --->

<\!\!\mathrm{owl}: \\ \mathbf{ObjectProperty} \quad \mathbf{rdf}: \mathbf{about} = \texttt{"http://www.onto.org/obda/basic-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-lattes-latte
        \hookrightarrow \ \ final\#temDetalhamentoDoArtigo">
        <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-

        ← final#Artigo"/>

        <rdfs:range rdf:resource="http://www.onto.org/obda/basic-lattes-

→ final#ArtigoPublicado"/>

</owl:ObjectProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#

→ temDetalhamentoDoEvento —>

<owl:ObjectProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#temDetalhamentoDoEvento">

        <rd>fs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-</rd>

        ← final#Artigo"/>

        <rdfs:range rdf:resource="http://www.onto.org/obda/basic-lattes-</pre>

→ final#TrabalhoEmEvento"/>

</owl: ObjectProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#temFormacaoAcademica
       → --->
```

```
<owl: ObjectProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#temFormacaoAcademica">

   <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-</pre>
       <rdfs:range rdf:resource="http://www.onto.org/obda/basic-lattes-

→ final#FormacaoAcademica"/>

</owl:ObjectProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#temPalavraChave -->
<owl: ObjectProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#temPalavraChave">

   <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-

→ final#ProducaoBibliografica"/>

    <rdfs:range rdf:resource="http://www.onto.org/obda/basic-lattes-

→ final#PalavraChave"/>

</owl:ObjectProperty>
<!--
// Data properties
<!-- http://www.onto.org/obda/basic-lattes-final#anoDoArtigo --->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#anoDoArtigo">
   <rd>fs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-</rd>

        ← final#Artigo"/>

    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
</owl:DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#cidadeNascimento -->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#cidadeNascimento">
   <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-</pre>

→ final#Pessoa"/>

    <rd>fs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
       → "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#doi --->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#doi">

   <rd>fs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-</rd>

        ← final#Artigo"/>

    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
       → "/>
</owl: DatatypeProperty>
```

```
<!-- http://www.onto.org/obda/basic-lattes-final#fasciculo --->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

        ← final#fasciculo">
    <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
        → "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#
    → flagDivulgacaoCientifica --->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#flagDivulgacaoCientifica">
    <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-

        ← final#Artigo"/>

    <rd>fs:range rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"
        → "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#flagRelevancia --->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#flagRelevancia">

    <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-
        <rd>fs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
        ∽ "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#homePageDoTrabalho --->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#homePageDoTrabalho">

    <rd>fs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-</rd>

    → final#Artigo"/>

    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
        → "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#idioma --->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

        ← final#idioma">

    <rd>fs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-</rd>

→ final#Artigo"/>

    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string</pre>
        → "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#issn --->
```

```
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#issn">

    <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-

→ final#ArtigoPublicado"/>

    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string</pre>
        \hookrightarrow "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#localDePublicacao -->
<owl: DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#localDePublicacao">

    <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-

→ final#ArtigoPublicado"/>

    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
        "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#meioDeDivulgacao -->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#meioDeDivulgacao">

    <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-</pre>

→ final#Artigo"/>

    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#nacionalidade -->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#nacionalidade">
    <rd>fs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-</rd>

→ final#Pessoa"/>

    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
        \hookrightarrow "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#natureza --->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>
   → final#natureza">
    <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-</pre>

→ final#Artigo"/>

    <rd>fs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
        → "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#nomeCompleto -->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#nomeCompleto">

    <rd>fs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-</rd>

→ final#Pessoa"/>
```

```
<rd>fs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
        → "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#nomeCompletoDoAutor
   \hookrightarrow --->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#nomeCompletoDoAutor">
    <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-

→ final#Coautor"/>

    <rd>fs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
        → "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#
    \hookrightarrow nomeDaAreaDConhecimento -->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#nomeDaAreaDConhecimento">

    <rd>fs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-</rd>
        \hookrightarrow final#AreaDeConhecimento"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
        \hookrightarrow "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#nomeDaEspecialidade
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#nomeDaEspecialidade">
    <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-

→ final#AreaDeConhecimento"/>

    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
        "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#
    \hookrightarrow nomeDaSubAreaDoConhecimento --->
<owl: DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>
    \hookrightarrow final#nomeDaSubAreaDoConhecimento">
    <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-

→ final#AreaDeConhecimento"/>

    <rd>fs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
        → "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#
    → nomeEmCitacoesBibliograficas --->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#nomeEmCitacoesBibliograficas">

    <rd>fs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-</rd>

→ final#Pesquisador"/>
```

```
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
       "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#
   \hookrightarrow nomeGrandeAreaDoConhecimento -->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>
   \hookrightarrow final#nomeGrandeAreaDoConhecimento">
   <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-

→ final#AreaDeConhecimento"/>

    <rd>fs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
       → "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#nomeParaCitacao -->
<owl: DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#nomeParaCitacao">

   <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-</pre>
       <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
       → "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#nroIdCnpq -->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#nroIdCnpq">

   <rd>fs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-</rd>

→ final#Coautor"/>

    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
       → "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#numeroIdentificador
   → --->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#numeroIdentificador">
   <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-</pre>

→ final#Pesquisador"/>

    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
       → "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#ordemDeAutoria --->

→ final#ordemDeAutoria">

   <rd>fs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-</rd>

→ final#Coautor"/>

   <\!rdfs:\!range\ rdf:\!resource="http://www.w3.org/2001/XMLSchema\#int"/>
</owl: DatatypeProperty>
```

```
<!-- http://www.onto.org/obda/basic-lattes-final#paginaFinal --->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#paginaFinal">

        <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-
                 <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#paginaInicial -->
<\!\!\mathrm{owl}: \mathtt{DatatypeProperty} \quad \mathtt{rdf}: \mathtt{about} = \texttt{"http://www.onto.org/obda/basic-lattes-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latter-latte

→ final#paginaInicial">
        <rd>fs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-</rd>

        ← final#Artigo"/>

        <\!rdfs:\!range \ rdf:\!resource="http://www.w3.org/2001/XMLSchema\#int"/\!>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#pais --->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

        ← final#pais">

        <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-

→ final#ArtigoPublicado"/>

         <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
                \hookrightarrow "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#paisDeNacionalidade
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#paisDeNacionalidade">

        <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-

→ final#Pessoa"/>

        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string</pre>
                 ∽ "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#paisDeNascimento -->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#paisDeNascimento">

        <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-

→ final#Pessoa"/>

         <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string</pre>
                → "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#permissaoDeDivulgacao
        → --->
```

```
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#permissaoDeDivulgacao">

    <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-</pre>

→ final#Pesquisador"/>

    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"
        → "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#serie --->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

    → final#serie">
    <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-

→ final#Artigo"/>

    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
        → "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#siglaPaisNacionalidade
<owl: DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#siglaPaisNacionalidade">
    <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-

    → final#Pessoa"/>

    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
        → "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#textoResumoCvRh -->
<owl: DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>
   \hookrightarrow final#textoResumoCvRh">
    <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-

→ final#Pesquisador"/>

    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
        → "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#textoResumoRvRhEn -->
<owl: DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#textoResumoRvRhEn">
    <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-</pre>

→ final#Pesquisador"/>

    <rd>fs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
        → "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#tipo --->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#tipo">

    <rd>fs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-</rd>

→ final#Artigo"/>
```

```
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
        "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#titulo --->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#titulo">

    <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-

→ final#Artigo"/>

    <rd>fs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
        "/>
</owl: DatatypeProperty>
< !--\ http://www.\ onto.\ org/obda/\ basic-lattes-final\#tituloDoArtigoIngles
   → -->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#tituloDoArtigoIngles">

    <rd>fs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-</rd>
        <rd>fs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
       \hookrightarrow "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#

→ tituloDoPeriodicoOuRevista 

→ > 

<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#tituloDoPeriodicoOuRevista">

    <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-

→ final#ArtigoPublicado"/>

    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
       "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#ufNascimento -->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#ufNascimento">

   <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-</pre>

    → final#Pessoa"/>

    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
        → "/>
</owl: DatatypeProperty>
<!-- http://www.onto.org/obda/basic-lattes-final#volume --->
<owl:DatatypeProperty rdf:about="http://www.onto.org/obda/basic-lattes-</pre>

→ final#volume">

    <rdfs:domain rdf:resource="http://www.onto.org/obda/basic-lattes-

→ final#Artigo"/>

    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"
        → "/>
</owl: DatatypeProperty>
```

```
// Classes
<!-- http://www.onto.org/obda/basic-lattes-final#AreaDeAtuacao --->
<owl: Class rdf:about="http://www.onto.org/obda/basic-lattes-final#</pre>
   → AreaDeAtuacao"/>
<!-- http://www.onto.org/obda/basic-lattes-final#AreaDeConhecimento --->
<owl:Class rdf:about="http://www.onto.org/obda/basic-lattes-final#</pre>

→ AreaDeConhecimento"/>

<!-- http://www.onto.org/obda/basic-lattes-final#Artigo --->
<owl: Class rdf:about="http://www.onto.org/obda/basic-lattes-final#</pre>
   → Artigo">
   <rdfs:subClassOf rdf:resource="http://www.onto.org/obda/basic-
       → lattes-final#ProducaoBibliografica"/>
</owl: Class>
<!-- http://www.onto.org/obda/basic-lattes-final#ArtigoPublicado -->
<owl: Class rdf:about="http://www.onto.org/obda/basic-lattes-final#</pre>
   → ArtigoPublicado">
   <rdfs:subClassOf rdf:resource="http://www.onto.org/obda/basic-
       → lattes-final#Artigo"/>
</owl: Class>
<!-- http://www.onto.org/obda/basic-lattes-final#Coautor -->
<owl: Class rdf:about="http://www.onto.org/obda/basic-lattes-final#</pre>

→ Coautor">

   <rdfs:subClassOf rdf:resource="http://www.onto.org/obda/basic-
       → lattes-final#Pessoa"/>
</owl: Class>
<!-- http://www.onto.org/obda/basic-lattes-final#FormacaoAcademica -->
<owl: Class rdf:about="http://www.onto.org/obda/basic-lattes-final#</pre>
   \hookrightarrow \ {\tt FormacaoAcademica"/>}
```

```
<!-- http://www.onto.org/obda/basic-lattes-final#PalavraChave -->
    <owl: Class rdf:about="http://www.onto.org/obda/basic-lattes-final#</pre>
        → PalavraChave"/>
    < !--\ http://www.onto.org/obda/basic-lattes-final \# Pesquisador --->
    <owl: Class rdf:about="http://www.onto.org/obda/basic-lattes-final#</pre>
        → Pesquisador">
        <rdfs:subClassOf rdf:resource="http://www.onto.org/obda/basic-
             → lattes-final#Pessoa"/>
    </owl: Class>
    <!-- http://www.onto.org/obda/basic-lattes-final#Pessoa --->
    <owl: Class rdf:about="http://www.onto.org/obda/basic-lattes-final#</pre>
        → Pessoa"/>
    <!-- http://www.onto.org/obda/basic-lattes-final#ProducaoBibliografica
        \hookrightarrow --->
    <\!\!\mathrm{owl}: \textcolor{red}{\mathbf{Class}} \quad \mathtt{rdf:about} = \texttt{"http://www.onto.org/obda/basic-lattes-final#}
        → ProducaoBibliografica"/>
    <!-- http://www.onto.org/obda/basic-lattes-final#TrabalhoEmEvento -->
    <owl: Class rdf:about="http://www.onto.org/obda/basic-lattes-final#</pre>
        → TrabalhoEmEvento">
        <rdfs:subClassOf rdf:resource="http://www.onto.org/obda/basic-
            → lattes -final#Artigo"/>
    </owl: Class>
</rdf:RDF>
<!-- Generated by the OWL API (version 4.2.8.20170104-2310) https://github.

→ com/owlcs/owlapi 
--->
```

#### Apêndice I

### Classes do modelo de objetos do OntoMongo

Arquivo I.1: Classe publication do modelo de objetos do OntoMongo

```
class Publication
     include Mongoid::Document
2
     include OntoMap
3
4
     embedded in :researchers
5
6
7
      field :nature, type: String
      field : title , type: String
8
      field : title en , type: String
10
      field : year, type: Integer
      field : country, type: String
12
      field :language, type: String
13
      field : medium, type: String
      field :doi, type: String
15
      field : coauthors, type: Array
16
17
   end
```

Arquivo I.2: Classes researcher do modelo de objetos do OntoMongo

```
1 \# coding: utf-8
   class Researcher
      include Mongoid::Document
3
      include OntoMap
4
5
     embeds many : publications
6
7
      field :name, type: String
8
      field : name in citations, type: String
10
      field :country, type: String
11
      field :resume, type: String
12
13
      validates : name, : uniqueness => true
14
      validates : name in citations, : uniqueness => true
     index({ 'name' => 1, 'name_in_citations' => 1}, { unique: true, drop_dups
         \hookrightarrow : true \})
16
17 end
```

#### Referências Bibliográficas

- Araujo et al. (2017) Thiago Henrique Dias Araujo, Barbara Tieko Agena, Kelly Rosa Braghetto e Renata Wassermann. OntoMongo Ontology-Based Data Access for NoSQL. Em Proceedings of the IX Seminar on Ontology Research in Brazil and I Doctoral and Masters Consortium on Ontologies, Brasília, Brazil, August 28th-30th, 2017., páginas 55–66. Citado na pág. xi, 2, 19, 25, 26, 32
- Bizer e Seaborne (2004) Christian Bizer e Andy Seaborne. D2RQ-treating non-RDF databases as virtual RDF graphs. Em *Proceedings of the 3rd international semantic web conference* (ISWC2004), volume 2004. Springer. Citado na pág. 8
- Botoeva et al. (2016) Elena Botoeva, Diego Calvanese, Benjamin Cogrel, Martin Rezk e Guohui Xiao. OBDA Beyond Relational DBs: A Study for MongoDB. Em Proceedings of the 29th International Workshop on Description Logics, Cape Town, South Africa, April 22-25, 2016. Citado na pág. 13, 15, 16, 17, 19, 25, 26, 29, 32, 43
- Calvanese et al. (2007) Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi e Riccardo Rosati. Ontology-based Database Access. Em Proceedings of the Fifteenth Italian Symposium on Advanced Database Systems, SEBD 2007, 17-20 June 2007, Torre Canne, Fasano, BR, Italy, páginas 324–331. Citado na pág. 1, 7
- Calvanese et al. (2013) Diego Calvanese, Martin Giese, Peter Haase, Ian Horrocks, Thomas Hubauer, Yannis E. Ioannidis, Ernesto Jiménez-Ruiz, Evgeny Kharlamov, Herald Kllapi, Johan W. Klüwer, Manolis Koubarakis, Steffen Lamparter, Ralf Möller, Christian Neuenstadt, T. Nordtveit, Özgür L. Özçep, Mariano Rodriguez-Muro, Mikhail Roshchin, Domenico Fabio Savo, Michael Schmidt, Ahmet Soylu, Arild Waaler e Dmitriy Zheleznyakov. Optique: OBDA Solution for Big Data. Em The Semantic Web: ESWC 2013 Satellite Events ESWC 2013 Satellite Events, Montpellier, France, May 26-30, 2013, Revised Selected Papers, páginas 293–295. Citado na pág. xi, 7
- Calvanese et al. (2017) Diego Calvanese, Benjamin Cogrel, Sarah Komla-Ebri, Roman Kontchakov, Davide Lanti, Martin Rezk, Mariano Rodriguez-Muro e Guohui Xiao. Ontop: Answering SPARQL queries over relational databases. Semantic Web, 8(3):471–487. Citado na pág. 3, 8, 43
- Curé et al. (2013) Olivier Curé, Myriam Lamolle e Chan Le Duc. Ontology Based Data Integration Over Document and Column Family Oriented NoSQL. CoRR, abs/1307.2603. Citado na pág. 44
- Das et al.(2012) Souripriya Das, Seema Sundara e Richard Cyganiak. R2RML: RDB to RDF mapping language, 2012. Acessado: 2017-08-07. Citado na pág. 9
- de Oliveira e Wassermann (2017) Rafael Brito de Oliveira e Renata Wassermann. Utilização de Ontologia para Busca em Base de Dados de Acórdãos do STF (Using an Ontology for Searching the Decisions of the Brazilian Supreme Court). Em Proceedings of the IX Seminar on Ontology Research in Brazil and I Doctoral and Masters Consortium on Ontologies, Brasília, Brazil, August 28th-30th, 2017., páginas 147–157. Citado na pág. 19

- Gruber (2009) Tom Gruber. Ontology. Em Ling Liu e M. Tamer Özsu, editors, *Encyclopedia of Database Systems*, páginas 1963–1965. Springer US. ISBN 978-0-387-35544-3. Citado na pág. 3
- Han et al. (2011) Jing Han, E Haihong, Guan Le e Jian Du. Survey on NoSQL database. Em 2011 6th International Conference on Pervasive Computing and Applications, páginas 363–366. IEEE. Citado na pág. 13
- Klyne e Carroll(2006) Graham Klyne e Jeremy J. Carroll. Resource Description Framework (RDF):Concepts and Abstract Syntax. Relatório técnico, W3C Consortium. Citado na pág. xi, 6
- Mena-Chalco e Junior (2009) Jesús P. Mena-Chalco e Roberto Marcondes Cesar Junior. ScriptLattes: an open-source knowledge extraction system from the Lattes platform. *Journal of the Brazilian Computer Society*, 15(4):31–39. Citado na pág. 26
- Michel et al. (2015) Franck Michel, Loic Djimenou, Catherine Faron-Zucker e Johan Montagnat. xR2RML: Non-relational databases to RDF mapping language. Relatório técnico, Signaux et Systèmes de Sophia-Antipolis. Citado na pág. 17
- Michel et al. (2016) Franck Michel, Catherine Faron-Zucker e Johan Montagnat. A Mapping-Based Method to Query MongoDB Documents with SPARQL. Em DEXA (2), volume 9828 of Lecture Notes in Computer Science, páginas 52–67. Springer. Citado na pág. 13, 17, 29, 32
- Ong et al. (2014) Kian Win Ong, Yannis Papakonstantinou e Romain Vernoux. The SQL++ Semistructured Data Model and Query Language: A Capabilities Survey of SQL-on-Hadoop, NoSQL and NewSQL databases. CoRR, abs/1405.3631. Citado na pág. 43
- Rodriguez-Muro e Calvanese (2011) Mariano Rodriguez-Muro e Diego Calvanese. Dependencies: Making Ontology Based Data Access Work in Practice. Em *AMW*, volume 749 of *CEUR Workshop Proceedings*. CEUR-WS.org. Citado na pág. 9
- Rodriguez-Muro e Calvanese (2012) Mariano Rodriguez-Muro e Diego Calvanese. Quest, a system for ontology based data access. Em *Proceedings of OWL: Experiences and Directions Workshop (OWLED)*, Heraklion. Citado na pág. 9
- Rodriguez-Muro et al. (2012) Mariano Rodriguez-Muro, Josef Hardi e Diego Calvanese. Quest: Efficient SPARQL-to-SQL for RDF and OWL. Em International Semantic Web Conference (Posters & Demos), volume 914 of CEUR Workshop Proceedings. CEUR-WS.org. Citado na pág. 9
- Rodriguez-Muro et al. (2013a) Mariano Rodriguez-Muro, Roman Kontchakov e Michael Zakharyaschev. Ontology-Based Data Access: Ontop of Databases. Em *International Semantic Web Conference* (1), volume 8218 of *Lecture Notes in Computer Science*, páginas 558–573. Springer. Citado na pág. 10
- Rodriguez-Muro et al. (2013b) Mariano Rodriguez-Muro, Martín Rezk, Josef Hardi, Mindaugas Slusnys, Timea Bagosi e Diego Calvanese. Evaluating SPARQL-to-SQL Translation in Ontop. Em Informal Proceedings of the 2nd International Workshop on OWL Reasoner Evaluation (ORE-2013), Ulm, Germany, July 22, 2013, páginas 94–100. Citado na pág. 8
- Sadalage e Fowler (2012) Pramod J. Sadalage e Martin Fowler. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. Addison-Wesley Professional, 1st edição. ISBN 0321826620, 9780321826626. Citado na pág. xi, 20
- Sequeda et al. (2014) Juan F. Sequeda, Marcelo Arenas e Daniel P. Miranker. OBDA: Query Rewriting or Materialization? In Practice, Both! Em The Semantic Web ISWC 2014 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I, páginas 535-551. Citado na pág. 8