

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

LUCAS BASSETTI RODRIGUES DA FONSECA

**MAPEANDO DADOS GOVERNAMENTAIS COM UMA ONTOLOGIA
DE ORGANIZAÇÕES**

VITÓRIA

2014

LUCAS BASSETTI RODRIGUES DA FONSECA

**MAPEANDO DADOS GOVERNAMENTAIS COM UMA ONTOLOGIA
DE ORGANIZAÇÕES**

Monografia apresentada ao Curso de Ciência da Computação do Departamento de Informática da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. João Paulo A. Almeida

Coorientador: MSc. Carlos Lins B. Azevedo

VITÓRIA

2014

LUCAS BASSETTI RODRIGUES DA FONSECA

**MAPEANDO DADOS GOVERNAMENTAIS COM UMA ONTOLOGIA
DE ORGANIZAÇÕES**

Monografia submetida ao Curso de Ciência da Computação do Departamento de Informática da Universidade Federal do Espírito Santo, como requisito parcial para a obtenção do Grau de Bacharel em Ciência da Computação.

Aprovada em 05 de Agosto de 2014.

COMISSÃO EXAMINADORA

Prof. Dr. João Paulo Andrade Almeida - Orientador
Universidade Federal do Espírito Santo

MSc. Carlos Lins Borges Azevedo - Coorientador
Universidade Federal do Espírito Santo

Prof. Dr. Vítor Estêvão Silva Souza
Universidade Federal do Espírito Santo

DEDICATÓRIA

Aos meus pais José e Marinete por tornar meu sonho realidade.

AGRADECIMENTOS

Primeiramente, gostaria de agradecer à minha família por todo apoio e incentivo de todas as formas possíveis para que eu concluísse mais esta etapa em minha vida.

Ao amigo Jocimar Nalesso que, sem ele, eu não teria chegado a UFES e provavelmente não teria conhecido todas as demais pessoas presentes abaixo neste agradecimento.

À minha namorada Juliana por todo amor, companheirismo, paciência, compreensão e incentivo (até mesmo a distância) nesta jornada.

Ao professor João Paulo pela oportunidade e orientação e ao Carlos pela coorientação e toda paciência em esclarecer minhas recorrentes dúvidas.

Ao professor Vítor por aceitar participar da banca deste trabalho.

Ao meu grande amigo Thiago Lima que foi meu parceiro nos momentos iniciais e que me ensinou a dar valor a vida e aproveitar cada momento dela.

Aos amigos no NEMO por toda amizade, auxílio e por aturarem minhas cantorias. Em especial, a todos os cafés proporcionados com todo amor por Laylla, Ariane e Jordana que me impulsionaram em momentos difíceis em meu trabalho. Ao Freddy por ser minha dupla nas cantorias e buscar cantigas do fundo do baú. Ao Fábio, Victor, Cássio, John, Sobral e Pedro Paulo pelos auxílios e esclarecimentos em diversos assuntos.

A toda turma de 2009 por companheirismo e união em todos os momentos. Em especial ao Judismar e Broetto pelas diversas dúvidas esclarecidas em trabalhos e atividades ao longo do curso. Aos amigos Fábio, Marcos, Rogner, Missael e Filipe pelos churrascos, viagens e peladas proporcionadas. “Que fase!”

RESUMO

Em 18 de novembro de 2011 foi sancionada a Lei de Acesso à Informação Pública (Lei 12.527/2011) que regula o acesso a dados e informações detidas pelo governo. Com essa lei em vigor, os dados governamentais passaram a ser disponibilizados na internet, o que deveria proporcionar ao cidadão um melhor entendimento do governo para acesso aos serviços públicos, ao controle das contas públicas e à participação no planejamento e desenvolvimento das políticas públicas. No entanto, os dados que são publicados advêm de diversas fontes nas quais existe pouca ou nenhuma integração. Os dados das várias organizações governamentais possuem padrões diferentes, o que dificulta a utilização destes pelos cidadãos e por sistemas computacionais. Uma abordagem recente que vem sendo proposta por órgãos de padronização para integrar esses dados é através do uso de ontologias e tecnologias semânticas. Este trabalho realiza um estudo de caso sobre a integração de dados governamentais utilizando a Ontologia de Organizações que é uma recomendação W3C. O trabalho aborda a obtenção dos dados, o mapeamento dos dados selecionados para a Ontologia de Organizações, a publicação dos dados em RDF e seu uso e potenciais. As dificuldades, benefícios e lições aprendidas em cada uma dessas fases no estudo de caso são apresentados.

LISTA DE FIGURAS

Figura 1 - Camadas da Web Semântica (BERNERS-LEE, 2000)	21
Figura 2 - Relação entre URI, URL e URN (W3C/IETF, 2001).....	23
Figura 3 – Formato de uma tripla em RDF (CYGANIAK; WOOD; LANTHALER, 2014).....	24
Figura 4 - Grafo informal de um conjunto de triplas (SCHREIBER; YVES, 2014).....	25
Figura 5 - Nuvem de dados estruturados (CYGANIAK; JENTZSCH, 2011).....	28
Figura 6 – ORG Ontology (REYNOLDS, 2014).....	33
Figura 7 – Etapas do Mapeamento.....	39
Figura 8 – Visão geral da R2RML (DAS; SUNDARA; CYGANIAK, 2012).....	40
Figura 9 – Representação referente ao mapeamento da relação entre servidores, cargos e organizações	46
Figura 10 – Representação referente ao mapeamento das relações entre organizações, unidades organizacionais e servidores	47
Figura 11 – Representação referente ao mapeamento da remuneração dos servidores	48
Figure 12 – Representação referente ao mapeamento das estruturas organizacionais	49
Figura 13 – Representação referente ao mapeamento das Unidades Federativas. .	50
Figura 14 – Representação do grafo temporário resultante de parte do mapeamento de servidores na linguagem R2RML.....	55
Figura 15 – Exemplo da representação de duas triplas da forma feita pelo grafo resultante do mapeamento (esquerda – vermelho) e da forma como deveria ser (direita – verde).....	56
Figura 16 - Exemplo de URI Derreferenciada.....	58
Figura 17 – Visualização em formulário de uma URI derreferenciada com “.html” ...	58
Figura 18 – Parte da visualização em RDF de uma URI derreferenciada com “.rdf”	59
Figura 19 – Descrição de <i>memberOf</i> na ORG Ontology	60
Figura 20 – Tela do SPARQL Query Editor	61
Figura 21 - Exemplo de resultado de uma consulta SPARQL com formado HTML ..	62

Figura 22 – Aba do iSPARQL para criação de consultas SPARQL com grafo	63
Figura 23 – Resultado em tabela de uma consulta SPARQL no iSPARQL	64
Figura 24 - Consulta no portal do SIORG aos códigos filhos do Ministério da Fazenda	70
Figura 25 – Exemplo de problema no mapeamento resultante de unidades organizacionais	72
Figura 26 - Exemplo de mapeamento resultante de servidor	73
Figura 27 - Exemplo do mapeamento resultante de Organização	73
Figura 28 – Exemplo de problema em mapeamento por nome em duas bases de dados diferentes	75
Figura 29 - Exemplo de integração entre aplicações com mesma classificação.....	77

LISTA DE TABELAS

Tabela 1 - Exemplo de um grafo RDF escrito usando a linguagem XML (HEATH; BIZER, 2011)	25
Tabela 2 - Exemplo de um grafo RDF escrito no formato Turtle (HEATH; BIZER, 2011).....	26
Tabela 3 – Exemplo de uma tripla RDF	27
Tabela 4 – Exemplo de um consulta SELECT em SPARQL	27
Tabela 5 – Resultado da Consulta SPAQL.....	27
Tabela 6 – Principais Namespaces e Prefixos dos Vocabulários presentes na ORG Ontology.....	35
Tabela 7 - Dados utilizados sobre servidores civis	36
Tabela 8 - Dados utilizados sobre remuneração dos servidores civis.....	37
Tabela 9 - Dados utilizados sobre estruturas organizacionais do Governo Federal .	37
Tabela 10 - Dados utilizados sobre Unidades Federativas	38
Tabela 11 – Namespace e prefixo da linguagem R2RML	41
Tabela 12 – Exemplo de Órgão de Lotação e Exercício	42
Tabela 13 – Mapeamento conceitual das colunas da base de dados de servidores com os termos da ORG Ontology.....	43
Tabela 14 - Mapeamento conceitual das colunas da base de dados de estruturas organizacionais com os termos da ORG Ontology.....	44
Tabela 15 - Namespaces e prefixos utilizados no mapeamento	44
Tabela 16 – Exemplo de Cargos semelhantes sem identificadores da base de dados de Servidores Civis	68
Tabela 17 - Exemplos da atividade “Assessor Chefe” representada com códigos diferentes	69
Tabela 18 - Diferentes identificadores para Ministério da Fazenda na base de dados de Estruturas Organizacionais.....	69
Tabela 19 – Representação para Presidência da República na base de dado de Servidores Civis	71
Tabela 20 – Representação para Presidência da República na base de dado de Estruturas Organizacionais.....	71

Tabela 21 – Consulta SPARQL para retornar todos os servidores da unidade organizacional Coordenação de Planejamento pertencente a Fundação Nacional de Saúde.....	73
Tabela 22 – Resultado da consulta SPARQL.....	74

LISTA DE SIGLAS/ACRÔNIMOS

BBC – British Broadcasting Corporation

CSV – Comma-Separated Values

FOAF – Friend of a Friend

HTML – HyperText Markup Language

HTTP – HyperText Transfer Protocol

IBGE – Instituto Brasileiro de Geografia e Estatística

IRI – Internationalized Resource Identifier

iSPARQL – Iterative SPARQL Protocol and RDF Query Language

iSQL – Iterative Structured Query Language

LDP – Linked Data Platform

LOD – Linked Open Data

MP – Ministério do Planejamento, Orçamento e Gestão

NS – Namespace

OWL – Web Ontology Language

R2RML – RDB to RDF Mapping Language

RDB – Relational Database

RDF – Resource Description Framework

SKOS – Simple Knowledge Organization System

SPARQL – SPARQL Protocol and RDF Query Language

SQL – Structured Query Language

UFES – Universidade Federal do Espírito Santo

URI – Uniform Resource Identifiers

URL – Uniform Resource Locator

URN – Uniform Resource Name

W3C – World Wide Web Consortium

XML – eXtensible Markup Language

SUMÁRIO

1	INTRODUÇÃO.....	15
1.1	Objetivo.....	16
1.2	Abordagem.....	17
1.2.1	Bases de dados escolhidas.....	17
1.2.2	Ferramentas/tecnologias.....	18
1.3	Organização do texto.....	18
2	REFERENCIAL TEÓRICO	20
2.1	Web Semântica	20
2.2	Identificador Uniforme de Recurso (URI).....	22
2.3	Framework de Descrição de Recurso (RDF).....	23
2.4	SPARQL.....	26
2.5	Dados Ligados (Linked Data).....	27
2.6	Ontologia	29
2.6.1	Linguagem de Ontologia para Web (OWL).....	30
2.7	Discussão.....	30
3	MAPEAMENTO.....	32
3.1	ORG Ontology	32
3.2	Bases de Dados Originais	36
3.3	Abordagem de Integração e Mapeamento	38
3.4	Linguagem de Mapeamento de RDB para RDF (R2RML).....	39
3.5	Mapeamento Conceitual para a ORG Ontology.....	41
3.5.1	Mapeamento Conceitual das bases de dados de Servidores e Remuneração	41
3.5.2	Mapeamento Conceitual das bases de dados de Estruturas Organizacionais e de Unidades Federativas.....	43
3.6	Mapeamento Conceitual Efetuado para a ORG Ontology	44
3.7	Implementação.....	51
4	APLICAÇÃO DO MAPEAMENTO E USO DOS DADOS MAPEADOS	57
4.1	URI Derreferenciada.....	57
4.2	SPARQL Endpoint.....	61
4.3	Inferência	64
5	Discussões	67

5.1	Dificuldades Encontradas	67
5.1.1	Problemas nas Bases de Dados	67
5.1.2	Problema da Falta de Identificação Única Entre Todos os Entes Federais	71
5.1.3	Problema de integração com identificadores diferentes entre entes do Governo Federal.....	74
5.2	Benefícios	75
5.2.1	Melhor integração com uso de ontologias bem definidas	76
5.2.2	Termos de vocabulários muito utilizados	77
5.2.3	Melhor visibilidade e consulta dos dados	77
5.2.4	Inferência	78
6	CONCLUSÃO.....	79
7	BIBLIOGRAFIA.....	81
8	APÊNDICE A.....	84

1 INTRODUÇÃO

Em 18 de novembro de 2011 foi sancionada a Lei de Acesso a Informação Pública (Lei 12.527/2011) que regula o acesso a dados e informações mantidas pelo governo. Essa lei constitui um marco para a democratização da informação pública. Com essa lei em vigor, os dados governamentais passam a ser disponibilizados na Internet, proporcionando ao cidadão um melhor entendimento do governo no acesso aos serviços públicos, no controle das contas públicas e na participação no planejamento e desenvolvimento das políticas públicas. O portal de Dados Abertos (disponível em <http://dados.gov.br>) e o Portal da Transparência (disponível em <http://www.portaltransparencia.gov.br>) são duas iniciativas do Governo Federal de publicação e disseminação de dados e informações públicas na Internet

Atualmente, os dados que são publicados advêm de diferentes fontes nas quais existe pouca ou nenhuma integração, ou seja, dados de várias organizações governamentais possuem padrões diferentes, o que dificulta a utilização destes por sistemas computacionais. Além disso, a publicação dos dados é quase sempre feita através de arquivos não estruturados (tais como planilhas e arquivos em PDF), o que dificulta a leitura e interpretação por humanos.

Uma forma de integração recente utilizada por órgãos de padronização para esses dados, será através de um mapeamento com base em ontologias e tecnologias semânticas como o Framework de Descrição de Recursos (RDF - *Resource Description Framework*). RDF é uma linguagem para representar informações na web (CYGANIAK; WOOD; LANTHALER, 2014). A transformação para RDF é chamada de triplificação. Na Web Semântica, as ontologias são descritas por meio de RDFS e da Linguagem de Ontologias para Web (OWL – *Web Ontology Language*) que é uma extensão do RDFS desenvolvida para aumentar a expressividade semântica disponível em XML, RDF e RDFS de forma a garantir maior interoperabilidade entre dados e facilitar o processamento de conteúdo da informação por máquinas (MCGUINNESS; HARMELEN, 2004). Neste trabalho, o mapeamento terá como base uma ontologia chamada *The Organization Ontology* (que será apresentada a partir daqui como *ORG Ontology*) (REYNOLDS, 2014).

A *ORG Ontology* foi criada com o intuito de prover um modelo genérico para publicação de informações sobre organizações e estruturas organizacionais, incluindo organizações governamentais. A Ontologia descreve a estrutura da organização bem como as pessoas envolvidas, informações sobre a localização da organização e histórico da organização (como fusões e mudanças de nomes) (REYNOLDS, 2014).

Neste trabalho é realizado um mapeamento utilizando alguns dados referentes a organizações governamentais (tais como cadastro de servidores ou informações organizacionais tais como informações de seus servidores e estruturas organizacionais) com à *ORG Ontology*. Com isso, intenciona-se verificar a dificuldade para o mapeamento e integração dos dados governamentais no escopo definido. Por fim, publicar os dados com formatos propostos pelo W3C de forma a aumentar a integração, visibilidade, acessibilidade e portabilidade dos dados.

1.1 OBJETIVO

O objetivo geral deste trabalho é mapear dados abertos governamentais disponibilizados na Web pelo governo brasileiro com base na *ORG Ontology* (REYNOLDS, 2014). Esse objetivo geral pode ser detalhado nos seguintes objetivos específicos:

- Analisar as dificuldades para mapear e integrar dados disponibilizados pelo governo brasileiro com a *ORG Ontology*.
- Integrar os dados governamentais advindos de diferentes fontes.
- Triplificar e publicar dados mapeados com base no Framework de Descrição de Recursos (RDF – *Resource Description Framework*) (CYGANIAK; WOOD; LANTHALER, 2014) e com base na *ORG Ontology*.
- Publicar dados mapeados usando um conjunto de regras para publicar dados na Web descrito em (BERNERS-LEE, 2006).

1.2 ABORDAGEM

Nesta seção serão retratadas as bases de dados escolhidas, bem como as ferramentas e tecnologias utilizadas para a realização do mapeamento.

1.2.1 Bases de dados escolhidas

No mapeamento realizado neste trabalho, foram utilizadas algumas bases de dados. A escolha baseou-se na relevância dos dados e em sua representatividade em relação a *ORG Ontology*, ou seja, se estes dados possuem interesse público, podem ser mapeados aos termos do vocabulário da *ORG Ontology* e se representam uma parte significativa de conceitos da *ORG Ontology*. Também foram escolhidos dados advindos de diversas fontes governamentais para verificar a integração existente entre os dados. Todas as bases de dados selecionadas pertencem a órgãos subordinados à República Federativa do Brasil. Neste trabalho foram utilizadas as seguintes bases de dados:

- **Servidores Civis (Servidores e Remuneração - Novembro/2013)**
 - Autor: Ministério do Planejamento, Banco Central, Comandos Militares, MRE, DPF, RFB, MAPA, DEST e Ministérios [Honorários (Jetons)]
 - Disponível em:
<http://www.portaltransparencia.gov.br/downloads/servidores.asp>
- **Sistema de Informações Organizacionais do Governo Federal (SIORG)**
 - Autor: Ministério do Planejamento, Orçamento e Gestão (MP).
 - Disponível em: <http://dados.gov.br/dataset/siorg>
- **Unidade Federativas**
 - Autor: Instituto Brasileiro de Geografia e Estatística (IBGE)
 - Disponível em:
<http://www.ibge.gov.br/home/geociencias/areaterritorial/principal.shtm>

1.2.2 Ferramentas/tecnologias

Para a realização do mapeando, algumas tecnologias e ferramentas foram necessárias, dentre elas temos:

- Tecnologias semânticas (RDF, OWL e SPARQL) para gerar os dados.
- Princípios dos Dados Ligados (*Linked Data*) para publicação das bases de dados.
- Ontologias como base do mapeamento e integração das bases de dados.
- Banco de dados Virtuoso¹ versão 6.1.7 *open-source* para armazenamento tanto das bases de dados originais quanto o grafo resultante do mapeamento.
- Linguagem de Mapeamento R2RML (DAS; SUNDARA; CYGANIAK, 2012) para mapear as bases de dados originais para RDF.

1.3 ORGANIZAÇÃO DO TEXTO

Esta monografia é estruturada em quatro capítulos e contém, além da presente introdução, os seguintes capítulos:

- Capítulo 2 – Apresenta o Referencial Teórico acerca de temas relevantes ao contexto deste trabalho, a saber: Web Semântica, URI, RDF, Dados Ligados (*Linked Data*) e Ontologias.
- Capítulo 3 – Apresenta a abordagem para realização do mapeamento, desde da etapa conceitual até sua implementação. Além disso, apresenta as bases de dados originais selecionadas e uma descrição da *ORG Ontology* e da linguagem R2RML, utilizadas para o mapeamento destas bases de dados.
- Capítulo 4 – Apresenta uma aplicação do mapeamento visando demonstrar as funcionalidades desenvolvida para uso dos dados mapeados.
- Capítulo 5 – Apresenta as discussões acerca do mapeamento, mostrando as dificuldades encontradas e os benefícios de ter dados mapeados utilizando tecnologia semântica.

¹ <http://sourceforge.net/projects/virtuoso/files/virtuoso/6.1.7/>

- Capítulo 6 – Apresenta as conclusões do trabalho, limitações e propostas de trabalhos futuros.

2 REFERENCIAL TEÓRICO

O mapeamento apresentado neste trabalho foi realizado com base em padrões da W3C. Ele envolve diretamente linguagens como RDF e SPARQL dentro de contextos como a Web Semântica, Dados Ligados (*Linked Data*) e Ontologias. Neste capítulo serão apresentados estes contextos bem como as linguagens envolvidas.

2.1 WEB SEMÂNTICA

Em (BERNERS-LEE; HENDLER; LASSILA, 2001), a Web Semântica é descrita como uma extensão da Web atual (Web de Documentos), em que a informação possui significado (semântica) bem definido, permitindo que pessoas e computadores trabalhem em cooperação, ou seja, que tanto humanos quanto computadores consigam interpretar os dados. Para isso, dados estruturados e um conjunto de regras de inferência com suporte a raciocínio automático devem ser acessíveis ao computador.

Na Web de Documentos a navegação é feita através de *links* entre documentos. Já na Web Semântica (também chamada de Web de Dados), a navegação é feita através de relações entre um conjunto de conceitos onde a Web de documentos também atua de forma subjacente (BERNERS-LEE *et al.*, 2006). De outro modo, enquanto a Web atual faz uso da Linguagem de Marcação de Hipertexto (HTML - *HyperText Markup Language*) para representação de um conteúdo que segue apenas regras sintáticas, a Web Semântica utiliza o Framework de Descrição de Recurso (RDF - *Resource Description Framework*) que é uma linguagem para representação dos dados interligando-os com outros conjuntos de dados ou outros domínios de conhecimento, conseguindo, assim, criar uma interoperabilidade entre os conteúdos publicados na Internet. Essa interoperabilidade deve ser perceptível tanto pelo usuário quanto pelos agentes de software. Tanto a Web atual quanto a Web Semântica baseiam-se no Protocolo de Transferência de Hipertexto HTTP (*HyperText Transfer Protocol*), que é o mecanismo de acesso universal, e nos Identificadores de Recursos Uniformes URIs

(*Uniform Resource Identifiers*), que é um mecanismo de identificação global e único comumente usado no formato de Localizadores de Recursos Uniformes URLs (*Uniform Resource Locator*).

A Figura 1 mostra as camadas propostas para a Web Semântica por Berners-Lee em (BERNERS-LEE, 2000). Este trabalho será tratado até o nível de Ontologias (Ontology). Portanto, os níveis além deste não serão retratados aqui.

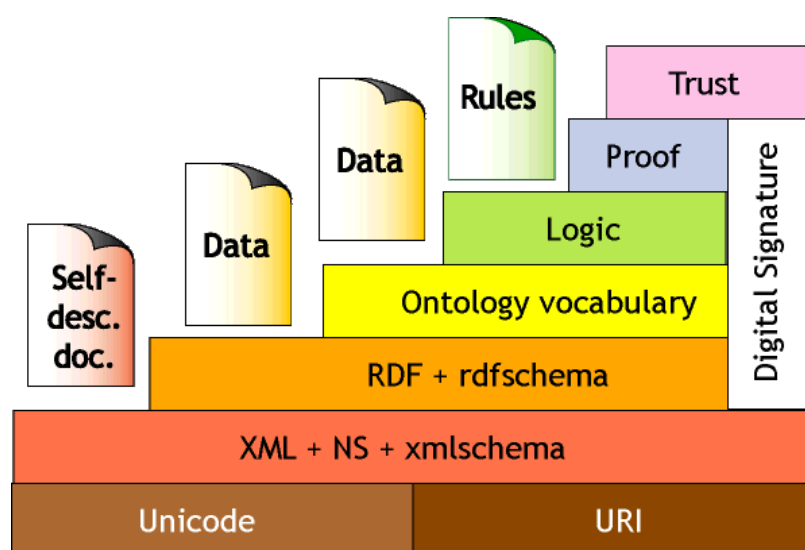


Figura 1 - Camadas da Web Semântica (BERNERS-LEE, 2000)

- I. **Unicode e URI** - Unicode é um padrão que permite aos computadores representar e manipular, de forma consistente, texto de qualquer sistema de escrita existente. URI é usado para identificar recursos a Web limitado a um subconjunto dos caracteres do Código de Padrão Americano para Intercâmbio de Informação (ASCII - *American Standard Code for Information Interchang*) (BERNERS-LEE; FIELDING; MASINTER, 2005).
- II. **XML, NS e XML Schema** – A Linguagem de Marcação Extensível (XML - *eXtensible Markup Language*) é uma linguagem para intercâmbio de dados no qual foi projetada para facilitar o compartilhamento de informação na Web (BRAY *et al.*, 2008). XML Schema é uma linguagem para definição de estruturas específicas em XML, ou seja, ela permite definir através de regras, a estrutura, o conteúdo e a semântica de um documento XML (GAO; SPERBERG-MCQUEEN; THOMPSON, 2012). Já *namespace* NS é uma *substring* em comum a um conjunto de

identificadores. O uso de *namespaces* aumenta a flexibilidade de XML Schema, permitindo a reutilização de definições feitas em outros esquemas. A utilização de um tipo definido em outro esquema é possível através da sua declaração e da associação de um prefixo a ele (BRAY *et al.*, 2006). Por exemplo, temos a definição do esquema `xmlns:ex="www.exemplo.com"` onde *ex* representa o prefixo associado ao *namespace* `www.exemplo.com`. Dessa forma, se quisermos indicar um elemento deste esquema, basta usar o prefixo mais o nome do elemento (por exemplo, `ex:Pessoa`).

- III. **RDF e RDF Schema** - A Linguagem de Descrição de Recurso RDF (*Resource Description Framework*) é uma linguagem para representar informações na Web (CYGANIAK; WOOD; LANTHALER, 2014). Semelhante ao XML Schema, o RDF Schema (RDFS) é uma extensão do RDF que prove um mecanismos para descrever esquemas, ou seja, descrever a estrutura dos dados e suas relações (BRICKLEY; GUHA, 2014).
- IV. **Ontologia (Ontology)** - Na Web Semântica, as ontologias são descritas por meio de RDFS e da Linguagem de Ontologias para Web (OWL – *Web Ontology Language*) que é uma extensão do RDFS desenvolvida para aumentar a expressividade semântica disponível em XML, RDF e RDFS de forma a garantir maior interoperabilidade entre dados e facilitar o processamento de conteúdo da informação por máquinas (MCGUINNESS; HARMELEN, 2004).

As camadas da Web Semântica usadas neste trabalho estão descritas mais detalhadamente nas seções 2.2, 2.3 e 2.6 a seguir.

2.2 IDENTIFICADOR UNIFORME DE RECURSO (URI)

Um Identificador Uniforme de Recurso (URI - *Uniform Resource Identifiers*) é uma cadeia de caracteres usada pra identificar ou denominar recurso na internet (BERNERS-LEE; FIELDING; MASINTER, 2005). Recursos podem ser qualquer coisa, incluindo objetos físicos, documentos, conceitos abstratos, números e strings

(CYGANIAK; WOOD; LANTHALER, 2014). URIs são fundamentais na arquitetura Web, pois fornecem uma identificação comum em toda Web (JACOBS; WALSH, 2004). Uma URI pode ser, como indicado na Figura 2, ou um Localizador de Recurso Uniforme (URL - *Uniform Resource Locator*), ou um Nome de Recurso Uniforme (URN - *Uniforme Resource Name*), ou ambos, ou nenhum deles (W3C/IETF, 2001).

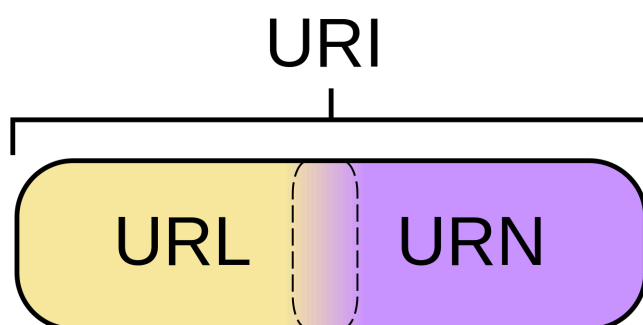


Figura 2 - Relação entre URI, URL e URN (W3C/IETF, 2001)

Um URL representa um endereço de um recurso, ou seja, um método para encontrá-lo (por exemplo, os protocolos HTTP, HTTPS e FTP). Já um URN define a identidade do item (como o nome de uma pessoa, por exemplo) (W3C/IETF, 2001). O Padrão Internacional de Número de Livro (ISBN - International Standard Book Number) (FOSTER, 1967) é um exemplo típico de um URN.

Um URI com protocolo HTTP pode ser descrito basicamente por: `http://<servidor>/<caminho>`. Esse caminho possui benefícios por ser simples, descentralizado e único (visto que o servidor é um nome registrado). Além de fornecer um meio para acessar um recurso.

2.3 FRAMEWORK DE DESCRIÇÃO DE RECURSO (RDF)

O Framework de Descrição de Recursos (RDF – *Resource Description Framework*) provê um framework para representar informações na Web tais como título, autor, e data de atualização de uma página Web. Na Web Semântica, o

significado é expresso com o framework RDF por meio de um conjunto de triplas, chamadas **triplas RDF** (*RDF triples*), onde cada tripla, como mostra a Figura 3, é composta de um **sujeito**, um **predicado** e um **objeto** (CYGANIAK; WOOD; LANTHALER, 2014).

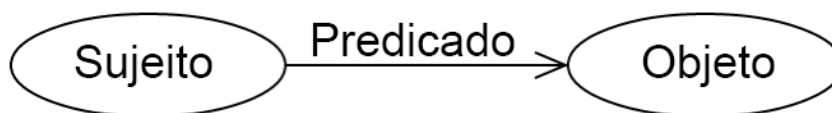


Figura 3 – Formato de uma tripla em RDF (CYGANIAK; WOOD; LANTHALER, 2014).

Uma declaração em RDF expressa uma relação entre dois **recursos**. O sujeito e o objeto representam os dois recursos a serem relacionados e o predicado representa uma relação direcional do sujeito para o objeto, denominado de **propriedade** (SCHREIBER; YVES, 2014). Abaixo podemos ver um exemplo de um conjunto de triplas RDF expressadas informalmente em pseudocódigo:

```
<Bob> <is a> <person>.
<Bob> <is a friend of> <Alice>.
<Bob> <is born on> <the 4th of July 1990>.
<Bob> <is interested in> <the Mona Lisa>.
<the Mona Lisa> <was created by> <Leonardo da Vinci>.
<the video 'La Joconde à Washington'> <is about> <the Mona Lisa>
```

No exemplo, Bob é o sujeito em quatro triplas e Mona Lisa é sujeito em uma tripla e objeto em duas outras. Essa possibilidade de ter um recurso como um sujeito em uma tripla e objeto em outra é uma importante parte do RDF que possibilita achar conexões entre as triplas (SCHREIBER; YVES, 2014). Um **grafo RDF** (*RDF graph*) representa um conjunto de triplas e pode ser visualizado como **nós** (sujeito ou objeto) conectados através de **arcos** (predicados), como mostra a Figura 4.

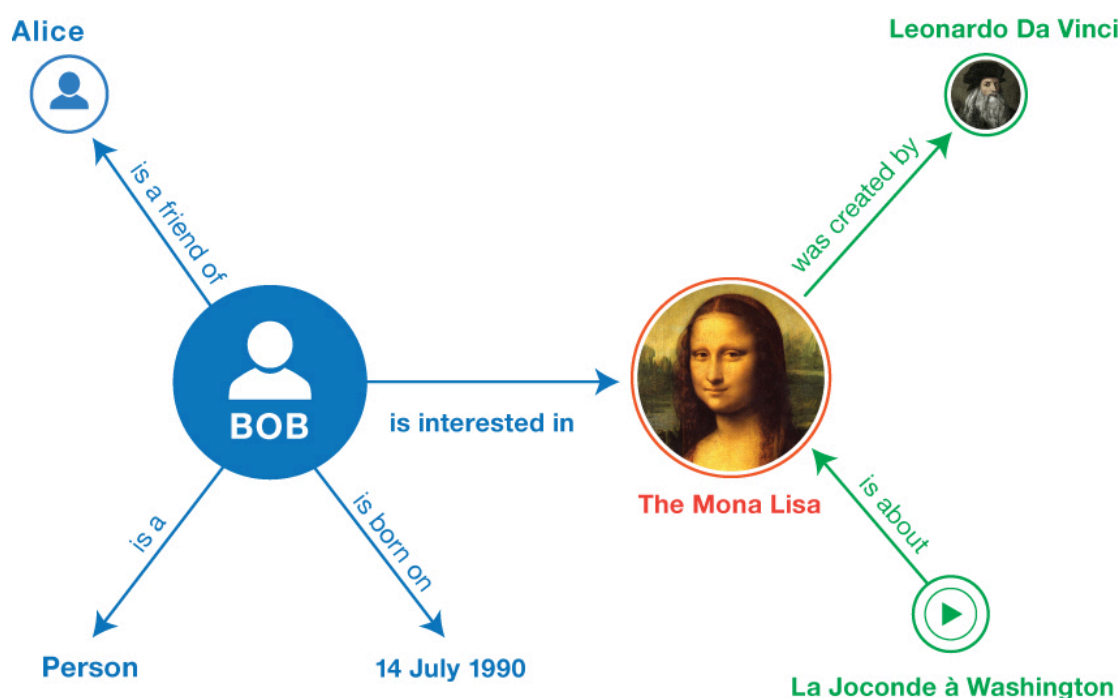


Figura 4 - Grafo informal de um conjunto de triplas (SCHREIBER; YVES, 2014)

O Grafo RDF pode ser escrito usando a Linguagem de Marcação Extensível (XML - *eXtensible Markup Language*) (BERNERS-LEE; HENDLER; LASSILA, 2001). XML foi projetada para facilitar o compartilhamento de informação na Web. Diversas aplicações têm suporte a XML pois é uma linguagem legível e fácil de criar e processar. A Tabela 1 apresenta um exemplo de um grafo RDF escrito com a linguagem XML.

Tabela 1 - Exemplo de um grafo RDF escrito usando a linguagem XML (HEATH; BIZER, 2011)

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/">

  <rdf:Description rdf:about="http://biglynx.co.uk/people/dave-smith">
    <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
    <foaf:name>Dave Smith</foaf:name>
  </rdf:Description>

</rdf:RDF>
```

Basicamente, o exemplo descreve um recurso sobre Dave Smith onde sua identificação é dada pelo URI <http://biglynx.co.uk/people/dave-smith> que é um tipo de Pessoa e possui um nome "Dave Smith". Apesar de ser melhor processável

por máquina o uso de RDF com XML é difícil a leitura e escrita por humanos. Por isso, foi criado o formato *Turtle* (BECKETT; BERNERS-LEE, 2011) no qual é menos estruturado para o computador e mais fácil para a leitura por humanos. A Tabela 2 mostra o mesmo exemplo anterior no formato *Turtle*.

Tabela 2 - Exemplo de um grafo RDF escrito no formato Turtle (HEATH; BIZER, 2011)

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

<http://biglynx.co.uk/people/dave-smith>
  rdf:type foaf:Person ;
  foaf:name "Dave Smith" .
```

Um grafo RDF pode conter um conjunto de URIs, no qual é chamado de **vocabulário RDF** (*RDF vocabulary*). Os URIs de um vocabulário RDF começam com um conjunto de caracteres em comum, chamado de **namespace**. O *namespace* pode ser abreviado em pequenos nomes, chamados de **prefixo** (*prefix*) (CYGANIAK; WOOD; LANTHALER, 2014). No exemplo, temos o prefixo `rdf` definido para o *namespace* `http://www.w3.org/1999/02/22-rdf-syntax-ns#`. Além disso, temos o nome “Dave Smith” que em RDF representa um **Literal**. Literais são usados para valores tais como *strings*, números e datas (CYGANIAK; WOOD; LANTHALER, 2014).

2.4 SPARQL

SPARQL é uma linguagem para consulta a dados em RDF. A maioria das consultas em SPARQL contém um conjunto de triplas similares as triplas RDF, exceto que o sujeito, predicado e objeto podem ser uma variável. Como resultado da consulta, temos um sub-grafo de um grafo RDF (HARRIS; SEABORNE, 2013). A Tabela 4 mostra um exemplo de uma consulta SPARQL utilizando os dados apresentados na Tabela 3.

Tabela 3 – Exemplo de uma tripla RDF

```
<http://example.org/book/book1> <http://purl.org/dc/elements/1.1/title> "SPARQL Example" .
```

Tabela 4 – Exemplo de um consulta SELECT em SPARQL

```
SELECT ?title
WHERE
{
    <http://example.org/book/book1> <http://purl.org/dc/elements/1.1/title> ?title .
}
```

A cláusula *SELECT* indica as variáveis resultantes da consulta (no caso, a variável *?title*). A cláusula *WHERE* fornece a consulta desejada, onde as variáveis são postas na posição desejada. A Tabela 5 apresenta o resultado da consulta.

Tabela 5 – Resultado da Consulta SPAQL

title
"SPARQL Example"

Se a cláusula *SELECT* for posta com símbolo "*" ao invés de variáveis, significa que o resultado trará toda a variável inserida na cláusula *WHERE*. Logo, no exemplo mostrado, se a variável *?title* for substituída pelo "*" na cláusula *SELECT* o resultado será o mesmo.

2.5 DADOS LIGADOS (LINKED DATA)

A Web Semântica trata de fazer ligações para que uma pessoa ou máquina possa explorar a Web de Dados (BERNERS-LEE, 2006). Dados representados utilizando um conjunto de boas práticas para publicar e conectar dados estruturas na Web, chamados de **Dados Ligados (*Linked Data*)** (BIZER; HEATH; BERNERS-LEE, 2009).

Berners-Lee (2006) define um conjunto de regras para publicação de dados na Web de forma que todos os dados publicados sejam unificados em único espaço global de dados. Estas regras são:

1. Usar URI como nome para as coisas.
2. Usar URIs HTTP para que as pessoas possam procurar por estes nomes.
3. Quando alguém visualiza um URI, fornecer informações uteis, usando os padrões RDF e SPARQL.
4. Incluir *links* para outros URI, para que eles possam descobrir mais coisas.

Estas regras fornecem os princípios básicos para a publicação e conexão de dados através da Web.

Atualmente, existem um projeto chamado *Linked Open Data* LOD que consiste em lançar conjunto de dados (*datasets*) de forma a gerar uma nuvem de dados estruturados. A Figura 5 mostra a última atualização da nuvem de dados estruturados disponível em (CYGANIAK; JENTZSCH, 2011).

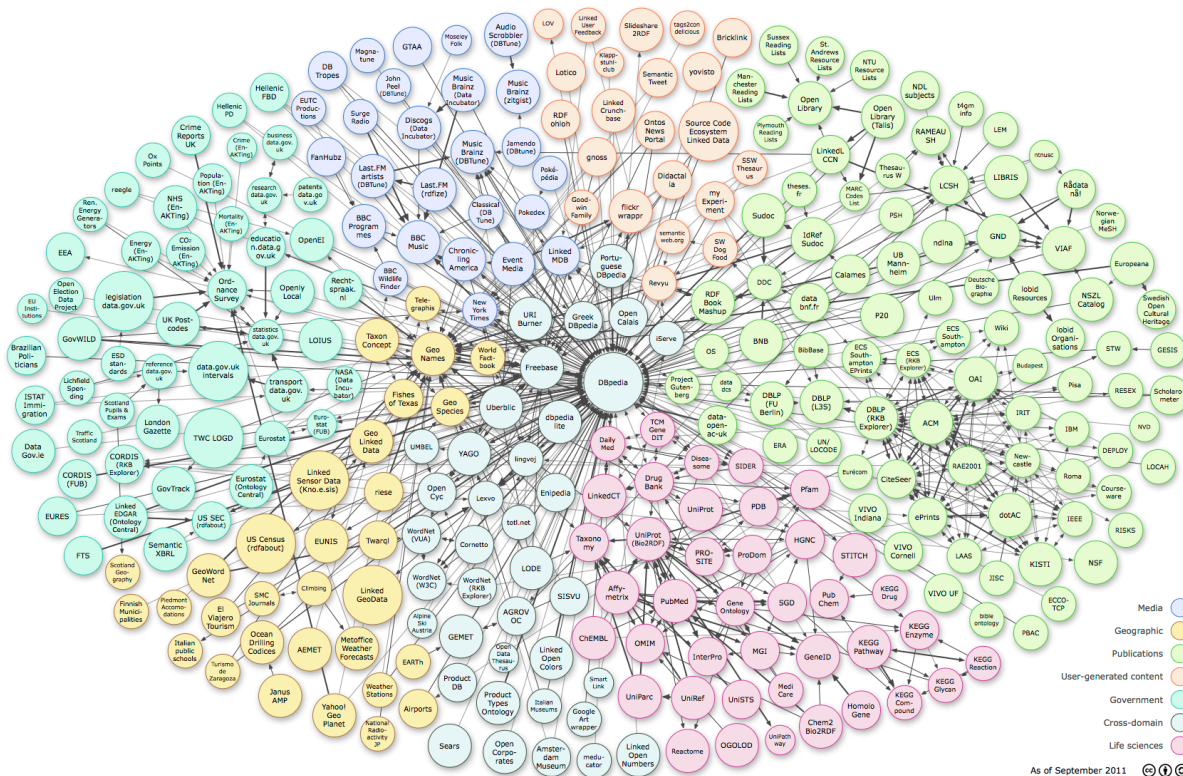


Figura 5 - Nuvem de dados estruturados (CYGANIAK; JENTZSCH, 2011)

O uso de RDF para dados ligados traz diversos benefícios, dentre eles temos:

- Referir-se, usando URIs, a qualquer que possa ser usado em escala global;
- Explorar uma gama de informação através da exploração de dados no qual os elementos das triplas se conectam;
- Ligar dados de diferentes fontes;
- Unir diversas triplas em um simples grafo, delimitando o escopo e misturando termos de diferentes vocabulários.

2.6 ONTOLOGIA

Segundo (GUARINO, 1998), ontologia, no sentido filosófico, define um sistema particular de categorias representado por uma determinada visão do mundo. Esse sistema não depende de uma linguagem particular, ou seja, a ontologia é sempre a mesma independentemente da linguagem usada para descrevê-la.

Na Ciência da Computação, ontologia tem sido reconhecida como um instrumento conceitual bastante útil desde o final dos anos sessenta, principalmente nas áreas de modelagem de dados e Inteligência Artificial (IA) (GUIZZARDI, 2005). Segundo (GRUBER, 1993), uma ontologia é “*uma especificação explícita de uma conceituação*”. Conceituação é uma visão simplificada do mundo representada para algum propósito. Já (GUARINO, 1998) refere-se a uma ontologia, em IA, como um artefato de engenharia, constituído por um vocabulário específico usado para descrever uma certa conceituação, além de um conjunto de suposições explícitas sobre o significado pretendido das palavras deste vocabulário.

As ontologias mais comuns na Web possuem uma **taxonomia** e um conjunto de **regras de inferência** (BERNERS-LEE; HENDLER; LASSILA, 2001). Taxonomia define classes de objetos e as relações entre eles. Por exemplo, “homem” e “mulher” podem ser definidos como tipo de “pessoa”, e “pessoa” pode ter a relação “casado” com outra “pessoa”. As regras de inferência permitem melhorar a qualidade da integração de dados na Web através da descoberta de novas relações, analisando automaticamente o conteúdo destes dados, além de detectar possíveis inconsistências neles. Por exemplo, vamos supor que exista informações na web sobre duas pessoas dizendo que ambos são filhos de um mesmo pai e mãe. No

entanto, não existe nenhuma relação dizendo que eles são irmãos. Então, através da relação deles com os pais é possível realizar inferência automática sobre estes dados e descobrir a relação de irmãos entre os dois filhos. Agora vamos supor que exista uma relação de irmãos entre os dois, mas ambos são filhos de pai e mãe diferentes. Assumindo que a relação entre irmãos não inclui a relação entre irmãos adotivos, é possível detectar uma inconsistência nesses dados uma vez que o neste caso o conjunto de declarações sobre as relações parentais é incompatível com o conjunto de relações fraternais.

2.6.1 Linguagem de Ontologia para Web (OWL)

A Linguagem de Ontologia para Web (OWL - *Web Ontology Language*) é uma extensão do *RDF Schema* que provê maior capacidade de descrição de classes e propriedades, incluindo a expressão de relações entre classes, restrições de cardinalidade, características de propriedades, classes enumeradas, etc. (MCGUINNESS; HARMELEN, 2004). Um exemplo desta capacidade é a especificação de relações inversas (*inverseOf*). Se dissermos que propriedade P1 é inversa de P2, então, se X for relacionado com Y através da propriedade P1, temos que Y é relacionado com X através da propriedade P2 (MCGUINNESS; HARMELEN, 2004). Isto permite inferências com dados parciais. Por exemplo, na *ORG Ontology* temos que a propriedade “membro de” (*memberOf*) é inversa da propriedade “tem membro” (*hasMember*). Logo, se tivermos que um servidor X é membro da organização Y, pode-se deduzir que Y tem o membro X.

2.7 DISCUSSÃO

De forma resumida, a Web Semântica oferece um novo e melhor caminho para criar e compartilhar informação Web. Para isso, um conjunto de boas práticas, chamados de dados ligados, devem ser seguidos para publicar e conectar os dados. Essas regras utilizam alguns padrões, tal como: URI, RDF, HTTP e SPARQL que são utilizados para nomear e rastrear os dados. Além disso, o uso de Ontologias é fundamental para estruturar a informação de forma a garantir interoperabilidade

entre os dados e possibilitar a utilização de regras de inferência para detecção de inconsistências e descoberta automática de novas relações entre os dados que antes não eram encontradas.

3 MAPEAMENTO

Este capítulo aborda o mapeamento de dados contidos nas bases governamentais selecionadas para dados estruturados de acordo com a *ORG Ontology*. Para tal foi utilizada uma linguagem de mapeamento de dados relacionais para RDF. O capítulo está organizado da seguinte forma: a Seção 3.1 descreve a *ORG Ontology*, que é base para o mapeamento; a Seção 3.2 apresenta as bases de dados utilizadas no mapeamento; a Seção 3.3 traz a abordagem utilizada para realização do mapeamento; a Seção 3.4 traz o mapeamento conceitual das bases de dados; a Seção 3.5 apresenta o mapeamento efetuado, ou seja, os modelos do mapeamento tendo como base a linguagem RDF. e; por fim, a Seção 3.6 aborda a implementação do mapeamento.

3.1 ORG ONTOLOGY

A *ORG Ontology* (*The Organizational Ontology*) descrita em (REYNOLDS, 2014) é uma recomendação W3C. Ela foi projetada para permitir a publicação de informações sobre as organizações e estruturas organizacionais, incluindo organizações não-governamentais. Ela fornece, como mostra a Figura 6, um modelo genérico e reutilizável que pode ser estendido ou especializado para uso em situações particulares. A ontologia descreve a estrutura da organização, bem como as pessoas envolvidas, informações sobre a localização da organização e o histórico da organização (como fusões e mudanças de nomes).

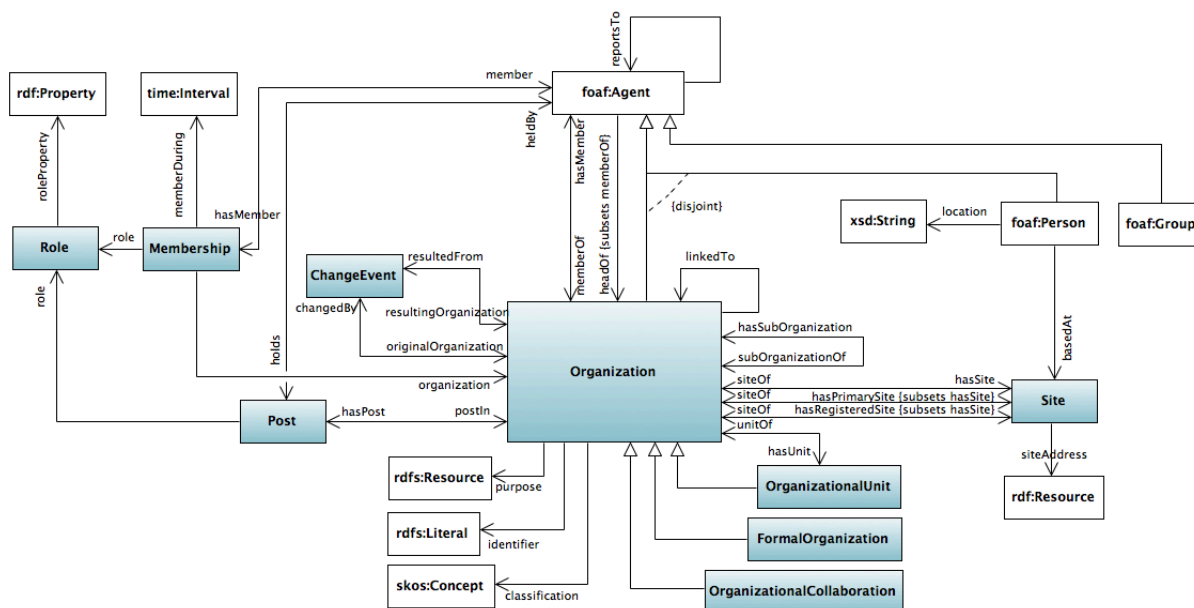


Figura 6 – ORG Ontology (REYNOLDS, 2014)

Conforme a figura, **Organização** (*Organization*) representa um conjunto de pessoas organizadas em uma comunidade ou outra estrutura social, comercial ou política. Uma **Organização** pode ser especializada em uma **Unidade Organizacional** (*OrganizationalUnit*), em uma **Organização Formal** (*FormalOrganization*) ou em uma **Colaboração Organizacional** (*OrganizationalCollaboration*). **Unidades Organizacionais** são entidades (como departamentos ou unidades de suporte) que pertencem a uma **Organização** e que não podem ser consideradas como uma entidade legal de direitos próprios. **Organizações Formais** são organizações reconhecidas mundialmente (como corporações, instituições de caridade, governo ou igreja) através de jurisdições legais, direitos e responsabilidades associadas. **Colaboração Organizacional** define um tipo de colaboração entre duas ou mais **Organizações** como um projeto. Ela atende aos critérios para ser uma organização na medida em que tem uma identidade e definição de propósito independente de seus membros em particular, mas não é nem uma entidade jurídica formalmente reconhecida nem uma unidade organizacional dentro de uma organização maior. Exemplos de **Colaboração Organizacional** são projetos conjuntos entre várias organizações ou consórcios de organizações, como os formados para licitações de grandes obras (REYNOLDS, 2014).

Adesão (*Membership*) de uma **Organização** representa o **Papel** (*Role*) que o **Agente** (*Agent*) tem na **Organização** durante um **Intervalo** (*Interval*) de tempo. **Agente** detém de um **Cargo** (*Post*) e pode ser uma **Organização** (exceto a qual ele é membro), uma **Pessoa** (*Person*) ou um **Grupo** (*Group*). **Pessoa** e **Organização** têm **Endereço** (*Site*). (REYNOLDS, 2014).

Organização pode ser alterada por **Eventos de Mudança** (*ChangeEvent*) que representam eventos que resultaram em grandes alterações para uma **Organização** como uma fusão ou reestruturação completa. Além disso, a **Organização** possui um **identificador** (*identifier*), um **propósito** (*purpose*), uma **classificação** (*classification*), essa última podendo variar dentro de algum esquema de classificação, e um ou vários **Endereços** (REYNOLDS, 2014).

O exemplo abaixo ilustra a pequeno fragmento da estrutura organizacional do gabinete oficial do Reino Unido retirado de (REYNOLDS, 2014).

```
<http://reference.data.gov.uk/id/department/co>
  rdf:type org:Organization , central-government:Department;
  skos:prefLabel "Cabinet Office" ;
  org:hasUnit
<http://reference.data.gov.uk/id/department/co/unit/cabinet-office-
communications> .

<http://reference.data.gov.uk/id/department/co/unit/cabinet-office-
communications>
  rdf:type org:OrganizationalUnit ;
  skos:prefLabel "Cabinet Office Communications" ;
  org:unitOf <http://reference.data.gov.uk/id/department/co> ;
  org:hasPost <http://reference.data.gov.uk/id/department/co/post/246> .

<http://reference.data.gov.uk/id/department/co/post/246>
  skos:prefLabel "Deputy Director, Deputy Prime Minister's
Spokesperson/Head of Communications" .
  org:postIn <http://reference.data.gov.uk/id/department/co/unit/cabinet-
office-communications> ;
  org:heldBy <#person161> .
```

Temos no exemplo que o “Gabinete Oficial” (*Cabinet Office*) é um tipo de Organização que possui a Unidade Organizacional “Gabinete Oficial de Comunicação” (*Cabinet Office Communication*), onde este último possui o cargo de Porta-Voz do Primeiro Ministro (*Deputy Prime Minister’s Spokesperson*).

A *ORG Ontology* faz uso de vocabulários de outras ontologias, os quais também são utilizados no mapeamento. A Tabela 6 mostra os principais *namespaces* e prefixos destes vocabulários.

Tabela 6 – Principais Namespaces e Prefixos dos Vocabulários presentes na ORG Ontology

Namespace URI	Prefixo	Referência
http://www.w3.org/ns/org#	org	(REYNOLDS, 2014)
http://xmlns.com/foaf/0.1/	foaf	(BRICKLEY; MILLER, 2014)
http://www.w3.org/1999/02/22-rdf-syntax-ns#	rdf	(CYGANIAK; WOOD; LANTHALER, 2014)
http://www.w3.org/2004/02/skos/core#	skos	(MILES; BECHHOFER, 2009)
http://www.w3.org/2002/07/owl#	owl	(MCGUINNESS; HARMELEN, 2004)
http://www.w3.org/2006/vcard/ns#	vcard	(IANNELLA; MCKINNEY, 2014)
http://purl.org/goodrelations/v1#	gr	(HEPP, 2011)

Friend of a Friend (FOAF) (BRICKLEY; MILLER, 2014) é uma ontologia para descrever pessoas, suas atividades e suas relações com outras pessoas e objetos. O Sistema Simples de Organização de Conhecimento (SKOS - *Simple Knowledge Organization System*) (MILES; BECHHOFER, 2009) fornece um modelo para representar estrutura básica e conteúdo de esquemas conceituais como tesouros, taxonomias, vocabulários controlados e sistemas de classificação de forma que sejam publicados na Web em formatos que possam ser lidos por máquinas. A *vCard Ontology* (IANNELLA; MCKINNEY, 2014) foi criada com o objetivo de utilizar o padrão vCard, de formato de arquivo para cartões de visita, para descrever pessoas e organizações na web. Por fim, temos a *GoodRelation* (HEPP, 2011) que é uma ontologia para descrever produtos, preços, lojas, e dados de empresas que podem ser embutidos em páginas web e lidos por máquinas. As demais ontologias usadas, RDF e OWL estão descritas nas seções 2.3 e 2.6.1.

3.2 BASES DE DADOS ORIGINAIS

Como discutido na seção 1.2.1, foram escolhidas bases de dados de diversas fontes governamentais (federais), a saber: base de dados sobre **Servidores Civis (Servidores e Remuneração)**, **Sistema de Informações Organizacionais do Governo Federal (SIORG)** e **Catálogo de Unidade Federativas**.

Nesta seção serão descritas as colunas que foram usadas de cada uma destas base de dados. Algumas colunas não foram utilizadas por não se encaixarem no escopo da ontologia.

A primeira base de dados apresenta dados de servidores civis sobre cargo, atividade, órgão e unidade organizacional de lotação e exercício e remuneração. A Tabela 7 indica os dados sobre servidores e a Tabela 8 sobre remuneração dos servidores utilizados para a realização do mapeamento.

Tabela 7 - Dados utilizados sobre servidores civis

SERVIDORES CIVIS	
COLUNA	DESCRIÇÃO
Id_SERVIDOR_PORTAL	Código identificador do servidor.
NOME	Nome do servidor.
DESCRICA0_CARGO	Descreve o cargo, ou seja, posição que o servidor ocupa dentro de uma empresa.
CODIGO_ATIVIDADE	Código identificador da atividade realizada pelo servidor.
ATIVIDADE	Descrição da atividade realizada pelo servidor, ou seja, o trabalho que ele desempenha.
UORG_LOTACAO	Unidade Organizacional em que o servidor está lotado, ou seja, local onde o servidor está efetivado.
COD_ORG_LOTACAO	Código identificado do órgão de lotação.
ORG_LOTACAO	Órgão em que o servidor está lotado.
COD_ORGSUP_LOTACAO	Código identificador do órgão superior de lotação.
ORGSUP_LOTACAO	Órgão de lotação superior ao órgão em que o servidor está lotado.
UORG_EXERCICIO	Unidade Organizacional em que o servidor está apresentando serviço.
COD_ORG_EXERCICIO	Código identificador do órgão de exercício.
ORG_EXERCICIO	Órgão de exercício do servidor.

COD_ORGSUP_EXERCICIO	Código identificador do órgão superior de exercício.
ORGSUP_EXERCICIO	Órgão superior ao órgão de exercício do servidor.

Tabela 8 - Dados utilizados sobre remuneração dos servidores civis

REMUNERAÇÃO DOS SERVIDORES CIVIS	
COLUNA	DESCRIÇÃO
ANO	Ano em que a remuneração foi efetuada.
MES	Mês em que a remuneração foi efetuada.
Id_SERVIDOR_PORTAL	Código identificador do servidor.
NOME	Nome do servidor.
REMUNERACAO_BASICA_BRUTA_BRL	Total bruto da remuneração recebida pelo servidor em real.
REMUNERACAO_BASICA_BRUTA_USD	Total bruto da remuneração recebida pelo servidor em dólar americano.

A segunda base de dados representa dados das estruturas organizacionais do Governo Federal sobre localização. A Tabela 9 apresenta os dados gerados e que foram utilizados no mapeamento.

Tabela 9 - Dados utilizados sobre estruturas organizacionais do Governo Federal

ESTRUTURAS ORGANIZACIONAIS DO GOVERNO FEDERAL	
COLUNA	DESCRIÇÃO
NOME	Nome do órgão.
CODIGO	Código identificador do órgão.
CODIGO_PAI	Código do órgão pai na hierarquia.
CODIGO_ORGAO_TOPO	Código do órgão que indica o topo da estrutura.
SIGLA	Sigla do órgão.
DDD	Prefixo DDD do órgão.
TELEFONE	Número(s) do(s) telefone(s) do órgão.
SITE	Endereço virtual do órgão.
EMAIL	Endereço de e-mail do órgão.
DESCRICAO_ENDERECO	Endereço da localização física do órgão.
NOME_PAIS	Nome do país do órgão.
NOME_CIDADE	Nome da cidade do órgão.
SIGLA_UF	Sigla do estado do órgão.
NUMERO_CEP	Número do Código de Endereçamento Postal (CEP) do órgão.

DESCRICA0_BAIRRO	Nome do bairro onde está localizado o órgão.
DESCRICA0_COMPLEMENTO	Complemento do endereço do órgão.

A terceira base de dados indica dados sobre as Unidades Federativas do Brasil. A Tabela 10 apresenta os dados utilizados no mapeamento.

Tabela 10 - Dados utilizados sobre Unidades Federativas

UNIDADES FEDERATIVAS	
COLUNA	DESCRIÇÃO
CODIGO_UF	Código identificador da unidade federativa.
NOME_UF	Nome da unidade federativa.
SIGLA_UF	Sigla da unidade federativa.

As bases de dados possuem diferentes formatos, onde a primeira base sobre Servidores Civis está no formato CSV (*Comma-Separated Values*) que é um formato de terminador de linha, a segunda do SIORG em XML e a última sobre Unidades Federativas em HTML. Com isso, uma transformação para uma formato em comum (no caso, o formato CSV) foi necessária para realização do mapeamento. Essa transformação é descrita com mais detalhes na seção 3.3.

3.3 ABORDAGEM DE INTEGRAÇÃO E MAPEAMENTO

No decorrer do mapeamento, várias etapas foram necessárias para gerar o Grafo RDF. Estas etapas são ilustradas na Figura 7.

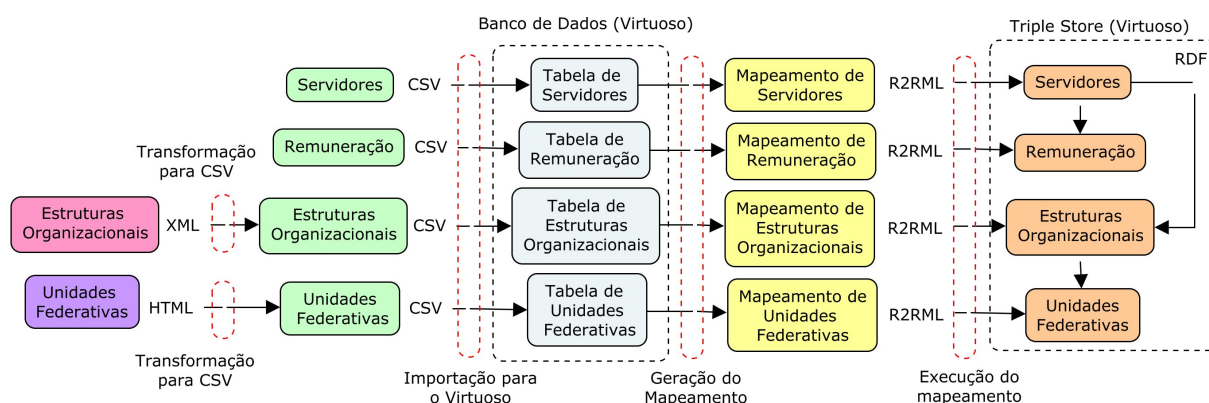


Figura 7 – Etapas do Mapeamento

Inicialmente, foi necessária a transformação de todas as bases de dados para um formato não estruturado único (no caso, o formato CSV) para poder realizar a importação no Virtuoso e conseguir efetuar o mapeamento. A transformação dos dados de Unidades Federativas foi feita manualmente copiando os dados e inserindo-os em um arquivo CSV. Já para os dados de Estruturas Organizacionais foi utilizada a ferramenta *Google Refine*² para converter o arquivo XML em CSV. Em seguida, todos os dados foram importados para o Virtuoso através de uma opção presente no banco que permite realizar esse tipo de operação, convertendo os dados em CSV para tabelas lógicas. Utilizando os dados importados no virtuoso, foi realizado um mapeamento utilizando a linguagem R2RML, que transforma o conteúdo das colunas usadas em RDF. Por fim, ao executar o mapeamento, é gerado o grafo RDF sobre o conteúdo das tabelas que é armazenada em um *Triple Store* no próprio Virtuoso.

3.4 LINGUAGEM DE MAPEAMENTO DE RDB PARA RDF (R2RML)

R2RML é uma linguagem para mapear dados de banco de dados relacionais para RDF. A R2RML é uma recomendação W3C e é descrita em (DAS; SUNDARA; CYGANIAK, 2012). A Figura 8 apresenta uma visão geral da linguagem.

² <https://code.google.com/p/google-refine/>

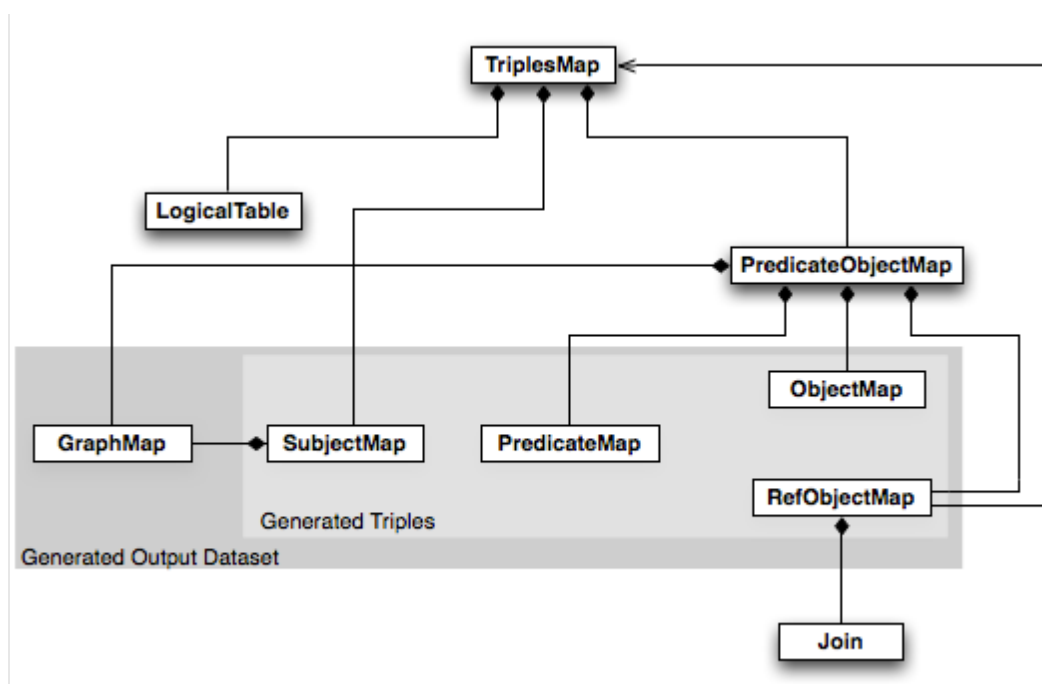


Figura 8 – Visão geral da R2RML (DAS; SUNDARA; CYGANIAK, 2012)

O **mapa de triplas** (*TripleMap*) especifica uma regra para traduzir cada linha de uma **tabela lógica** (*LogicalTable*) em triplas RDF. A **tabela lógica** representa o nome da tabela no banco de dados relacional no qual o dado será mapeado. O **mapa de sujeitos** (*SubjectMap*) especifica a regra para gerar o sujeito de uma tripla RDF que é gerado pelo *mapa de tripla*. O **mapa de predicados e objetos** (*PredicateObjectMap*) é uma função que cria um ou mais pares de predicados e objetos para cada linha da **tabela lógica**. Ele é usado em conjunto com um mapa de sujeito para gerar a tripla RDF do mapa de triplas (DAS; SUNDARA; CYGANIAK, 2012).

Tanto um **mapa de sujeitos** quanto o **mapa de predicados e objetos** podem conter um ou vários **mapas de grafo** (*GraphMap*), que representa o grafo nomeado no qual as triplas resultantes do mapeamento possam ser acessadas. Caso não seja inserido o grafo, o resultado será inserido no grafo predefinido da linguagem. Cada **mapa de predicados e objetos** contém um **mapa de predicado** (*PredicateMap*), que representa o predicado de uma tripla RDF, e o **mapa de objeto** (*ObjectMap*) ou uma **mapa de objeto referenciado** (*RefObjectMap*) que representam o objeto da tripla RDF. O mapa de objeto referenciado indica um objeto na tripla RDF que é representado por um sujeito de outro **mapa de tripla**. Além disso, ele é composto de

uma **condição de junção** (*Join*) que deve associar uma coluna da tabela do **mapa de tripla** que faz a referência com uma coluna da tabela do **mapa de tripla** referenciada (DAS; SUNDARA; CYGANIAK, 2012).

A linguagem R2RML possui um vocabulário cujo seu *namespace* e prefixo são apresentados na Tabela 11.

Tabela 11 – Namespace e prefixo da linguagem R2RML

Namespace URI	Prefixo
http://www.w3.org/ns/r2rml#	rr

3.5 MAPEAMENTO CONCEITUAL PARA A ORG ONTOLOGY

Esta seção discute as classes e relações da *ORG Ontology* que serão instanciadas para representar elementos constantes nas bases de dados originais.

3.5.1 Mapeamento Conceitual das bases de dados de Servidores e Remuneração

A base de dados de servidores possui colunas sobre servidores, cargos e atividades e diversas colunas sobre organizações. A coluna Servidores representa os servidores de um órgão. Como nessa base de dados servidores precisam ser pessoas, essa coluna foi mapeada para o conceito de **Pessoa** (*Person*) na *ORG Ontology*. Já as colunas Cargo e Atividade, apesar de possuir significados diferentes (onde o primeiro representa um conjunto de atribuições inerentes ao agente público e o segundo um conjunto de atribuições inerentes ao exercício de funções especiais, chefia e assessoramento), ambos são mapeados ao conceito genérico de **Cargo** (*Post*) na *ORG Ontology*. Esse tipo de opção de um mapeamento ao conceito genérico foi realizada, pois não faz parte do escopo do trabalho propor extensões para a ontologia.

Para as organizações, temos diversas colunas sobre unidades organizacionais, órgãos e órgãos superiores. Tanto as unidades organizacionais

quanto órgãos e órgãos superiores podem ser tanto de lotação quanto de exercício. O Órgão de Lotação representa a unidade em que o servidor foi empossado. Já o Órgão de Exercício corresponde à unidade em que o servidor exerce suas atribuições. A mesma definição vale para Unidades Organizacionais (de Lotação e de Exercício). A Tabela 12 mostra um exemplo de um Órgão e Lotação e Órgão e Exercício. No caso, representam, por exemplo, um servidor X que está lotado no Ministério da Agricultura, mas exerce suas atribuições no Governo do Estado do Mato Grosso do Sul.

Tabela 12 – Exemplo de Órgão de Lotação e Exercício

ORG_LOTACAO	ORG_EXERCICIO
MINIST.DA AGRICULTURA,PECUARIA E ABAST.	GOVERNO DO ESTADO DO MATO GROSSO DO SUL

A ontologia, assim como cargo e atividade, não faz distinção entre a relação de Lotação e a aquela de Exercício. Portanto, o mapeamento foi feito para o supertipo comum dos elementos. No caso, **Organização Formal** (*FormalOrganization*) para os Órgãos de Lotação e Exercício e **Unidade Organizacional** (*OrganizationalUnit*) para as Unidades Organizacionais de Lotação e Exercício.

No mapeamento de remuneração dos servidores, a ontologia sugere o uso do vocabulário *GoodRelation* (HEPP, 2011) para esse tipo de representação, mas sem restringir a outros tipos de vocabulários. Verificou-se, no entanto, que esse vocabulário foi criado para descrever preços e produtos, não cobrindo elementos relativos a remuneração, tais como remuneração líquida e bruta, gratificações, impostos, etc. Dessa forma, levando em consideração que não é proposta do trabalho realizar extensões, o mapeamento foi realizado apenas com a remuneração bruta do servidor e para o conceito **Especificação de Preço** (*PriceSpecification*) da ontologia, interpretando que o trabalho do servidor possui uma especificação de preço para o órgão governamental contratante.

A Tabela 13 apresenta o mapeamento conceitual das colunas da base de dados de servidores com os termos da *ORG Ontology*.

Tabela 13 – Mapeamento conceitual das colunas da base de dados de servidores com os termos da ORG Ontology

Servidores e Remuneração	Termo da ORG Ontology
Servidor	Pessoa (<i>Person</i>)
Cargo	Cargo (<i>Post</i>)
Atividade	Cargo (<i>Post</i>)
Órgão de Lotação	Organização Formal (<i>FormalOrganization</i>)
Órgão de Exercício	Organização Formal (<i>FormalOrganization</i>)
Unidade Organizacional de Lotação	Unidade Organizacional (<i>OrganizationalUnit</i>)
Unidade Organizacional de Exercício	Unidade Organizacional (<i>OrganizationalUnit</i>)
Remuneração Bruta	Especificação de Preço (<i>PriceSpecification</i>)

3.5.2 Mapeamento Conceitual das bases de dados de Estruturas Organizacionais e de Unidades Federativas

Assim como na base de dados de servidores, na base de dados de estruturas organizacionais é feito o mapeamento de cada órgão como sendo uma **Organização Formal**. Já dados sobre endereço de cada órgão são mapeados como sendo um tipo de **Endereço** (*Site*) na ontologia. Os atributos da organização como: sigla, nome, código são relacionados utilizando termos da SKOS (MILES; BECHHOFFER, 2009) e, para atributos como: telefone e e-mail e endereço (bairro, cidade, etc.) são relacionados utilizando termos da vCard (IANNELLA; MCKINNEY, 2014).

Por fim, os elementos da base de dados de Unidades Federativas são mapeadas utilizando a ontologia SKOS para nome, sigla e código e a ontologia vCard para associar o endereço da organização a Unidade Federativa que ele pertence.

A Tabela 14 apresenta o mapeamento conceitual das colunas da base de dados de estruturas organizacionais com os termos da *ORG Ontology*.

Tabela 14 - Mapeamento conceitual das colunas da base de dados de estruturas organizacionais com os termos da ORG Ontology

Estruturas Organizacionais e Unidades Federativas	Termo da ORG Ontology
Órgão	Organização Formal (<i>FormalOrganization</i>)
Endereço do Órgão	Endereço (<i>Site</i>)
Nome, Sigla e Código do Órgão e da Unidade Federativa	Termos da SKOS (prefLabel, altLabel e notation)
E-mail, Telefone e Endereço do Órgão	Termos da vCard (hasEmail, hasTelephone street-address, country-name, postal-code, etc.)

3.6 MAPEAMENTO CONCEITUAL EFETUADO PARA A ORG ONTOLOGY

Nesta seção são descritos os mapeamentos das bases de dados descritas anteriormente com os elementos da *ORG Ontology*.

Com o intuito de prover um mapeamento adequado aos princípios dos dados ligados, foi criado um *namespace* URI com protocolo HTTP, servidor local e caminhos específicos para representar os elementos das bases de dados como páginas acessíveis na web. A Tabela 15 apresenta os *namespaces* e prefixos criados no mapeamento. Os *namespaces* e prefixos indicados na Tabela 15 representam respectivamente: Dados de organizações, servidores e remuneração da base de dados de servidores; dados de organizações e endereços da base de dados de estruturas organizacionais; dados de estados brasileiros da base de dados do IBGE.

Tabela 15 - Namespaces e prefixos utilizados no mapeamento

Namespace URI	Prefixo
http://localhost:8080/NOM/resource/organization/	org-r
http://localhost:8080/NOM/resource/person/	person
http://localhost:8080/NOM/resource/remuneration/	rem
http://localhost:8080/NOM/resource/organization-structure/	org-s
http://localhost:8080/NOM/resource/organization-structure/address/	org-a

http://localhost:8080/NOM/resource/region/	Region
--	--------

Para representar os padrões de mapeamento aqui descritos, foi criada uma representação visual, em que um diagrama representa um “template” de um grafo que será gerado na saída do mapeamento. Este template é parametrizado com variáveis que representam dados originários das bases de dados de entrada.

São representados nesta notação os seguintes elementos:

- **Classes das Ontologias (laranja):** Representa os termos dos vocabulários das ontologias contendo o prefixo da ontologia mais o termo do vocabulário.
 - **Representação:** <Prefixo>:<Termo do vocabulário>
- **Entidades (azul):** Representa uma URI contendo o prefixo criado para a base de dados mais uma variável dada pelo valor da coluna mapeada.
 - **Representação:** <Prefixo>:<{Variável}>
- **Literais (verde):** Representa o valor da coluna mapeada.
 - **Representação:** <”Variável”>

Inicialmente, temos na Figura 9 o mapeamento da relação entre servidores, cargos e órgão de lotação e exercício.

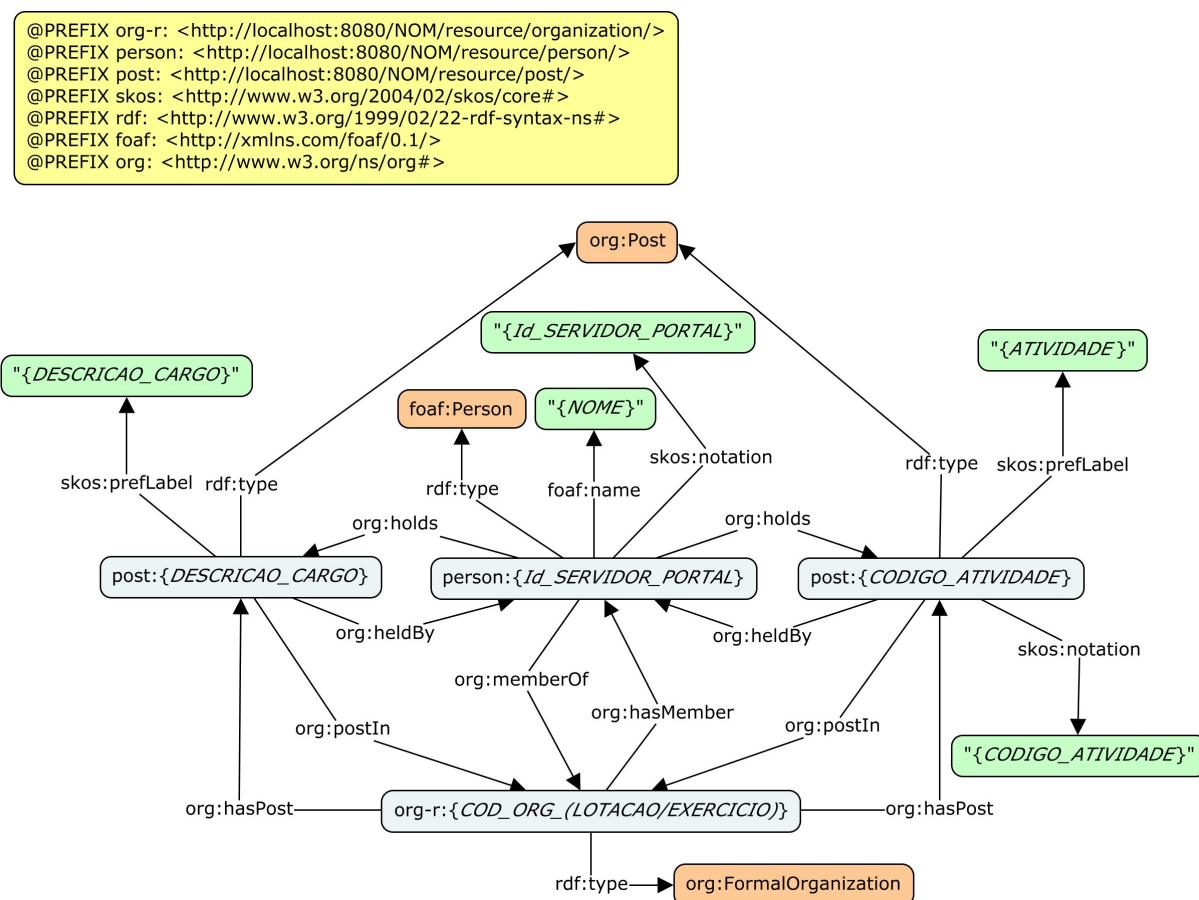


Figura 9 – Representação referente ao mapeamento da relação entre servidores, cargos e organizações

Cada servidor é representado pelo seu identificador na base de dados (Id_SERVIDOR_PORTAL). Servidores são um tipo (*rdf:type*) de Pessoa (*foaf:Person*) e possuem um nome (*foaf:name*) representado pela coluna “NOME” e o identificador único (*skos:notation*) representado pela coluna “Id_SERVIDOR_PORTAL”. Servidores têm (*org:holds*) cargo ou atividade e são membros (*org:memberOf*) de um Órgão de Lotação e Exercício. Como a base de dados não possui identificadores para cargos, sua URI e seu rótulo preferencial (*skos:prefLabel*) são criados utilizando a coluna “DESCRICAO_CARGO”. Já para atividades, temos que a URI utiliza o código (CODIGO_ATIVIDADE) e seu rótulo preferencial (*skos:prefLabel*) é representado pela coluna “ATIVIDADE”. Além disso, Atividade possui o identificador único (*skos:notation*) representado pela coluna “CODIGO_ATIVIDADE”. Ambos cargos e atividades são mapeados como um mesmo tipo (*rdf:type*) Cargo (*org:Post*) e são cargos (*org:postIn*) dentro de um órgão

de lotação e exercício. Tanto Órgão de Lotação quanto de Exercício possuem membros (*org:hasMember*) e cargos e atividades (*org:hasPost*).

A Figura 10 indica uma continuação da representação anterior mostrando as relações entre organizações, unidades organizacionais e servidores.

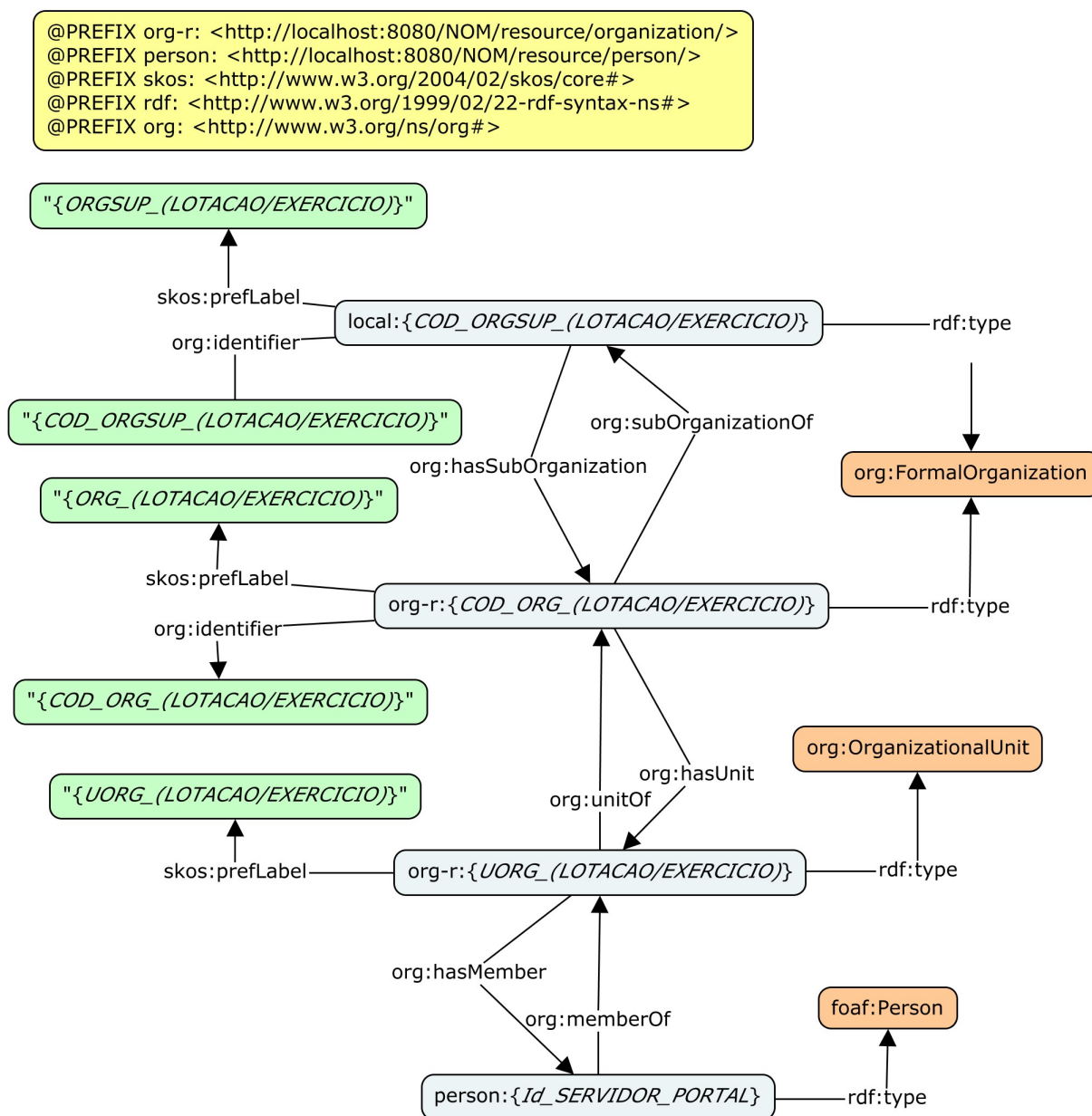


Figura 10 – Representação referente ao mapeamento das relações entre organizações, unidades organizacionais e servidores

Tanto Órgão de Lotação quanto Órgão de Exercício são tipos (*rdf:type*) de Organização Formal (*org:FormalOrganization*) e têm suas URIs representadas por

um código identificador (COD_ORG_LOTACAO e COD_ORG_EXERCICIO). Ambas possuem rótulos preferenciais (*skos:prefLabel*) representados pelas colunas “ORG_LOTACAO” e “ORG_EXERCICIO” e identificadores únicos (*org:identifier*) representados pelas colunas “COD_ORG_LOTACAO” e “COD_ORG_EXERCICIO”. Órgão de Lotação e Órgão de Exercício são, respectivamente, sub-organizações de (*org:subOrganizationOf*) um Órgão Superior de Lotação e de um Órgão Superior de Exercício e podem ter, respectivamente, uma ou várias Unidades Organizacionais de Lotação e Exercício. Da mesma forma, Órgão superiores têm suas URIs representadas por um código identificador (COD_ORGSUP_LOTACAO e COD_ORGSUP_EXERCICIO), possuem rótulos preferenciais (*skos:prefLabel*) representados pelas colunas “ORGSUP_LOTACAO” e “ORGSUP_EXERCICIO” e identificadores únicos (*org:identifier*) representados pelas colunas “COD_ORGSUP_LOTACAO” e “COD_ORGSUP_EXERCICIO”. Unidades Organizacionais de Lotação e Exercício são tipos (*rdf:type*) de Unidades Organizacionais (*org:OrganizationalUnit*) e por ter como membros (*org:hasMember*) um ou vários servidores.

A Figura 11 mostra o mapeamento realizado para remuneração dos servidores.

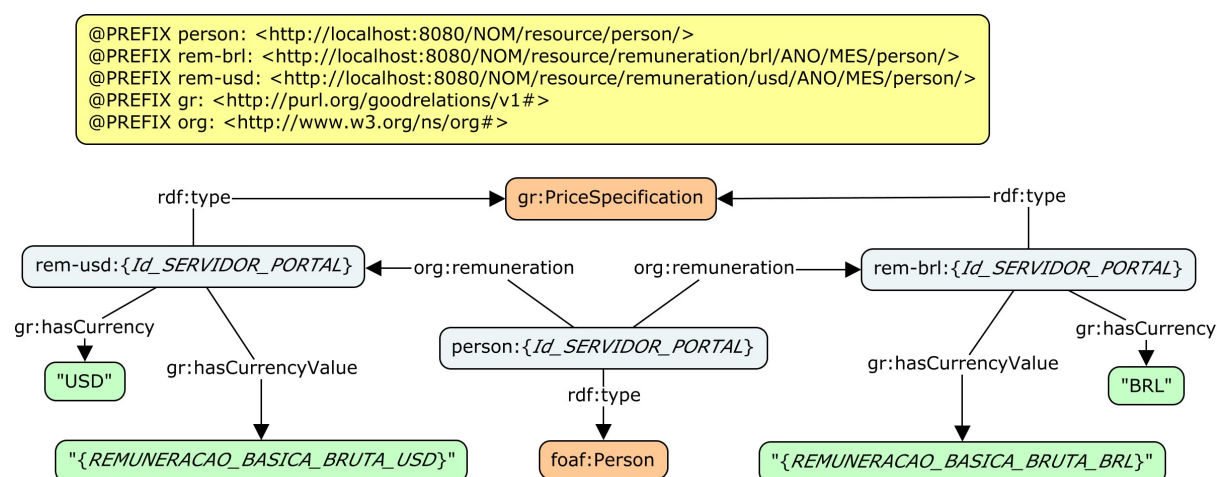


Figura 11 – Representação referente ao mapeamento da remuneração dos servidores

Basicamente, servidores possuem remuneração (*org:remuneration*) que pode ser tanto em real quanto em dólar americano. Cada remuneração, tanto em real quanto dólar americano, é um tipo de Especificação de Preço (*gr:PriceSpecification*)

e tem sua URI representada pelas colunas “ANO”, “MES” para ter algo temporal e pela coluna “Id_SERVIDOR_PORTAL” para indicar o servidor remunerado. A remuneração em real tem valor (*gr:hasCurrencyValue*) representado pela coluna “REMUNERACAO_BASICA_BRUTA_BRL” e a remuneração em dólar americano representado pela coluna “REMUNERACAO_BASICA_BRUTA_USD”.

Em seguida, temos a Figure 12 que apresenta o mapeamento de estruturas organizacionais.

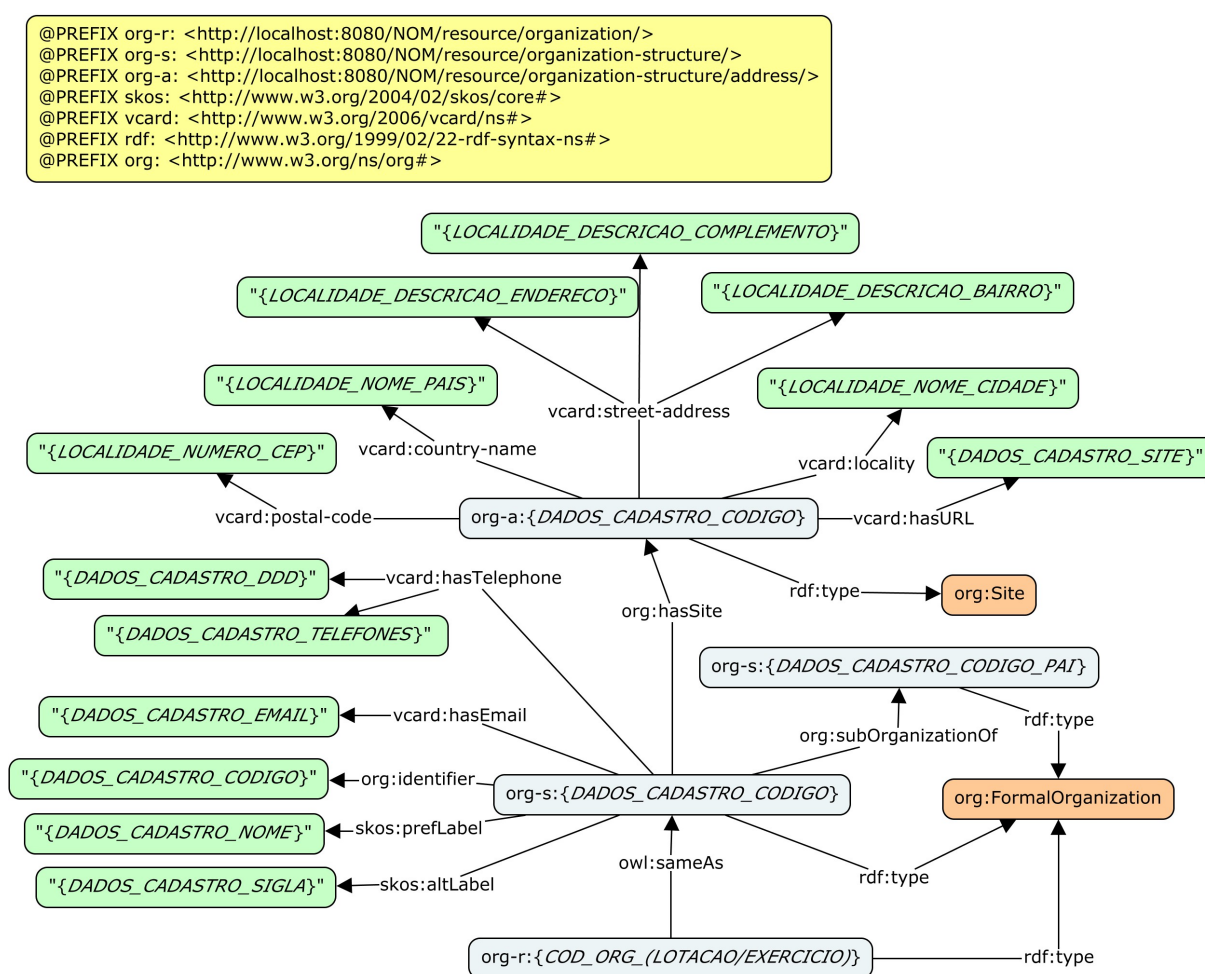


Figure 12 – Representação referente ao mapeamento das estruturas organizacionais

Temos primeiramente que a URI da organização é representada por um código identificador (DADOS_CADASTRO_CODIGO) e é o mesmo (*owl:sameAs*) que a URI da organização de mesmo nome na base de dados de servidores (COD_ORG_LOTACAO e COD_ORG_EXERCICIO). Em seguida, temos que a organização é um tipo (*rdf:type*) de Organização Formal (*org:FormalOrganization*), possui um identificador único (*org:identifier*) representado pela coluna

“DADOS_CADASTRO_CODIGO”, um rótulo preferencial (*skos:prefLabel*) e um rótulo alternativo (*skos:altLabel*) representado, respectivamente, pelas colunas “DADOS_CADASTRO_NOME” e “DADOS_CADASTRO_SIGLA”, nenhum ou vários telefones (*vcard:hasTelephone*) representados pelas colunas “DADOS_CADASTRO_DDD” e “DADOS_CADASTRO_TELEFONE”, nenhum ou vários e-mails (*vcard:hasEmail*) representados pela coluna “DADOS_CADASTRO_EMAIL” e nenhum ou vários endereços (*vcard:hasSite*) que é uma URI representado pelo mesmo identificador da organização, só que com outro *namespace*. Este endereço é um tipo de Endereço (*org:Site*) e possui um código postal (*vcard:postal-code*) representado pela coluna “LOCALIDADE_NUMERO_CEP”, um país (*vcard:country-name*) representado pela coluna “LOCALIDADE_NOME_PAIS”, um endereço (*vcard:street-address*) representado pelas colunas “LOCALIDADE_DESCRICA_O_ENDERECO”, “LOCALIDADE_DESCRICA_O_COMPLEMENTO” e “LOCALIDADE_DESCRICA_O_BAIRRO”, uma cidade (*vcard:locality*) representado pela coluna “LOCALIDADE_NOME_CIDADE” e nenhum ou vários endereços eletrônicos (*vcard:hasURL*) representado pela coluna “DADOS_CADASTRO_SITE”.

Por fim, a Figura 13 mostra o mapeamento de Unidades Federativas.

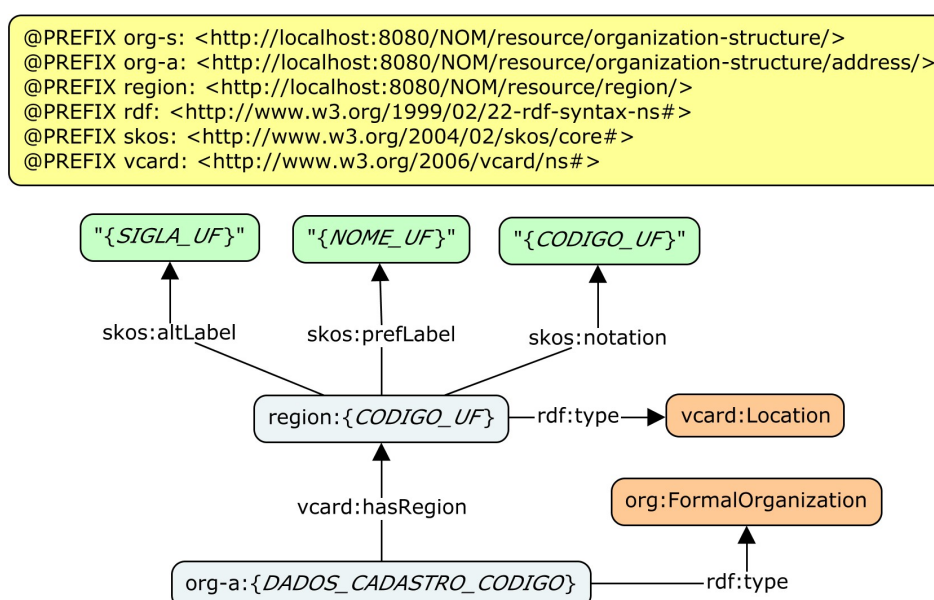


Figura 13 – Representação referente ao mapeamento das Unidades Federativas.

O endereço de uma organização (representada no modelo anterior) possui um estado (*vcard:hasRegion*) que é uma URI criada através do código identificador do estado (CODIGO_UF). Cada estado possui um código único (*skos:notation*) representado pela coluna “CODIGO_UF”, um rótulo preferencial (*skos:prefLabel*) representado pela coluna “NOME_UF” e um rótulo alternativo (*skos:altLabel*) representado pela coluna “SIGLA_UF”.

3.7 IMPLEMENTAÇÃO

Nesta seção será mostrada parte da implementação em R2RML do mapeamento. A implementação completa é apresentada no APÊNDICE A. Para a implementação foi utilizado o banco de dados Virtuoso versão 6.1.7 *open-source* (<http://sourceforge.net/projects/virtuoso/files/virtuoso/6.1.7/>) que possui suporte a RDF e R2RML. O Virtuoso provê um console SQL (iSQL – *Iterative Structured Query Language*) onde é possível executar os códigos descritos aqui para realização do mapeamento.

Alguns ajustes nos dados foram necessários para que o mapeamento fosse possível tanto na sua realização quanto no desempenho para executá-lo. Dentre os ajuste temos:

- Alteração nos caracteres na base de dados de estruturas organizacionais para se adequar a base de dados de servidores. Com isso, os caracteres passaram a ser todos maiúsculos e sem acentos assim como na base de servidores. Casos em que ocorra alguma diferença de tipografia ou espaçamento, exceto acentuação ou diferenciação entre letras maiúsculas e minúsculas não serão mapeados como o mesmo elemento, pois a verificação é feita através de uma comparação de igualdade entre todos os caracteres dos elementos. Isso implica que elementos iguais são tomados como diferentes, o que empobrece a geração e estatísticas e relatórios e a consequente tomada de decisões da organização, além, obviamente, de dificultar que o cidadão possua o entendimento das informações governamentais.

- Divisão dos dados da base de dados de servidores em várias outras bases para que as colunas que representam URIs tenham uma redução no número de elementos repetidos e uma melhora no desempenho na execução do mapeamento. No caso, a separação gerou as seguintes bases:
 - Agentes (servidores);
 - Organizações;
 - Órgãos superiores;
 - Unidades organizacionais.
- Mapeamento em somente uma direção, ou seja, sem relações inversas. Por exemplo, temos o mapeamento onde servidor é membro de uma organização, no entanto, não se tem o inverso em que organização tem como membro o servidor. Isso foi necessário devido a separação da bases de dados descrita acima que impossibilita, devido a exclusão de elementos repetidos, criar a relação inversa. Por exemplo, como existe várias organizações com vários servidores, ao excluir as organizações repetidas, só é possível mapear o servidor a organização, e não o contrário. No entanto, veremos em seções subsequentes que é possível criar esta relação inversa através de inferência.

Os dados ajustados estão disponíveis para *download* no seguinte *link*:

http://www.4shared.com/zip/Hayvjac5ba/ArquivosMapeamento_LucasBasset.html

Para a realização do mapeamento foi utilizado o computador com as seguintes configurações:

Modelo: Mac OS X – Versão 10.9.4
Processador: 2.4 GHz Intel Core i5
Memória: 8 GB 1600 MHz DDR3

O mapeamento durou em média 30 minutos para um total de quase 9,5 milhões de triplas.

Abaixo temos o código que apresenta parte do mapeamento de servidores civis:

<pre>DB.DBA.TTLP (' @prefix rr: <http://www.w3.org/ns/r2rml#> .</pre>

```

@prefix org: <http://www.w3.org/ns/org#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .

<http://localhost:8080/NOM/resource#TriplesMapPerson>
  a rr:TriplesMap;

  rr:logicalTable
  [ rr:tableSchema "CSV"; rr:tableOwner "DBA"; rr:tableName "Agentes_csv" ];

  rr:subjectMap
  [
    rr:template "http://localhost:8080/NOM/resource/person/{Id_SERVIDOR_PORTAL}";
    rr:class foaf:Person;
    rr:graph <http://localhost:8080/NOM/graph#>;
  ];

  rr:predicateObjectMap
  [
    rr:predicate foaf:name;
    rr:objectMap [ rr:column "NOME" ];
  ];

  rr:predicateObjectMap [
    rr:predicate org:memberOf;
    rr:objectMap [
      rr:parentTriplesMap <http://localhost:8080/NOM/resource#TriplesMapOrganization>;
      rr:joinCondition [
        rr:child "COD_ORG_LOTACAO";
        rr:parent "COD_ORG_LOTACAO";
      ];
    ];
  ]

;
', 'http://temp/person', 'http://temp/person');

```

Inicialmente são definidos os prefixos dos vocabulários RDF. O prefixo *rr* representa o vocabulário da linguagem R2RML. Em seguida, é definido um mapa de tripla como um tipo de mapa de triplas (*rr:TripleMap*). Após isso, é indicada a base de dados (*rr:logicalTable*) que será mapeada. Então, é definido o sujeito da tripla (*rr:subjectMap*) com um URI padrão (*rr:template*) usando um ou vários dados da base de dados, no caso, o identificador do servidor (Id_SERVIDOR_PORTAL). É definido então o tipo (*rr:class* que no resultado do mapeando vira um *rdf:type*) do URI e o grafo RDF (*rr:graph*) no qual será inserido as triplas RDF. Adiante, temos os mapeamentos de predicados e objetos (*rr:predicateObjectMap*) que serão ligados ao sujeito da tripla RDF. O predicado (*rr:predicate*) será um URI de algum vocabulário. E o objeto pode ser um literal que é uma coluna (*rr:column*) da base de dados ou

uma outra URI de outro mapa de triplas. Neste ultimo caso, é feita uma junção (*rr:joinCondition*) com a outra base de dados através da coluna da base atual (*rr:child*) com a coluna da base do outro mapa de triplas (*rr:parent*). Ele consiste em ligar o sujeito do mapa atual com o sujeito do outro mapa (*rr:parentTripleMap*) quando os valores das duas colunas forem iguais. Neste caso, o mapeamento ocorre com dois mapas que utilizam bases de dados iguais e, por isso, as colunas possuem o mesmo nome (COD_ORG_LOTACAO). O código abaixo mostra um exemplo no qual a junção é feita com duas bases diferentes onde a coluna das duas bases, apesar de representarem a mesma coisa, possuem nomes diferentes.

```
rr:predicateObjectMap [
  rr:predicate owl:sameAs;
  rr:objectMap [
    rr:parentTriplesMap <http://localhost:8080/NOM/resource#TriplesMapOrganizationStructure>;
    rr:joinCondition [
      rr:child "ORG_LOTACAO";
      rr:parent "DADOS_CADASTRO_NOME";
    ];
  ];
]
```

Por fim, o mapeamento é criado em um grafo temporário (<http://temp/person>). A Figura 14 mostra uma representação do grafo temporário resultante dessa parte do mapeamento.

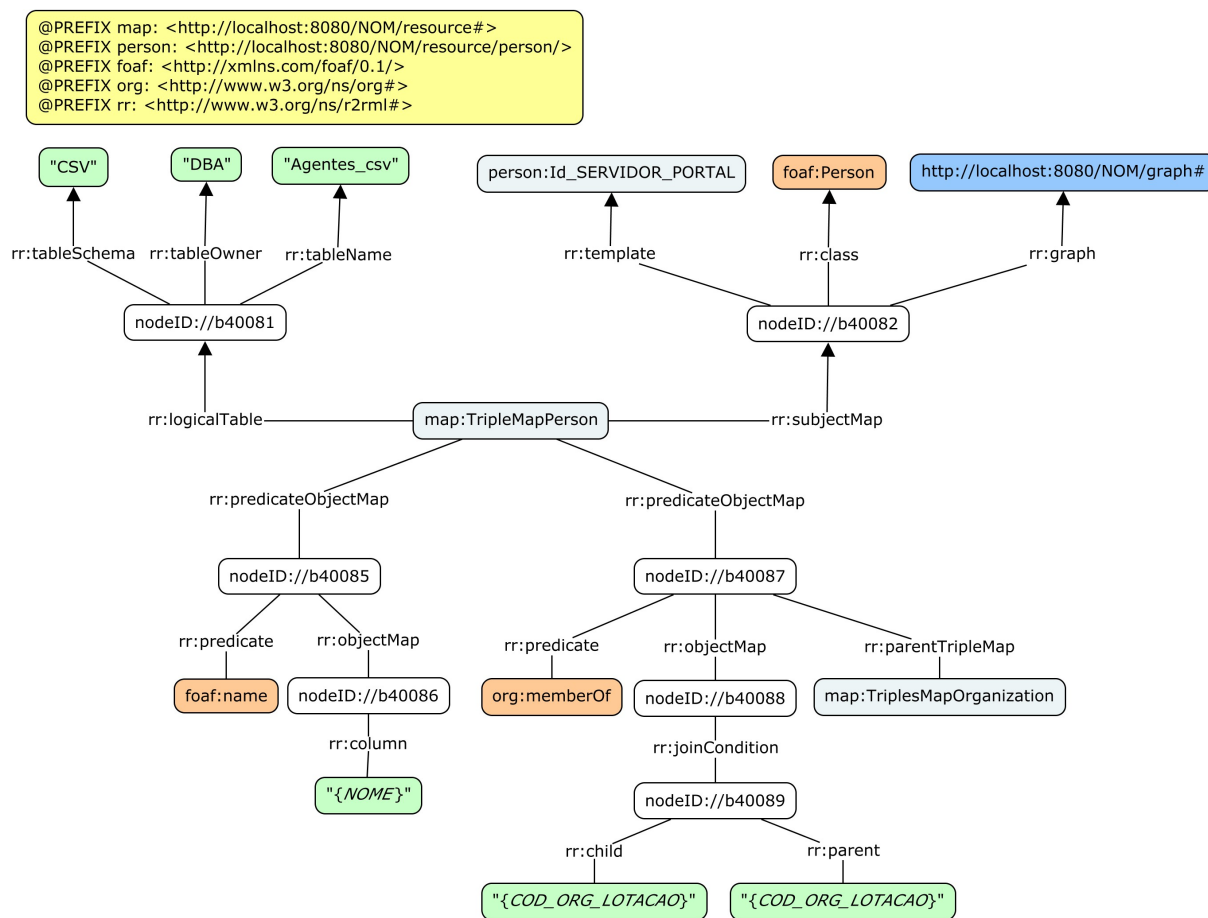


Figura 14 – Representação do grafo temporário resultante de parte do mapeamento de servidores na linguagem R2RML

Os nós representados pelo prefixo *nodeID* representam nós vazios que são criados automaticamente para conectar os elementos do grafo. Ao executar o mapeamento no Virtuoso, o resultado será inserido no grafo representado pelo grafo padrão `http://localhost:8080/NOM/graph#` indicado no mapa do sujeito. No entanto, esse grafo apenas simula um RDF *dataset*, ou seja, ele apenas busca dados em tabelas relacionais e transforma para triplas RDF. Isso faz com que, quando um elemento conecta a vários outros, as consultas retornem várias URIs iguais para este elemento ao invés de apenas uma que se conecta aos vários outros. Resumindo, esse grafo retorna triplas, mas essas triplas não estão conectadas como um grafo. A Figura 15 ilustra um exemplo de duas pessoas (P1 e P2) ligadas a mesma instituição UFES, onde do lado esquerdo (vermelho) apresenta a forma das triplas pelo grafo resultante do mapeamento e, a direita (verde), a forma como deveria ser representado.



Figura 15 – Exemplo da representação de duas triplas da forma feita pelo grafo resultante do mapeamento (esquerda – vermelho) e da forma como deveria ser (direita – verde)

Da forma apresenta a esquerda (vermelho) temos que UFES não representa um elemento único, ou seja, ele é apenas o valor relativo a uma linha de uma determinada coluna em uma tabela. Para se adequar a forma correta, ou seja, o elemento UFES passar a ser realmente uma URI que representa-o unicamente, é necessário inserir via SPARQL todas as triplas em um RDF *dataset* como indicado abaixo.

```
SPARQL INSERT IN GRAPH <http://localhost:8080/NOM/resource#>
  { ?sujeito ?predicado ?objeto }
FROM <http://localhost:8080/NOM/graph#>
WHERE { ?sujeito ?predicado ?objeto } ;
```

A inserção consiste em usar uma tripla com três variáveis (sujeito, predicado e objeto) que busca todas as triplas do grafo mapeado e insere em um RDF *dataset* definido por um grafo padrão (no caso, `http://localhost:8080/NOM/resource#`). Todas as triplas criadas no formato descrito anteriormente são inseridas de forma que não se tenha triplas repetidas. Se tivermos o caso de dois servidores ligados a uma mesma instituição, a ligação será na URI que representa unicamente esta instituição como ilustrado no grafo da direita (verde) do exemplo da Figura 15 apresentado acima. Logo, Assim, temos que a busca é feita em um grafo RDF e os elementos retornados são realmente URIs.

4 APLICAÇÃO DO MAPEAMENTO E USO DOS DADOS MAPEADOS

Através da implementação do mapeamento foi realizada uma aplicação local para publicar os dados, onde boa parte dos elementos gerados são representados por meio de URIs (em forma de URLs) utilizando o protocolo HTTP. A aplicação é capaz de derreferenciar estas URIs, ou seja, transmitir algum resultado quando as URIs são pesquisadas. Além disso, o banco de dados Virtuoso utilizado para o mapeamento possui um *SPARQL Endpoint* que pode ser acessado de forma programática (com consultas através de código) e de forma interativa (com consultas através de grafos visuais). Assim, foram realizadas consultas específicas sobre os dados, além de inferências. Este capítulo apresenta a aplicação para acessar as URLs e o conjunto de funcionalidades para consulta dos dados publicados.

4.1 URI DERREFERENCIADA

Com o intuito de demonstrar como os dados poderiam ser publicados, uma aplicação local foi desenvolvida de forma que seja possível derreferenciar as URIs (em forma de URLs). Ao acessar a URI são mostrados todas os elementos que se conectam diretamente a ela. Além disso, é possível que o usuário navegue entre estes elementos conectados obtendo as suas conexões. O usuário também pode navegar entre as conexões do elemento principal mostrado e, assim, ir descobrindo uma gama de informações. A Figura 16 mostra um exemplo de um derreferenciamento da URI seguinte, que trás os elementos conectados diretamente a ela:

```
http://localhost:8080/NOM/resource/person/1001212
```

Property	Value
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://xmlns.com/foaf/0.1/Person
http://www.w3.org/2004/02/skos/core#notation	1001212
http://xmlns.com/foaf/0.1/name	ANA CRISTINA RIBEIRO ALVIM
http://www.w3.org/ns/org#memberOf	http://localhost:8080/NOM/resource/organization/26411
http://www.w3.org/ns/org#memberOf	http://localhost:8080/NOM/resource/organizational-unit/CAMPUS%20BCENA%20-%20DIRETORIA%20ADMIN%20E%20PLANEJA
http://www.w3.org/ns/org#holds	http://localhost:8080/NOM/resource/post/CONTADOR
http://www.w3.org/ns/org#remuneration	http://localhost:8080/NOM/resource/remuneration/brl/2013/11/person/1001212
http://www.w3.org/ns/org#remuneration	http://localhost:8080/NOM/resource/remuneration/usd/2013/11/person/1001212

Figura 16 - Exemplo de URI Derreferenciada

Como resultado, observa-se que a primeira coluna (*Property*) trás os predicados de uma tripla e a segunda coluna (*Value*) os objetos. Os links (elementos em azul) representam outras URIs que podem ser também acessadas e, em caso de URIs que representam elementos mapeados das bases de dados (URI com o *localhost*) o resultado será uma tabela semelhante a apresentada na figura, com as conexões referentes ao elemento buscado. Além disso, para as URIs locais existe a possibilidade de ter a informação apresentada de outras formas. Por exemplo, ao acessar a mesma URI do exemplo anterior colocando no final “.html” ou “.rdf” ela resultará, respectivamente, em uma visualização em HTML (em forma de formulário ao invés de tabela) e RDF. As Figura 17 e Figura 18 exemplificam essa situação.

- PESSOA

NOME: ANA CRISTINA RIBEIRO ALVIM

ID: 1001212

ORGANIZAÇÃO: INSTITUTO FED.DO SUDESTE DE MINAS GERAIS

UNIDADE ORGANIZACIONAL: CAMPUS BCENA - DIRETORIA ADMIN E PLANEJA

CARGO: CONTADOR

REMUNERAÇÃO: R\$ 4080,31

Figura 17 – Visualização em formulário de uma URI derreferenciada com “.html”

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:skos="http://www.w3.org/2004/02/skos/core#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:org="http://www.w3.org/ns/org#" >
  <rdf:Description rdf:about="http://localhost:8080/NOM/resource/person/1001212">
    <org:memberOf rdf:resource="http://localhost:8080/NOM/resource/organization/264" />
    <foaf:name>ANA CRISTINA RIBEIRO ALVIM</foaf:name>
    <org:remuneration rdf:resource="http://localhost:8080/NOM/resource/remunerati" />
    <org:holds rdf:resource="http://localhost:8080/NOM/resource/post/CONTADOR" />
    <skos:notation rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1001212</skos:notation>
    <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Person" />
    <org:remuneration rdf:resource="http://localhost:8080/NOM/resource/remunerati" />
    <org:memberOf rdf:resource="http://localhost:8080/NOM/resource/organizational-u" />
  </rdf:Description>
</rdf:RDF>

```

Figura 18 – Parte da visualização em RDF de uma URI derreferenciada com “.rdf”

Essas formas de apresentação dos dados são podem ser úteis para situações específicas. Por exemplo, quem deseja trabalhar com dados em uma outra aplicação, pode buscar pelo dado em RDF, de manipulação mais fácil por ser melhor estruturado. Já um cidadão pode visualizar a informação de uma forma mais agradável, buscando pelo conteúdo em HTML na forma de formulário.

URIs que representam elementos dos vocabulários das ontologias, trarão como resultado suas definições diretamente ou, em alguns casos, todos os elementos da ontologia onde é possível buscar a definição do elemento utilizado. Por exemplo, ao acessar na URI *org:memberOf* o resultado será toda a *ORG Ontology* em RDF. Então, é possível verificar na ontologia a descrição do predicado *memberOf* como mostra a Figura 19.

```

org:memberOf a owl:ObjectProperty, rdf:Property;

    rdfs:label "member of"@en;
    rdfs:label "membre de"@fr;
    rdfs:label "membro di"@it;

    rdfs:domain foaf:Agent;
    rdfs:range org:Organization;

    rdfs:comment """Indicates that a person is a member of the Organization with no indication of
the nature of that membership or the role played. Note that the choice of property name is not meant
to limit the property to only formal membership arrangements, it is also intended to cover related
concepts such as affiliation or other involvement in the organization. Extensions can specialize
this relationship to indicate particular roles within the organization or more nuanced relationships
to the organization. Has an optional inverse, `org:hasmember`."""@en;
    rdfs:comment """Indique qu'une personne est membre de l'Organisation sans précision sur la
nature de cet engagement ou du rôle joué. Notez que le choix du nom de cette propriété ne vise pas à
la limiter aux seuls engagements formels, elle peut également couvrir des concepts reliés comme
l'affiliation ou le bénévolat. Des extensions peuvent spécialiser cette relation pour indiquer des
rôles particuliers au sein de l'Organisation or des relations plus nuancées avec elle. Possède une
propriété inverse optionnelle, `org:hasmember`."""@fr;
    rdfs:comment """Indica che una persona è membro di una Organization senza una precisa
indicazione sulla natura di questa appartenenza e sul suo ruolo. Si noti che la scelta del nome di
questa proprietà non intende limitarla alla sola rappresentazione formalmente di un'appartenenza.
La proprietà può coprire anche altri coinvolgimenti nell'organizzazione. Questa proprietà può essere
specializzata per indicare ruoli all'interno organizzazione o relazioni di diverse tipologie. Ha
`org:hasmember` come proprietà inversa opzionale."""@it;
    rdfs:comment "エージェント（人または他の組織）が組織のメンバーであることを示します。ただし、その構成員の本質や担
う役割は示しません。プロパティ名の選択は、プロパティを正式な構成員配置のみに制限することが目的ではないことに注意して
ください。所属や組織へのその他の関与などの関連する概念をカバーすることも意図されています。拡張により、この関係を特殊化
し、組織内の特定の役割やよりニュアンスを含んだ組織との関係を示すことができます。"@ja;
    rdfs:isDefinedBy <http://www.w3.org/ns/org> ;
    .

org:memberOf rdfs:label "es miembro de"@es ;
    rdfs:comment "Persona que pertenece a la organización o es miembro de la misma, sin que conste
la naturaleza de dicha pertenencia o el papel que desempeña. Se debe tener en cuenta que la elección
de una denominación para esta propiedad no significa que la propiedad esté limitada a ciertos tipos
de pertenencia formales, sino que pretende cubrir conceptos relacionados como el de afiliación u
otras formas de participación en la organización. Se puede hacer uso de extensiones para
especializar esta relación de forma que incluya tipos específicos de pertenencia a las
organizaciones o relaciones especiales con la organización."@es .

```

Figura 19 – Descrição de *memberOf* na ORG Ontology

Através disso, é possível ver como o termo é definido na *ORG Ontology* e ter um entendimento melhor sobre os dados publicados. Por exemplo, ao acessar alguma URI de uma organização e ver que ela é um tipo da classe Organização Formal da *ORG Ontology* é possível saber, através de sua definição, que aquela organização específica representa uma organizações reconhecidas mundialmente (como corporações, instituições de caridade, governo ou igreja) através de jurisdições legais, direitos e responsabilidades associadas. Dessa forma, quem deseja integrar outras bases a estas, por exemplo, teria maior facilidade para realizar esse procedimento visto que seria necessário associar os elementos das novas bases aos termos das ontologias e, com isso, saber a qual grupo ele pertence.

4.2 SPARQL ENDPOINT

Para quem deseja buscar um conjunto de informação específicas, é possível realizar consultas SPARQL através de um *SPARQL Endpoint*. O banco de dados Virtuoso fornece um *SPARQL Query Editor* que pode ser acessado via uma URL específica (por exemplo, <http://localhost:8080/NOM/sparql>). Nele é possível realizar diversas consultas e ter o resultado em diversos formatos (RDF, HTML, CSV, etc.). A Figura 20 apresenta a tela do *SPARQL Query Editor*.

Virtuoso SPARQL Query Editor

[About](#) | [Namespace Prefixes](#) | [Inference rules](#) | [iSPARQL](#)

Default Data Set Name (Graph IRI)

Query Text

Sponging:

Results Format:

Execution timeout: milliseconds (values less than 1000 are ignored)

Options: Strict checking of void variables

(The result can only be sent back to browser, not saved on the server, see [details](#))

Copyright © 2014 [OpenLink Software](#)
 Virtuoso version 06.01.3.127 on Darwin (x86_64-apple-darwin13.1.0), Single Server Edition

Figura 20 – Tela do SPARQL Query Editor

Inicialmente, é fornecida a URI que contém o grafo do mapeamento onde será buscada a informação da consulta realizada. Em seguida, o editor da consulta permite criar a consulta SPARQL que retornará a informação desejada. Por fim, há opções avançadas, com as quais é possível definir o formato em que será retornado o resultado da consulta. Assim, como dito na seção anterior, os resultados podem ser disponibilizados em diversos formatos como: RDF, CSV e HTML (tabelas) como mostra a Figura 21 para o resultado de uma consulta em formato HTML com tabelas.

person	name
http://localhost:8080/NOM/resource/person/1725833	OSVALDO CRUZ
http://localhost:8080/NOM/resource/person/1915869	OSVALDO CRUZ
http://localhost:8080/NOM/resource/person/1005584	ANTONIO DIAS
http://localhost:8080/NOM/resource/person/1927118	LUIZ CONCEICAO
http://localhost:8080/NOM/resource/person/1021937	RICARDO SILVA PEREIRA

Figura 21 - Exemplo de resultado de uma consulta SPARQL com formato HTML

O resultado em HTML trará as URIs, que são derreferenciadas pela aplicação, se acessadas, fornecendo as sua ligações, como apresentado na seção anterior.

A Figura 20 apresenta no canto superior direito um link para o SPARQL interativo (iSPARQL). O iSPARQL é semelhante ao SPARQL Query Editor com uma nova interface e novas opções. Nele é possível importar esquemas e construir consultas visuais com grafos. A Figura 22 mostra a aba em que é possível criar esse tipo de consulta.

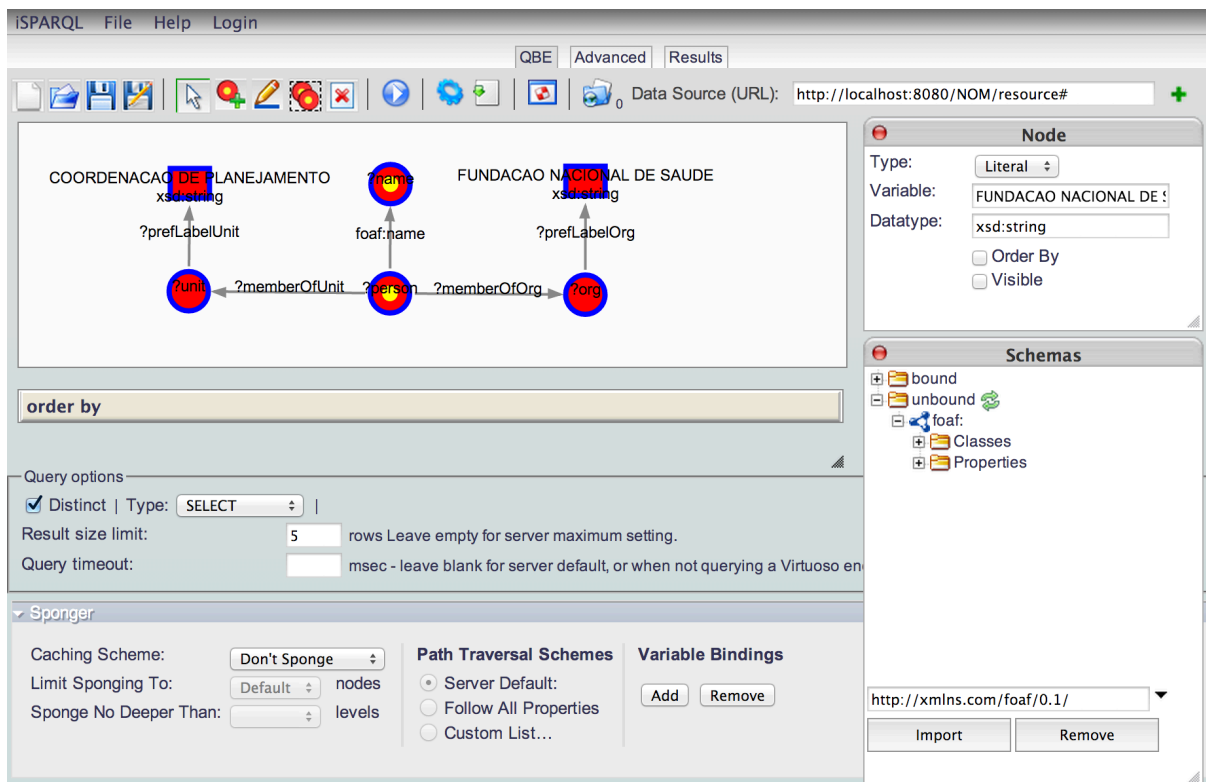


Figura 22 – Aba do iSPARQL para criação de consultas SPARQL com grafo

Nesta aba, é possível realizar consulta visual inserindo os vértices (sujeito e objeto) e arestas (predicados). Na paleta Node é possível editar os vértices e arestas com os valores desejados (tipo, varável, tipo de dado, ordenado por e visível). Já na paleta Schema é possível importar as ontologias e utilizadas na consulta com seus prefixos. Além disso, a consulta possui opções como: Consulta distinta (*Distinct*), tipo da consulta (*SELECT*, *INSERT*, *DELETE*, etc.) e limite da quantidade de elementos resultantes (*Result size limit*) e do tempo para realizar a consulta (*Query timeout*). Então, temos como resultado uma tabela com os valores das variáveis desejadas. A Figura 23 mostra o resultado para a consulta descrita na Figura 22, que retorna a URI e o nome dos servidores que são membros da Coordenação de Planejamento da Fundação Nacional de Saúde.

name	personURI
ANTONIO SERGIO SUZANO LOQUES BARROSO	http://localhost:8080/NOM/resource/person/1849666
CICERO ALEXANDRE GOSS	http://localhost:8080/NOM/resource/person/1948260
FABRICIO ROCHA CORE	http://localhost:8080/NOM/resource/person/1934274
MARIOSAN MACIEL	http://localhost:8080/NOM/resource/person/1000032

Figura 23 – Resultado em tabela de uma consulta SPARQL no iSPARQL

Alterando a modo de derreferenciamento (*Derreferencing*) de “*SPARQL Describe*” para “*SELECT IRI as S or O*” é possível navegar sobre os elementos de forma semelhante a aplicação criada, ou seja, ao clicar em uma URI ele retorna os elementos ligados diretamente à URI consultada. O iSPARQL não permite trazer a consulta em outros formatos (como RDF ou CSV) como o *SPARQL Query Editor*, dessa forma, ele é apenas uma forma diferente de visualização e criação de consultas.

4.3 INFERÊNCIA

Por fim, temos a possibilidade de realizar inferência sobre os dados, ou seja, realizar raciocínio automático através de regras definidas para descobrir novas relações, i.e., relações não anteriormente mostradas entre os elementos do grafo. Essa descoberta é feita através de análise sobre os dados existentes e buscando, através de suas relações, novas triplas com base na regra definida. Por exemplo, se existe uma regra que define que a relação “filho de” é inversa da relação “pai de” é possível descobrir novas relações apenas com triplas que utilizem um desses predicados. Logo, se tivermos, por exemplo, a tripla “João – filho de – Pedro” é possível inferir a tripla “Pedro – pai de – João” através da regra de inversa definida.

Como exemplo no mapeamento realizado, temos que não é possível saber, através de uma consulta direta, tomando como sujeito a organização, os membros

de uma determinada organização. Isso se deve ao fato da implementação ser feita em apenas uma direção, ou seja, sem a relação inversa entre os elementos. Para saber quem são os membros de uma determinada organização é necessário consultar tomando como sujeito as pessoas e utilizando o predicado “membro de” (*org:memberOf*) como demonstrado abaixo.

```
PREFIX org: <http://www.w3.org/ns/org#>
select count(?person) where {
  ?person org:memberOf ?org .
  ?org skos:prefLabel "FUNDAÇÃO NACIONAL DE SAÚDE" .
}
```

Essa consulta pode ser feita com uso de inferência tomando a organização como sujeito para saber quantos membros ela tem. Com isso, a inferência seria realizada criando a relação inversa de “membro de” entre organização e pessoa (definida na Ontologia de Organização), ou seja, criando a relação “tem membro” (*org:hasMember*). O virtuoso permite inserir grafos que possuem regras para realização de consultas SPARQL com inferência. Abaixo temos uma consulta onde é definido na regra de inferência o grafo da *ORG Ontology* (<http://www.w3.org/ns/org#>) que possui as relações inversas dos predicados usados no mapeamento.

```
DEFINE input:inference <http://www.w3.org/ns/org#>
PREFIX org: <http://www.w3.org/ns/org#>
SELECT COUNT(?person) WHERE {
  ?org org:hasMember ?person .
  ?org skos:prefLabel "FUNDAÇÃO NACIONAL DE SAÚDE" .
}
```

Logo, com a regra definida, a consulta trará como resultado o mesmo valor que a consulta anterior. No caso, 13254 pessoas que são membro da organização Fundação Nacional de Saúde.

Além das inferências nas consultas, é possível realizar inferências com base nos dados em RDF através do *Framework Apache Jena* (disponível em <https://jena.apache.org/>) para aplicações na linguagem *Java* usando o *Pellet*

(disponível em <http://clarkparsia.com/pellet/>) para raciocínio automático. Com esta ferramenta é possível realizar raciocínio sobre todo o grafo, gerando um novo grafo com o adicional das relações geradas. Com isso, é possível realizar consultas SPARQL a partir do novo grafo gerado.

5 DISCUSSÕES

Neste capítulo é apresentada uma discussão sobre as dificuldades encontradas na abordagem para realizar a integração dos dados governamentais brasileiros com o uso de mapeamento para a *ORG Ontology* utilizando-se tecnologias semânticas. Os benefícios encontrados e outros potenciais benefícios desse tipo de abordagem são discutidos nessa seção.

5.1 DIFICULDADES ENCONTRADAS

No decorrer deste trabalho, algumas dificuldades para realização de um mapeamento mais preciso foram identificadas principalmente devido a problemas nas bases de dados decorrentes da falta de padronização para o seu conteúdo, o que dificulta uma análise e entendimento mais preciso dos dados e a pouca (ou nenhuma) integração entre as diferentes bases de dados selecionadas. Esta seção apresenta os problemas encontrados para realização do mapeamento.

5.1.1 Problemas nas Bases de Dados

Inicialmente, temos que a base de dados de servidores civis não possui identificadores em alguns de seus dados. Dessa forma, ao mapear bases de dados diferentes e tentar buscar elementos que representam as mesmas entidades (órgãos, pessoas, etc.) em ambas bases, é necessário realizar uma comparação entre dados iguais de cada base e que, no caso de dados sem identificadores, deverá ser feita pelo nome do elemento. Com isso, existe a possibilidade de elementos que potencialmente denotam a mesma coisa, porém são representados de alguma forma diferente (abreviações ou algum caractere diferente por exemplo) não serem mapeados como o mesmo elemento, pois a comparação de nomes entre eles será distinta. Esse mesmo problema pode ocorrer com elementos de uma mesma base que tem denotações diferentes, mas que podem representar a mesma coisa. Como exemplo, temos 4 casos indicados na Tabela 16 em que cargos da

base de dados de Servidores Civis são mapeados como sendo diferentes, mas que aparentemente representam a mesma coisa, possuindo diferença na sua digitação.

Tabela 16 – Exemplo de Cargos semelhantes sem identificadores da base de dados de Servidores Civis

CARGO
AGENTE DE SEGURANCA FERROVIARIA
AGENTE DE SEGURANCA FERROVIARIO
AGENTE DE TELEC E ELETRICIDADE
AGENTE DE TELEC ELETRICIDADE
AGENTE DE TELECOMUNI E ELETRICIDADE
AGENTE DE TELECOMUNIC E ELETRICIDADE
AGENTE DE TRANSP MARITIMO E FLUVIAL
AGENTE DE TRANSP MARITIMO FLUVIAL

Esse acontecimento é visto como um problema, pois não é possível saber exatamente se os elementos representam a mesma coisa ou não. Isso diminui a precisão de relatórios e estatísticas para esses dados, Além de prejudicar o mapeamento com outras bases, dado que a representação em ambas pode apresentar distinções nos caracteres.

Uma abordagem para resolver esse tipo de problema, seria o uso de identificadores, onde elementos semanticamente iguais teriam o mesmo identificador mesmo que com nomes diferentes e elementos diferentes seriam identificados de forma diferente, mesmo que tenham nomes iguais. Por exemplo, caso Agente de Segurança Ferroviária e Agente de Segurança Ferroviário representem o mesmo cargo, ambos poderiam ter um mesmo identificador e, com isso, serem tratados como o mesmo elemento.

No entanto, em alguns casos, apenas o uso de identificadores não seria o suficiente para distinguir elementos diferentes ou iguais. A Tabela 17 ilustra essa situação. A atividade de “Assessor Chefe” da Agência Nacional do Cinema é representada com código diferente da atividade de “Assessor Chefe” do Instituto

Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis, supostamente representando atividades diferentes apesar de possuírem nomes iguais.

Tabela 17 - Exemplos da atividade “Assessor Chefe” representada com códigos diferentes

CODIGO_ATIVIDADE	ATIVIDADE	ORG_LOTACAO
25	ASSESSOR CHEFE	AGENCIA NACIONAL DO CINEMA
3675	ASSESSOR CHEFE	INST. BR. MEIO AMB. REC. NAT. RENOVAVEIS

Um forma de diferenciar estes elementos seria através da descrição do conjunto de atribuições dos ocupantes destes cargos ou atividades. Tais atribuições são definidas por lei, no entanto, não são apresentadas nos dados publicados pelo governo. Se as atribuições fossem publicadas, a semântica dos dados seria melhor representada e conseqüentemente, seria mais fácil para distinguir de forma mais precisa os dados como os do exemplo apresentado.

Um outro exemplo da situações retratada na Tabela 17 pode ser visto na base de dados de Estruturas Organizacionais, indicado na Tabela 18. Neste exemplo, o Ministério da Fazenda é cadastrado com diversos códigos.

Tabela 18 - Diferentes identificadores para Ministério da Fazenda na base de dados de Estruturas Organizacionais

DADOS_CADASTRO_NOME	DADOS_CADASTRO_CODIGO	DADOS_CADASTRO_CODIGO_PAI
MINISTERIO DA FAZENDA	1929	26
MINISTERIO DA FAZENDA	118699	1929
MINISTERIO DA FAZENDA	118700	1929
MINISTERIO DA FAZENDA	118701	1929
MINISTERIO DA FAZENDA	118702	1929
MINISTERIO DA FAZENDA	118703	1929
MINISTERIO DA FAZENDA	118704	1929
MINISTERIO DA FAZENDA	118708	1929
MINISTERIO DA FAZENDA	118709	1929
MINISTERIO DA FAZENDA	118710	1929

Nesta situação, temos aparentemente vários “Ministérios da Fazenda” cadastrados com diferentes códigos. Não é possível saber se os elementos representam a mesma coisa ou coisas diferentes. A coluna

DADOS_CADASTRO_CODIGO_PAI leva a crer que os registros descritos da segunda linha em diante são filhos (sub organizações) do Ministério da Fazenda descrito com o identificador 1929. No portal do SIORG verifica-se que as demais linhas representam órgãos vinculados ao Ministério da Fazenda, como ilustra a Figura 24 para o caso do Órgão de identificador 118699.

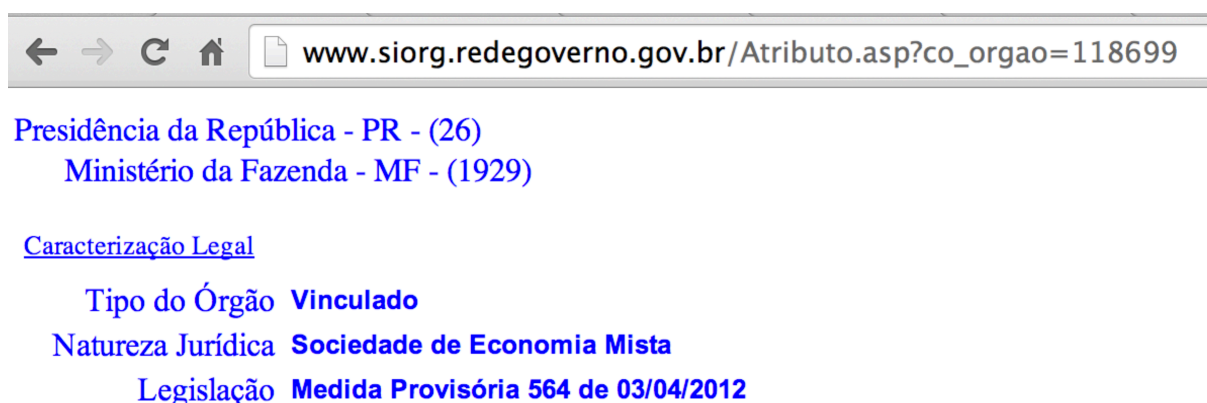


Figura 24 - Consulta no portal do SIORG aos códigos filhos do Ministério da Fazenda

Esse tipo de representação gera outro problema que é a integração com outras bases de dados. Esse problema é ainda agravado quando os elementos a serem mapeados possuem identificadores diferentes, ou seja, quando em uma base o dado possui um determinado número identificador e na outra base o dado que representa o mesmo elemento possui um outro número identificador. Isso ocorre nas bases de dados de estruturas organizacionais e servidores civis onde a primeira representa seus órgão com identificadores diferentes dos identificadores representados na segunda. Logo, para mapear dados destas bases é necessário verificar se os nomes dos órgãos são iguais e, então, criar uma relação entre os elementos das diferentes bases que indica que os elementos são os mesmos. Dessa forma, novos problemas são encontrados. Primeiro, a verificação dos nomes é feita com uma comparação de igualdade entre todos os caracteres, ou seja, qualquer diferença na acentuação, tipografia ou espaçamento no nome tornam esses elementos diferentes e o mapeamento desses dados como sendo iguais não é efetuado. Segundo, dados como os mostrados na Tabela 18, que possuem nomes iguais, mas identificadores diferentes, acabam sendo mapeados como sendo um mesmo elemento de outra base devido a comparação ser feita através do nome. Reitera-se que isso empobrece as comparações resultantes possíveis, pois torna

inacessíveis informações sobre os órgãos subordinados (anteriormente referenciados como ‘filhos’) ao órgão superior.

Dificuldades também ocorreram na realização do mapeamento em relação às bases de dados. No estudo apresentado, um problema ocorreu ao integrar as bases de servidores e estruturas organizacionais, onde na primeira os dados são todos em letras maiúsculas e sem acentos, e na segunda os dados possuem acentos e letras tanto em maiúsculo quanto em minúsculo. A Tabela 19 e a Tabela 20 mostra o exemplo em que a Presidência da República é descrita de forma diferente na base de dados de Servidores Civis e na base de dados de Estruturas Organizacionais.

Tabela 19 – Representação para Presidência da República na base de dado de Servidores Civis

DADO SERVIDORES CIVIS	
20101	PRESIDENCIA DA REPUBLICA

Tabela 20 – Representação para Presidência da República na base de dado de Estruturas Organizacionais

DADO ESTRUTURAS ORGANIZACIONAIS	
26	Presidência da República

Com isso, uma abordagem para realizar o mapeamento seria realizando a adequação das bases entre si. A forma para realização da adequação, foi efetuar a verificação e alteração dos caracteres como detalhado na seção 3.7. Neste caso, para realização do mapeamento foi feita a alteração na segunda base de dados para se adequar a primeira.

5.1.2 Problema da Falta de Identificação Única Entre Todos os Entes Federais

Outro problema encontrado, foi de mapear de maneira correta as unidades organizacionais devido a falta de elementos que as ligam (como identificadores por exemplo). Dessa forma, como o mapeamento é feito através do nome das unidades organizacionais, quando ligadas a sua instituição, várias unidades organizacionais de mesmo nome, porém de instituições diferentes, são tratadas como uma só. A

Figura 25 ilustra esse situação, onde é mostrado o resultado de uma consulta SPARQL que retorna todos os elementos ligados com a seguinte URI:

<http://localhost:8080/NOM/resource/organizational-unit/COORDENACAO%20DE%20PLANEJAMENTO>.

Property	Value
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/ns/org#OrganizationalUnit
http://www.w3.org/2004/02/skos/core#prefLabel	COORDENACAO DE PLANEJAMENTO
http://www.w3.org/ns/org#unitOf	http://localhost:8080/NOM/resource/organization/36205
http://www.w3.org/ns/org#unitOf	http://localhost:8080/NOM/resource/organization/25000
http://www.w3.org/ns/org#unitOf	http://localhost:8080/NOM/resource/organization/40112
http://www.w3.org/ns/org#unitOf	http://localhost:8080/NOM/resource/organization/45203
http://www.w3.org/ns/org#unitOf	http://localhost:8080/NOM/resource/organization/20224
http://www.w3.org/ns/org#unitOf	http://localhost:8080/NOM/resource/organization/28000
http://www.w3.org/ns/org#unitOf	http://localhost:8080/NOM/resource/organization/26442

Figura 25 – Exemplo de problema no mapeamento resultante de unidades organizacionais

A figura mostra que o termo “Coordenação de Planejamento” é mapeado com 7 organizações diferentes. Isso implica que não é possível saber de fato a qual organização ela pertence, dado que podem existir várias instituições diferentes que possuem uma coordenação de planejamento.

Com o problema apresentado acima, seria inviável saber a partir dos dados qual organização cuja unidade organizacional um ou vários servidores pertencem, pois as unidades organizacionais são mapeadas como sendo a mesma e acabam por ser ligadas a várias organizações como dito anteriormente.

No entanto, através de uma consulta que retorne os elementos ligados ao servidor, temos a unidade organizacional e organização que ele pertence. Com isso, é possível deduzir que a unidade organizacional pertence a esta organização. A Figura 26 exemplifica essa situação onde é feito uma consulta que retorna todos os elementos ligados a seguinte URI:

<http://localhost:8080/NOM/resource/person/1000032>

Property	Value
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://xmlns.com/foaf/0.1/Person
http://xmlns.com/foaf/0.1/name	MARIOSAN MACIEL
http://www.w3.org/2004/02/skos/core#notation	1000032
http://www.w3.org/ns/org#memberOf	http://localhost:8080/NOM/resource/organizational-unit/COORDENACAO%20DE%20PLANEJAMENTO
http://www.w3.org/ns/org#memberOf	http://localhost:8080/NOM/resource/organization/36205
http://www.w3.org/ns/org#holds	http://localhost:8080/NOM/resource/post/ADMINISTRADOR
http://www.w3.org/ns/org#remuneration	http://localhost:8080/NOM/resource/remuneration/brl/2013/11/person/1000032
http://www.w3.org/ns/org#remuneration	http://localhost:8080/NOM/resource/remuneration/usd/2013/11/person/1000032

Figura 26 - Exemplo de mapeamento resultante de servidor

Observando a imagem, é possível ver que o servidor Mariosan Maciel é membro da Coordenação de Planejamento que é unidade organizacional da organização Fundação Nacional de Saúde que é representada pela URI <http://localhost:8080/NOM/resource/organization/36205> como mostra a Figura 27.

Property	Value
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/ns/org#FormalOrganization
http://www.w3.org/2002/07/owl#sameAs	http://localhost:8080/NOM/resource/organization-structure/2207
http://www.w3.org/2004/02/skos/core#prefLabel	FUNDACAO NACIONAL DE SAUDE
http://www.w3.org/2004/02/skos/core#notation	36205
http://www.w3.org/ns/org#subOrganizationOf	http://localhost:8080/NOM/resource/organization/25000

Figura 27 - Exemplo do mapeamento resultante de Organização

Com isso, é possível saber por exemplo, todos os servidores que estão lotados na unidade organizacional Coordenação de Planejamento da Fundação Nacional de Saúde mesmo que essa Coordenação do Planejamento seja representada como sendo a mesma unidade de várias organizações. A Tabela 21 mostra a consulta SPARQL que retorna o nome e o URI dos servidores do exemplo apresentado.

Tabela 21 – Consulta SPARQL para retornar todos os servidores da unidade organizacional Coordenação de Planejamento pertencente a Fundação Nacional de Saúde

```
SELECT ?name ?personURI
WHERE {
  ?personURI ?memberOfUnit ?unit .
  ?unit ?prefLabelUnit "COORDENACAO DE PLANEJAMENTO" .
  ?personURI ?memberOfOrg ?org .
```

```

?org ?prefLabelOrg "FUNDAÇÃO NACIONAL DE SAÚDE" .
?personURI foaf:name ?name .
}
ORDER BY (?name)

```

A Tabela 22 apresenta o resultado da consulta.

Tabela 22 – Resultado da consulta SPARQL

name	personURI
ANTONIO SERGIO SUZANO LOQUES BARROSO	http://localhost:8080/NOM/resource/person/1849666
CICERO ALEXANDRE GOSS	http://localhost:8080/NOM/resource/person/1948260
FABRICIO ROCHA CORE	http://localhost:8080/NOM/resource/person/1934274
MARIOSAN MACIEL	http://localhost:8080/NOM/resource/person/1000032

Uma solução para o problema potencial seria uma alteração dos dados pelo seu provedor (ou seja, os órgão que fornecem estes dados) de forma a adicionar um identificador para cada unidade organizacional, onde, na realização do mapeamento, seria possível diferenciar as organizações às quais cada unidade organizacional é subordinada, resultado assim em um mapeamento mais preciso. Com isso seria possível descobrir com uma consulta simples (1 tripla apenas) qual unidade organizacional determinado servidor é membro.

5.1.3 Problema de integração com identificadores diferentes entre entes do Governo Federal

Outro problema encontrado no mapeamento é na integração de diferentes bases de dados nas quais elementos semelhantes de ambas as bases apresentam identificadores distintos. Dessa forma, ao realizar o mapeamento, necessita ser feita uma comparação entre os nomes dos elementos. Isso acarreta em um problema ao integrar bases diferentes, pois, elementos identificados diferentes em uma mesma base, mas que apresentam o mesmo nome, são mapeados como sendo o mesmo elemento da outra base.

A Figura 28 ilustra esse problema onde Ministério da Fazenda da base de dados de servidores é dito como igual a todos os elementos representados como Ministérios da Fazenda da base de dados de estruturas organizacionais.

Property	Value
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/ns/org#FormalOrganization
http://www.w3.org/2002/07/owl#sameAs	http://localhost:8080/NOM/resource/organization-structure/1929
http://www.w3.org/2002/07/owl#sameAs	http://localhost:8080/NOM/resource/organization-structure/118703
http://www.w3.org/2002/07/owl#sameAs	http://localhost:8080/NOM/resource/organization-structure/118704
http://www.w3.org/2002/07/owl#sameAs	http://localhost:8080/NOM/resource/organization-structure/118699
http://www.w3.org/2002/07/owl#sameAs	http://localhost:8080/NOM/resource/organization-structure/118700
http://www.w3.org/2002/07/owl#sameAs	http://localhost:8080/NOM/resource/organization-structure/118701
http://www.w3.org/2002/07/owl#sameAs	http://localhost:8080/NOM/resource/organization-structure/118708
http://www.w3.org/2002/07/owl#sameAs	http://localhost:8080/NOM/resource/organization-structure/118709
http://www.w3.org/2002/07/owl#sameAs	http://localhost:8080/NOM/resource/organization-structure/118710
http://www.w3.org/2004/02/skos/core#prefLabel	MINISTERIO DA FAZENDA
http://www.w3.org/2004/02/skos/core#notation	17000
http://www.w3.org/ns/org#subOrganizationOf	http://localhost:8080/NOM/resource/organization/17000

Figura 28 – Exemplo de problema em mapeamento por nome em duas bases de dados diferentes

Nessa situação, temos vários elementos diferentes, advindos da base de estruturas organizacionais, com o mesmo nome Ministério da Fazenda (como mostrado na Tabela 18). Com isso, ao mapear o Ministério da Fazenda da base de servidores, ele é ligado como sendo igual (*owl:sameAs*) a todos os Ministérios da Fazenda da base de estruturas organizacionais (representados com URI contendo o caminho “*./organization-structure/.*”), pois ambos tem o mesmo nome. Isso prejudica a semântica do mapeamento, pois, mesmo sabendo que estes elementos são diferentes, eles são mapeados como sendo iguais devido a identificação ser diferente e o mapeamento ser realizado através da comparação dos nomes.

5.2 BENEFÍCIOS

Com a realização do mapeamento, diversos benefícios foram encontrados. Dentre eles temos: melhor integração devido ao uso de ontologias bem definidas,

uso de termos de vocabulários que são amplamente utilizados, visibilidade e consulta melhor dos dados e a possibilidade de realização de inferência sobre os dados. Esta seção apresenta os benefícios com a realização do mapeamento feito neste trabalho.

5.2.1 Melhor integração com uso de ontologias bem definidas

A integração é o maior benefício do mapeamento. Dados ligados a ontologias possuem semântica tipicamente melhor definida. Como ontologias possuem uma taxonomia que define classes de objetos e as relações entre eles, ao associar elementos das bases de dados com essas classes é possível criar relações entre os dados através das relações definidas pela ontologia, ou seja, dados que são mapeados a classes que se relacionam na ontologia, passam a ter essa mesma relação entre eles. Imagine que temos que integrar duas aplicações diferentes sobre organizações, com características diferentes, mas que usam a *ORG Ontology* para ligar os seus termos. Então, temos que ambas tem elementos sobre organizações pelo qual estão representadas como uma **Organização Formal**. Assim, conclui-se que esses elementos representam organizações reconhecidas mundialmente (como corporações, instituições de caridade, governo ou igreja) através de jurisdições legais, direitos e responsabilidades associadas. Logo, ao criar uma nova aplicação que une essas duas, saberemos que esses elementos devem ser tratados da mesma forma. A Figura 29 ilustra essa situação, onde organizações de aplicações diferentes são mapeadas ao conceito de **Organização Formal** (*Formal Organization*) da *ORG Ontology* e, ao unir as bases, os elementos associados ao conceito são unidos e tratados da mesma forma.

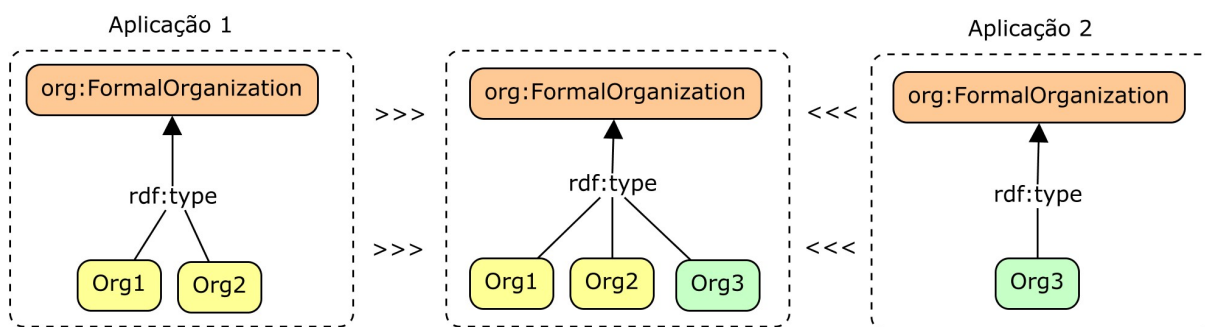


Figura 29 - Exemplo de integração entre aplicações com mesma classificação

A mesma situação pode ocorrer entre aplicações ou dados sem essas classificações como os utilizados neste trabalho. Basta utilizar as ontologias para classificar ambos vocabulários e depois unir os elementos que possuem a mesma classificação.

5.2.2 Termos de vocabulários muito utilizados

Outro benefício do mapeamento é ter os dados classificados com termos de diversos vocabulários bem definidos e que são amplamente utilizados. Isso facilita a recuperação, entendimento e integração de dados com outras fontes. Vocabulários de ontologias como FOAF, SKOS e GoodRelation são exemplos de vocabulários utilizados no mapeamento que possuem seus termos relativamente bem definidos e utilizados em larga escala¹. Logo, é possível ao consultar um dado, verificar em qual dessas ontologias ele é classificado e, assim, buscar a definição no vocabulário correto. Além disso, como essas ontologias são amplamente utilizadas³ a chance de mais pessoas entenderem o que representa aquela informação publicada é maior.

5.2.3 Melhor visibilidade e consulta dos dados

Com o uso de URIs para os elementos, é possível ver com mais facilidade cada entidade unicamente e os elementos com os quais ela se conecta. Para um usuário comum, isso torna a busca de conteúdo simples (como o endereço de algum lugar,

³ Grandes empresas Google, BBC e BestBuy são exemplos de utilizadores.

ou o local de trabalho de determinado servidor) melhor, pois não há a necessidade de baixar todo o arquivo para obter determinada informação. Além do conteúdo apresentar ser visivelmente melhor, pois ele poderia acessar com a URI na forma com o final “.html”, descrita na seção 4.1, e ter uma informação clara dos dados. Já usuários experientes poderiam também utilizar o *SPARQL Endpoint* para buscar uma gama de informações desejadas sem ter a necessidade de, também, ter que baixar todo o conteúdo das bases de dados.

5.2.4 Inferência

Por fim, temos a possibilidade de realizar inferência sobre os dados, ou seja, realizar raciocínio automático através de regras definidas de forma a descobrir novas relações entre os elementos do grafo. As regras indicam como os dados serão analisados e o que pode ser gerado através deles. Dessa forma, é possível descobrir relações entre dados que se conectam ou não, diretamente ou indiretamente, permitindo buscar conexões, em alguns casos, que não são tão simples de serem observadas. Dessa forma, a inferência é uma ferramenta poderosa, pois permite gerar uma gama de informação automaticamente e, conseqüentemente, ter uma análise mais precisa dos dados.

6 CONCLUSÃO

Dados governamentais são frequentemente publicados de forma não estruturada ou em formatos não padronizados. Além disto, muitas vezes diferentes bases de dados utilizam diferentes formatos, havendo pouca ou nenhuma integração entre elas. O uso de ontologias, que são utilizadas por órgãos internacionais de padronização, traz uma forma de solução para a integração dos dados governamentais na web. Como ontologias definem classes de objetos e suas relações, ao se associar elementos das bases de dados com classes definidas em ontologias é possível definir ligações entre estes elementos através das relações definidas pela ontologia. Isso permitiria uma melhor integração entre bases de dados diferentes ao associar elementos que estão ligados a uma mesma classe. A publicação dos dados é feita na web semântica utilizando-se os princípios dos dados ligados definidos por Berners-Lee (2006). Estes princípios definem que as instâncias sejam representadas por URIs com protocolo HTTP para que sejam acessíveis na web. Os *links* para outros recursos também são representados com URIs, permitindo que as pessoas possam navegar pelos dados e descobrir informações.

Através da linguagem de mapeamento R2RML, foi possível gerar um grande grafo RDF (com mais de 9,4 milhões de triplas) a partir das bases de dados selecionadas. No entanto, a precisão do grafo gerado foi comprometida devido a limitações nas bases de dados, por possuírem um conteúdo confuso e pouca ou nenhuma integração. Isso mostrou a necessidade de melhorias na publicação dos dados pelo governo, uma vez que a padronização para geração do conteúdo e a interoperabilidade entre os sistemas das organizações responsáveis por gerar estes dados é pouca ou inexistente. Foram definidas estratégias para execução do mapeamento; para reduzir o tempo de execução do mapeamento, houve a necessidade de dividir uma base de dados em várias outras com o intuito de eliminar uma quantidade de dados repetidos e, conseqüentemente, reduzir o número de comparações feitas durante a execução do mapeamento.

Apesar disso, a realização do mapeamento trouxe diversos benefícios. A integração foi o principal deles, pois os dados foram mapeados a termos de

ontologias bem definidas e que são recomendações do W3C, o que permite entendimento e possibilidade de uso do RDF gerado pela comunidade. Além disso, os dados passam a ter uma melhor visibilidade e consulta, onde, através de uma aplicação, é possível, sem a necessidade de baixar todo o arquivo existente, visualizar conteúdo simples sobre uma entidade por meio do derreferenciamento de seu URI ou buscar uma gama de informações em diferentes formatos usando um SPARQL *Endpoint*. Além disto, é possível realizar consultas com inferências sobre os dados, resultando em triplas geradas dinamicamente por meio de regras definidas e possibilitando a descoberta de novas relações entre as informações existentes.

Como perspectivas futuras, um aumento do escopo do mapeamento de dados que inclua outros elementos além de servidores e organizações (como gastos e convênios, por exemplo) poderia ser realizado gerando uma nuvem de informações governamentais estruturadas. Para o governo de um país que prega transparência da informação, seria interessante ter os dados publicados na forma apresentada neste trabalho para que qualquer pessoa ou máquina possa extrair a informação desejada e analisá-la com maior facilidade. Isto viabilizaria também que essa gama de informações se torne cada vez maior através de aplicações externas ligadas aos dados governamentais. Isso proporcionaria ao cidadão e aos gestores públicos um melhor acesso à informação de qualidade, tanto no que diz respeito às tecnologias para consumo de dados, assim como no que diz respeito à semântica dos dados consumidos.

7 BIBLIOGRAFIA

BECKETT, D.; BERNERS-LEE, T. **Turtle - Terse RDF Triple Language**. Disponível em: <<http://www.w3.org/TeamSubmission/turtle/>>. Acesso em: 8 jun. 2014.

BERNERS-LEE, T. **Semantic Web on XML**. Disponível em: <<http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html>>. Acesso em: 5 maio. 2014.

BERNERS-LEE, T. **Linked Data**. Disponível em: <<http://www.w3.org/DesignIssues/LinkedData.html>>. Acesso em: 30 abr. 2014.

BERNERS-LEE, T. et al. **Tabulator: Exploring and analyzing linked data on the semantic web** Proceedings of the 3rd International Semantic Web User Interaction Workshop. **Anais...**2006

BERNERS-LEE, T.; FIELDING, R.; MASINTER, L. **Uniform resource identifier (URI): Generic syntax**. Disponível em: <<http://tools.ietf.org/html/rfc3986>>. Acesso em: 30 abr. 2014.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. The Semantic Web. **Scientific american**, v. 284, n. 5, p. 28–37, 2001.

BIZER, C.; HEATH, T.; BERNERS-LEE, T. Linked Data - The Story So Far. **International journal on semantic web and information systems**, v. 5, n. 3, p. 1–22, 2009.

BRAY, T. et al. **Namespaces in XML 1.1 (Second Edition)**. Disponível em: <<http://www.w3.org/TR/xml-names11/>>. Acesso em: 6 jun. 2014.

BRAY, T. et al. **Extensible Markup Language (XML) 1.0 (Fifth Edition)**. Disponível em: <<http://www.w3.org/TR/xml/>>. Acesso em: 6 jun. 2014.

BRICKLEY, D.; GUHA, R. V. **RDF Schema 1.1**. Disponível em: <<http://www.w3.org/TR/rdf-schema/>>. Acesso em: 30 abr. 2014.

BRICKLEY, D.; MILLER, L. **FOAF Vocabulary Specification 0.99**. Disponível em: <<http://xmlns.com/foaf/spec/>>. Acesso em: 30 abr. 2014.

CYGANIAK, R.; JENTZSCH, A. **The Linking Open Data cloud diagram**. Disponível em: <<http://lod-cloud.net/>>. Acesso em: 7 maio. 2014.

CYGANIAK, R.; WOOD, D.; LANTHALER, M. **RDF 1.1 Concepts and Abstract Syntax**. Disponível em: <<http://www.w3.org/TR/rdf11-concepts/>>. Acesso em: 30 abr. 2014.

DAS, S.; SUNDARA, S.; CYGANIAK, R. **R2RML: RDB to RDF Mapping Language**. Disponível em: <<http://www.w3.org/TR/r2rml/>>. Acesso em: 29 abr. 2014.

FOSTER, F. G. **Standard Numbering in the Book Trade: A Report**. Great Britain: Publishers Association, 1967.

GAO, S.; SPERBERG-MCQUEEN, C. M.; THOMPSON, H. S. **W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures**. Disponível em: <<http://www.w3.org/TR/xmlschema11-1/>>. Acesso em: 30 abr. 2014.

GRUBER, T. R. A translation approach to portable ontology specifications. **Knowledge acquisition**, v. 5, n. 2, p. 199–220, 1993.

GUARINO, N. **Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy**. Amsterdam: IOS Press, 1998. p. 3–15

GUIZZARDI, G. **Ontological foundations for structural conceptual models**. The Netherlands: CTIT, Centre for Telematics and Information Technology, 2005.

HARRIS, S.; SEABORNE, A. **SPARQL 1.1 Query Language**. Disponível em: <<http://www.w3.org/TR/sparql11-query/>>. Acesso em: 30 abr. 2014.

HEATH, T.; BIZER, C. **Linked Data Envolving the Web into a Global Data Space**. 1. ed. [s.l.] Morgan & Claypool, 2011.

HEPP, M. **GoodRelations Language Reference**. Disponível em: <<http://www.heppnetz.de/ontologies/goodrelations/v1.html>>. Acesso em: 30 maio. 2014.

IANNELLA, R.; MCKINNEY, J. **vCard Ontology - for describing People and Organizations**. Disponível em: <<http://www.w3.org/TR/vcard-rdf/>>. Acesso em: 30 maio. 2014.

JACOBS, I.; WALSH, N. **Architecture of the World Wide Web, Volume One**. Disponível em: <<http://www.w3.org/TR/webarch/>>. Acesso em: 29 abr. 2014.

MCGUINNESS, D. L.; HARMELEN, F. VAN. **OWL Web Ontology Language Overview**. Disponível em: <<http://www.w3.org/TR/owl-features/>>. Acesso em: 7 maio. 2014.

MILES, A.; BECHHOFFER, S. **SKOS Simple Knowledge Organization System Reference**. Disponível em: <<http://www.w3.org/TR/skos-reference/>>. Acesso em: 30 abr. 2014.

REYNOLDS, D. **The Organization Ontology**. Disponível em: <<http://www.w3.org/TR/vocab-org/>>. Acesso em: 29 abr. 2014.

SCHREIBER, G.; YVES, R. **RDF 1.1 Primer**. Disponível em: <<http://www.w3.org/TR/rdf11-primer/>>. Acesso em: 29 abr. 2014.

W3C/IETF, U. P. I. G. **URIs, URLs, and URNs: Clarifications and Recommendations 1.0**. Disponível em: <<http://www.w3.org/TR/uri-clarification/>>. Acesso em: 29 abr. 2014.

8 APÊNDICE A

```

sparql clear graph <http://temp/organization>;

DB.DBA.TTLP ('
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix org: <http://www.w3.org/ns/org#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .

<http://localhost:8080/NOM/resource#TriplesMapOrganization>
  a rr:TriplesMap;

  rr:logicalTable
  [
    rr:tableSchema "CSV";
    rr:tableOwner "DBA";
    rr:tableName "OrgLotacao_csv"
  ];

  rr:subjectMap
  [
    rr:template "http://localhost:8080/NOM/resource/organization/{COD_ORG_LOTACAO}";
    rr:class org:FormalOrganization;
    rr:graph <http://localhost:8080/NOM/graph#>;
  ];

  rr:predicateObjectMap
  [
    rr:predicate skos:prefLabel;
    rr:objectMap [ rr:column "ORG_LOTACAO" ];
  ];

  rr:predicateObjectMap
  [
    rr:predicate org:identifier;
    rr:objectMap [ rr:column "COD_ORG_LOTACAO" ];
  ];

  rr:predicateObjectMap
  [
    rr:predicate org:subOrganizationOf;
    rr:objectMap [
      rr:parentTriplesMap
      <http://localhost:8080/NOM/resource#TriplesMapOrganizationSuperior>;
      rr:joinCondition [
        rr:child "COD_ORGSUP_LOTACAO";
        rr:parent "COD_ORGSUP_LOTACAO";
      ];
    ];
  ];

  rr:predicateObjectMap [
    rr:predicate owl:sameAs;
    rr:objectMap [
      rr:parentTriplesMap
      <http://localhost:8080/NOM/resource#TriplesMapOrganizationStructure>;

```

```

        rr:joinCondition [
            rr:child "ORG_LOTACAO";
            rr:parent "DADOS_CADASTRO_NOME";
        ];
    ];
]
.
', 'http://temp/organization', 'http://temp/organization' );

sparql clear graph <http://temp/organization2>;

DB.DBA.TTLP ('
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix org: <http://www.w3.org/ns/org#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .

<http://localhost:8080/NOM/resource#TriplesMapOrganization2>
  a rr:TriplesMap;

  rr:logicalTable
  [
    rr:tableSchema "CSV";
    rr:tableOwner "DBA";
    rr:tableName "OrgExercicio_csv"
  ];

  rr:subjectMap
  [
    rr:template "http://localhost:8080/NOM/resource/organization/{COD_ORG_EXERCICIO}";
    rr:class org:FormalOrganization;
    rr:graph <http://localhost:8080/NOM/graph#>;
  ];

  rr:predicateObjectMap
  [
    rr:predicate skos:prefLabel;
    rr:objectMap [ rr:column "ORG_EXERCICIO" ];
  ];

  rr:predicateObjectMap
  [
    rr:predicate org:identifier;
    rr:objectMap [ rr:column "COD_ORG_EXERCICIO" ];
  ];

  rr:predicateObjectMap
  [
    rr:predicate org:subOrganizationOf;
    rr:objectMap [
      rr:parentTriplesMap
<http://localhost:8080/NOM/resource#TriplesMapOrganizationSuperior2>;
      rr:joinCondition [
        rr:child "COD_ORGSUP_EXERCICIO";
        rr:parent "COD_ORGSUP_EXERCICIO";
      ];
    ];
  ];

  rr:predicateObjectMap [
    rr:predicate owl:sameAs;
    rr:objectMap [
      rr:parentTriplesMap
<http://localhost:8080/NOM/resource#TriplesMapOrganizationStructure>;
      rr:joinCondition [

```

```

                rr:child "ORG_EXERCICIO";
                rr:parent "DADOS_CADASTRO_NOME";
            ];
        ];
    ];
;
', 'http://temp/organization2', 'http://temp/organization2' );

sparql clear graph <http://temp/organization-address>;

DB.DBA.TTLP ('
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix org: <http://www.w3.org/ns/org#> .
@prefix vcard: <http://www.w3.org/2006/vcard/ns#> .

<http://localhost:8080/NOM/resource#TriplesMapOrganizationAddress>
  a rr:TriplesMap;

  rr:logicalTable
  [
    rr:tableSchema "CSV";
    rr:tableOwner "DBA";
    rr:tableName "Organizacoes_csv"
  ];

  rr:subjectMap
  [
    rr:template "http://localhost:8080/NOM/resource/organization-
structure/address/{DADOS_CADASTRO_CODIGO}";
    rr:class org:Site;
    rr:graph <http://localhost:8080/NOM/graph#>;
  ];

  rr:predicateObjectMap
  [
    rr:predicate vcard:postal-code;
    rr:objectMap [ rr:column "LOCALIDADE_NUMERO_CEP" ];
  ];

    rr:predicateObjectMap
  [
    rr:predicate vcard:country-name;
    rr:objectMap [ rr:column "LOCALIDADE_NOME_PAIS" ];
  ];

    rr:predicateObjectMap
  [
    rr:predicate vcard:locality;
    rr:objectMap [ rr:column "LOCALIDADE_NOME_CIDADE" ];
  ];

    rr:predicateObjectMap
  [
    rr:predicate vcard:street-address;
    rr:objectMap [ rr:column "LOCALIDADE_DESCRICAO_ENDERECO" ];
  ];

    rr:predicateObjectMap
  [
    rr:predicate vcard:street-address;
    rr:objectMap [ rr:column "LOCALIDADE_DESCRICAO_BAIRRO" ];
  ];

    rr:predicateObjectMap
  [

```

```

rr:predicate vcard:street-address;
rr:objectMap [ rr:column "LOCALIDADE_DESCRICAO_COMPLEMENTO" ];
];

rr:predicateObjectMap [
rr:predicate vcard:hasRegion;
rr:objectMap [
rr:parentTriplesMap <http://localhost:8080/NOM/resource#TriplesMapRegion>;
rr:joinCondition [
rr:child "LOCALIDADE_SIGLA_UF";
rr:parent "SIGLA_UF";
];
];
];

', 'http://temp/organization-address', 'http://temp/organization-address' );

sparql clear graph <http://temp/organizational-unit>;

DB.DBA.TTLP ('
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix org: <http://www.w3.org/ns/org#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .

<http://localhost:8080/NOM/resource#TriplesMapOrganizationalUnit>
a rr:TriplesMap;

rr:logicalTable
[
rr:tableSchema "CSV";
rr:tableOwner "DBA";
rr:tableName "UOrgLotacao_csv"
];

rr:subjectMap
[
rr:template "http://localhost:8080/NOM/resource/organizational-unit/{UORG_LOTACAO}";
rr:class org:OrganizationalUnit;
rr:graph <http://localhost:8080/NOM/graph#>;
];

rr:predicateObjectMap
[
rr:predicate skos:prefLabel;
rr:objectMap [ rr:column "UORG_LOTACAO" ];
];

rr:predicateObjectMap [
rr:predicate org:unitOf;
rr:objectMap [
rr:parentTriplesMap <http://localhost:8080/NOM/resource#TriplesMapOrganization>;
rr:joinCondition [
rr:child "COD_ORG_LOTACAO";
rr:parent "COD_ORG_LOTACAO";
];
];
];

', 'http://temp/organizational-unit', 'http://temp/organizational-unit' );

sparql clear graph <http://temp/organizational-unit2>;

DB.DBA.TTLP ('
@prefix rr: <http://www.w3.org/ns/r2rml#> .

```

```

@prefix org: <http://www.w3.org/ns/org#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .

<http://localhost:8080/NOM/resource#TriplesMapOrganizationalUnit2>
  a rr:TriplesMap;

  rr:logicalTable
  [
    rr:tableSchema "CSV";
    rr:tableOwner "DBA";
    rr:tableName "UOrgExercicio_csv"
  ];

  rr:subjectMap
  [
    rr:template "http://localhost:8080/NOM/resource/organizational-unit/{UORG_EXERCICIO}";
    rr:class org:OrganizationalUnit;
    rr:graph <http://localhost:8080/NOM/graph#>;
  ];

  rr:predicateObjectMap
  [
    rr:predicate skos:prefLabel;
    rr:objectMap [ rr:column "UORG_EXERCICIO" ];
  ];

  rr:predicateObjectMap [
    rr:predicate org:unitOf;
    rr:objectMap [
      rr:parentTriplesMap <http://localhost:8080/NOM/resource#TriplesMapOrganization2>;
      rr:joinCondition [
        rr:child "COD_ORG_EXERCICIO";
        rr:parent "COD_ORG_EXERCICIO";
      ];
    ];
  ];

', 'http://temp/organizational-unit2', 'http://temp/organizational-unit2' );

sparql clear graph <http://temp/organization-structure>;

DB.DBA.TTLP ('
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix org: <http://www.w3.org/ns/org#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix vcard: <http://www.w3.org/2006/vcard/ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .

<http://localhost:8080/NOM/resource#TriplesMapOrganizationStructure>
  a rr:TriplesMap;

  rr:logicalTable
  [
    rr:tableSchema "CSV";
    rr:tableOwner "DBA";
    rr:tableName "Organizacoes_csv"
  ];

  rr:subjectMap
  [
    rr:template "http://localhost:8080/NOM/resource/organization-
structure/{DADOS_CADASTRO_CODIGO}";
    rr:class org:FormalOrganization;
    rr:graph <http://localhost:8080/NOM/graph#>;
  ];

```

```

    rr:predicateObjectMap
  [
    rr:predicate org:identifier;
    rr:objectMap [ rr:column "DADOS_CADASTRO_CODIGO" ];
  ];

  rr:predicateObjectMap
  [
    rr:predicate skos:prefLabel;
    rr:objectMap [ rr:column "DADOS_CADASTRO_NOME" ];
  ];

    rr:predicateObjectMap
  [
    rr:predicate skos:altLabel;
    rr:objectMap [ rr:column "DADOS_CADASTRO_SIGLA" ];
  ];

    rr:predicateObjectMap
  [
    rr:predicate vcard:hasTelephone;
    rr:objectMap [ rr:column "DADOS_CADASTRO_TELEFONES" ];
  ];

    rr:predicateObjectMap
  [
    rr:predicate vcard:hasEmail;
    rr:objectMap [ rr:column "DADOS_CADASTRO_EMAIL" ];
  ];

    rr:predicateObjectMap
  [
    rr:predicate vcard:hasURL;
    rr:objectMap [ rr:column "DADOS_CADASTRO_SITE" ];
  ];

    rr:predicateObjectMap
  [
    rr:predicate org:subOrganizationOf;
    rr:objectMap [ rr:template "http://localhost:8080/NOM/resource/organization-
structure/{DADOS_CADASTRO_CODIGO_PAIS}" ];
  ];

    rr:predicateObjectMap [
    rr:predicate owl:sameAs;
    rr:objectMap [
      rr:parentTriplesMap <http://localhost:8080/NOM/resource#TriplesMapSIORG>;
      rr:joinCondition [
        rr:child "DADOS_CADASTRO_CODIGO";
        rr:parent "DADOS_CADASTRO_CODIGO";
      ];
    ];
  ];

    rr:predicateObjectMap [
    rr:predicate org:hasSite;
    rr:objectMap [
      rr:parentTriplesMap
<http://localhost:8080/NOM/resource#TriplesMapOrganizationAddress>;
      rr:joinCondition [
        rr:child "DADOS_CADASTRO_CODIGO";
        rr:parent "DADOS_CADASTRO_CODIGO";
      ];
    ];
  ];

```

```

];

rr:predicateObjectMap [
  rr:predicate owl:sameAs;
  rr:objectMap [
    rr:parentTriplesMap <http://localhost:8080/NOM/resource#TriplesMap0organization>;
    rr:joinCondition [
      rr:child "DADOS_CADASTRO_NOME";
      rr:parent "ORG_LOTACAO";
    ];
  ];
];

rr:predicateObjectMap [
  rr:predicate owl:sameAs;
  rr:objectMap [
    rr:parentTriplesMap <http://localhost:8080/NOM/resource#TriplesMap0organization2>;
    rr:joinCondition [
      rr:child "DADOS_CADASTRO_NOME";
      rr:parent "ORG_EXERCICIO";
    ];
  ];
];

rr:predicateObjectMap [
  rr:predicate owl:sameAs;
  rr:objectMap [
    rr:parentTriplesMap
<http://localhost:8080/NOM/resource#TriplesMap0organizationSuperior>;
    rr:joinCondition [
      rr:child "DADOS_CADASTRO_NOME";
      rr:parent "ORGSUP_LOTACAO";
    ];
  ];
];

rr:predicateObjectMap [
  rr:predicate owl:sameAs;
  rr:objectMap [
    rr:parentTriplesMap
<http://localhost:8080/NOM/resource#TriplesMap0organizationSuperior2>;
    rr:joinCondition [
      rr:child "DADOS_CADASTRO_NOME";
      rr:parent "ORGSUP_EXERCICIO";
    ];
  ];
];

;
', 'http://temp/organization-structure', 'http://temp/organization-structure' );

sparql clear graph <http://temp/organization-superior>;

DB.DBA.TTLP ( '
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix org: <http://www.w3.org/ns/org#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .

<http://localhost:8080/NOM/resource#TriplesMap0organizationSuperior>
  a rr:TriplesMap;

  rr:logicalTable
  [
    rr:tableSchema "CSV";

```

```

rr:tableOwner "DBA";
rr:tableName "OrgSupLotacao_csv"
];

rr:subjectMap
[
rr:template "http://localhost:8080/NOM/resource/organization/{COD_ORGSUP_LOTACAO}";
rr:class org:FormalOrganization;
rr:graph <http://localhost:8080/NOM/graph#>;
];

rr:predicateObjectMap
[
rr:predicate org:identifier;
rr:objectMap [ rr:column "COD_ORGSUP_LOTACAO" ];
];

rr:predicateObjectMap
[
rr:predicate skos:prefLabel;
rr:objectMap [ rr:column "ORGSUP_LOTACAO" ];
];

rr:predicateObjectMap [
rr:predicate owl:sameAs;
rr:objectMap [
rr:parentTriplesMap
<http://localhost:8080/NOM/resource#TriplesMapOrganizationStructure>;
rr:joinCondition [
rr:child "ORGSUP_LOTACAO";
rr:parent "DADOS_CADASTRO_NOME";
];
];
];

;
', 'http://temp/organization-superior', 'http://temp/organization-superior' );

sparql clear graph <http://temp/organization-superior2>;

DB.DBA.TTLP ('
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix org: <http://www.w3.org/ns/org#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .

<http://localhost:8080/NOM/resource#TriplesMapOrganizationSuperior2>
a rr:TriplesMap;

rr:logicalTable
[
rr:tableSchema "CSV";
rr:tableOwner "DBA";
rr:tableName "OrgSupExercicio_csv"
];

rr:subjectMap
[
rr:template "http://localhost:8080/NOM/resource/organization/{COD_ORGSUP_EXERCICIO}";
rr:class org:FormalOrganization;
rr:graph <http://localhost:8080/NOM/graph#>;
];

rr:predicateObjectMap
[
rr:predicate org:identifier;

```

```

    rr:objectMap [ rr:column "COD_ORGSUP_EXERCICIO" ];
  ];

  rr:predicateObjectMap
  [
    rr:predicate skos:prefLabel;
    rr:objectMap [ rr:column "ORGSUP_EXERCICIO" ];
  ];

    rr:predicateObjectMap [
      rr:predicate owl:sameAs;
      rr:objectMap [
        rr:parentTriplesMap
<http://localhost:8080/NOM/resource#TriplesMapOrganizationStructure>;
          rr:joinCondition [
            rr:child "ORGSUP_EXERCICIO";
            rr:parent "DADOS_CADASTRO_NOME";
          ];
        ];
      ];
    ];

;
', 'http://temp/organization-superior2', 'http://temp/organization-superior2' );

sparql clear graph <http://temp/person>;

DB.DBA.TTLP ('
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix org: <http://www.w3.org/ns/org#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .

<http://localhost:8080/NOM/resource#TriplesMapPerson>
  a rr:TriplesMap;

  rr:logicalTable
  [
    rr:tableSchema "CSV";
    rr:tableOwner "DBA";
    rr:tableName "Agentes_csv"
  ];

  rr:subjectMap
  [
    rr:template "http://localhost:8080/NOM/resource/person/{Id_SERVIDOR_PORTAL}";
    rr:class foaf:Person;
    rr:graph <http://localhost:8080/NOM/graph#>;
  ];

    rr:predicateObjectMap
  [
    rr:predicate skos:notation;
    rr:objectMap [ rr:column "Id_SERVIDOR_PORTAL" ];
  ];

  rr:predicateObjectMap
  [
    rr:predicate foaf:name;
    rr:objectMap [ rr:column "NOME" ];
  ];

    rr:predicateObjectMap [
      rr:predicate org:memberOf;
      rr:objectMap [
        rr:parentTriplesMap <http://localhost:8080/NOM/resource#TriplesMapOrganization>;
        rr:joinCondition [

```

```

                rr:child "COD_ORG_LOTACAO";
                rr:parent "COD_ORG_LOTACAO";
            ];
        ];
    ];

    rr:predicateObjectMap [
        rr:predicate org:memberOf;
        rr:objectMap [
            rr:parentTriplesMap
<http://localhost:8080/NOM/resource#TriplesMapOrganizationalUnit>;
            rr:joinCondition [
                rr:child "UORG_LOTACAO";
                rr:parent "UORG_LOTACAO";
            ];
        ];
    ];

    rr:predicateObjectMap [
        rr:predicate org:memberOf;
        rr:objectMap [
            rr:parentTriplesMap <http://localhost:8080/NOM/resource#TriplesMapOrganization2>;
            rr:joinCondition [
                rr:child "COD_ORG_EXERCICIO";
                rr:parent "COD_ORG_EXERCICIO";
            ];
        ];
    ];

    rr:predicateObjectMap [
        rr:predicate org:memberOf;
        rr:objectMap [
            rr:parentTriplesMap
<http://localhost:8080/NOM/resource#TriplesMapOrganizationalUnit2>;
            rr:joinCondition [
                rr:child "UORG_EXERCICIO";
                rr:parent "UORG_EXERCICIO";
            ];
        ];
    ];

    rr:predicateObjectMap [
        rr:predicate org:holds;
        rr:objectMap [
            rr:parentTriplesMap <http://localhost:8080/NOM/resource#TriplesMapPost>;
            rr:joinCondition [
                rr:child "DESCRICAO_CARGO";
                rr:parent "DESCRICAO_CARGO";
            ];
        ];
    ];

    rr:predicateObjectMap [
        rr:predicate org:holds;
        rr:objectMap [
            rr:parentTriplesMap <http://localhost:8080/NOM/resource#TriplesMapPost2>;
            rr:joinCondition [
                rr:child "CODIGO_ATIVIDADE";
                rr:parent "CODIGO_ATIVIDADE";
            ];
        ];
    ];

    rr:predicateObjectMap [
        rr:predicate org:remuneration;

```

```

        rr:objectMap [
            rr:parentTriplesMap
<http://localhost:8080/NOM/resource#TriplesMapRemunerationBRL>;
            rr:joinCondition [
                rr:child "Id_SERVIDOR_PORTAL";
                rr:parent "Id_SERVIDOR_PORTAL";
            ];
        ];
];

rr:predicateObjectMap [
    rr:predicate org:remuneration;
    rr:objectMap [
        rr:parentTriplesMap
<http://localhost:8080/NOM/resource#TriplesMapRemunerationUSD>;
        rr:joinCondition [
            rr:child "Id_SERVIDOR_PORTAL";
            rr:parent "Id_SERVIDOR_PORTAL";
        ];
    ];
];

;
', 'http://temp/person', 'http://temp/person' );

sparql clear graph <http://temp/post>;

DB.DBA.TTLP ('
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix org: <http://www.w3.org/ns/org#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .

<http://localhost:8080/NOM/resource#TriplesMapPost>
a rr:TriplesMap;

rr:logicalTable
[
    rr:tableSchema "CSV";
    rr:tableOwner "DBA";
    rr:tableName "Cargos_csv"
];

rr:subjectMap
[
    rr:template "http://localhost:8080/NOM/resource/post/{DESCRICA0_CARGO}";
    rr:class org:Post;
    rr:graph <http://localhost:8080/NOM/graph#>;
];

rr:predicateObjectMap
[
    rr:predicate skos:prefLabel;
    rr:objectMap [ rr:column "DESCRICA0_CARGO" ];
];

;
', 'http://temp/post', 'http://temp/post' );

sparql clear graph <http://temp/post2>;

DB.DBA.TTLP ('
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix org: <http://www.w3.org/ns/org#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .

<http://localhost:8080/NOM/resource#TriplesMapPost2>

```

```

a rr:TriplesMap;

rr:logicalTable
[
  rr:tableSchema "CSV";
  rr:tableOwner "DBA";
  rr:tableName "Atividades_csv"
];

rr:subjectMap
[
  rr:template "http://localhost:8080/NOM/resource/post/{CODIGO_ATIVIDADE}";
  rr:class org:Post;
  rr:graph <http://localhost:8080/NOM/graph#>;
];

  rr:predicateObjectMap
  [
    rr:predicate skos:notation;
    rr:objectMap [ rr:column "CODIGO_ATIVIDADE" ];
  ];

rr:predicateObjectMap
[
  rr:predicate skos:prefLabel;
  rr:objectMap [ rr:column "ATIVIDADE" ];
]

.
', 'http://temp/post2', 'http://temp/post2' );

sparql clear graph <http://temp/region>;

DB.DBA.TTLP ('
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix vcard: <http://www.w3.org/2006/vcard/ns#> .

<http://localhost:8080/NOM/resource#TriplesMapRegion>
a rr:TriplesMap;

rr:logicalTable
[
  rr:tableSchema "CSV";
  rr:tableOwner "DBA";
  rr:tableName "EstadosIBGE_csv"
];

rr:subjectMap
[
  rr:template "http://localhost:8080/NOM/resource/region/{CODIGO_UF}";
  rr:class vcard:Location;
  rr:graph <http://localhost:8080/NOM/graph#>;
];

  rr:predicateObjectMap
  [
    rr:predicate skos:prefLabel;
    rr:objectMap [ rr:column "NOME_UF" ];
  ];

rr:predicateObjectMap
[
  rr:predicate skos:notation;
  rr:objectMap [ rr:column "CODIGO_UF" ];
];

```

```

];

    rr:predicateObjectMap
  [
    rr:predicate skos:altLabel;
    rr:objectMap [ rr:column "SIGLA_UF" ];
  ]

;
', 'http://temp/region', 'http://temp/region' );

sparql clear graph <http://temp/person-remuneration-brl>;

DB.DBA.TTLP ('
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix gr: <http://purl.org/goodrelations/v1#> .

<http://localhost:8080/NOM/resource#TriplesMapRemunerationBRL>
  a rr:TriplesMap;

  rr:logicalTable
  [
    rr:tableSchema "CSV";
    rr:tableOwner "DBA";
    rr:tableName "Remuneracao112013_csv"
  ];

  rr:subjectMap
  [
    rr:template
"http://localhost:8080/NOM/resource/remuneration/brl/{ANO}/{MES}/person/{Id_SERVIDOR_PORTAL}";
    rr:class gr:PriceSpecification;
    rr:graph <http://localhost:8080/NOM/graph#>;
  ];

    rr:predicateObjectMap
  [
    rr:predicate gr:hasCurrencyValue;
    rr:objectMap [ rr:column "REMUNERACAO_BASICA_BRUTA_BRL" ];
  ]

;
', 'http://temp/person-remuneration-brl', 'http://temp/person-remuneration-brl' );

sparql clear graph <http://temp/person-remuneration-usd>;

DB.DBA.TTLP ('
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix gr: <http://purl.org/goodrelations/v1#> .

<http://localhost:8080/NOM/resource#TriplesMapRemunerationUSD>
  a rr:TriplesMap;

  rr:logicalTable
  [
    rr:tableSchema "CSV";
    rr:tableOwner "DBA";
    rr:tableName "Remuneracao112013_csv"
  ];

  rr:subjectMap
  [
    rr:template
"http://localhost:8080/NOM/resource/remuneration/usd/{ANO}/{MES}/person/{Id_SERVIDOR_PORTAL}";
    rr:class gr:PriceSpecification;

```

```

rr:graph <http://localhost:8080/NOM/graph#>;
];

rr:predicateObjectMap
[
rr:predicate gr:hasCurrencyValue;
rr:objectMap [ rr:column "REMUNERACAO_BASICA_BRUTA_USD" ];
]

;
', 'http://temp/person-remuneration-usd', 'http://temp/person-remuneration-usd' );

sparql clear graph <http://temp/organization-siorg>;

DB.DBA.TTLP ('
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix org: <http://www.w3.org/ns/org#> .

<http://localhost:8080/NOM/resource#TriplesMapSIORG>
a rr:TriplesMap;

rr:logicalTable
[
rr:tableSchema "CSV";
rr:tableOwner "DBA";
rr:tableName "Organizacoes_csv"
];

rr:subjectMap
[
rr:template
"http://www.siorg.redegoverno.gov.br/Atributo.asp?co_orgao={DADOS_CADASTRO_CODIGO}";
rr:class org:FormalOrganization;
rr:graph <http://localhost:8080/NOM/graph#>;
]

;
', 'http://temp/organization-siorg', 'http://temp/organization-siorg' );

sparql clear graph <http://temp/mix>;
sparql clear graph <http://localhost:8080/NOM/graph#>;
sparql clear graph <http://localhost:8080/NOM/resource#>;

sparql insert in graph <http://temp/mix> { ?s ?p ?o } from <http://temp/region> where { ?s
?p ?o } ;
sparql insert in graph <http://temp/mix> { ?s ?p ?o } from <http://temp/organization-siorg>
where { ?s ?p ?o } ;
sparql insert in graph <http://temp/mix> { ?s ?p ?o } from <http://temp/organization-
address> where { ?s ?p ?o } ;
sparql insert in graph <http://temp/mix> { ?s ?p ?o } from <http://temp/organization-
structure> where { ?s ?p ?o } ;
sparql insert in graph <http://temp/mix> { ?s ?p ?o } from <http://temp/organization-
superior> where { ?s ?p ?o } ;
sparql insert in graph <http://temp/mix> { ?s ?p ?o } from <http://temp/organization-
superior2> where { ?s ?p ?o } ;
sparql insert in graph <http://temp/mix> { ?s ?p ?o } from <http://temp/organizational-unit>
where { ?s ?p ?o } ;
sparql insert in graph <http://temp/mix> { ?s ?p ?o } from <http://temp/organization> where
{ ?s ?p ?o } ;
sparql insert in graph <http://temp/mix> { ?s ?p ?o } from <http://temp/organizational-
unit2> where { ?s ?p ?o } ;
sparql insert in graph <http://temp/mix> { ?s ?p ?o } from <http://temp/organization2> where
{ ?s ?p ?o } ;
sparql insert in graph <http://temp/mix> { ?s ?p ?o } from <http://temp/post> where { ?s ?p
?o } ;

```

```
sparql insert in graph <http://temp/mix> { ?s ?p ?o } from <http://temp/post2> where { ?s ?p ?o } ;
sparql insert in graph <http://temp/mix> { ?s ?p ?o } from <http://temp/person-remuneration-brl> where { ?s ?p ?o } ;
sparql insert in graph <http://temp/mix> { ?s ?p ?o } from <http://temp/person-remuneration-usd> where { ?s ?p ?o } ;
sparql insert in graph <http://temp/mix> { ?s ?p ?o } from <http://temp/person> where { ?s ?p ?o } ;

exec ('sparql ' || DB.DBA.R2RML_MAKE_QM_FROM_G ('http://temp/mix')) ;

sparql insert in graph <http://localhost:8080/NOM/resource#> { ?s ?p ?o } from <http://localhost:8080/NOM/graph#> where { ?s ?p ?o } ;
```