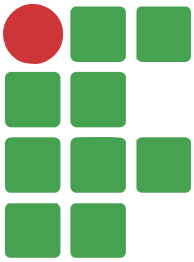


INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
DEPARTAMENTO DE TELEMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Gabriel Lopes

# MAURA: Um Framework baseado em Mediador Semântico para construção eficiente de Linked Data Mashups

Fortaleza – CE

21 de fevereiro de 2017

Gabriel Lopes

# MAURA: Um Framework baseado em Mediador Semântico para construção eficiente de Linked Data Mashups

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Web Semântica

Orientador: Antônio Mauro Barbosa de Oliveira

Coorientador: Vânia Maria Ponte Vidal

Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)

Programa de Pós-Graduação em Ciência da Computação (PPGCC)

Fortaleza – CE

21 de fevereiro de 2017

Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)
Programa de Pós-Graduação em Ciência da Computação (PPGCC)

Gabriel Lopes

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação, sendo aprovada pela Coordenação do Programa de Pós-Graduação em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará e pela banca examinadora:

---

**Orientador: Antônio Mauro Barbosa de Oliveira**
Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)

---

**Coorientador: Vânia Maria Ponte Vidal**
Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)

---

**Cidcley Teixeira de Souza**
Instituto Federal do Ceará (IFCE)

Fortaleza – CE
21 de fevereiro de 2017

*Dedico este trabalho à Tim Berner's Lee, que entre a vaidade e a luxúria, optou pelo razoável.*

# Agradecimentos

Primeiramente, agradeço à minha mãe, que, independente do quão difícil a situação fosse, nunca pensou em desistir. Uma guerreira que onde muitos fracassariam, foi vitoriosa. Me ensinou o quão forte uma pessoa pode ser. Muito obrigado mãe.

Ao Prof. Glauber Cintra. Com seu profissionalismo exemplar, representou minha primeira grande motivação, durante a graduação, para seguir a carreira acadêmica. Obrigado professor.

Ao Prof. Cidcley Teixeira de Souza, quem me iniciou na carreira acadêmica, acreditando no meu potencial talvez até mais do que eu mesmo. Penso que poucas coisas na vida são mais prazerozas do que a sensação de poder agradecer àlguém que, de alguma forma, fez a diferença na sua vida. O sr. fez na minha. Se hoje tenho a possibilidade de terminar o mestrado, muito devo ao sr. Um orientador, professor e o mais importante: um amigo.

Agradeço, é claro, à minha querida co-orientador Profa. Vânia Vidal, que me mostrou o magnífico mundo da *Web Semântica* e do *Linked Data*, me motivando a seguir uma área que será meu foco acadêmico por vários e vários anos. Obrigado professora, a senhora me motivou a sempre querer aprender mais. Obrigado por todos puxões de orelha e por toda a paciência. Não tenha dúvidas, seus ensinamentos ajudarão a moldar o pesquisador que um dia eu serei. Obrigado, é uma honra poder ser co-orientado pela sra. Espero que essa parceria ainda dê muitos e muitos frutos!

Aos meus grandes amigos do Mestrado PPGCC 2014.2. Vocês fizeram parte de uma das melhores épocas da minha vida! Torço muito pelo sucesso de cada um.

Obrigado também aos meus amigos do laboratório ARIDA: Thiago Pequeno, Narciso Vidal, Salomão Magalhaes, Arlino Magalhães, Bruno Leal, Lívia Almada e Ticiane Linhares. Obrigado especial ao Narciso Vidal, que por diversas vezes teve a paciência de me ajudar a entender os diversos conceitos da *Web Semântica*. Obrigado meus amigos, principalmente pelos cafés de toda tarde! Vocês foram essenciais na minha formação.

Ao apoio financeiro recebido pela Fundação Cearense de Apoio ao Desenvolvimento Científica e Tecnológico (FUNCAP) e pelos auxílios da CAPES e CNPq que permitiram a minha ida à vários congressos.

Finalmente, agradeço ao Prof. Luiz Fernando Gomes Soares, o orientador do meu orientador. A vida é efêmera. É de uma inteligência incrível perceber que, durante o pouco tempo que passamos nela, temos que dar o nosso melhor e sempre tentar fazer a diferença: para uma cidade, uma pessoa ou uma nação. O sr. fez. O maior feito que alguém pode

obter durante uma vida é o de alcançar a imortalidade. O sr. a alcançou. Obrigado pela contribuição científica; pela contribuição na sociedade; pelos ensinamentos ; pelas risadas e pelo meu orientador e amigo, Prof. Antônio Mauro Barbosa de Oliveira, essencial na minha formação pessoal e acadêmica.

*“ As pessoas que são loucas o suficiente para achar que podem mudar o mundo são as que, de fato, o mudam.”*

*(Steve Jobs)*

# Resumo

GISSA é um projeto de pesquisa e de desenvolvimento que conta com o suporte da Financiadora de Estudos e Projetos (FINEP). O objetivo definido no GISSA é auxiliar os diversos atores da área de saúde (paciente, agente de saúde, médicos, prefeitos, secretários de estado, etc.) nos diversos processos de tomadas de decisão envolvidos no contexto do programa Rede Cegonha do Ministério da Saúde (MS). Para tanto, o projeto faz uso de informações oriundas de bases de dados do Sistema Único de Saúde brasileiro (SUS) e sua prova de conceito está sendo realizada na cidade de Tauá/CE - Brasil, desde 2015. Para identificar as causas de óbitos-infantis e partos prematuros, os gestores recorrem, em geral, às informações disponíveis sobre as mães, tais como: uso de álcool e drogas durante a gestação; doenças crônicas, como diabetes e hipertensão; situação socioeconômica, dentre outras. Porém, no SUS, tais informações estão distribuídas em bancos de dados relacionais heterogêneos, que dificultam a conciliação sintática e semântica necessária à integração de dados. Se de um lado a heterogeneidade sintática tem sido tratada pelo Departamento de Informática do MS (DATASUS) com tecnologias clássicas (barramento SOA, por ex.) a questão semântica resta ainda como um desafio. Esta é uma tarefa complexa, haja vista que em bancos relacionais os dados estão dispostos em tabelas, com pouca ou nenhuma semântica sobre a informação. O *Linked Data* se apresenta como solução atraente ao trato desta problemática, sendo uma mudança de paradigma na forma como as fontes de dados são representadas: fontes isoladas no formato de tabelas cedem lugar à grafos RDF interligados. Porém, a construção de uma visão homogeneizada sobre fontes *Linked Data* distintas, visão *Linked Data Mashup* (LDM), ainda não é uma tarefa trivial. Esta tarefa exige o uso de *frameworks* que requerem conhecimentos específicos em integração de dados e nas tecnologias da Web Semântica. O presente trabalho especifica e implementa o *MAshUp mediador for RDF Applications* (MAURA), um framework baseado em mediador semântico para facilitar a construção de LDMs. O MAURA permite que usuários de propósito geral, sem conhecimentos específicos, criem seus próprios *Linked Data Mashups* baseados em parâmetros, de forma rápida e intuitiva. Para tanto, o MAURA reutiliza uma especificação de LDM na criação de novos mashups, realizando um processo de mediação que materializa apenas os dados relevantes para o usuário. Além desta nova funcionalidade, o MAURA cria o conceito em que um *Linked Data Mashup* pode ser buscado, incrementado, i.e. ter novas fontes agregadas, e depositado novamente na *Web*. Isso possibilita que equipes, como a do projeto GISSA, possam economizar tempo no processo de integração de dados. Também foi criado um guia sobre a implementação do MAURA, contendo seus principais algoritmos e um modelo conceitual. Além disso, um protótipo foi desenvolvido, ilustrando os conceitos propostos no MAURA.

**Palavras-chave**: linked data mashup. web semantica. apoio a tomada de decisão.

# Abstract

GISSA is a project of research and development that counts with financial support from Financier of Studies and Projects (FINEP). The aim of this project is to support the health actors (e.g. patient, health agent, mayors, health managers) in the decision-making processes involved in the *Rede Cegonha*'s program from brazilian government. For this, GISSA uses informations from brazilian Public Health System (SUS) and it's proof of concept is being applied in the city of Tauá/CE - Brazil, since 2015. In order to identify the factors involved in infant deaths and premature births, managers usually analyze the mother's information, as: drugs and alcohol usage; cronical diseases, like diabetes and hypertension; socioeconomic situation, among others. But, in SUS, these infromations are distributed over heterogeneous relational databases, that difficults the sintatic and semantic conciliation, needed for data integration. For one side, the sintatic problem is being addressed by Government's Informatics Departament (DATASUS), with classical approaches (e.g. SOA), but on the other side, the semantic problem is still an issue. This is a complex task to solve, once relational databases store their data as tables, with no semantics about of the information stored. Linked Data is being seeing by the data integration community as a possible answer for the semantic problem, as it represents a new paradigm for datasets representation as it promotes the publication of previously isolated databases as interlinked RDF datasets. But, the development of an homogeneized view of sources in Linked Data, Linked Data Mashup view (LDM), isn't an easy task. For this, it is needed the use of specifics frameworks that requires specific knowledge in data integration and semantic web technologies. This work specifics and implements the *MAshUp mediador for RDF Applications* (MAURA), a framework based on semantic mediation to ease the process of construction of Linked Data Mashups. MAURA allows general purpose users, without specific knowledge, to create their own Linked Data Mashups based on parameters, in a easy, fast and intuitive way. For that, MAURA reuses a LDM specification in order to create new mashups, doing a mediation process that materializes only the relevant data to the user. Furthermore, MAURA creates the concept that a Linked Data Mashup can be searched on the web, incremented and deployed again on the web. This concept allows that third groups, like GISSA, can spend much less time on data integration process. In this work, we present an implementation guide by showing the algorithms needed in MAURA and a conceptual model of the approach. Finally, we also have developed a prototype, that shows the main new concepts of the framework MAURA.

**Keywords**: linked data mashup. linked data. semantic web. data integration. support decision-making

# Lista de ilustrações


Figura 1 – Exemplo de tabela em relacional. . . . . 16
Figura 2 – Relação de um indivíduo e suas doenças . . . . . 17
Figura 3 – Primeira página Web da história . . . . . 23
Figura 4 – Camadas da Web Semântica . . . . . 26
Figura 5 – Exemplo de Grafo RDF . . . . . 28
Figura 6 – Relacionamento de RDF e RDFS . . . . . 31
Figura 7 – Exemplo de inconsistência perceptível ao OWL . . . . . 33
Figura 8 – Exemplo de ação do raciocinador . . . . . 33
Figura 9 – Representatividade das sub-linguagens de OWL como Diagramas de *Venn* 34
Figura 10 – SPARQL Endpoint DBPedia. . . . . 35
Figura 11 – Processo genérico para integração de dados . . . . . 38
Figura 12 – Representação gráfica das abordagens OWA e CWA. . . . . 40
Figura 13 – Representação gráfica da diferença dos enfoques virtual e materializado. 40
Figura 14 – Arquitetura de um mediador genérico. . . . . 41
Figura 15 – Manutenção incremental em ambiente materializado. . . . . 42
Figura 16 – Arquitetura genérica de um Data Warehouse. . . . . 43
Figura 17 – Exemplo de tabela em relacional. . . . . 43
Figura 18 – Outra possibilidade de representação. . . . . 44
Figura 19 – Relação de um indivíduo e suas doenças . . . . . 45
Figura 20 – Nuvem Linked Open Data . . . . . 46
Figura 21 – Processo para definição dos Termos de Pesquisa . . . . . 52
Figura 22 – Resumo do processo de busca e seleção . . . . . 57
Figura 23 – Artigos para análise x ano de publicação . . . . . 58
Figura 24 – Porcentagem de inclusão de artigos x base de dados . . . . . 58
Figura 25 – Framework 3 Camadas . . . . . 64
Figura 26 – Visão da base de dados SINASC . . . . . 67
Figura 27 – Visão da base de dados e-SUS . . . . . 67
Figura 28 – Ontologia de Domínio Datasus_hub . . . . . 68
Figura 29 – Instâncias das Visões Exportadas . . . . . 71
Figura 30 – Instâncias da Visão de Mashup materializada . . . . . 72
Figura 31 – Arquitetura 4 Camadas do Mediador Semântico . . . . . 75
Figura 32 – Modelo Conceitual Mediador Semântico . . . . . 83
Figura 33 – Exemplo de independência de API . . . . . 89

# Lista de tabelas

Tabela 1 – Decomposição das questões de pesquisa . . . . . 52
Tabela 2 – Definição da *string* de busca . . . . . 52
Tabela 3 – *Strings* para busca automática nas bases . . . . . 54
Tabela 4 – Resultado das buscas manuais . . . . . 55
Tabela 5 – Mapeamentos SINASC . . . . . 69
Tabela 6 – Mapeamentos *Esus_EV'* . . . . . 79
Tabela 7 – Mapeamentos *DBPedia_EV* . . . . . 80

# Lista de Algoritmos

1 Reescrita de Especificação . . . . . 85
2 Interseção de Ontologias . . . . . 86
3 Adiciona Filtros aos Mapeamentos . . . . . 86
4 Remove Regras de Fusão . . . . . 87

# Sumário


**1 INTRODUÇÃO . . . . . 16**
**1.1 Motivação do Trabalho . . . . . 16**
**1.2 Descrição do Problema . . . . . 18**
**1.3 Objetivo Geral e Específicos . . . . . 19**
**1.4 Produção científica . . . . . 20**
**1.5 Estrutura da Dissertação . . . . . 20**

**2 FUNDAMENTAÇÃO TEÓRICA . . . . . 22**
**2.1 Introdução . . . . . 22**
**2.2 Evolução da Web: Web 1.0 a Web Semântica . . . . . 22**
2.2.1 Web 1.0 . . . . . 22
2.2.2 Web 2.0 . . . . . 23
2.2.3 Problemas da Web Sintática . . . . . 24
2.2.4 Web 3.0 - Web Semântica . . . . . 25
**2.3 Tecnologias da Web Semântica . . . . . 25**
2.3.1 Ontologias . . . . . 26
2.3.2 RDF . . . . . 26
2.3.2.1 Serialização RDF . . . . . 29
2.3.2.2 RDF-Schema . . . . . 30
2.3.3 OWL . . . . . 31
2.3.3.1 Racionadores OWL . . . . . 32
2.3.4 SPARQL . . . . . 34
2.3.5 R2RML . . . . . 35
2.3.6 Implementação de ontologias . . . . . 35
**2.4 Integração de Dados . . . . . 36**
2.4.1 Motivação para Integrar Dados . . . . . 37
2.4.1.1 Cenário do Sistema de Saúde brasileiro . . . . . 37
2.4.2 Abordagens e Desafios . . . . . 37
2.4.3 Visão Integrada . . . . . 39
2.4.3.1 Abordagem Virtual . . . . . 40
2.4.3.1.1 Mediadores . . . . . 41
2.4.3.2 Abordagem Materializada . . . . . 42
2.4.3.2.1 Data Warehouse . . . . . 42
**2.5 Web Semântica para Integração de Dados . . . . . 44**
2.5.1 Linked Data . . . . . 45

2.5.2 Linked Open Data . . . 46
2.5.3 Linked Data Mashup . . . 47
**2.6 Conclusão . . . 48**

**3 REVISÃO BIBLIOGRÁFICA . . . 49**
**3.1 Introdução . . . 49**
**3.2 Revisão Sistemática . . . 49**
3.2.1 Questão de Pesquisa . . . 50
3.2.2 Estratégia de Busca . . . 50
3.2.2.1 Termos de pesquisa . . . 51
3.2.2.2 Escopo de busca e bases digitais . . . 53
3.2.2.3 Processo de Busca . . . 54
3.2.3 Estratégia para Seleção . . . 56
**3.3 Discussão dos Resultados . . . 57**
3.3.1 Visão Geral . . . 58
3.3.2 *QP1: Como é realizado o processo de integração de dados?* . . . 59
3.3.3 *QP2: A construção de um Linked Data Mashup pode auxiliar na construção de um outro mashup sobre as mesmas fontes?* . . . 61
3.3.4 *QP3: Para construir um mashup, são necessários conhecimentos específicos em Web Semântica?* . . . 61
3.3.5 *QP4: Os autores ainda mantém este framework?* . . . 61
3.3.6 *QP5: Quais são as principais ferramentas para construção de Linked Data Mashups?* . . . 62
**3.4 Conclusão . . . 62**

**4 ESPECIFICAÇÃO DE LINKED DATA MASHUP . . . 63**
**4.1 Introdução . . . 63**
**4.2 Especificação de Mashup . . . 63**
4.2.1 Visão Geral . . . 63
4.2.2 Arquitetura 3 - Camadas . . . 64
4.2.3 Especificação das Visões Exportadas . . . 65
4.2.4 Especificação dos Links Semânticos . . . 65
4.2.5 Especificação das Regras de Fusão . . . 65
4.2.6 Materialização de Aplicações de mashups . . . 66
**4.3 Datasus_HUB . . . 66**
4.3.1 Fontes de Dados . . . 67
4.3.2 Ontologia de Domínio . . . 68
4.3.3 Visões Exportadas . . . 68
4.3.4 Links Semânticos . . . 69
4.3.4.1 Fusão e Qualidade . . . 69

**4.4 Construção de Aplicações de Mashups com Datasus_hub . . . . . 70**
4.4.1 SOS:Gestacao . . . . . 70
4.4.1.1 Materialização das Visões Exportadas . . . . . 70
4.4.1.2 Materialização dos Links Semânticos . . . . . 70
4.4.1.3 Materialização da Visão de Mashup . . . . . 71
**4.5 Conclusão . . . . . 72**

**5 MAURA: CONSTRUÇÃO DE LINKED DATA MASHUPS . . . . . 73**
**5.1 Introdução . . . . . 73**
**5.2 MAURA - Mediador Semântico . . . . . 74**
5.2.1 Visão geral . . . . . 74
5.2.2 Arquitetura 4-Camadas . . . . . 75
5.2.3 Construção de uma Visão de Aplicação . . . . . 76
5.2.3.1 Geração da Especificação de $V$ sobre $M$ . . . . . 76
5.2.4 Geração da Especificação de $V$ sobre as Fontes de Dados . . . . . 76
**5.3 Casos de Uso . . . . . 78**
5.3.1 Caso de uso 1 : SOS:Gestacao . . . . . 78
5.3.2 Caso de Uso 2: Integração com DBPedia . . . . . 79
5.3.3 Reutilização de Especificações para Impsulsionar Estudos . . . . . 80
**5.4 Conclusão . . . . . 81**

**6 A IMPLEMENTAÇÃO DE UM PROTÓTIPO PARA O MAURA . . 82**
**6.1 Introdução . . . . . 82**
**6.2 Modelagem de MAURA . . . . . 82**
6.2.1 Modelo Conceitual . . . . . 82
6.2.1.1 Visão Integrada Semanticamente . . . . . 83
6.2.1.2 Visão de Aplicação . . . . . 84
6.2.2 Diagrama de Fluxo . . . . . 84
**6.3 Mediador Semântico: Implementação . . . . . 84**
6.3.1 Reescrita da especificação . . . . . 84
6.3.1.1 Interseção entre as Ontologias . . . . . 85
6.3.1.2 Adição dos Filtros aos Mapeamentos . . . . . 86
6.3.1.3 Novas Regras de Fusão . . . . . 87
**6.4 Mediador Semântico: Protótipo . . . . . 87**
6.4.1 Tecnologias . . . . . 88
6.4.1.1 API RDF e OWL . . . . . 88
6.4.1.2 Linguagem . . . . . 88
6.4.1.3 Padrões de Projeto . . . . . 88
6.4.2 Visão Integrada - Mashup . . . . . 89
6.4.3 Visão de Aplicação . . . . . 91

6.4.4 Reescrita de Especificação . . . . . 91
6.4.5 Materialização . . . . . 92
**6.5 Conclusão . . . . . 92**

**7 CONCLUSÃO . . . . . 93**
**7.1 Considerações Finais . . . . . 93**
**7.2 Trabalhos Futuros . . . . . 94**

**REFERÊNCIAS . . . . . 95**

**APÊNDICES 105**

**ANEXOS 106**

# 1 Introdução

Na Área da Saúde, existem diversos fatores que devem ser levados em consideração durante uma tomada de decisão. Geralmente, muitas dessas informações estão armazenadas em bases de dados distribuídas, dificultando uma análise integrada dessas informações. Para se determinar as causas de um óbito materno, por exemplo, um gestor de saúde pode precisar analisar informações sobre a mãe e o recém-nascido. No Sistema Único de Saúde brasileiro (SUS), tais informações estão distribuídas em fontes de dados heterogêneas: óbito, Sistema de Informações sobre Mortalidade (SIM); informações sobre a mãe, como uso de álcool ou drogas, SUS eletrônico (e-SUS); informações do recém-nascido, como peso e possiveis anomalias, Sistema de Informações sobre Nascidos Vivos (SINASC), etc.

Nesse contexto, Sistemas Clínicos para o Apoio a Tomada de Decisões (CDSS) desempenham um papel essencial no auxílio a gestores. Tais sistemas auxiliam na visualização das informações por meio de gráficos e *dashboards*. Porém, uma problemática comum em CDSS é a integração de dados. Como no exemplo do óbito-materno, comumente as informações precisam ser integradas para, então, serem analisadas pelos gestores. Atualmente, o tipo mais utilizado para armazenamento de dados ainda é o formato relacional (DB-ENGINES, 2017) e, com isso, tais sistemas geralmente utilizam abordagens como o *data warehouse* para integração de dados.

## 1.1 Motivação do Trabalho

A maior problemática na integração de dados é a conciliação semântica das informações em bases distintas (HULL, 1997). Essa conciliação é realizada por meio da construção de um *Esquema Global*. Para isso, os dados têm que ser analisados, entendidos e então mapeados nesse esquema global (LENZERINI, 2002). Em bancos de dados relacionais, os dados são representados por tabelas, e a semântica, i.e. o significado, das informações é limitada por seu conteúdo e pelos nome das colunas. Por exemplo, a tabela mostrada na Figura 17 representa a relação de um indivíduo com suas doenças em um banco de dados relacional.

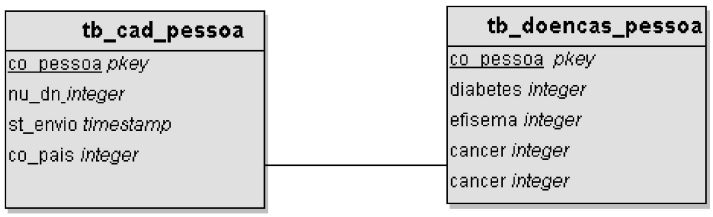



Figura 1 – Exemplo de tabela em relacional.

Sem um conhecimento do domínio da base de dados, não é óbvio entender, na Figura 17, que *"cad"*, por exemplo, é uma abreviação de *"cadastro"*. Embora seja fácil de entender que a coluna *"co_pais"* se trata do *"código de um país"*, a coluna *"st_envio"* não é tão óbvia assim. Assim, a semântica em bancos relacionais está limitada ao conteúdo e também à nomenclatura que fica a critério do desenvolvedor de suas tabelas. Em bancos de dados relacionais, uma conciliação semântica pode não ter uma solução viável em projetos de grandes instituições, por exemplo, onde os bancos de dados tendem a ser enormes, com tabelas com até centenas de colunas.

Web Semântica se apresenta como solução atraente ao trato desta problemática. Os dados na Web Semântica deixam de ser representados por tabelas e colunas, para serem representados no formato de triplas **RDF**: *sujeito, predicado* e *objeto* (MANOLA; MILLER, 2004). Em uma tripla, todas as informações são consideradas *recursos* e possuem uma URI[1] associada. Essa URI é única na *Web* e contém informações sobre um dado recurso, permitindo que, além de humanos, *softwares* também sejam capazes de entender a informação (HEATH; BIZER, 2011). A Figura 19 demonstra o exemplo da Figura 17 expresso nas tecnologias da Web Semântica.

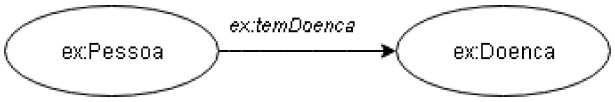



Figura 2 – Relação de um indivíduo e suas doenças

,

Na Fig. 19, *"ex:Pessoa"*, *"ex:temDoenca"* e *"ex:Doenca"* são recursos. Cada recurso é associado a uma URI, abreviada por *"ex:"* no exemplo, que contém detalhes sobre cada recurso. Por exemplo, a URI de *ex:temDoenca* pode descrever que essa propriedade conecta objetos do tipo *ex:Pessoa* com objetos do tipo *ex:Doenca*.

Com o avanço das tecnologias da Web Semântica, foi criada a iniciativa *Linked Data* (BIZER; HEATH; Berners-Lee, 2009), um conjunto de princípios e de boas práticas

[1] Com exceção dos literais, estes não possuem URIs

para publicação de dados na *Web*. O *Linked Data* promove a publicação de fontes de dados anteriormente isoladas como fontes RDF interligadas. Em RDF as informações concretas ou abstratas são representadas por **Ontologias**(GRUBER, 1993), que representam a base da *Web Semântica*. Uma ontologia é descrita por regras formais, o que permitem computadores raciociocinarem sobre elas. Além de permitir o descobrimento *on-the-fly* de fontes possivelmente relevantes, uma das boas práticas do *Linked Data* é a reutilização de ontologias bem definidas[2]. Assim, a problemática de falta de semântica nas informações, enfrentada por bancos de dados relacionais, diminui consideravelmente.

O projeto GISSA (GISSA, 2015) é um CDSS que utiliza mecanismos inteligentes, como mineração e integração de dados, para prover uma informação mais rica ao gestor de saúde. O GISSA está realizando sua prova de conceito (PoC) na cidade de Tauá/CE - Brasil. Num determinado mês, os gestores da prefeitura dessa cidade perceberam um número incomum de casos de óbito-infantil. A fim de investigar as causas de tais óbitos, recorreram às informações das suas mães, como: uso de drogas, tabaco ou álcool durante a gravidez; doenças crônicas, como diabetes e hipertensão, dentre outros. Porém, no SUS, tais informações estão distribuídas em bancos de dados relacionais heterogêneos, que dificultam a conciliação sintática e semântica necessária à integração de dados. Se de um lado a heterogeneidade sintática tem sido tratada pelo Departamento de Informática do MS (DATASUS) com tecnologias clássicas (barramento SOA, por ex.) a questão semântica resta ainda como um desafio. Um dos principais objetivos dessa dissertação é fornecer ao GISSA uma visão integrada sobre as fontes heterogêneas.

## 1.2 Descrição do Problema

Com o *Linked Data*, a *web* evoluiu de uma *Web de Documentos*, onde os dados são inteligíveis apenas por humanos, para uma *Web de Dados*, com fontes de dados RDF interligadas, inteligíveis também por máquinas (HEATH; BIZER, 2011).

Embora um dos princípios do *Linked Data* seja reutilizar ontologias já definidas, nem sempre isso é possível. Comumente existem necessidades específicas de determinado domínio que não estão descritas na ontologia em questão. Nestes casos, o provedor de dados tem que desenvolver uma própria, criando a heterogeneidade na *Web de Dados*. Uma visão integrada sobre fontes *Linked Data*, *Linked Data Mashup* (LDM), pode ser utilizada para a criação de aplicações que, diferentemente de abordagens convencionais, utiliza dados da *Web* para enriquecimento dos dados, tratando, desta forma, a heterogeneidade na *Web de Dados*. Porém, desenvolver um *Linked Data Mashup* não é uma tarefa trivial. Segundo (VIDAL et al., 2015), existem 4 desafios principais para criação de *mashups* em *Linked Data*: (i) seleção das fontes *linked data* relevantes para a aplicação; (ii) extração e tradução

[2] http://5stardata.info/en/

de fontes de dados distintas para uma ontologia comum; (iii) identificação de *links* que denotam a similaridade entre instâncias em fontes distintas e, finalmente, (iv) combinação e fusão de múltiplas representações de um mesmo objeto do mundo real numa única representação. As abordagens atuais para construção de *Linked Data Mashups* requerem conhecimentos específicos em *Web Semântica*, impossibilitando um usuário de propósito geral construir seus próprios *mashups*. Além disso, por conta da falta de abordagens que materializem os dados de forma parametrizada, diversas informações em um *mashup* podem não ser relevantes para o usuário. Assim, faz-se necessária uma solução que crie *Linked Data Mashups* de forma eficiente, baseados nos parâmetro do usuário.

## 1.3 Objetivo Geral e Específicos

Este trabalho especifica o MAURA (*MAshUp mediator for RDF Applications*), *framework* baseado em mediador semântico para construção de *Linked Data Mashups* de forma eficiente, baseado em parâmetros, possibilitando um usuário de propósito geral criar um *mashup* de acordo com suas necessidades. Como prova de conceito, é implementado um protótipo destinado ao GISSA, um projeto que ambiciona auxiliar todos os atores da área da saúde (paciente, agente de saúde, médicos, prefeitos, secretários de estado, etc.) nos diversos processos de tomadas de decisão envolvidos no contexto do programa Rede Cegonha do Ministério da Saúde - MS.

Para tanto, o MAURA reutiliza especificações de LDM na criação de novos mashups, realizando um processo de mediação que materializa apenas os dados relevantes para o usuário. Além desta nova funcionalidade, o MAURA cria o conceito em que um *Linked Data Mashup* pode ser buscado, incrementado, i.e. agregar novas fontes, e depositado novamente na *Web*. Essa abordagem é chamada, no presente trabalho, de *pay-as-you-go*. Isso possibilita que equipes, como a do projeto GISSA, possam economizar tempo no processo de integração de dados. Também foi criado um guia sobre a implementação do novo *framework* contendo seus principais algoritmos e um modelo conceitual. Um protótipo foi desenvolvido, ilustrando os conceitos propostos no MAURA.

Os objetivos específicos são:

- Com o auxílio de um estudo de Revisão Sistemática, apresentar os principais *frameworks* para construção de *Linked Data Mashups*, abordando suas principais características e funcionalidades;
- Construir uma *framework* baseado em mediador semântico para ambientes heterogêneos na *Web de Dados*;
- Implementar o primeiro protótipo de um *framework* para especificação de *Linked Data Mashups*;
- Permitir, com a abordagem proposta, que usuários sem conhecimentos específicos

em integração de dados ou Web Semântica criem seus próprios *Linked Data Mashups*, de acordo com seus parâmetros;
- Incorporar o *framework* ao GISSA como um de seus módulos de inteligência, disponibilizando aos gestores de saúde uma visão integrada sobre os dados anteriormente isolados;
- Demonstrar, mediante estudos de caso, como o *Linked Data* por ser utilizado para agregar valor em sistemas de apoio a tomada de decisão;
- Definir o conceito de reutilização da especificação de *Linked Data Mashups* e discutir como esse conceito pode impulsionar estudos em diversas áreas, principalmente em integração em *Linked Data*;

## 1.4 Produção científica

Durante este projeto de mestrado, os seguintes trabalhos científicos foram publicados e apresentados, a saber:

- **Gabriel Lopes, Vânia Vidal, and Mauro Oliveira**. *A framework for creation of linked data mashups: A case study on healthcare.* In Proceedings of the 22Nd Brazilian Symposium on Multimedia and the Web, Webmedia 2016, pages 327–330, New York, NY, USA, 2016. ACM
- **Gabriel Lopes, Vânia Vidal, Mauro Oliveira and Odorico Andrade**. *LAIS: Towards to a Linked Data Framework to Support Decision-Making on Healthcare.* In 5th ADVANCE 2017, Evry Val d'Essonne, France.
- **Gabriel Lopes, Vânia Vidal, and Mauro Oliveira**. *Construção de Linked Data Mashup para Integração de Dados da Saúde Pública.* In Proceedings of 31th Brazilian Symposium on Databases.

## 1.5 Estrutura da Dissertação

Essa dissertação possui 8 capítulos, contando com o capítulo introdutório. Os restantes são descritos a seguir.

**O Capítulo 2** - Fundamentação Teórica - apresenta uma síntese dos assuntos mais relevantes que servem de fundamentação para o entendimento dos demais capítulos desta dissertação. Nele a trajetória da *web* desde sua concepção até os dias atuais é descrita. Também são expostas as principais tecnologias da Web Semântica, como: RDF, Ontologias e SPARQL. Também são discutidas as abordagens e os desafios para integrar dados. Finalmente, foi mostrado como a Web Semântica, junto com a iniciativa *Linked Data* trazem um novo paradigma para integração de dados.

**O Capítulo 3** - *Frameworks* para Linked Data Mashup: Uma Revisão Sistemática - apresenta um estudo secundário na forma de revisão sistemática, cujo objetivo é conhecer as principais ferramentas para construção de *Linked Data Mashups* atualmente. Além disso, também é enfatizado quais as diferenças de cada *framework* com a abordagem proposta.

**O Capitulo 4** - GISSA: Governança Inteligente em Saúde - descreve a arquitetura, as funcionalidades e os componentes do sistema GISSA. Também é discutido como a abordagem proposta nessa dissertação pode agregar valor ao GISSA.

**O Capítulo 5** - *Framework* para especificação de Linked Data Mashups - apresenta o *framework* conceitual que originou o *framework* proposto nessa dissertação. Nele é descrito como um *Linked Data Mashup* pode ser formalmente especificado com o auxílio de visões exportadas, visões de *links* semânticos e regras de fusão. Além disso, para demonstrar a aplicabilidade do *framework*, é desenvolvido um estudo de caso que integra dados na Saúde Pública.

**O Capítulo 6** - Mediador Semântico - descreve o *framework* proposto nessa dissertação. Nele é discutido como um usuário de propósito geral pode construir *Linked Data Mashups* sem conhecimentos específicos em *Linked Data* ou em integração de dados. Também é descrito o processo reescrita, necessário para a reutilização dos *Linked Data Mashups*. Por fim, são apresentados casos de uso demonstrando as características do *framework*

**O Capítulo 7** - Mediador Semântico: Implementação - apresenta um guia de como especificar o *framework* descrito formalmente no Cáp. 5. Para isso, são expostos um modelo conceitual sobre a abordagem, bem como os principais algoritmos necessários pelo *framework*. Além disso, é discutido o processo de implementação de um protótipo da abordagem.

Por fim, **O Capítulo 8** - Conclusão - descreve os principais objetivos do *framework*. É discutido, como espera-se que a abordagem proposta agregue valor tanto à comunidade científica, com ênfase em *Linked Data*, quanto ao sistema GISSA. Além disso, os trabalhos e desafios futuros são pontuados.

# 2 Fundamentação Teórica

## 2.1 Introdução

Neste capítulo é apresentada, inicialmente, uma breve discussão sobre a evolução da *Web*. Em seguida, são discutidos alguns dos métodos mais comuns para integração de dados, bem como algumas de suas problemáticas. Nas subseções seguintes, são apresentadas as principais tecnologias da *Web Semântica*: RDF, SPARQL e OWL. Além disso, também é discutida a iniciativa *Linked Data* e suas abordagens para integração de dados.

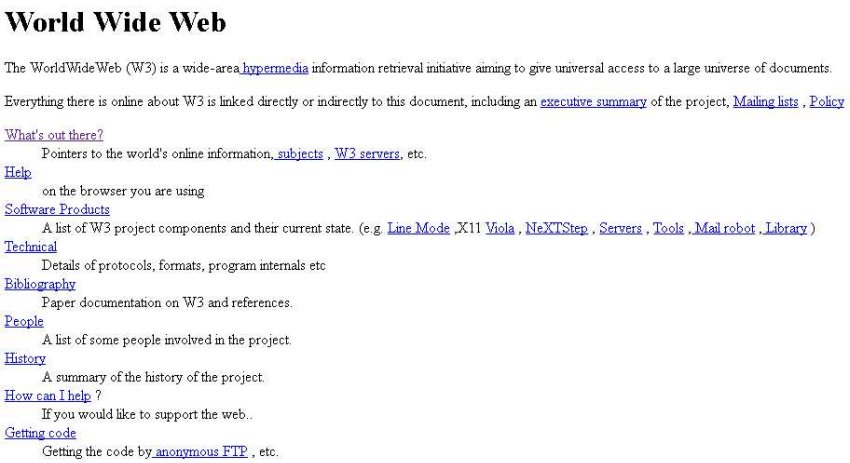
## 2.2 Evolução da Web: Web 1.0 a Web Semântica

Uma das dificuldades observadas por Tim Berner's Lee, durante o final da década de 80, era o fato das máquinas computacionais não encontrarem *links* entre objetos diferentes (LEE, 1998). Por exemplo, em uma empresa, não era uma tarefa trivial denotar que dois registros em máquinas distintas estão relacionados, como uma pessoa e a descrição do cargo que desempenha. Isso impedia o compartilhamento de informações entre os computadores de uma organização. Alguns dos desafios encontrados que impediam o acesso global às informações eram: (i) falta de um formato padrão para que diferentes sistemas operacionais pudessem entender a informação; (ii) definição de nomes comuns à entidades que desejava-se compartilhar e (iii) dificuldade em definir protocolos de envio e recebimento de dados.

Frente a esses desafios, Tim Berner's Lee propôs o *World Wide Web*: um "sonho", como abordado na época, de criar um espaço global que possibilitasse o compartilhamento de informações. Nesse *espaço global*, cada registro (documento) é dotado de um ***Identificador Universal de Documentos*** (UDI, posteriormente URI), que o permite ser acessado de qualquer local dentro desse espaço (BERNERS-LEE et al., 1992). O UDI faz com que a informação seja apresentada apenas uma vez e, a partir de então, criam-se *links* para referenciá-lo. Além disso, também foram propostos: a ***Linguagem de Marcação Hipertextual*** (HTML), uma linguagem para que diferentes sistemas entendam a informação e o ***Protocolo de Transferência de Hipertexto*** (HTTP), um protocolo para permitir a transferência de informações HTML dentro da rede (FIELDING et al., 1999). Esses foram apenas os passos iniciais para a grande evolução que estava por vir e da concretização da *Web* que conhecemos hoje.

### 2.2.1 Web 1.0

A primeira geração da *Web*, chamada de *Web 1.0*, também conhecida como *Web Cognitiva*, representava um espaço para compartilhamento de informações com pouca ou nenhuma interação com o usuário. As páginas *Web* eram (e ainda são) construídas utilizando a linguagem HTML. Essa página é então interpretada por um *Browser* que transforma o código HTML em componentes visuais intuitivos para os usuários. A Figura 3 apresenta a primeira página *web* criada[1].


**World Wide Web**

The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary of the project, Mailing lists , Policy

What's out there?
Pointers to the world's online information, subjects , W3 servers, etc.
Help
on the browser you are using
Software Products
A list of W3 project components and their current state. (e.g. Line Mode ,X11 Viola , NeXTStep , Servers , Tools , Mail robot , Library )
Technical
Details of protocols, formats, program internals etc
Bibliography
Paper documentation on W3 and references.
People
A list of some people involved in the project.
History
A summary of the history of the project.
How can I help ?
If you would like to support the web..
Getting code
Getting the code by anonymous FTP , etc.

Figura 3 – Primeira página Web da história

As palavras em azul na Figura 3 representam os *links*, chamados de ***hyperlinks***, entre duas páginas *Web*. A partir de um "clique"no *hyperlink*, o usuário é redirecionado à página referente a este link. Nesse estágio, ainda não havia contribuição por parte do usuário, assim, as páginas *Web* eram similares a um jornal, porém em HTML. Também foi na *Web 1.0* que surgiram os primeiros sites de *e-commerce*.

### 2.2.2 Web 2.0

A *Web 2.0* trouxe uma maior participação do usuário que, até então, se limitava a buscar e ler os conteúdos na *Web*. Nessa etapa, diversas tecnologias foram desenvolvidas, e.g. ***XML***(BRAY et al., 2008), uma linguagem de marcação com objetivo de promover a interoperabilidade entre sistemas; ***Google Web Toolkit***[2], um *framework* aberto para auxiliar o desenvolvimento de aplicações *Web*; ***FLEX***[3], um *kit* de desenvolvimento (SDK) que permite a troca de conteúdo entre aplicações *Web*. Essa evolução permitiu uma maior globalização da *Web* e uma maior participação do usuário. Nessa época foram criados

[1] http://info.cern.ch/hypertext/WWW/TheProject.html
[2] http://www.gwtproject.org/?csw=1
[3] http://www.adobe.com/products/flex.html

os *Feeds*, as redes sociais e os primeiros ***mashups***, que representam a combinação de informações heterogêneas. Os *mashups* de dados são detalhados na subseção 2.5.3.

### 2.2.3 Problemas da Web Sintática

Em um artigo publicado em 1998, o *Realising the Full Potential of The Web* (LEE, 1997), Tim Berner's Lee descreve algumas das problemáticas da *Web 2.0*. Segundo ele, a *Web* é um espaço com várias informações valiosas que podem auxiliar em diversas pesquisas, como: cura de doenças; previsões no mercado financeiro e na tomada de decisões. Porém, a maioria dessas informações está em um formato inteligível apenas para humanos, i.e., os dados estão dispostos numa estrutura sintática (e.g. HTML e XML) sem uma semântica definida, o que torna inviável para um programa de computador compreender e utilizar esses dados. Além disso, a *Web* vem crescendo de forma "desenfreada", i.e. as informações contidas nas páginas não atendem à um formato padrão de publicação. Assim, o usuário é livre para escolher como publicar esses dados. Por isso, esses dados, em sua grande maioria, estão armazenados em fontes isoladas umas das outras, onde a única ligação entre duas fontes é realizada na forma de *hyperlinks*, um recurso utilizado para levar o usuário humano à outra página. Os *hyperlinks* são recursos intuitivos para humanos, porém de difícil compreensão para *softwares*.

O problema de semântica nos documentos da *Web* também pode ser compreendido pela falta de mecanismos de buscas que utilizem palavras-chave. Nesse tipo de busca, conceitos com significados iguais, mas escritos de forma distintas, retornam resultados diferentes numa consulta. Isso acontece porque as *engines* de busca, e.g. Google e Yahoo!, frequentemente utilizam as palavras-chave da consulta sem relação com seus significados para realizar determinada busca. Por exemplo, a página "*www.exemplo.com.br/Pessoas/Gabriel*"representa, intuitivamente, uma página HTML contendo um texto descritivo sobre o indivíduo Gabriel, como o perfil acadêmico e o endereço do local de trabalho. Esta página podem conter *hyperlinks* que levam um usuário humano a outra página, como a página da instituição onde trabalha, o Instituto Federal do Ceará[4](IFCE), por exemplo. Como esses dados estão num formato textual sem uma semântica definida, uma *engine* de busca por palavra-chave não seria capaz de responder a seguinte consulta:

*"Quais são os profissionais que fazem parte do Programa de Mestrado no Instituto Federal do Ceará e que trabalham com Web Semântica?*

Percebe-se que, se a página do IFCE não contiver as palavras *"Gabriel"*, *"Profissionais"*, *"Mestrado"* e *"Web Semântica"* relacionadas, a busca não retornará o esperado.

Para solver esse problema da falta de semântica nas informações, podem ser citadas duas abordagens. A primeira é utilizar algoritmos de Inteligência Artificial e Aprendizagem

[4] http://www.ifce.edu.br

de Máquina para interpretar os textos descritos nos arquivos HTML. Os maiores desafios dessa abordagem são manter uma taxa de erro aceitável, diante do constante e rápido crescimento da *Web*, e, como a extração da semântica será feita a partir da palavra, devem ser desenvolvidos interpretadores para cada linguagem (CHAU; CHEN, 2008), (ZHOU; MASHUQ, 2013). Esta segunda abordagem é expressar o significado do conteúdo de uma página *Web* de forma compreendível também por máquinas. Assim, com um formato padrão para descrever a semântica do conteúdo de sites *Web*, algoritmos poderiam recuperar dados desses sites para criar novas aplicações. A segunda abordagem é o conceito base da *Web Semântica*

### 2.2.4 Web 3.0 - Web Semântica

A terceira geração da Web, conhecida como *Web Semântica*, tem como objetivo descrever o significado dos dados publicados na *Web* de uma forma inteligível tanto para humanos quanto para computadores, facilitando o processamento e a integração de dados (BERNERS-LEE et al., 2001). A ideia básica da Web Semântica é utilizar padrões para descrever semanticamente objetos do mundo real publicados na *Web* e atribuir *links* entre eles, permitindo que um computador compreenda o significado das informações e consiga fazer descobertas de conteúdo em tempo de execução (AGHAEI; ALI; KHOSRAVI, 2012). A princípio, pode parecer que a Web Semântica tem como objetivo *substituir* a *Web* atual, porém, a Web Semântica representa uma camada acima, i.e. não interfere nos dados já publicados, mas atua dando uma maior utilidade à *Web* convencional.

Na Web Semântica, uma *homepage* deixa de ser representada apenas por um conjunto de códigos HTML, para também utilizar um arquivo de descrição, responsável por atribuir significado às informações contidas na *homepage*. Dessa forma, no exemplo descrito na subseção 2.2.3, se a página "*www.exemplo.com.br/Pessoas/Gabriel*"tivesse sido publicada utilizando os padrões da Web Semântica, ela iria conter um conjunto de *metadados* que descreve o objeto real, o indivíduo Gabriel, de uma forma inteligível para um computador, utilizando uma linguagem padrão para que outros computadores ao redor do mundo também possam acessá-lo. Além disso, esse arquivo também conteria *links* lógicos, interligando o objeto Gabriel à outros objetos-reais, como a instituição onde trabalha e atividades que desempenha. Assim, um interpretador semântico facilmente responderia a pergunta do exemplo. As tecnologias necessárias para que a Web Semântica desempenhe o papel que promete são discutidas na Seção a seguir.

## 2.3 Tecnologias da Web Semântica

Nesta seção, serão apresentadas as principais tecnologias da Web Semântica. Primeiramente é descrito o *RDF* e *RDFS*, tecnologias-chave para descrever objetos do mundo

real. Em seguida, é discutido o conceito de *ontologias*, que é a base da infraestrutura da Web Semântica, e a linguagem de consultas sobre bases *RDF*, *SPARQL*.

A Web Semântica descreve uma *Web* de dados ao invés de uma *Web* de documentos. Para isso, são necessárias linguagens com poder de representatividade suficiente para descrever *quaisquer* informações na *Web*. Nesse sentido, a W3C[5], um consórcio entre empresas e laboratórios de pesquisas para descrever padrões na *Web*, desempenha um papel fundamental no desenvolvimento da Web Semântica. Com o apoio do W3C, diversos padrões já foram definidos. Esses padrões, bem como outras tecnologias utilizadas na Web Semântica, são resumidos na Figura 4. Esse famoso diagrama, conhecido como "Bolo de Noiva"(tradução aproximada de *layer-cake*), resume as tecnologias da Web Semântica(W3C, 2000).

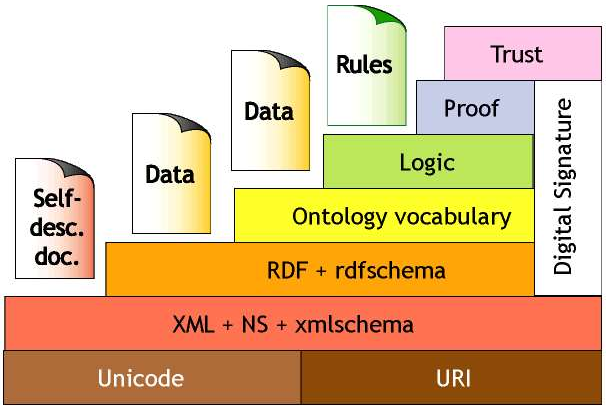



Figura 4 – Camadas da Web Semântica

### 2.3.1 Ontologias

Ontologia é um conceito original da Filosofia, inicialmente usado para conceituar coisas, elementos da natureza e pensamentos (GRUBER, 1993). A Computação importou esse conceito com o propósito de representar o conhecimento na forma de modelos, a fim de permitir inferências e a interoperabilidade entre sistemas heterogêneos (CHANDRASEKARAN; JOSEPHSON; BENJAMINS, 1999).

O fato de ontologias, na computação, serem representadas por definições formais, significa que computadores podem realizar raciocínios sobre elas. Dessa forma, o uso de ontologias pode melhorar a acurácia de buscas, uma vez que *engines* podem recuperar dados de um determinado conceito. Na Web Semântica, ontologias representam a base arquitetural da Web Semântica. A seguir são apresentadas tecnologias necessárias para descrever as ontologias.

[5] https://www.w3.org/

### 2.3.2 RDF

No princípio da Web, um dos desafios no compartilhamento de informações era a falta de uma linguagem padrão para permitir que diferentes sistemas entendessem a informação. Na época, a interoperabilidade foi alcançada com o HTML, uma linguagem padrão para construir páginas na Web, onde um programa cliente (*browser*) a interpreta. Na Web Semântica o desafio está em descrever o significado dos dados num formato comum.

O **Resource Description Framework**(RDF) é uma linguagem baseada em XML com objetivo de promover uma padronização para descrever semanticamente dados na Web. Inicialmente proposto em 1999 (LASSILA; SWICK, 1999), RDF é uma recomendação da W3C. Também pode ser considerado um modelo de dados, pois também auxilia na modelagem conceitual dos dados (W3C, 2004). Diferentemente de outras linguagens, o *RDF* é capaz de descrever um objeto à nível semântico, possibilitando que um algoritmo consuma esses dados e compreenda o significado do objeto. Nele podemos definir que um dado objeto do mundo real possui determinada propriedade, definindo *links* entre objetos reais. Em *RDF*, os dados são escritos no formato de triplas: **sujeito, predicado e objeto**. Além disso, o RDF trata todas as informações na Web como **recursos** .

O **sujeito** geralmente representa um objeto real, como um indivíduo, uma instituição, um local ou um alimento. O **objeto** pode representar um objeto real ou um literal, e.g. um valor, um texto (*string*) ou uma data. O **predicado** faz o papel de relacionar um sujeito à seu objeto. Tanto o objeto quanto o sujeito são representados através de um ***Universal Resource Identifier*** (URI), que os identifica de forma única em toda a *Web*. Dessa forma, o indivíduo *Gabriel* pode ser representado pela URI *"www.example.com.br/Pessoas/Gabriel"* e ser acessado por qualquer sistema na *Web*. Por exemplo, a expressão

*"Gabriel Lopes é um aluno de mestrado do IFCE e trabalha com Web Semântica"*

pode ser escrita em RDF como:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:ex="http://www.example.com.br/vocab#">
    <rdf:Description
       rdf:about=www.example.com.br/Pessoas/Gabriel>
           <ex:nome> Gabriel Lopes </ex:nome>
           <ex:curso rdf:resource = ex:Mestrado>
           <ex:estudaEm rdf:resource=
               www.ifce.edu.br#>
           <ex:trabalhaCom rdf:resource = ex:WebSemantica>
    </rdf:Description>
</rdf>
```

Onde o **sujeito** é `www.example.com.br/Pessoas/Gabriel`; os **predicados** são `ex:nome`, `ex:curso`, `ex:estudaEm` e `ex:trabalhaCom` e, finalmente, os **objetos** são `Gabriel Lopes` (*string*), `ex:Mestrado`, `www.ifce.edu.br` e `ex:WebSemantica`. Essa tripla RDF também pode ser representada em formato de grafo, onde os **nós** representam os sujeitos e objetos, enquanto as **arestas** representam os predicados. A Figura 5 representa um exemplo de tripla RDF representada no formato de grafo.

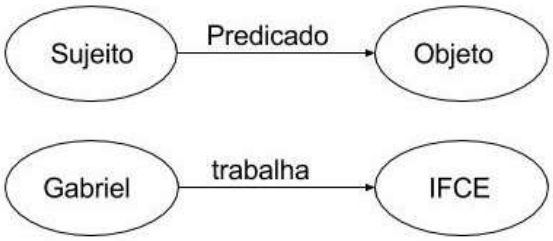



Figura 5 – Exemplo de Grafo RDF

A princípio, pode parecer que a mesma expressão também poderia ser escrita em XML. No entanto, em XML não há como definirmos qual palavra deve ser utilizada para representar um conceito. No exemplo acima, foi utilizado o predicado `ex:estudaEm` para identificar o local onde o indivíduo estuda. Esse predicado tem diversos sinônimos, como `ex:éAlunoDe`, e, sem uma padronização nas nomenclaturas dos recursos, não teríamos uma interoperabilidade semântica, visto que cada provedor de dados poderia descrevê-los em um formato próprio. Além disso, graças a flexibilidade do XML, uma informação pode ser escrita de diversas formas. Por exemplo, a seguir temos 2 exemplos de descrever a afirmação "Gabriel Lopes trabalha no IFCE".

```
<pessoa href="#Gabriel">
 <detalhes>
   <nome>Gabriel Lopes</nome>
   <trabalho>IFCE</trabalho>
 </detalhes>
</pessoa>
...
<pessoa>
 <uri>www.ifce.edu.br/Gabriel#</uri>
  <trabalhaEm>IFCE</trabalhaEm>
</pessoa>
```

Note que ambas maneiras estão perfeitamente corretas e serão interpretadas normalmente por um interpretador XML. Ao interpretar esse XML, o interpretador constrói uma árvore de informações, onde, por exemplo, no primeiro exemplo teríamos que `nome` é um elemento filho de `pessoa` e assim por diante. A partir de então, *softwares* podem analisar essa informação e extrair conhecimento dela. Porém, graças à flexibilidade do XML, temos várias possibilidades para representar conceitos e, portanto, voltamos à problemática da dificuldade em padronização.

Já em RDF, tudo é considerado um **recurso**: sujeitos, predicados e objetos. Cada recurso está associado a uma URI, que identifica de forma única um recurso na Web. Dessa forma, é possível utilizar um mesmo conceito em diversas aplicações distintas. No nosso exemplo, o predicado `ex:estudaEm`, que é um recurso, é identificado por uma URI, por exemplo `www.example.com.br/vocab#estudaEm`, que pode ser acessada por qualquer computador na Web. Além disso, os vocabulários são definidos por meio de ontologias, que ajudam a contextualizar o significado do termo. O trecho a seguir foi retirado documento RDF[6] do vocabulário FOAF e define a propriedade `foaf:name`. As propriedades `RDFS:range` (linha 7) e `RDFS:domain` (linha 6) definem, respectivamente, quais as classes do objeto e sujeito esperados pela propriedade. Por exemplo, ao encontrar a propriedade `foaf:name`, um computador pode dereferenciá-lo a partir de sua URI e descobrir que essa propriedade faz a ligação de uma `owl:Thing` com um `owl:Literal` (BRICKLEY; MILLER, 2010).

```
<rdf:Property rdf:about="http://xmlns.com/foaf/0.1/name"
    vs:term_status="testing" RDFS:label="name" RDFS:comment="A name for some
    thing.">

   <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <RDFS:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <RDFS:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
    <RDFS:isDefinedBy rdf:resource="http://xmlns.com/foaf/0.1/"/>
    <RDFS:subPropertyOf
        rdf:resource="http://www.w3.org/2000/01/rdf-schema#label"/>
</rdf:Property>
```

#### 2.3.2.1 Serialização RDF

Dados RDF podem ser escritos em diversos formatos, conhecidos como serializações. As principais serializações RDF são: ***Turtle*** (CAROTHERS; PRUD'HOMMEAUX, 2014), ***N-triples*** (GRANT; BECKET, 2004) e RDFa. Dentre essas, cada uma tem suas características: *N-triples* é mais facilmente lida por um computador; *Turtle* apresenta uma

[6] http://xmlns.com/foaf/spec/index.rdf

descrição de RDF mais legível na percepção humana, sendo mais didática, enquanto *RDFa* permite embarcar código RDF em uma página HTML. Nesta dissertação, em geral, será utilizada a serialização *Turtle* em exemplos. Uma possível representação do exemplo RDF/XML anterior em *Turtle*, é:

```
@prefix ex: <http://www.example.com.br/vocab#> .

<http://www.example.com.br/Pessoas/Gabriel>
    ex:nome "Gabriel Lopes" ;
    ex:estudaEm <http://www.ifce.edu.br> ;
    ex:trabalhaCom ex:WebSemantica .
```

#### 2.3.2.2 RDF-Schema

A propriedade `ex:estudaEm`, embora identificada por uma URI, não provê semântica o suficiente para um computador entender seu significado. Um usuário humano, ao ler a tripla `"Gabriel" ex:estudaEm "www.ifce.edu"` percebe, intuitivamente, o seu significado, pois conseguimos extrair a semântica por meio do texto da propriedade sem ter que analisar o contexto. Um computador, porém, precisa de mais informações, como: "Quais tipos de objetos esse predicado faz ligação"?; "Essa propriedade pode receber um inteiro como objeto?". Caso contrário, `ex:estudaEm` continua sendo apenas um texto e continuaríamos com os problemas do XML e HTML: falta de homogeneidade entre os termos utilizados para denotar conceitos e falta de semântica nos dados.

A **Linguagem para Definição de Vocabulários RDF** (RDFS) (MCBRIDE, 2004) é uma extensão do RDF e provê a base da interoperabilidade semântica na Web. O RDFS permite descrever os recursos na forma de **classes**, **propriedades** e **valores**, fornecendo um modelo para os objetos reais. Com o RDFS, podemos definir que toda propriedade `ex:estudaEm` faz o relacionamento entre um indivíduo do tipo `ex:Aluno` e uma entidade do tipo `ex:Instituição`, por exemplo. Dessa forma, um computador pode compreender que a propriedade `ex:estudaEm` define a ligação entre um aluno e uma instituição. Em RDF, definimos expressões relacionadas a um objeto em específico, como o indivíduo `Gabriel Lopes`. Em RDFS, as expressões são generalizadas. Por exemplo, a expressão

*"O Aluno estuda em uma Instituição de Ensino"*

é definida em RDFS da seguinte forma:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix RDFS: <http://www.w3.org/2000/01/rdf-schema#>
@prefix foaf: <http://xmlns.com/foaf/0.1>
@prefix ex: <http://www.example.com.br/vocab#>
```

```
ex:estudaEm rdf:type rdf:Property;
      RDFS:domain foaf:Person;
      RDFS:range ex:InstituicaoEnsino;
```

A Figura 6 representa graficamente o relacionamento entre RDF e RDFS.

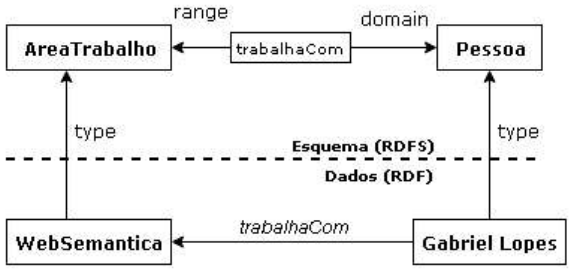



Figura 6 – Relacionamento de RDF e RDFS

Os termos *domain* e *range* determinam, respectivamente, o domínio de uma propriedade e os tipos com quem esta pode se relacionar.

Um modelo definido com o RDFS é chamado de **vocabulário** e é acessado por meio de uma URI, podendo ser reutilizado em outras aplicações. Ao acessar essa URI, usuários e computadores podem ter acesso a todas propriedades definidas nesse vocabulário. Dentro de um arquivo RDF, essa URI pode ser abreviada, gerando um **prefixo**, representado por *@prefix*, que torna o código RDF mais legível. O termo `ex:` do exemplo apresentado é, na verdade, a abreviação da URI do vocabulário `example`, onde são definidas todas as propriedades e classes usadas no arquivo RDF. Assim, quando utilizamos a propriedade, por exemplo `ex:curso`, estamos acessando a URI `http://www.example.com.br/vocab#curso`, que contém a descrição semântica dessa propriedade.

**Vocabulários** são a peça-chave para a homogeneidade de termos na Web. Uma das boas práticas na Web Semântica define que devemos, sempre que possível, reutilizar vocabulários conhecidos. No exemplo apresentado, foi utilizado o vocabulário *Friend of a Friend* (FOAF) (BRICKLEY; MILLER, 2010). *Friend of a Friend* (FOAF), amplamente difundido na Web Semântica para descrever pessoas: relacionamentos sociais e profissionais, características, dentre outros. O FOAF define conceitos concretos, como uma Empresa (*foaf:Organization*) ou uma Pessoa (*foaf:Person*); bem como conceitos abstratos, como o ato de conhecer outra pessoa: "Pessoa1 *conhece* (*foaf:knows*) Pessoa2".

### 2.3.3 OWL

Na Web Semântica, os dados são descritos por ontologias, o que permite o raciocínio sobre os dados, i.e. descobrir conceitos a partir do que já existe. O raciocínio lógico

sobre os dados é feito utilizando um *software* chamado de **raciocinador** (ou *reasoner*, no inglês). Para isso, são necessárias linguagens com poder de representatividade o suficiente para descrever os conceitos de uma ontologia. Embora o RDFS tenha um poder de representatividade maior que o RDF, ainda não é capaz de descrever expressões mais complexas, presentes em ontologias, como a relação entre coisas em vocabulários diferentes ou de definir um grupo sobre classes distintas. Por exemplo, a expressão

*"Toda Instituição de Ensino que contém Mestrado e Doutorado são consideradas Universidades"*

não pode ser definida em RDFS, pois não contém uma propriedade de cardinalidade.

Nesse contexto, existem diversas linguagens para descrever ontologias, onde **Web Ontology Language - OWL** (MCGUINNESS; HARMELEN, 2004) é a mais difundida. OWL, assim como RDFS e RDF, é um padrão W3C[7] desde 2003 (HORI; EUZENAT; PATEL-SCHNEIDER, 2004) e representa uma extensão do RDFS, permitindo definir diversos conceitos adicionais, como:

- **Operações de conjuntos**. Com OWL, é possível definir operações como união (*owl:unionOf*), interseção (*owl:intersectionOf*) e disjunção (*owl:disjointWith*) entre classes.
- **Similaridade entre objetos distintos**. Também é possível definir a similaridade de duas instâncias de objetos com vocabulários distintos. A propriedade *owl:sameAs* define que duas instâncias representam um mesmo objeto do mundo real.
- **Cardinalidade**. Com a propriedade *owl:cardinality* é possível definirmos uma restrições de cardinalidade em uma ontologia.

#### 2.3.3.1 Racionadores OWL

OWL permite descrever de forma lógica e formal uma ontologia. Essa característica do OWL, faz com que um *software* seja capaz de *raciocinar* sobre os dados, permitindo a descoberta de novas entidades ou de descobrir conceitos falhos, i.e. verificar se a integridade da ontologia foi comprometida pelos dados. Para isso, são utilizados os **Raciocinadores Semânticos**: *softwares* especializados em inferir sequências lógicas a partir de um conjunto de **fatos**, os **Axiomas** em ontologias(SIRIN et al., 2007). Uma lista com vários raciocinadores semânticos pode ser encontrada em (MANCHESTER, 2016). A seguir, são apresentados dois exemplos de atuação dos raciocinadores.

Suponha que uma determinada ontologia defina que a propriedade `ex:estudaEm` relaciona entidades do tipo `foaf:Person` com `ex:InstituicaoEnsino`. Se em um arquivo RDF

[7] https://www.w3.org/OWL/

contiver uma tripla que não atenda a esse requisito, o raciocinador OWL consegue descobrir tal inconsistência. Nesse exemplo, a tripla `"Gabriel"(tipo` ***foaf:Person*** `ex:estudaEm "IFCE"(`***string***`)` é inconsistente. A Figura 7 representa este cenário.

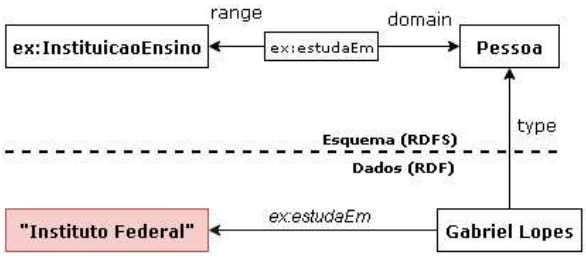



Figura 7 – Exemplo de inconsistência perceptível ao OWL

O raciocinador também é capaz de inferir conceitos implícitos em uma ontologia. Por exemplo, é definido em uma ontologia que toda Instituição de Ensino com cursos de Mestrado e Doutorado são consideradas Universidades. O raciocinador então examina as triplas RDF, a fim de descobrir se há alguma Instituição que atenda aos requisitos de ser uma Universidade, mas que não esteja classificada como uma. As Figuras 8a e 8b representam graficamente o antes e depois da ação do raciocinador, respectivamente.

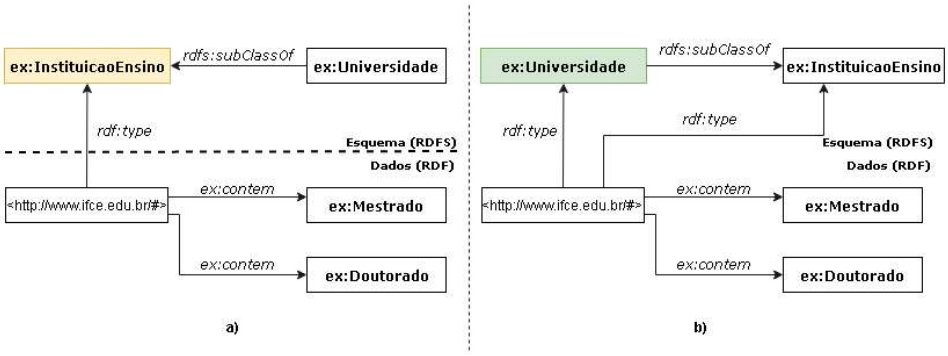



Figura 8 – Exemplo de ação do raciocinador

Inserir a capacidade de realizar raciocínios lógicos em uma linguagem para descrição de ontologias não é uma tarefa trivial. Deve seve ser levado em consideração que a capacidade de representatividade de uma linguagem é inversamente proporcional ao seu desempenho (HEFLIN et al., 2007), i.e. quanto mais conceitos podem ser representados por uma linguagem, menos desempenho ela terá. Por conta disso, OWL oferece 3 sub-linguagens (SMITH; WELTY; MCGUINNESS, 2004):

- **OWL Lite**: Possui um subconjunto limitado de construtores OWL. É destinada à usuários com necessidades simples de modelagem. A maioria das ontologias é descritas utilizando essa linguagem.

- **OWL DL**: Baseada em lógica descritiva de primeira ordem, chamada em inglês de *Description Logic*. Destinada à usuários que não desejam perder o processamento computacional, uma vez que tudo descrito nessa em OWL DL é garantido de ser computado.
- **OWL Full**: Destinada à usuários que querem o máximo poder de representatividade de ontologias. Diferentemente de *OWL DL*, não há garantias formais de que tudo descrito nessa linguagem será computado.

As linguagens possuem representatividade acumulativa, i.e. OWL *Full* pode representar tudo que OWL DL e OWL Lite são capazes, e assim por diante. A Figura 9 representa em forma de diagramas o poder de representatividade de cada linguagem OWL.


Figura 9 – Representatividade das sub-linguagens de OWL como Diagramas de *Venn*

Mesmo a linguagem mais básica do OWL, OWL *Lite*, já possui um bom poder de representatividade. Em 2006, Wang *et al.* (WANG; PARSIA; HENDLER, 2006) analisou 1275 ontologias na *Web* e descobriu que 924 delas estavam em OWL *Full*. Entretanto, a maioria dessas ontologias poderia ser automaticamente convertidas para OWL *Lite* ou OWL DL. Após as conversões, apenas 61 ontologias permaneceram em OWL Full. Com isso, Wang argumentou que a maioria das ontologias não precisa de tanta expressividade adicional.

### 2.3.4 SPARQL

Até essa Seção, foram vistas tecnologias capazes de descrever semanticamente diversos conhecimentos na Web, transformando esse conhecimento em dados RDF. Esses dados, assim como diversos outros, podem ser armazenados em *softwares* de bancos de dados. Bancos de Dados relacionais, por exemplo, armazenam os dados em formatos de *tuplas* em linhas e colunas, permitindo consultas sobre as tabelas de dados. Os dados RDF, por sua vez, podem ser armazenados como grafos em memória, arquivos de texto ou em *frameworks* específicos para armazenamento de triplas RDF, chamados de **RDF Stores**, *Triple Stores*

ou *Quad Stores*. Um *RDF Store* provê mecanismos para o armazenamento persistente dos dados e de acesso aos grafos RDF. Existem diversas *Triple Stores*, como o *Virtuoso*[8] (SEVERAL, 2009) e *Fuseki*[9].

Além disso, tais *frameworks* também disponibilizam uma *interface* para consultas sobre as bases RDFS, chamadas de **SPARQL Endpoint**. **SPARQL** (PRUD'HOMMEAUX; HARRIS; SEABORNE, 2013) é uma linguagem de consultas sobre bases RDF. Assim como as demais tecnologias apresentadas, também é um padrão W3C. A Figura 10 representa um exemplo de SPARQL *Endpoint*. Este *Endpoint* pertence à DBPedia[10].

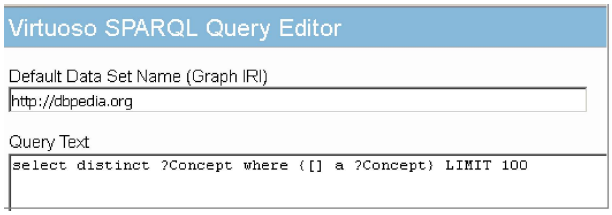



Figura 10 – SPARQL Endpoint DBPedia.

Além disso, SPARQL não é apenas uma linguagem para consulta sobre os dados, mas também é um protocolo usado para enviar consultas e recuperar resultados por meio do protocolo HTTP.

### 2.3.5 R2RML

Sempre que um modelo de dados tiver de ser transformado em outro, o processo intuitivo é o de criar *mapeamentos* entre as entidades. Por exemplo, se um sistema quiser migrar o modelo de dados relacional para o RDF, têm-se, primeiramente, que criar mapeamentos entre os dois modelos. Para isso, tem-se a linguagem R2RML (W3C, 2016). Assim como o SPARQL e RDF, R2RML é um padrão W3C para mapeamentos de dados relacionais para RDF.

Mapeamentos R2RML são definidos na forma de declarações. Cada declaração é uma *TripleMap* e define o mapeamento entre entidade do banco relacional com um recurso no RDF. Em uma *TripleMap*, são definidos:

- *Logical Table*. Referente à tabela lógica no banco de dados relacional;
- *SubjectMap*, que determina o sujeito nos dados RDF;
- *PredicateObjectMap*. Mapeamentos dos predicados e objetos referentes ao sujeito (*subjectmap*).

No exemplo abaixo, uma entidade do tipo *"tb_pessoa"*, de um banco relacional, é mapeada numa entidade do tipo *"ex:Pessoa"*, de um arquivo RDF.

[8] http://virtuoso.openlinksw.com/
[9] https://jena.apache.org/documentation/serving_data/
[10] http://dbpedia.org/sparql

### 2.3.6 Implementação de ontologias

Existem diversos *frameworks* que auxiliam na manipulação de ontologias e arquivos RDF em ambientes de desenvolvimento. Alguns deles são mostradas a seguir.

O **Jena** (CARROLL et al., 2004) é um *framework* originalmente criado pela *HP Labs*, atualmente mantido pela Apache. É um *framework* para JAVA e fornece APIs para que *softwares* possam manipular arquivos RDF e manipular ontologias. As principais características do Jena são:

- Conta com a RDF API, que suporta os formatos de serialização RDF mais populares, como *Turtle*, N3 e *N-Triples*;
- Permite trabalhar com as principais tecnologias da Web Semântica: RDF, RDFS, RDFa e OWL;
- Possui uma *engine*, Jena ARQ, que permite realizar consultas sobre dados RDF;
- Por meio da *Inference API*, Jena inclui suporte à diversos algoritmos de inferência.
- Capaz de disponibilizar um SPARQL *Endpoint* (Jena TDB ou Fuseki).

**OWL API** é uma API livre, também para desenvolvimento em Java, que auxilia na criação, manipulação e serialização de ontologias OWL (HORRIDGE; BECHHOFER, 2011). É mantida pela Universidade de Manchester e conta com diversos colaboradores, como a emprsa *Stardog* [11] e Universidade de Ulm [12]. Suas principais características são:

- Contém analisadores (*parser*) e escritores (*writer*) OWL/XML e RDF/XML;
- Dá suporte aos formatos *Turtle*, N3 e *N-Triples*;
- *Parser* e *Writer* OWL/XML;
- Trabalha com diversos dos principais raciocinadores *(reasoners)* usados atualmente, como FaCT++ (TSARKOV; HORROCKS, 2006), HermiT (SHEARER; MOTIK; HORROCKS, 2008), Pellet (PARSIA; SIRIN, 2003) e Racer (HAARSLEV; MÖLLER, 2001).

## 2.4 Integração de Dados

Nessa seção, serão abordados os conceitos de Integração de Dados, as abordagens existentes e seus desafios. Também serão discutidas as motivações e os benefícios para integrar dados.

[11] http://stardog.com/
[12] http://www.informatik.uni-ulm.de/ki/noppens.html

Integração de dados é o processo de combinar dados de fontes distribuídas, possivelmente heterogêneas, a fim de prover ao usuário uma **visão integrada** sobre esses dados, lhe dando a impressão de interagir sobre uma única base de dados (LENZERINI, 2002). Um **banco de dados distribuído** é um conjunto de dados que, possivelmente, pertence à uma mesma instituição, mas seus dados estão espalhados fisicamente na forma de *bases de dados locais* (CERI G. PELAGATTI, 1981). Uma *visão integrada* pode unir informações complementares, que, ao serem combinadas, pode originar novos fatos. No contexto de Banco de Dados, uma **visão** é a representação da estrutura de um conjunto de dados.

Além disso, um banco de dados distribuído pode ser classificado em duas categorias: **homogêneo**, quando suas bases locais estão descritas num mesmo formato e por um modelo de dados comum, ou **heterogêneo**, quando o formato e/ou modelo de suas bases locais são distintos. Segundo (BATINI; LENZERINI; NAVATHE, 1986), bases de dados distribuídas acontecem principalmente por dois motivos: (i) a estrutura de um banco de dados para grandes instituições é muito complexa para ser modelada como uma única visão e (ii) grupos de usuários e/ou empresas comumente operam independentemente, desenvolvendo seus próprios *softwares* e bases de dados. A seguir, são discutidas as motivações para integração de dados, as principais abordagens e os desafios.

### 2.4.1 Motivação para Integrar Dados

A integração de dados é essencial em diversos cenários. Na área de Inteligência Empresarial (*Business Intelligence*), por exemplo, a integração de dados pode ser usada para: consultas e criação de relatórios; análises estatísticas, *online analytical processing* (OLAP); mineração de dados; dentre outros (ZIEGLER; DITTRICH, 2007). A integração de dados promove à empresa uma visualização em diferentes perspectivas sobre os dados, tendo como objetivo promover vantagens competitivas nos negócios.

#### 2.4.1.1 Cenário do Sistema de Saúde brasileiro

A integração de dados também é fundamental no apoio a tomada de decisões. Diferentes fontes de dados podem conter informações complementares que, quando combinadas, podem dar origem à novos fatos e contextualizar vários problemas. Por exemplo, no Sistema de Saúde brasileiro (SUS), existem diversas bases de dados isoladas umas das outras. Numa delas, o e-SUS, há informações sobre os indivíduos que utilizaram um posto de saúde ou hospital público. Essa base contém informações sobre o indivíduo, como: o uso de drogas, de álcool e tabaco; doenças crônicas, como diabetes e câncer; dentre outras. Em outra base de dados do SUS, o Sistema de Informações sobre Nascidos Vivos (SINASC), há informações sobre gestações. Nessa base, há informações como: a quantidade de consultas pré-natal que a gestante realizou durante a gravidez; complicações em gestações anteriores; informações sobre o recém-nascido, como peso e possíveis anomalias; dentre outras. Essas duas bases de

dados, e-SUS (1) e SINASC (2), são isoladas uma da outra, i.e. suas informações, apesar de complementares, estão em formatos distintos. Assim, um gestor de Saúde é impossibilitado de consultar, por exemplo, se a mãe de um recém-nascido (2) é fumante ou usuária de drogas (1).

### 2.4.2 Abordagens e Desafios

Desde a década de 80 que a área de Integração de Dados é foco ativo de pesquisas. Trabalhos da época já abordavam os conceitos, benefícios e as problemáticas para integração de dados heterogêneos. Em (BATINI; LENZERINI; NAVATHE, 1986) é descrito um *framework* genérico para integração de dados, resumido na Figura 11. Nesse método, que continua influenciando pesquisas até os dias atuais, os dados das fontes heterogêneas são conciliados por meio de um *Esquema Global*. Além disso, devem ser criados mapeamentos entre os *Esquemas Locais*, referentes às fontes, para o esquema global. Ao realizar uma consulta sobre o esquema global, o *framework* utiliza os mapeamentos definidos para traduzir a consulta nos formatos das bases locais.

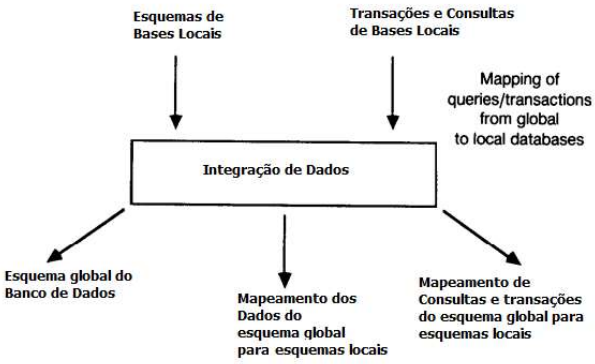



Figura 11 – Processo genérico para integração de dados

Extendendo essa abordagem genérica, (LENZERINI, 2002) formaliza uma integração de dados, $\zeta$, como uma tupla no formato $\zeta = \{G, S, M\}$, onde:

- **G** é o esquema global das fontes de dados;
- **S** é o esquema de uma base local;
- **M** é o conjunto de mapeamentos entre $S$ e $M$, constituída por assertivas na forma de $q_s \rightarrow q_g$.

Além disso, também discute sobre duas possíveis abordagens para definição dos mapeamentos. Na primeira, chamada de *Local-as-View* - LAV (visão local), os mapeamentos de $M$ associam cada elemento $s \in S$ a uma consulta $q$ sobre $G$. Nessa abordagem, tem-se a ideia que as fontes de dados devem ser expressas num modelo conceitual comum, como um modelo de dados numa empresa ou uma ontologia. Essa abordagem, que segue um modelo *bottom-up*, permite uma maior extensibilidade para novas fontes de dados e é comumente

utilizada quando as fontes são conhecidas e deseja-se integrá-las. A outra abordagem, chamada de *Global-as-View* - GAV (visão global), os mapeamentos de $M$ associam cada elemento $g \in G$ a uma consulta $q$ sobre $S$. Nessa abordagem, um esquema global é criado quando ainda não se conhecem as fontes a serem integradas. Essa abordagem segue um modelo *top-down* e é comumente utilizada, por exemplo, em ambientes *Web*, quando primeiro modela-se o problema para então achar as fontes de dados. Os modelos *bottom-up* e *top-down* são brevemente discutidos na próxima subseção.

### 2.4.3 Visão Integrada

Independente da abordagem para integração utilizada, existem problemas comuns ao tentar interoperar os dados. A heterogeneidade das bases de dados podem ocorrer de diversas formas, desde *hardwares* e *softwares* que o banco de dados é baseado, até modelos de dados, esquemas e formatos (e.g. texto, vídeo) distintos. Dentre os tipos de heterogeneidade, destaca-se a **heterogeneidade semântica** pela sua complexidade. Essa heterogeneidade representa a divergência de como um mesmo objeto pode ser representado em duas bases de dados distintas (HAMMER; MCLEOD, 1993; HULL, 1997). Ao tentar solver esse tipo de heterogeneidade, vários outros desafios são criados, como: (i) criar um *Esquema Global* (ou *Visão Integrada*) que concilie as demais visões num ambiente heterogêneo; (ii) manter a visão integrada atualizada; (iii) unir representações distintas de um mesmo objeto do mundo real; (iv) definir mapeamentos; dentre outros (HULL, 1997). O processo de resolução dessa heterogeneidade é chamado de **Integração Semântica**.

Uma visão integrada pode ser construída utilizando uma abordagem *bottom-up* ou *top-down*. Na primeira, a visão integrada é construída a partir dos esquemas das fontes de dados subjacentes já conhecidas. Nessa abordagem, apenas os fatos descritos pelos dados são verdadeiros, seguindo a definição de (REITER, 1984) para *Closed World Assumption* - CWA (Hipótese de Mundo Fechado). Na segunda abordagem, *top-down*, a visão integrada é construída antes de se conhecer os esquemas das fontes de dados. As fontes de dados são expressas na forma de subconjuntos dessa visão integrada (ULLMAN, 2000). Ao contrário da abordagem *bottom-up*, essa visão integrada segue um modelo *Open World Assumption* - OWA (Hipótese de Mundo Aberto), i.e. um fato só é falso se os dados o definem como falso, se não, é verdadeiro ou *desconhecido*. Essa abordagem é comumente utilizada quando nem todos os dados são explicitados, como é o caso da Web Semântica. A Figura 12 representa graficamente a diferença entre os modelos OWA e CWA.

Além disso, uma visão pode ser **virtual** ou **materializada**. Uma visão virtual não contém instâncias de um banco de dados, mas apenas a definição de um modelo de dados. Nesse enfoque, os dados são obtidos diretamente das fontes apenas no momento da consulta. Uma visão materializada, porém, contém, fisicamente, as instâncias de um banco de dados. A Figura 13 demonstra graficamente a diferença dos enfoques.

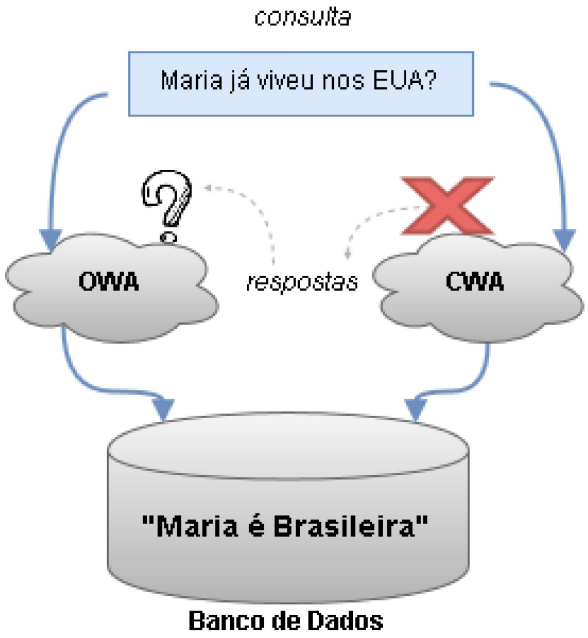



Figura 12 – Representação gráfica das abordagens OWA e CWA.

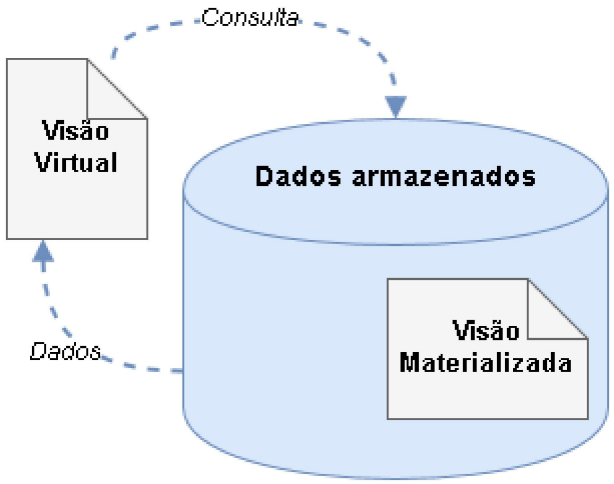



Figura 13 – Representação gráfica da diferença dos enfoques virtual e materializado.

A seguir, tais enfoques são brevemente discutidos. Além disso, também são apresentados exemplos de abordagens para integração de dados que utilizam tais enfoques. Essas abordagens são classificadas como *read-only* (apenas para leitura), i.e. a visão integrada suporta apenas consulta sobre os dados, não permitindo inserções diretas à ela.

#### 2.4.3.1 Abordagem Virtual

No enfoque virtual, utiliza-se uma *visão integrada* para representar a união das informações contidas nas bases de dados distribuídas. Geralmente, abordagens que utilizam um enfoque virtual precisam fazer um processo de *reescrita de consulta*, necessário para recuperação dos dados nas fontes. Uma *vantagem* do enfoque virtual é que os dados sempre estarão atualizados, uma vez que são obtidos em tempo de consulta. Porém, essa vantagem

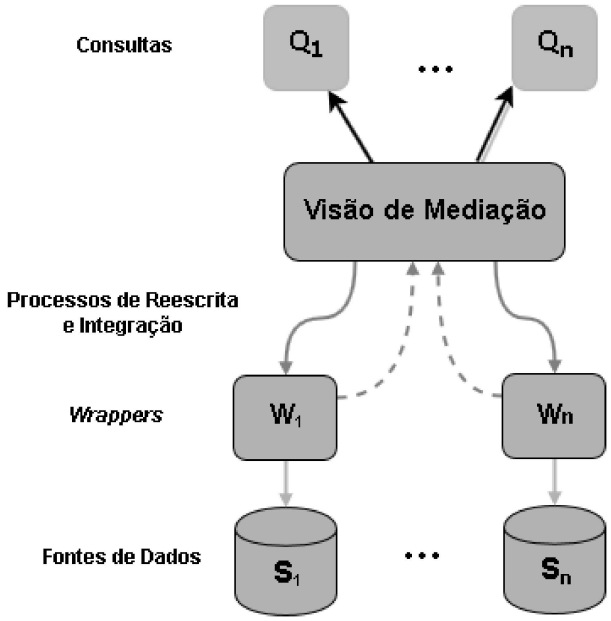



Figura 14 – Arquitetura de um mediador genérico.

trás um custo computacional maior, comprometendo o desempenho. Um exemplo dessa abordagem são os **Mediadores**.

#### 2.4.3.1.1 Mediadores

(WIEDERHOLD, 1992), em 92, apontou a necessidade de haver "um conjunto de módulos de *software* que faça a mediação entre aplicações e bases de dados". Segundo o autor, uma camada de abstração é importante para facilitar a compreensão dos dados, auxiliando na tomada de decisões e análise dos dados. Esse *software*, chamado de **Mediador**, utiliza uma visão integrada, também chamada de *Visão de Mediação*, para conciliar semanticamente diversas fontes de dados. As consultas não são realizadas diretamente nas bases de dados, mas sim na visão de mediação. Ao receber uma consulta sobre a visão, o mediador a decompõe e a reescreve na forma de subconsultas que serão executadas nas diversas fontes de dados. O mediador recebe o resultado dessas consultas, integra, e retorna como o resultado da consulta. Todas essas etapas ocorrem em tempo de execução. Visto que as fontes de dados podem estar em formatos, modelos e tecnologias distintas, normalmente um mediador faz uso de *wrappers* (tradutores). Um *wrapper* é responsável por realizar a tradução de uma fonte de dados para um formato comum conhecido pelo Mediador. Em resumo, as principais características de um mediador são: (i) simplificar; (ii) abstrair; (iii) reescrever e (iv) integrar dados. A Figura 14 representa uma arquitetura genérica de um mediador.

A construção de um Mediador não é um processo trivial. Dentre os principais desafios, podem ser destacados: (i) otimização e reescrita de consultas e (ii)fusão dos dados retornados. No processo de fusão, representações distintas de um mesmo objeto do mundo real


Figura 15 – Manutenção incremental em ambiente materializado.

têm que ser combinadas numa única representação. Além disso, uma ordem de execução das consultas deve ser estabelecida para não gerar resultados inconsistentes ou gargalos de processamento. Por exemplo, considere o mediador $M$ que contém uma visão integrada $V_m$ sobre as fontes de dados $S_1$ e $S_2$. Considere também que é realizada uma consulta $q_1$ sobre $V_m$, tal que $q_1$ = "Retorne os indivíduos de $S_1$ maiores de 20 anos e que também estão em $S_2$". Suponha $q_{m1}$ e $q_{m2}$ como as subconsultas geradas por $M$ para $S_1$ e $S_2$ respectivamente. Note que, se $q_{m2}$ for executada antes de $q_{m1}$, todos os indivíduos de $S_2$ serão retornados, contrapondo o filtro inicial, "indivíduos maiores que 20 anos". Apesar de não gerar um resultado errôneo, essa abordagem gera um custo computacional possivelmente insatisfatório. Basta imaginar um ambiente onde $S_2$ contenha milhões de registros.

#### 2.4.3.2 Abordagem Materializada

Diferentemente do enfoque virtual, nessa abordagem os dados são mantidos fisicamente na visão integrada, i.e. as consultas são realizadas diretamente à visão, chamada de *Visão Materializada* (ZHUGE et al., 1995). Dessa forma, não há necessidade de um processo de reescrita de consulta ou de otimização de consultas, o que resulta num melhor desempenho. Porém, em visões materializadas ainda é um desafio manter os dados atualizados em relação às fontes. Uma forma de tratar a manutenção é atualizar os dados periodicamente, por exemplo, uma vez ao dia. Outra estratégia é detectar as modificações realizadas nas fontes e usá-las para atualizar a visão integrada. Algoritmos que utilizam essa estratégia, conhecida como **manutenção incremental** (GUPTA; MUMICK, 1999), normalmente se baseiam no *Paradigma de Heraclitus* (HULL; JACOBS, 1991). Esse paradigma utiliza *"deltas"*, que representam a diferença entre os estados do bancos de dados, para atualizar uma base materializada. Esses *deltas* são transmitidos das fontes para a visão materializada sempre que o estado dos dados for modificado (ZHUGE et al., 1995; COLBY et al., 1996). A Figura 15, de (ZHUGE et al., 1995), representa um fluxo dos "deltas"num ambiente materializado.

Existem várias ocasiões onde uma abordagem materializada pode ser superior do que uma virtual, e.g. em ambientes distribuídos em que a rede não seja confiável, ou onde seja mais barato computacionalmente realizar uma manutenção incremental do que recomputar a base inteira sempre que uma consulta for requisitada (ZHOU; HULL; KING, 1996). Um exemplo de integração de dados que utiliza visões materializadas é o *data warehousing*.

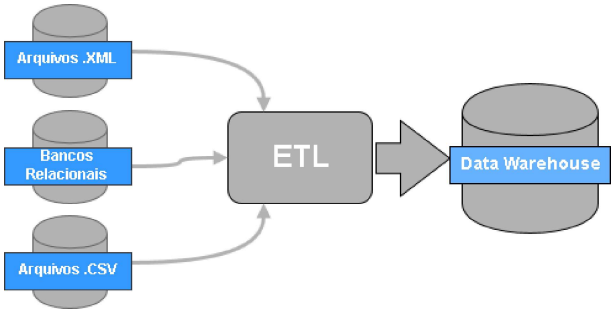



Figura 16 – Arquitetura genérica de um Data Warehouse.

#### 2.4.3.2.1 Data Warehouse

Um **Data Warehouse** (DW) (armazém de dados) faz a coleta de informações de múltiplas fontes de dados, integra e as armazena em um repositório de dados, possibilitando consultas e análises para, por exemplo, suporte a tomada de decisão, OLAP, mineração de dados, dentre outros (INMON; KELLEY, 1993). O processo de *data warehousing*, i.e. construção de um DW, envolve os passos de: (i) extração da informação; (ii) tradução e filtragem dos dados; (iii) fusão com a visão já integrada (WIDOM, 1995). Assim como os Mediadores, *data warehouses* comumente utilizam *wrappers* para realizar a tradução dos dados das fontes. Esse processo é comumente chamado de *Extraction, Transformation & Load* - ETL (KIMBALL; ROSS, 2002). A Figura 16, de (WIDOM, 1995), representa uma arquitetura genérica de DW.

O processo de construção de um DW, assim como os Mediadores, não é uma tarefa trivial. O maior desafio na construção de um DW, assim como descrito na subseção de Visões Materializadas, é a construção de um esquema global para conciliar a heterogeneidade semântica das múltiplas fontes de dados. Entretanto, *data warehouses* são projetados, geralmente, utilizando bancos de dados relacionais, onde os dados são armazenados na forma de tabelas. Essas tabelas contém pouca ou nenhuma semântica sobre as informações, uma vez que a única semântica que tem-se sobre os dados são os nomes das colunas. Assim, como discutido em (AN; BORGIDA; MYLOPOULOS, 2006), muitas vezes é necessária a construção de modelos conceituais e/ou ontologias sobre esses dados relacionais. Por exemplo, a tabela a seguir representa a relação de um indivíduo com suas doenças em um banco de dados relacional.

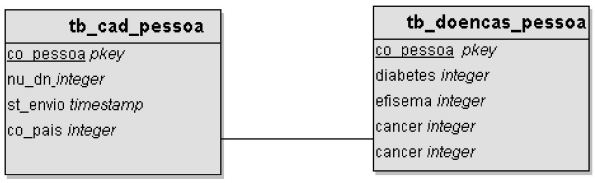



Figura 17 – Exemplo de tabela em relacional.

É importante notar que essa não é a única forma de representarmos essa relação. Uma outra possibilidade seria criarmos uma relação $N : N$ por meio de uma tabela intermediária, *"rl_pessoa_doenca"*. A Figura 18 demonstra essa possibilidade.

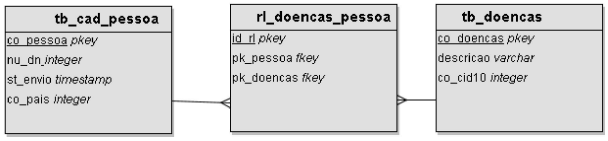



Figura 18 – Outra possibilidade de representação.

É importante também notar que, sem um entendimento do domínio, não é óbvio entender que *"cad"*, por exemplo, é uma abreviação de *"cadastro"*. Embora seja fácil de entender que a coluna *"co_pais"* se trata do *"código de um país"*; a coluna *"st_envio"* não é tão óbvia assim. A semântica em bancos relacionais está limitada ao conteúdo e a nomenclatura, que fica a critério do desenvolvedor, de suas tabelas. Dessa forma, para extrair o significado dessas tabelas são necessárias análises minuciosas sobre os dados e as colunas. Não é difícil imaginar que essa tarefa pode ser inviável para integração de bases em grandes empresas, que chegam a ter milhares de tabelas. Além disso, a construção do esquema global ainda conta com mais um desafio: manter a consistência e integridade (possivelmente) existente nas fontes de dados (ZIEGLER; DITTRICH, 2007).

Outro problema enfrentado na construção de um DW é o de extensibilidade. Um DW tem que ser extensível tanto à novas fontes de dados quanto à modificações nos esquemas das fontes. Entretanto, as etapas para construção de um DW, e.g. construção de visões; transformação dos dados; inclusão de novas informações, são, geralmente, construídas via *scripts* SQL. Assim, sempre que uma informação for adicionada ou modificada, tem-se que alterar ou adicionar códigos *SQL*. Existem ferramentas que auxiliam no processo de desenvolvimento de um DW, e.g. Pentaho *Data Integration* [13]. Tais ferramentas descrevem um processo de *workflow*, abstraindo diversas etapas que teriam de ser codificadas manualmente. Porém, por serem ferramentas de uso geral, comumente o usuário tem que adicionar funcionalidades para se adaptar à sua solução.

A próxima Seção discute sobre uma mudança de paradigma para integração de dados, utilizando tecnologias da Web Semântica.

## 2.5 Web Semântica para Integração de Dados

Nessa Seção, é discutido sobre a mudança de paradigma que a Web Semântica propõe para integração de dados. Também é demonstrado como as tecnologias da Web Semântica podem auxiliar nessa tarefa, bem como seus desafios.

[13] http://www.pentaho.com/product/data-integration

Uma das maiores problemáticas ao se integrar dados em bancos relacionais é o problema de *heterogeneidade semântica*. Essa heterogeneidade, como já mencionado, é o problema de duas ou mais fontes de dados serem capazes de representar um mesmo objeto do mundo real de maneiras distintas. Como também discutido na subseção 2.4.3.2.1, a semântica das tabelas em dados relacionais limita-se à nomenclatura de suas colunas.

Na Web Semântica (Seção 2.2.4), porém, todos os *recursos*, i.e. objetos, predicados e e sujeitos, possuem uma URI[14], que os possibilitam de ser acessados em qualquer local da *Web*. Além disso, cada recurso que contém uma URI faz parte de um vocabulário, modelado por uma ontologia, também acessível por uma URI. Nessa ontologia, temos informações como: qual o domínio e a imagem de determinada propriedade; quais os tipos de atributos de determinada classe; dentre outros. Dessa forma, a Web Semântica consegue prover uma semântica dos dados, inteligível tanto para humanos quanto por máquinas. A Figura 19 demonstra o exemplo da Figura 17 expresso nas tecnologias da Web Semântica.

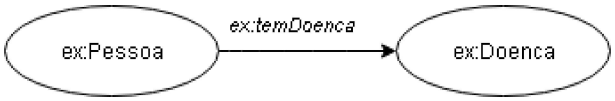



Figura 19 – Relação de um indivíduo e suas doenças

,

Como discutido, cada recurso numa tripla possui uma URI única na *web*, que o detalha semanticamente.

### 2.5.1 Linked Data

**Linked Data** (BIZER; HEATH; Berners-Lee, 2009) é um conjunto de princípios e melhores práticas baseadas em tecnologias da Web Semântica, que usa o RDF (*Resource Description Framework*) (W3C, 2004) para publicação de dados estruturados na *Web*. O *Linked Data* propõe que, assim como existem *hyperlinks* entre páginas HTML, devem ser criados *links* entre as fontes de dados publicadas na *Web*. Esses *links* permitem a descoberta em tempo de execução de novas fontes de dados. Dessa forma, uma das metas do *Linked Data* é promover a evolução de uma *Web* de Documentos, onde os dados são publicados na forma de documentos HTML, entendíveis apenas por humanos; para uma *Web de Dados*, com fontes de informações interligadas, num formato também entendível por máquinas (HEATH; BIZER, 2011). Para isso, *Linked Data* define 4 regras para promover o crescimento da *Web* de Dados:

1. Usar URI para nomear coisas;

[14] Com exceção dos literais. Estes não possuem uma URI

2. Usar o protocolo HTTP para permitir que pessoas possam dereferenciar as URIs criadas;

3. Prover informação útil por meio das URIs dereferenciadas. Utilize padrões, como RDF e SPARQL;

4. Inclua *links* RDF para outras fontes de dados.

O *Linked Data* também representa uma mudança de paradigma nas páginas da *Web* e na descoberta de informações. Essa abordagem utiliza o fato de que a *Web*, apeasr de ser rica em informações, seus formatos estão em formatos inconsistentes, que dificulta ou impossibilita o uso de algoritmos analisar esses dados. Com essa abordagem, a *Web* deixa de ser apenas um ambiente para interação humana, para se tornar um ambiente onde *softwares* podem fazer descobertas de informações em tempo de execução. Atualmente, existe uma iniciativa para publicação de dados abertos na *Web*, chamada *Linked Open Data*, apresentada na subseção a seguir.

### 2.5.2 Linked Open Data

*Linked Open Data*[15] (LOD) é uma iniciativa de publicação de dados abertos, acessíveis e de fácil integração, a fim de promover o enriquecimento de fontes de dados. A LOD é representada por um grafo onde seus nós denotam fontes de dados publicadas em formato aberto. A Figura 20 demonstra o estado atual do grafo.

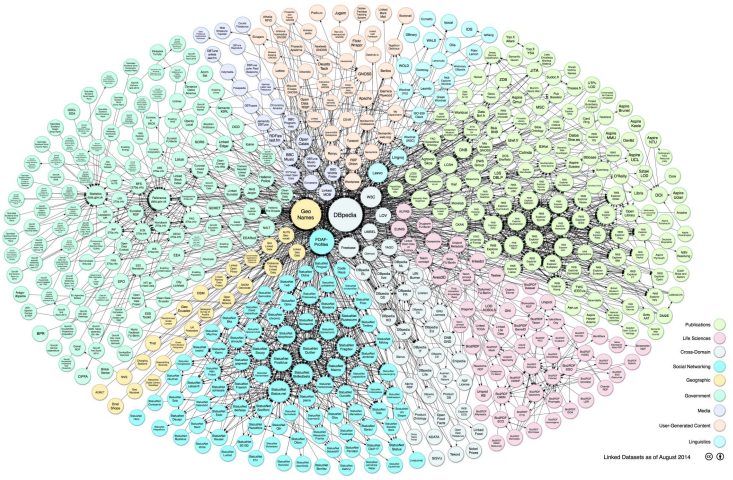



Figura 20 – Nuvem Linked Open Data

Tim Berner's Lee, inventor da *Web* e idealizador do *Linked Data*, propôs em 2010 o plano "5 estrelas *Linked Data*"[16] para publicação de dados. Esse plano tem o objetivo

[15] http://lod-cloud.net/
[16] http://5stardata.info/en/

de encorajar pessoas, governos e grande instituições a publicarem os dados no formato *Linked Data*. Segundo descrito em sua página pessoal (LEE, 2009), as "estrelas"são dadas da seguinte forma:

1. **Uma estrela**: disponibilizar dados na *Web*;
2. **Duas estrelas**: Disponibilizar os dados num formato interpretável por máquinas, e.g. excel em vez de um *scan* de uma imagem;
3. **Três estrelas**: Mesmo que (2), mas prover num formato não proprietário, e.g. CSV;
4. **Quatro estrelas**: Mesmo que (3), porém em formatos padrões da W3C, e.g. RDF e SPARQL, para identficar coisas;
5. **Cinco estrelas**: Além dos itens anteriores, crie *links* para outras fontes de dados, para que pessoas e algoritmos possam descobrir novas informações.

Existem diversos trabalhos que utilizam a LOD para enriquecer as informações de uma aplicação (PAULHEIM, 2013; KOUKOURIKOS; VOUROS; KARKALETSIS, 2012). Em (KOUKOURIKOS; VOUROS; KARKALETSIS, 2012), os autores foram capazes de utilizar informações textuais contidas em fontes na LOD para enriquecer uma ontologia. Já em (GRAY et al., 2012), os autores descreveram uma plataforma *Linked Data* que usa dados da LOD para enriquecer uma base de informações sobre medicamentos. Segundo os autores, esse trabalho foi motivado pela dificuldade em responder perguntas complexas por meio de consultas convencionais, i.e. utilizando dados relacionais. Foram coletadas 83 perguntas, feitas a companhias farmacêuticas, como "me retorne todas informações sobre aspirina"; "para uma dada doença, retorne todos os compostos que podem combatê-la", dentre outras.

### 2.5.3 Linked Data Mashup

Um **mashup** é uma aplicação *Web* (ou não) que utiliza dados integrados para prover um novo serviço. Um *mashup* proporciona a impressão de haver uma *visão integrada* sobre as fontes distribuídas, anteriormente isoladas. Uma **Aplicação de Mashup**, por sua vez, é uma aplicação que consome um *mashup* para determinado fim. Analogamente, *Linked Data Mashups* (LDM) são *mashups* sobre fontes de dados RDF, a fim de combinar e transformar dados de diferentes fontes heterogêneas (HOANG et al., 2014). Quando um *mashup* é construído sobre visões de fontes de dados, é chamado de *Visão de Mashup*. Analogamente, uma *Visão de Linked Data Mashup* é um LDM construído sobre visões das fontes de dados. Nessa dissertação, é utilizado o termo *Visão de Aplicação de Mashup* para denotar uma visão criada sobre um *mashup* que pode ser utilizada para a construção de aplicações.

Em integração de dados, *Linked Data* tem sido alvo constante de pesquisas em diversas áreas. Em (JENTZSCH et al., 2009), um dos primeiros artigos a tratar o problema de integração de dados com *Linked Data*, os autores ligaram diversas fontes de dados farmacêuticos, e.g. DrugBank (WISHART et al., 2008) e Bio2RDF (BELLEAU et al., 2008), para auxiliar gestores de companhias farmacêuticas na tomada de decisão. Existem diversos trabalhos utilizando *Linked Data* para integração de dados nas mais diversas áreas, e.g. Música (DING; SUN; SINGHI, 2010); e-Commerce (MATA; PIMENTEL; ZEPEDA, 2010) e Medicina (KOZáK et al., 2013). Frente a isso, alguns dos fatores que contribuem para o sucesso de Linked Data para integrar dados são:

- **Formato padrão**. A padronização de um formato remove um dos mais comuns de heterogeneidade nos dados;
- **Semântica nos dados**. Diferentemente dos bancos de dados relacionais, fontes em *Linked Data* utilizam RDFS, RDF e OWL para definir semanticamente os dados. Como resultado, o processo de criação de um esquema global para integrar fontes heterogêneas é simplificado;
- **Descoberta de novas fontes**. A tecnologia RDF permite a criação de *links* para outras fontes de dados. Com isso, novas informações podem ser facilmente adicionadas à um *Mashup* já existente.

Uma das contribuições dessa dissertação é a construção de um *Linked Data Mashup* para auxiliar gestores na Saúde Pública.

## 2.6 Conclusão

Este capítulo apresentou uma síntese dos assuntos mais relevantes que servem de fundamentação para o entendimento dos demais capítulos desta dissertação. Foi exposta as principais motivações para a criação da *Web*, bem como sua trajetória até a Web Semântica. Também foram abordadas as tecnologias do *layer-cake*, bolo-de-noiva em tradução livre, que compõem a Web Semântica, como: ontologias, RDF e SPARQL. Além disso, foram discutidas as abordagens e os desafios para integrar dados. Foi mostrado que apesar de ser um tema bem antigo, ainda hoje existem desafios. Por fim, foi mostrado como a Web Semântica, junto com a iniciativa *Linked Data*, trouxeram uma mudança de paradigmas em Integração de Dados.

# 3 Revisão Bibliográfica

## 3.1 Introdução

Esta dissertação propõe um *framework* baseado em mediador semântico para facilitar a criação de *Linked Data Mashups*. Antes do desenvolvimento dessa proposta, necessitou-se realizar um estudo da arte acerca dos principais *frameworks* existentes para criação de *mashups* em Linked Data. Para tanto, na literatura, há um método de estudo que consiste em identificar, avaliar e interpretar vários dos principais estudos acerca de determinado assunto, área ou fenômeno de interesse: Revisão Sistemática (KITCHENHAM, 2004). Uma revisão sistemática propõe uma sumarização das evidências existentes sobre determinado método, tratamento ou tecnologia, provendo uma base consistente de estudo e auxiliando novas possibilidades de pesquisa.

Esse capítulo descreve uma revisão sistemática sobre as principais ferramentas para integração de dados com fontes *Linked Data* que sejam similares à abordagem proposta desta dissertação.

## 3.2 Revisão Sistemática

Uma Revisão sistemática é um estudo secundário com objetivo de identificar, avaliar e analisar diversos estudos primários, i.e. estudo que gera resultado, acerca de determinado tópico, área de estudo ou fenômeno seguindo uma metodologia rigorosa(MALLETT et al., 2012). Como destacado em (KITCHENHAM; CHARTERS, 2007), existem diversas motivações para se conduzir uma revisão sistemática, como:

- Promover a ênfase na importância de evidências empíricas sobre conhecimento prévio;
- Identificar lacunas de conhecimento no tema da pesquisa abordada, destacando deficiências e, portanto, sugerir áreas para investigações futuras;
- Uma revisão sistemática propõe disponibilizar um vasto estudo da arte sobre determinado assunto. Portanto, pode auxiliar e impulsionar estudos sobre determinada área.

Um elemento crítico ao realizar uma revisão sistemática é o desenvolvimento de um protocolo. O protocolo especifica todos os passos realizados durante a revisão e é o elemento que que determina o grau de confiabilidade do estudo. No protocolo são especificados, por exemplo, as seguintes metodologias: (i) como encontrar estudos relevantes? (ii) Quais

estudos estão ou não relacionados com a temática? (iii) Quais estudos devem ser analisados? Dentre outros.

O protocolo é iniciado definindo-se as *Questões de Pesquisa* (QP). Em seguida, a *estratégia para busca manual e automática* dos artigos primários é definida, de modo que a maior quantidade de artigos primários seja encontrada. Após encontrar uma gama de estudos, é apresentado um método para *seleção dos estudos* encontrados, levando-se em consideração critérios para inclusão e exclusão de tais estudos. Também são definidos critérios para avaliar a *qualidade* dos estudos, levando em consideração suas procedências. Finalmente, é apresentada uma estratégia para selecionar o que deve ser extraído de cada estudo, a fim de realizar uma discussão sobre eles.

### 3.2.1 Questão de Pesquisa

Para a definição das questões de pesquisa abordadas nesta revisão sistemática, foi utilizado o método *Goal-Question-Metric* (GQM) (BASILI; CALDIERA; ROMBACH, 1994). Este método define que para encontrar questões de pesquisas, devem ser especificados *Purpose, Issue, Object* e *Viewpoint* (PIOV); propósito, assunto, objecto e ponto de vista, respectivamente:

*Purpose*: analisar as principais características; *Issue*: integração de dados em *Linked Data*; *Object*: ferramentas de *Linked Data Mashups*. *Viewpoint*: do ponto de vista do usuário, do pesquisador e do sistema.

Assim, foram definidas as seguintes questões de pesquisa:

- **QP1**: Como é realizado o processo de integração de dados?
- **QP2**: A construção de um *Linked Data Mashup* pode auxiliar na construção de um outro *mashup* sobre as mesmas fontes?
- **QP3**: Para construir um *mashup*, são necessários conhecimetnos específicos em Web Semântica?
- **QP4**: Os autores ainda mantém este *framework*?
- **QP5**: Quais são as principais ferramentas para construção de *Linked Data Mashups?*

### 3.2.2 Estratégia de Busca

Nesta etapa, é definida qual metodologia utilizada para encontrar os artigos discutidos nesta revisão sistemática. Primeiramente, foi realizada uma etapa de busca preliminar com as *strings "Linked Data Mashups"* e *"Linked Data Integration"* nas fontes definidas na subseção 3.2.2.2, com o objetivo de conhecer os principais artigos primários e secundários sobre o tema. Nesse processo, foram encontrados os estudos (TRAN et al., 2014; HOANG

et al., 2014; SCHULTZ et al., 2011; VIDAL et al., 2015; LOPES; VIDAL; OLIVEIRA, 2016)

Após, sobre cada artigo encontrado foi realizada uma busca em suas referências e palavras-chave, a fim de conhecer os principais termos.

Também foram realizadas *buscas automáticas e manuais*. Existem diversas bases de dados na *web* que dão suporte à buscas automáticas, em que o usuário utiliza termos para encontrar os artigos. Porém, nem todo artigo publicado, seja em periódico ou conferência, está em uma base de dados com suporte à buscas automáticas. Desse modo, também foram realizadas buscas manuais nas bases de conferências relevantes. Nesse tipo de pesquisa, o usuário, sem o uso de termos, procura pelos artigos em uma base de dados. Em resumo, a estratégia utilizada foi construída da seguinte forma:

1. Buscar primeiramente por estudos secundários e artigos relevantes sobre a temática (e.g. (HOANG et al., 2014));
2. Contatar pesquisadores experientes na área a fim de conhecer os principais periódicos e as conferências;
3. Conduzir estudos preliminares com o intuito de conhecer os principais termos da problemática;
4. Definir com os demais pesquisadores os termos de pesquisa que serão utilizados na busca automática;
5. Acordar com os demais pesquisadores as principais bases de dados digitais para a busca automática;
6. Realizar buscas experimentais em tais bases com a *string* de busca definida;
7. Além das buscas automáticas, realizar pesquisas manuais em conferências e periódicos em que não haja um sistema de busca;
8. Com os estudos encontrados, realizar buscas sobre suas referências e palavras-chave, a fim de descobrir algum novo termo a ser incluído na pesquisa;

#### 3.2.2.1 Termos de pesquisa

Para definição dos termos, foi definida uma estratégia cíclica, descrita a seguir. Primeiro, foi definida uma *string de busca* inicial para a realização de *buscas preliminares*. Para tanto, a Questão de Pesquisa (subseção 3.2.1) foi decomposta em termos principais. Esses termos foram concatenados utilizando o operador boleano *AND*. Além disso, cada termo principal foi concatenado com seu(s) sinônimo(s) usando o operador boleano *OR*. Com a *string* inicial gerada, foram realizadas buscas preliminares nas bases de dados já definidas

Tabela 1 – Decomposição das questões de pesquisa

| **Termo principal** | **Sinônimos** |
|---|---|
| Ferramentas | Tool, framework, prototype |
| Construção | Construction, development, creation, generation, building |
| Linked Data | Semantic Web, Semantic |
| Mashup | integration, mashup, mashup, meshup, mesh-up |

com suporte à busca automática. O objetivo das buscas preliminares é, utilizando as palavras-chave dos artigos encontrados, descobrir termos que não fazem parte da *string inicial* para, então, adicioná-los à *string*. Esse processo foi realizado até que não fossem descobertos novos termos. A Figura 21 resume esse processo.

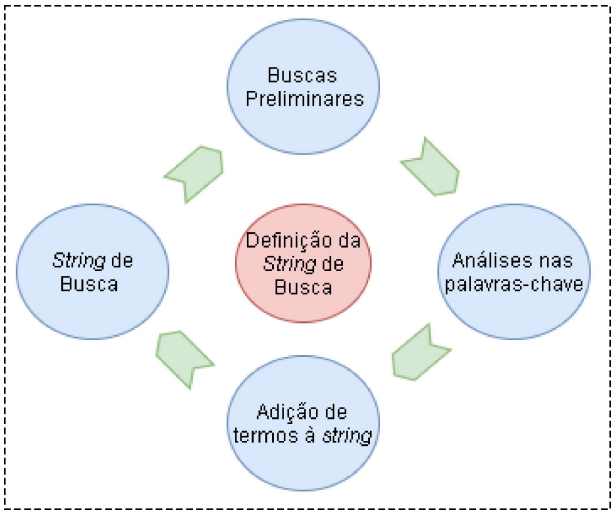



Figura 21 – Processo para definição dos Termos de Pesquisa

A decomposição das questões de pesquisa é resumida na Tabela 1, mostrada a seguir.

Assim, ao final do processo de concatenação, foram encontrados a *string* de busca mostrada na Tabela 2 :

Tabela 2 – Definição da *string* de busca

| |
|---|
| (ferramenta OR tool OR framework OR prototype) AND (construcao OR construction OR development OR creation OR generation) AND (linked data OR semantic web OR semantic) AND (mashup OR data integration OR mashup OR mash OR mesh) |

#### 3.2.2.2 Escopo de busca e bases digitais

Para a construção deste estudo, foram incluídos apenas os artigos do ano de 2008 até Novembro de 2016. Essa data foi escolhida porque um dos trabalhos inspiradores para construção de mashups em Linked Data foi o Yahoo! Pipes, de 2007.

Para definir as bases de dados com busca automática a serem utilizadas, foi realizada uma pesquisa por revisões sistemáticas relevantes na área da computação com a string *"Systematic Review"* sobre a base *"Google Scholar"*. As revisões sistemáticas (WEGELER et al., 2013; KITCHENHAM; MENDES; TRAVASSOS, 2006; MAHDAVI-HEZAVEHI; GALSTER; AVGERIOU, 2013; JULA; SUNDARARAJAN; OTHMAN, 2014) foram selecionadas por conta de suas grandes quantidades de estudos envolvidos e citações. Para encontrar as bases de dados para busca manual, foi um site[1], mantido pela Universidade Federal do Mato Grosso - Brasil(UFMT), que lista todas as conferências e os periódicos da área da computação. Assim, foi utilizado o *Qualis* como critério de inclusão de uma conferência ou um periódico. Além disso, também foi consultado o site *A Wiki for Call for Papers* (WikiCFP)[2] para descobrir possíveis conferências ou periódicos relevantes que não estejam listados no *Qualis*. As bases encontradas foram:

- Bases de dados
    1. IEEExplore[3]
    2. ACM Digital Library[4]
    3. Science Direct[5]
- Periódicos e conferências
    1. ISWC [6]

[1] http://qualis.ic.ufmt.br/
[2] http://www.wikicfp.com/cfp/
[3] http://ieeexplore.ieee.org
[4] http://dl.acm.org/
[5] http://www.sciencedirect.com/
[6] http://swsa.semanticweb.org/content/international-semantic-web-conference-iswc

2. ESWC [7]
3. Semantic Web Journal (SWJ)[8]
4. WWW [9]

Várias das fontes encontradas nos artigos de parâmetro não foram utilizadas por conta da sua pouca associação com estudos em *Linked Data*, como *Web of Science*. A *Springer Link*[10] não foi incluída por conta da inviabilidade de obter seus artigos.

#### 3.2.2.3 Processo de Busca

Para realizar a busca automática sobre as bases definidas, a *string* de busca (subseção 3.2.2.1) teve que ser transformada para um formato particular de cada uma das bases. Embora o termo *"ferramenta"* tenha sido considerado, primeiramente, como uma palavra-chave, ela foi retirada da *string* final de busca. A concatenação desse termo e seus sinônimos ao restante da *string* chegou a diminuir a quantidade de resultados retornados em até 600%, e.g. ACM. Assim, a Tabela 3 mostra as *strings finais* desenvolvidas para sua respectiva base e a quantidade de artigos retornados por cada.

Tabela 3 – *Strings* para busca automática nas bases

| **Base de Dados** | **String de busca** | **Resultado** |
|---|---|---|
| IEEExplore | AND ("Document Title":construction OR "Document Title":development OR "Document Title":creation) AND ("Document Title":"linked data"OR "Document Title":semantic) AND ("Document Title": mashup OR "Document Title":mesh OR "Document Title": integration) | 8 |
| ACM | acmdlTitle:(construction development creation generation) AND acmdlTitle:(linked data semantic) AND acmdlTitle:(integration mashup mesh meshup) | 68 |
| Science Direct | TITLE(construction OR creation OR creation OR development OR generation) AND TITLE(linked data OR semantic) AND TITLE(mashup OR mash OR mesh OR integration) | 0 |

[7] http://eswc-conferences.org/
[8] http://www.semantic-web-journal.net/
[9] http://www.www2017.com.au/
[10] http://link.springer.com/

A princípio, as bases ISWC, ESWC e WWW foram classificadas como bases para pesquisa manual. Porém, ambas conferências foram transformadas em grafos RDF e dispõem de um SPARQL Endpoint[11]. Assim, foi criada uma consulta SPARQL que abrangesse os resultados que a *string* de busca retornaria. O código SPARQL utilizado é apresentado a seguir.

```
PREFIX person: <https://w3id.org/scholarlydata/person/>
PREFIX conf: <https://w3id.org/scholarlydata/ontology/conference-ontology.owl#>
SELECT DISTINCT ?paper ?title

WHERE{
   {  ?paper a conf:InProceedings;
       conf:title ?title .

       FILTER regex(?title, "semantic","i") .
       FILTER regex(?title, "mash","i") .

   } UNION {
      ?paper a conf:InProceedings;
       conf:title ?title .

       FILTER regex(?title, "semantic","i") .
       FILTER regex(?title, "integration","i") .
   } UNION {
      ?paper a conf:InProceedings;
       conf:title ?title .

       FILTER regex(?title, "linked data","i") .
       FILTER regex(?title, "mash","i") .

   } UNION {
      ?paper a conf:InProceedings;
       conf:title ?title .

       FILTER regex(?title, "linked data","i") .
       FILTER regex(?title, "integration","i") .
   }
}
```

Os resultados das buscas manuais sobre as conferências e periódicos definidos são

[11] http://www.scholarlydata.org/

resumidos na Tabela 4.

Tabela 4 – Resultado das buscas manuais

| **Base de Dados** | **Resultado** |
|---|---|
| SWJ | 3 |
| ISWC | 23 |
| ESWC | 10 |
| WWW | 4 |

### 3.2.3 Estratégia para Seleção

Nesta etapa, é discutida uma estratégia para seleção e exclusão dos artigos primários retornados pela etapa de busca (subseção 3.2.2). Os critérios que foram julgados relevantes para a realização desta revisão sistemática foram:

**Inclusão:**

- **I1**: O artigo descreve ou cita uma ferramenta ou abordagem que auxilia na integração de dados em *Linked Data*.

Notou-se que, devido a baixa quantidade de artigos retornados, os critérios de inclusão não deveriam ser tão rigorosos. Por isso, para a escolha dos critérios de inclusão, foi decidido apenas incluir todos os artigos que descrevessem uma ferramenta que auxilie a construção de *Linked Data Mashups*. **Exclusão:**

- **E1**: O artigo está datado antes de 2008;
- **E2**: A abordagem para integração descrita no artigo não leva em conta problemáticas comuns em integração de dados, como: definição dos mapeamentos, criação de *links owl:sameAs* e fusão de dados;
- **E3**: O artigo não trata sobre integração de dados em *Linked Data*;
- **E4**: Não descreve o processo de integração de dados.

Como discutido na introdução, esta revisão sistemática tem como objetivo descrever abordagens similares ao *framework* proposto nesta dissertação. Por isso, foram excluídos os artigos (e.g. (LE-PHUOC et al., 2009)) que tratam a integração como uma construção unificada de triplas, sem abordar problemáticas como: mapeamentos e fusão de dados. Para definir se um artigo deviam ou não ser incluído na síntese dos resultados (Sec. 3.3), os critérios foram aplicados mediante leitura do título e resumo do artigo. Se após a leitura desses componentes ainda não fosse possível aplicar os critérios, as Seções de *Introdução* e *Conclusão* foram lidas. Finalmente, se nenhum dos componentes foi suficiente, o restante do artigo era lido. A Figura 22 resume o processo de busca e seleção dos artigos primários.

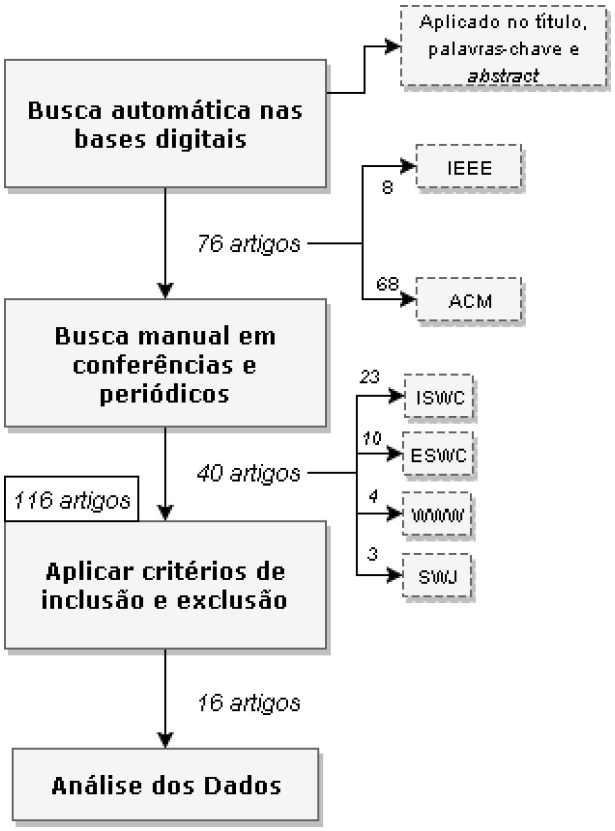



Figura 22 – Resumo do processo de busca e seleção

Durante a seleção dos artigos, foram encontrados diversos trabalhos (e.g. (TOUMA; ROMERO; JOVANOVIC, 2015)) sobre *Ontology Merging and Alignment* (NOY; MUSEN, 2000). Aqui vale frisar que este assunto está fora do escopo tanto desta dissertação quanto desta revisão sistemática. Dois dos artigos selecionados ((LOPES; VIDAL; OLIVEIRA, 2016; VIDAL et al., 2015)) são estudos que dão origem a esta dissertação, portanto foram descartados.

Para realização desse processo, foi utilizada a ferramenta StArt[12] (*State of the Art through systematic review*) desenvolvida pelo laboratório LaPES (*Laboratory of Research on Software Engineering (LaPES)*). Com essa ferramenta, foi possível gerenciar os diversos artigos encontrados e aplicá-los os critérios de inclusão e exclusão. A seguir, os artigos selecionados são discutidos.

## 3.3 Discussão dos Resultados

Ao fim do processo de busca e seleção dos artigos, há o processo de extração e discussão dos resultados. Aqui as informações são extraídas dos artigos a fim de responder as perguntas de pesquisa definidas na subseção 3.2.1. A seguir é mostrado uma visão geral

[12] http://lapes.dc.ufscar.br/

sobre os artigos encontrados. Então, cada pergunta de pesquisa é respondida mediante análise dos estudos.

### 3.3.1 Visão Geral

O processo de busca mostrou a grande variedade de aplicações em *Linked Data* na área de *mashups*. Foram encontrados diversos estudos utilizando fontes *Linked Data* para agregar valor em empresas, sistemas de apoio à tomada de decisão e em aplicativos. Foi possível notar o crescimento de aplicações *mobile* que utilizam tecnologias da web semântica como diferencial para o usuário. Também notou-se que estudos mais atuais estão mais focados em aplicações do que em propor novas abordagens. A Figura 23 relaciona a quantidade de artigos aceitos com o ano de publicação.

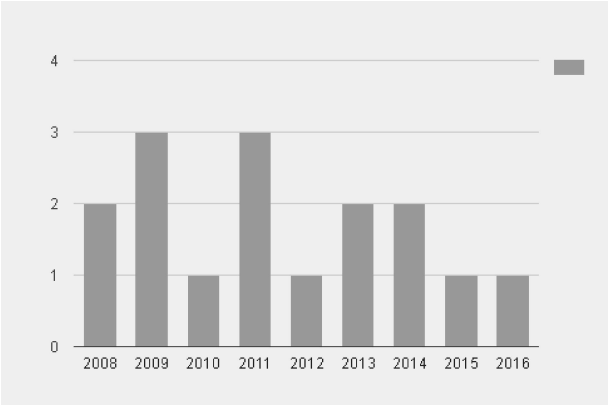



Figura 23 – Artigos para análise x ano de publicação

Além disso, também foi possível verificar que, para esta revisão sistemática, as buscas manuais foram bem mais eficientes do que as realizadas de forma automática. A Figura 24 retrata esse cenário.

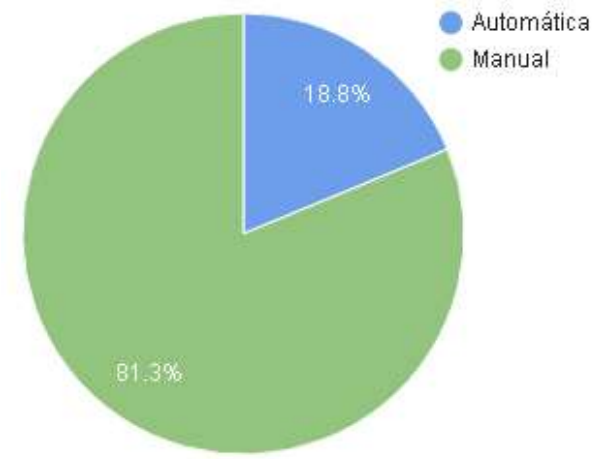



Figura 24 – Porcentagem de inclusão de artigos x base de dados

### 3.3.2 *QP1: Como é realizado o processo de integração de dados?*

Parte dos estudos selecionados utilizaram uma abordagem empírica, i.e. sem auxílio de um *framework*, para integração de dados. Em (FOX et al., 2007) os autores utilizaram uma abordagem própria para registro de dado em 3 etapas: (i) Registro de *metadata*; (ii) registro de *schema* e (iii) registro de item de dado. Na abordagem utilizada, porém, não são definidos *links* entre as fontes heterogêneas. Talvez por conta da época em que foi publicado (2008), quando *engines* de descoberta de *links owl:sameAs* como o SILK (BIZER et al., 2009).

Em (PENG et al., 2010), os autores apresentam uma arquitetura para ser usada em um aplicativo de celular. Essa arquitetura descreve o processo de mapeamento e de inclusão de novas fontes. Porém, também não é especificado como ocorre a criação dos *links*.

O *Linked Data Integration Framework* (LDIF)(SCHULTZ et al., 2011) descreve um processo de integração de dados em 4 etapas: (i) acesso aos dados; (ii) definição dos mapeamentos; (iii) criação dos *links owl:sameAs* e (iv) fusão dos dados. Nesse processo são utilizadas ferramentas específicas para cada etapa. A Figura resume o funcionamento do LDIF.

Em (MéDINI et al., 2014) os autores propõem o *SRM Framework*. Este *framework* representa a união de outros duas ferramentas também proposta pelos autores: RDF-REST, uma Web API para serviços em RDF (CHAMPIN, 2013) e Dataconf, um *Linked Data Mashup* sobre dados de conferências. O *framework* proposto, *SRM Framework*, tem o objetivo de resgatar grafos RDF de fontes distintas, aplicar regras de *matching* sobre eles e representá-los como um grafo unificado. Para tanto, o serviço é dividido em etapas, responsáveis por identificar o recurso requisitado, otimizar a rota e realizar o *mashup*. Porém, talvez por ser um artigo-curto, não são dados detalhes de como o *framework* constrói o *mashup* das informações.

Em (HARTH et al., 2013) é proposto um *framework*, *Karma*, para integração *on-the-fly* de fontes de dados heterogêneas mediante uma *interface web*. Neste *framework* são levados em consideração: Ontologia de domínio, fontes de dados; fontes de dados como fontes *linked data*; modelos próprios do Karma baseados na ontologia de domínio e um programa para integração dos dados, baseado em regras e consultas. Os modelos Karma são criados pela combinação das fontes de dados com a ontologia de domínio. Nessa abordagem, um programa de integração de dados pode ser definido para ocorrer periodicamente, de segundo até meses. Para adicionar uma nova fonte de dados, o usuário deve modelá-la no formato do Karma e especificar em um módulo, Data-Fu, as regras para obtenção dos dados. Quando uma nova fonte é adicionada, esta deve ser mapeada, mediante a *interface*, para o vocabulário das bases já integradas. O módulo Data-Fu é responsável por interpretar a consulta, decompô-la e construir as triplas para retornar ao usuário.

A abordagem descrita em (GRACIA; D'AQUIN; MENA, 2009) ataca os problemas de redundância e escalabilidade utilizando o sistema Watson (D'AQUIN et al., 2007) para identificar objetos similares. O sistema Watson serve como um *gateway* da web semântica: é responsável por identificar os itens e disponibilizá-los por um único ponto de saída. A abordagem proposta no artigo aplica algoritmos de clusterização baseados em ontologias para classificar os itens oriundos do Watson. Para a integração dos itens, é utilizada uma abordagem que identifica se dois objetos referem-se à um mesmo conceito. É criada então uma nova ontologia em que os dois objetos serão mapeados à ela. Diferentemente da abordagem proposta nessa dissertação, (GRACIA; D'AQUIN; MENA, 2009) propõe a integração de termos na *web semântica*;

No trabalho descrito em (LE-PHUOC et al., 2009), é proposto um sistema, *Semantic Web Pipes* (SWP) que utiliza o conceito de *Semantic Pipe*. Inspirados pelo Yahoo! Pipes (PRUETT, 2007), um *semantic pipe* recebe um dado, texto, XML ou RDF, e processa de acordo com o *pipe* utilizado. Com auxílio dos demais componentes e operadores da ferramenta, as triplas são construídas utilizando o CONSTRUCT do SPARQL.

Em (GREEN et al., 2008), é apresentada uma arquitetura para integração de dados utilizando ontologias de aplicação, que reflete as necessidades da aplicação, de domínio e das fontes. Para agregar uma nova fonte, deve ser construída uma ontologia correspondente e então mapeada, utilizando D2RQ (BIZER, 2004). Porém, o artigo não descreve como é feito nem o processo de fusão dos dados nem de criação dos *links*.

Em (BORAN et al., 2011), os autores propõem um *framework* que conta com 3 abordagens para integrar dados: por consulta, por regras ou axiomas. Na primeira, os dados integrados são construídos via SPARQL CONSTRUCTs. Na segunda, são utilizadas regras SWRL (*Semantic Web Rule Language*) (HORROCKS et al., 2004) para combinar os dados em OWL. Finalmente, na última abordagem são utilizados axiomas para integração dos dados. Porém, o artigo não trata problemáticas como a criação de *links* entre as fontes heterogêneas nem a problemática de fusão dos dados. O artigo também não dá detalhes de como as 3 abordagens realizam a integração dos dados.

(LANGEGGER; WöSS; BLöCHL, 2008) propõe um mediador para consultas em fontes virtuais de grafos RDF na web semantica. A heterogeneidade dos dados é tratada com *wrappers* D2RQ, que ficam acoplados às fontes RDF heterogêneas. Neste *framework*, SemWIQ (*Semantic Web Integrator and Query Engine*), o mediador utiliza os *wrappers* D2RQ na tradução das fontes RDF para um formato comum do mediador. Essa tradução é realizada *on-the-fly*, i.e. os grafos RDF são virtuais e só são aplicadas as regras de mapeamento e a materialização quando requisitado, mediante uma consulta, por um cliente.

No trabalho apresentado em (KÄMPGEN et al., 2014), os autores desenvolveram o FIOS (*Financial Information Observation System*): um sistema que utiliza dados integrados

de fontes *Linked Data*. Embora esse trabalho não proponha um novo *framework* para guiar a construção de *mashups*, ele detalha o processo de integração de dados realizado durante o trabalho. Resumidamente, a integração de dados foi feita seguindo os passos: (i) Identificação e aquisição dos dados distribuídos; (ii) modelagem de uma ontologia; (iii) criação de *links owl:sameAs* e *skos:narrower*; (iv) etapa de consolidação, onde todas entidades com *links* serão combinadas.

### 3.3.3 *QP2: A construção de um Linked Data Mashup pode auxiliar na construção de um outro mashup sobre as mesmas fontes?*

(FOX et al., 2007), neste estudo, para a construção de uma nova integração, o processo de registro (metadata, schema e dado) deve ser realizado novamente. (SCHULTZ et al., 2011), para a construção de um mashup, cada etapa deve ser especificada. Caso um novo mashup seja construído, novas especificações para cada etapa devem ser criadas., (MéDINI et al., 2014) é uma abordagem virtual, a medida que novas fontes sãoa gregadas, devem ser geradas novas regras. (HARTH et al., 2013), é utilizada uma abordagem virtual: os dados só são materializados quando requisitados pelo usuário. Caso um *mashup* já tenha sido especificado (modelos e regras criadas), o *mashup* é realizado em tempo de execução. Em (GRACIA; D'AQUIN; MENA, 2009), sempre que um novo termo for adicionado à integração, um novo processo de clusterização deve ser realizado. Já em (LE-PHUOC et al., 2009), são utilizados *semantic pipes* para construir triplas unificando dois grafos virtuais heterogêneos. Se o *mashup* das fontes RDF já foi especificado, i.e. regras e mapeamentos criados, um novo *mashup* pode ser construído em tempo de execução. No mediador para consulta sobre fontes de grafos virtuais RDF, (LANGEGGER; WöSS; BLöCHL, 2008), se os mapeamentos dos *wrappers* D2RQ já foram definidos, um *mashup* é construído em tempo de execução. Nos demais estudos, não ficou clara a resposta de *QP2*. Em (KÄMPGEN et al., 2014), caso um novo *mashup* necessite ser criado sobre as mesmas fontes, novas especificações devem ser definidas para cada uma das 4 etapas (identificação e aquisição, modelagem, links e consolidação).

### 3.3.4 *QP3: Para construir um mashup, são necessários conhecimentos específicos em Web Semântica?*

Identificou-se que em todas as abordagens são necessários conhecimentos específicos em pelo menos uma tecnologia da web semântica. O Karma, (HARTH et al., 2013), porém, conta com uma interface gráfica bastante intuitiva, mas que ainda sim requer que o usuário tenha conhecimentos em algumas das etapas de integração de dados, como em mapeamentos. Diferentemente da abordagem proposta nesta dissertação, onde novos *mashups* podem ser construídos sem conhecimentos específicos.

### 3.3.5 *QP4: Os autores ainda mantém este framework?*

Das abordagens listadas, apenas Karma (HARTH et al., 2013) e LDIF(SCHULTZ et al., 2011) ainda são utilizados pela comunidade.

### 3.3.6 *QP5: Quais são as principais ferramentas para construção de Linked Data Mashups?*

Segundo esta revisão sistemática, as ferramentas mais relevantes encontradas foram Karma (HARTH et al., 2013) e LDIF(SCHULTZ et al., 2011)

## 3.4 Conclusão

Neste capítulo foi abordado um estudo de revisão sistemática sobre as principais *ferramentas* ou abordagens para construção de *Linked Data Mashups*. O objetivo deste capítulo é identificar as similaridades, vantagens e desvantanges das abordagens atuais para a proposta nesta dissertação. Foi possível identificar que em diversos estudos, o processo de *mashup* é realizado de forma empírica, sem um *framework* para guiar o processo de integração. Com isso, a criação de novos *mashups*, mesmo sobre fontes já integradas, pode requerer certo re-trabalho. Além disso, não foi possível identificar abordagens que utilizem o conceito de depositar e consultar especificações de *mashup*, método proposto nesta dissertação.

# 4 Especificação de Linked Data Mashup

## 4.1 Introdução

Uma das problemáticas ao integrar dados é a heterogeneidade semântica contida nas fontes. Como discutido em 2, esse é o problema de quando um mesmo objeto do mundo real é representado de formas distintas em fontes de dados. Isso acontece porque, geralmente, os provedores de dados, e.g. instituições e empresas, desenvolvem seus próprios bancos de dados e *softwares* (HAMMER; MCLEOD, 1993). Sendo assim, é comum que cada provedor represente um mesmo objeto do mundo real, e.g. um indivíduo, em um formato próprio, ocasionando na heterogeneidade semântica. Uma das dificuldades em conciliar semanticamente duas fontes de dados é a falta de semântica nas informações, o que, comumente, torna inviável o processo de entendimento das bases e construção de um novo modelo.

A Web Semântica trouxe um novo paradigma na forma com que os dados são visualizados: deixam de ser representados por tabelas, com pouca ou nenhuma semântica sobre as informações; para serem representados por recursos associados a uma URI, única na *web*. Apesar da troca de paradigma, a mudança das tecnologias trouxeram diversos desafios para integração de dados. Segundo (VIDAL et al., 2015) criar uma visão homogeneizada sobre fontes de dados em *Linked Data*, ou *visão de Linked Data Mashup*, é uma tarefa complexa que envolve 4 desafios principais: (i) seleção das fontes *linked data* relevantes para a aplicação; (ii) extração e tradução de fontes de dados distintas para um vocabulário comum; (iii) identificação de *links* que denotam a similaridade entre instâncias de fontes distintas e, finalmente, (iv) combinação e fusão de múltiplas representações de um mesmo objeto do mundo real numa única representação.

Dessa forma, esse Capítulo descreve o *framework* conceitual apresentado em (VIDAL et al., 2015) para especificação de visões de *Linked Data Mashup* - visões LDM. Nessa abordagem, uma visão LDM é especificada com o auxílio de *Visões Exportadas, Visões de Links Semânticos*, *Regras de Fusão e de Normalização*. Para demonstrar a aplicabilidade do *framework*, foi construído um estudo de caso que integra dados na Saúde Pública.

## 4.2 Especificação de Mashup

### 4.2.1 Visão Geral

A abordagem de (VIDAL et al., 2015) descreve um *framework* de 3 camadas baseado em ontologias, resumido na figura 25, para especificação de *mashups* em *Linked Data*. Uma

especificação de *mashup* descreve formalmente como construir um *mashup* sobre duas fontes heterogêneas. Esse *mashup* pode ser materializado e então utilizado na criação de aplicações de *mashup* ou então, pode ser utilizado para auxiliar no desenvolvimento de *data warehouses*.

### 4.2.2 Arquitetura 3 - Camadas

Na **Camada de Integração Semântica**, a *Ontologia de Domínio* $O_M$ representa um modelo conceitual, responsável por conciliar a heterogeneidade semântica possivelmente presente nas fontes de dados $S_1, ..., S_n$. Para isso, $O_M$ define o vocabulário a ser utilizado pelas fontes de dados.

Na **Camada de Dados**, cada fonte $S_i$ é descrita por uma ontologia $O_{Si}$ e exporta uma ou mais visões, chamadas de *Visões Exportadas*. Na camada de **Visões Exportadas**, cada visão $E_i$ é composta por uma *Ontologia Exportada* $O_{Ei}$, cujo vocabulário é um subconjunto de $O_M$, e um conjunto de mapeamentos $M_{Ei}$, que mapeia os conceitos de $O_{Si}$ em $O_{Ei}$. Também é na Camada de Visões Exportadas que são definidas as regras para descobertas de **Links Semânticos**, $EL_1, ..., EL_m$. Os *Links Semânticos* definem a similaridade entre duas representações distintas de um mesmo objeto do mundo real.

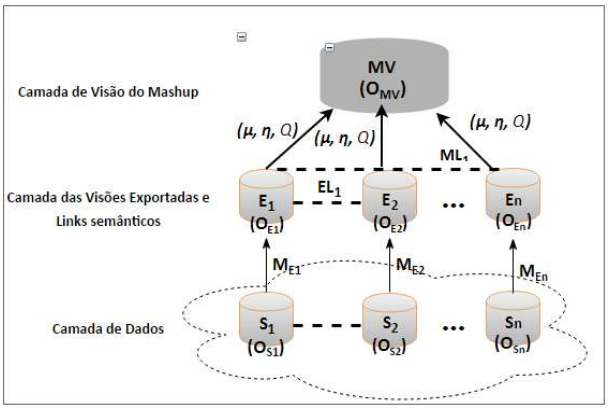



Figura 25 – Framework 3 Camadas

A especificação de uma integração semântica é definida por uma tupla $\lambda_m = \{M, O_M, E_M, EL_M, \mu_m$ onde:

- M é o *nome* da visão integrada;
- $O_M$ é a *ontologia de domínio*;
- $E_M$ representa o conjunto das *visões exportadas*;
- $EL_M$ define o conjunto de regras para descoberta dos *links semânticos*;
- $\mu_m$ determina as *regras de fusão*;
- $Q$ determina os critérios para avaliação de qualidade das fontes de dados.

As etapas para criação de $\lambda_m$ são detalhadas nas subseções a seguir.

### 4.2.3 Especificação das Visões Exportadas

Nesse *framework*, cada fonte de dados $S_i$ *exporta* uma visão no formato da ontologia de domínio $O_M$. A especificação de uma *Visão Exportada*, $E_i \in E_M$, é uma quíntula, $\{E, S, O_S, O_E, M_E\}$, onde :

- $E$ é o nome da visão;
- $S$ é a fonte de dados;
- $O_S$ é a ontologia descrevendo a fonte de dados $S$;
- $O_E$ representa a ontologia exportada de $S$ e descreve a fonte de dados $S$ no vocabulário de $O_M$;
- $M_E$ é um conjunto de mapeamentos entre $O_{Si}$ e $O_E$.

As fontes de dados são escolhidas de acordo com a relevância para a aplicação. Por exemplo, se a aplicação requer dados geográficos, GeoNames[1] pode ser uma escolha. Além disso, deve haver uma ontologia $O_{Si}$ descrevendo cada fonte $S_i$.

### 4.2.4 Especificação dos Links Semânticos

O *framework* apresentado nesse Capítulo contém dois tipos de links semânticos: (1) *Links* já existentes nas fontes de dados e (2) *Links* de um *mashup*. Aqui será discutido apenas *links* do segundo tipo.

Um *Link Semântico* determina que duas instâncias em fontes de dados distintas referem-se a um mesmo objeto do mundo real. A especificação de um *link* semântico é definida pela tupla $\{ML, T, U, C, p_1, ..., p_n, \mu\}$, onde:

- ML é o nome da visão;
- $(T, S_T, O_{ST}, O_T, M_T)$ e $(U, S_U, O_{SU}, O_U, M_U)$ são visões exportadas;
- $C$ é uma classe presente em ambos vocabulários das ontologias exportadas $O_T$ e $O_U$;
- $p_1, ..., p_n$ são propriedades da classe C, também presente nas ontologias exportadas;
- $\mu$ é chamado de *regra de correspondência*, responsável por definir a similaridade entre os objetos de $T$ e $U$.

### 4.2.5 Especificação das Regras de Fusão

Além das regras de fusão, o *framework* também define funções de **Normalização**. Tais regras remapeiam as URIs declaradas nas visões exportadas, pertencentes à um mesmo

[1] http://www.geonames.org/

objeto, em uma única URI. Entretanto, por não ser o foco dessa dissertação, aqui são apresentadas apenas as regras de fusão.

As regras de fusão definem como duas representações distintas de um mesmo objeto do mundo real serão combinadas em uma única representação. Cada regra é definida como uma *assertiva de fusão de propriedade* (FPA - *fusion property assertion*). Uma FPA é definida sobre as propriedades de uma classe $C$, em que, dada duas visões exportadas distintas, $EV_i$ e $EV_j$, se $\exists$ (t,*rdf:type*,C) e (u,*rdf:type*,C), onde t e u são instâncias da classe $C$, para $t \in EV_i$, $u \in EV_j$ e $\exists$ (t,*owl:sameAs*,u). Portanto, uma FPA é dada na forma $\psi : P[C] \equiv f/Q$ , onde:

- $\psi$ é o nome da assertiva de fusão;
- $C$ é uma classe pertencente ao vocabulário de $O_M$;
- $P$ e $Q$ são propriedades de $C$;
- $f$ é uma função de fusão que determina métricas para avaliação das propriedades.

### 4.2.6 Materialização de Aplicações de mashups

A especificação criada pode ser materializada, permitindo a criação de aplicações de *mashups*. A materialização da especificação $\lambda$ é realizada em 3 passos:

1. **Materialização das Visões Exportadas**: nessa etapa as fontes de dados $S_{Vi}$ são traduzidas para o vocabulário de $O_{EVi}$, utilizando os mapeamentos $M_{EVi}$.

2. **Materialização dos Links Semânticos**: dada uma especificação de *links* $EL_{Vi}$ sobre um conjunto $E_V$ de visões exportadas, essa etapa computa os *links owl:sameAs* entre entidades similares, mas em visões exportadas distintas.

3. **Materialização da Visão Aplicação Mashup**: essa etapa materializa a visão de *mashup* $V$ aplicando as regras de fusão sobre a materialização das visões exportadas e dos links semânticos. Nessa etapa serão avaliadas as fontes de dados, segundo critérios de qualidade e também serão resolvidas possíveis inconsistências.

Para isso, podem ser utilizadas ferramentas especializadas para cada etapa. Basta que as regras sejam traduzidas para o formato aceito em cada ferramenta. Por exemplo, o SILK (BIZER et al., 2009) é uma ferramenta especializada na descoberta de *links owl:sameAs* em fontes *Linked Data*. Há também o SIEVE (MENDES; MüHLEISEN; BIZER, 2012), utilizado para realizar a fusão de fontes *Linked Data*.

## 4.3 Datasus_HUB

Nessa Seção, a aplicabilidade do *framework* apresentado é demonstrada por meio da especificação de uma visão de *mashup*, chamada *Datasus_Hub*. Essa visão integra semanticamente visões de duas fontes de dados do Sistema Único de Saúde brasileiro (SUS): Sistema de Informações sobre Nascidos Vivos (SINASC) e SUS eletrônico (e-SUS). Também é demonstrado, com o auxílio de um estudo de caso, como essa especificação pode ser utilizada para a criação de aplicações de *mashups*. As etapas para criação da especificação são discutidas nas subseções a seguir.

### 4.3.1 Fontes de Dados

As fontes de dados selecionadas foram SINASC e e-SUS, respectivamente representadas por $S_{sinasc}$ e $S_{esus}$. Ambas as fontes estão em um formato de dados relacional e, com isso, suas ontologias, *Sinasc_OWL* ($O_{S_{sinasc}}$ e *Esus_OWL* ($O_{S_{esus}}$), são representadas por visões de suas tabelas no banco de dados relacional. As visões *Sinasc_view* e *Esus_view* são representadas nas Figuras 26 e 27 respectivamente.

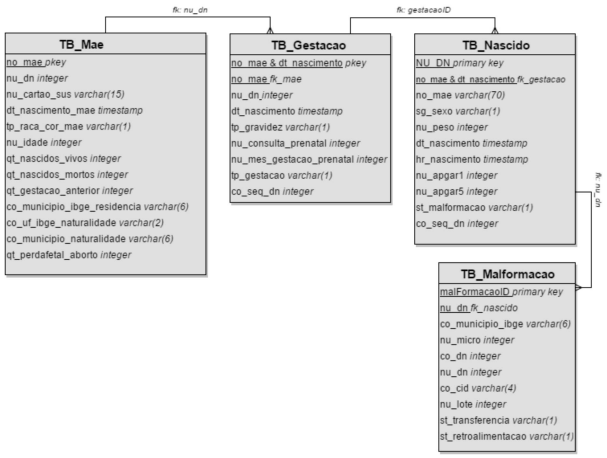



Figura 26 – Visão da base de dados SINASC

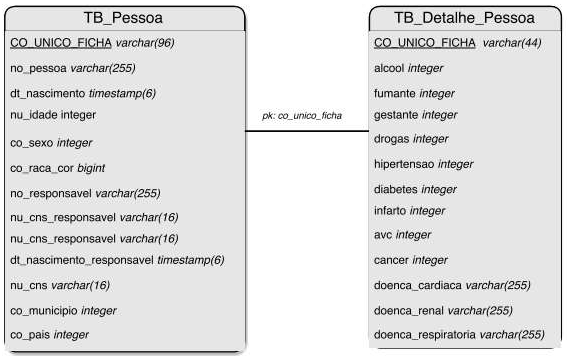



Figura 27 – Visão da base de dados e-SUS

A base SINASC contém informações sobre uma gestação e sobre o recém-nascido, como: a quantidade de consultas pré-natal da gestante; a quantidade de abortos em gestações anteriores; o peso do recém-nascido e possíveis anomalias-congênitas. A base e-SUS contém informações sobre um indivíduo, como hábitos e doenças. Nessa base é informado, por exemplo, se determinado indivíduo é fumante, usa drogas ou é diabético.

### 4.3.2 Ontologia de Domínio

A ontologia desenvolvida, *Datasus_OWL*, representada na Figura 28, reutiliza termos de vocabulários bem definidos, como *foaf (Friend of a Friend)*[2], *dbo (DBpedia Ontology)*[3] e *bio (Biographical Information)*[4]. Além disso, foi criado o vocabulário *gissa* para representação de novos termos, e.g. *"gissa:Gestacao"*.

[2] http://xmlns.com/foaf/spec/
[3] http://dbpedia.org/ontology/
[4] http://purl.org/vocab/bio/0.1/

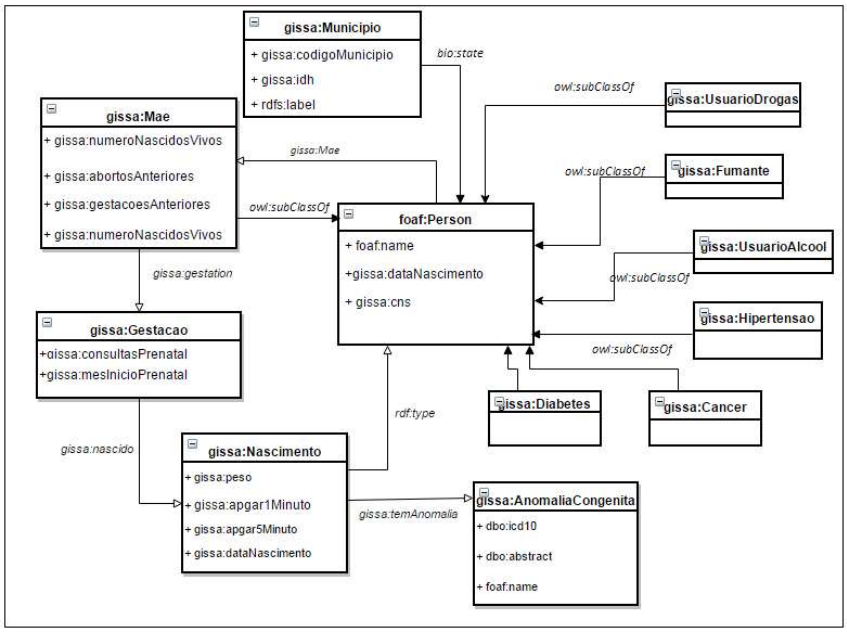



Figura 28 – Ontologia de Domínio Datasus_hub

### 4.3.3 Visões Exportadas

As visões exportadas *Sinasc_EV*, $E_{sinasc}$, e *Esus_EV*, $E_{esus}$, são definidas pelas regras de mapeamento entre $O_{S_{sinasc}}$ e $O_{S_{esus}}$ e a ontologia *Datasus_OWL*. Note que a aplicação de tais regras de mapeamentos resultam nas ontologias exportadas $O_{E_{sinasc}}$ e $O_{Eesus}$. As regras de mapeamento $M_{E_{sinasc}}$ e $M_{E_{esus}}$ foram definidas seguindo o formalismo de (VIDAL et al., 2014), que descreve regras para mapear dados relacionais em RDF. Um trecho dos mapeamentos do SINASC é apresentado nas Tabela 5.

Tabela 5 – Mapeamentos SINASC

| | |
|---|---|
| **ACC1** | gissa:Pessoa ≡ TB_Pessoa[CO_UNICO] |
| **ACC2** | gissa:Mulher ≡ TB_Detalhe_Pessoa[CO_UNICO] [no_sexo = 'F'] |
| **ACC3** | gissa:UsuarioDrogas ≡ TB_Detalhe_Pessoa[CO_UNICO] [drogas = '1'] |
| **ACC4** | gissa:UsuarioAlcool ≡ TB_Detalhe_Pessoa[CO_UNICO] [alcool = '1'] |

A especificação das visões exportadas *Sinasc_EV* e *Esus_EV* são, respectivamente:

- $E_{sinasc} = \{Sinasc\_EV; S_{sinasc}; O_{sinasc}; O_{E_{sinasc}}; M_{E_{sinasc}}\}$, tal que, $O_{E_{sinasc}}$: *{gissa:Mae, gissa:Gestacao, gissa:Nascimento, gissa:AnomaliaCongenita}*.
- $E_{esus} = \{Esus\_EV; S_{esus}; O_{esus}; O_{E_{esus}}; M_{E_{esus}}\}$, tal que, $O_{E_{sinasc}}$: *{foaf:Person, gissa:Municipio, gissa:Diabetes, gissa:Cancer, gissa:Hipertensao, gissa:UsuarioAlcool, gissa:Fumante, gissa:UsuarioDrogas}*.

### 4.3.4 Links Semânticos

A especificação dos *links semânticos* entre $E_{sinasc}$ e $E_{esus}$ foi definida sobre a classe *"foaf:Person"*, utilizando as propriedades: *foaf:name* e *gissa:cns*, que representa um identificador do cidadão nas bases da Saúde e *gissa:dataNascimento*. Para verificação da similaridade das instâncias, foi definida a seguinte regra:

Considere $t \in E_{sinasc}$ e $u \in E_{esus}$ instâncias, tal que existam as triplas ($t$, *rdf:type*, *foaf:Person*) e ($u$, *rdf:type*, *foaf:Person*). Tome também $w_1, w_2$ e $w_3$ como objetos das triplas ($t$, *foaf:name*, $w_1$); ($t$, *gissa:cns*, $w_2$) e ($t$, *gissa:dataNascimento*, $w_3$), respectivamente. A definição de $v_1, v_2$ e $v_3$ para $u$ é análoga. Sendo assim,

$\exists(t, owl : sameAs, u)$, sse $\sigma(v_i, w_i) > \alpha$, para i = 1, 2 e 3, onde $\sigma$ = distância 3-gram (KONDRAK, 2005) e $\alpha > 0.5$.

#### 4.3.4.1 Fusão e Qualidade

Para quantificar a qualidade das bases, utilizamos alguns dos critérios abordados em (PIPINO; LEE; WANG, 2002), como: (i) quantidade de registros ausentes nas bases de dados; (ii) quantidade de registros duplicados e (iii) quantidade de registros com erros, como nomes de pessoas com erros de escrita, por exemplo. Atribuímos peso 1 para cada critério, e a fonte de dados com maior pontuação representa a menos confiável. Essa métrica foi denominada de *KeepValueByReputation* (KVBR) (guarde o valor pela reputação). Como a única classe que contém *links owl:sameAs* é *foaf:Person*, as FPAs definidas são sobre as propriedades dessa classe.

1. $FPA_1$: *gissa:nomeCompleto*[*foaf:Person*] $\equiv$ *KVBR/gissa:nomeCompleto*;
2. $FPA_2$: *gissa:dataNascimento*[*foaf:Person* $\equiv$ *KVBR/gissa:dataNascimento*;

## 4.4 Construção de Aplicações de Mashups com Datasus_hub

Nessa Seção, foi utilizado o *mashup Datasus_hub* para criar uma aplicação de *mashup*. Nesse exemplo fictício, um gestor da saúde no Brasil quer alertar a população de seu município sobre os perigos dos maus-hábitos durante a gravidez. Para isso, o gestor quer correlacionar o uso de drogas, do tabaco e de álcool, durante a gestação, com a malformação em recém-nascidos e partos prematuros. No Brasil, essas informações estão distribuídas em fontes de dados heterogêneas. Para isso, *Datasus_hub* foi materializado para permitir a criação da aplicação de mashup *SOS:Gestacao*, descrita a seguir.

### 4.4.1 SOS:Gestacao

Essa aplicação vai permitir que gestores possam relacionar fatores dos óbitos-infantis, e.g. prematuriedade e malformação, com hábitos e doenças da mãe durante a gravidez. Essa aplicação consome os dados do *mashup Datasus_hub*. As etapas necessárias para a materialização, descritas em 4.2.6, são detalhadas nas subseções seguintes.

#### 4.4.1.1 Materialização das Visões Exportadas

Nessa etapa, os mapeamentos $M_{E_{sinasc}}$ e $M_{E_{esus}}$ foram traduzidos para a linguagem padrão[5] para mapeamentos de dados relacionais em RDF, R2RML (W3C, 2016). Esses mapeamentos são processados pela *engine* da ferramenta *D2R-Server*. Essa ferramenta recebe os mapeamentos R2RML e materializa as visões RDF correspondentes.

#### 4.4.1.2 Materialização dos Links Semânticos

A materialização dos links semânticos foi feita utilizando a ferramenta SILK (BIZER et al., 2009). A heurística de descoberta de *links owl:sameAs* descrita em 4.3.4. O SILK recebe como parâmetro de configuração um arquivo XML onde são descritas quais propriedades serão avaliadas e quais métricas devem ser aplicadas para descoberta dos links. Uma representação gráfica das instâncias das visões exportadas *Sinasc_EV* e *Esus_EV* já com os *links* criados pode ser vista na Figura 29.

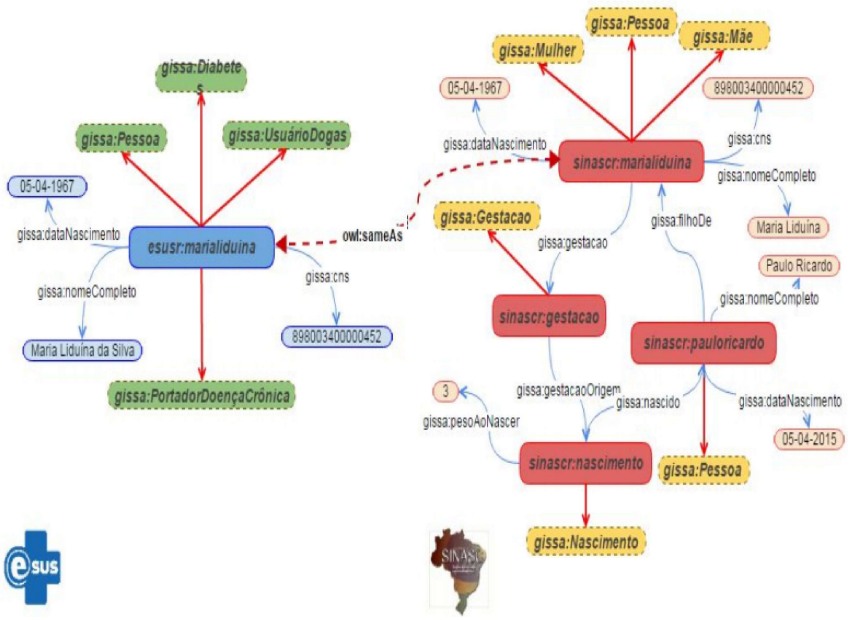



Figura 29 – Instâncias das Visões Exportadas

[5] Padrão W3C

#### 4.4.1.3 Materialização da Visão de Mashup

Finalmente, as visões exportadas *Esus_EV* e *Sinasc_EV* devem ter seus indivíduos com *links owl:sameAs*, porém representados nas visões de formas possivelmente distintas, unificados numa única representação. Para isso, foi utilizado a ferramenta SIEVE (MENDES; MüHLEISEN; BIZER, 2012). Nessa ferramenta, o usuário especifica, por meio de um arquivo XML, métricas para avaliar a qualidade das fontes de dados. Além disso, também são especificadas quais propriedades serão utilizadas na fusão, bem como uma função para avaliar a qualidade das fontes. Nesse estudo de caso, foi utilizado o critério de maior confiabilidade nas fontes. Além do arquivo de configuração XML, o SIEVE recebe as visões exportadas RDF e um conjunto de *links owl:sameAs*, retornado pelo SILK. A Figura 30 representa a fusão de mashup.

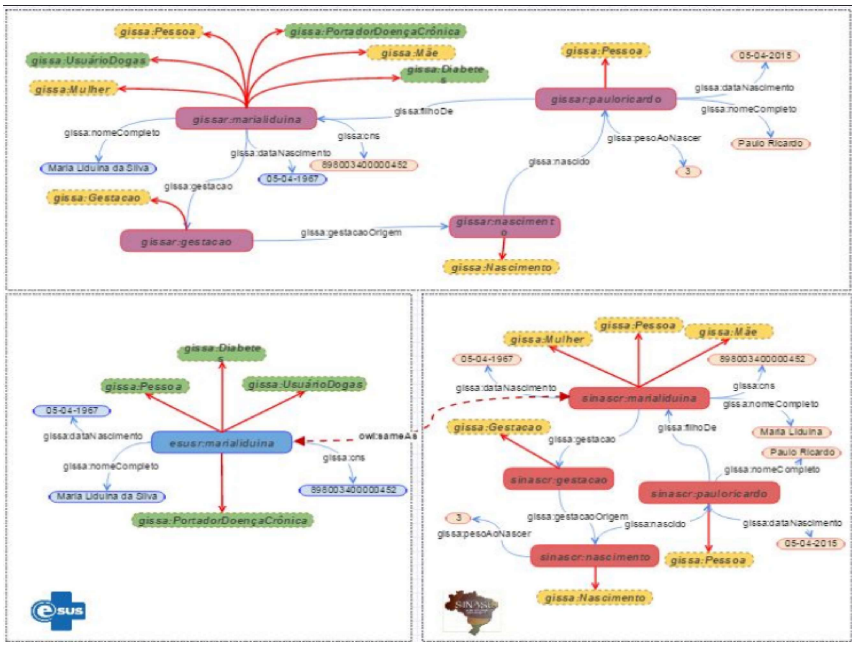



Figura 30 – Instâncias da Visão de Mashup materializada

Com a materialização da visão integrada, podem ser aplicadas técnicas de *visualização de dados*, e.g. *dashboards* e gráficos, para melhorar a visibilidade das informações e auxiliar gestores na tomada de decisão. Também podem ser aplicadas técnicas de *mineração de dados* para descobrir fatos escondidos nas informações, como: *"60% das mães que fumaram durante a gravidez geraram um filho com peso ao nascer abaixo de 2Kg"*[6]

## 4.5 Conclusão

Nesse Capítulo foi apresentado um *framework* conceitual que especifica formalmente um *Linked Data Mashup*. Essa especificação pode ser utilizada para a construção de aplicações

[6] Essa afirmação é apenas uma suposição e não deve ser tomada como um fato.

(aplicações de *mashup*) ou para auxiliar no desenvolvimento de *data warehouses*. Esse *framework* é essencial para o desenvolvimento da proposta dessa dissertação, apresentada no próximo Capítulo.

# 5 MAURA: Construção de Linked Data Mashups

## 5.1 Introdução

A maior problemática na integração de dados é a conciliação semântica das informações em bases distintas. Essa conciliação é realizada por meio da construção de um *Esquema Global*. Para isso, os dados têm que ser analisados, entendidos e então mapeados nesse esquema global. Em bancos de dados relacionais, os dados são representados por tabelas, onde ou se tem pouca ou nenhuma semântica sobre as informações. Em bancos relacionais, a conciliação semântica pode não ter uma solução viável em projetos de grandes instituições, por exemplo, onde os bancos de dados tendem a ser enormes.

Nesse contexto, a Web Semântica representa um novo paradigma na integração de dados. Os dados na Web Semântica são representados no formato de triplas e cada informação é referenciada por uma URI, única na *Web*, que contém informações sobre aquele dado. No Capítulo 4, foi discutido um *framework* para especificação de *Linked Data Mashups* (LDM). Embora a mudança de paradigma represente mais facilidade para integrar dados, a especificação de um *mashup* em fontes *Linked Data* ainda não é uma tarefa trivial. Para construir um *mashup*, são exigidos conhecimentos específicos sobre as tecnologias da Web Semântica, como RDF e OWL, e em integração de dados, como construção de um modelo global e definição de mapeamentos. Além disso, como demonstrado no Capítulo de revisão sistemática, 3, as especificações de um *mashup* não são reutilizadas, i.e. sempre que houver a necessidade de construir um novo *mashup*, a especificação tem que ser alterada. Ainda segundo o Capítulo 3, nas abordagens atuais, a alteração dessa especificação exige conhecimentos específicos, dificultando para um usuário de propósito geral, e.g. um gestor, de criar seu próprio *mashup*.

Esse capítulo apresenta o *MAshUp mediador for RDF Applications* (MAURA), um *framework* baseado em *Mediador Semântico* para facilitar a criação de *Linked Data Mashups* (LDM) mediante a reutilização de especificações de *mashups*. Um dos objetivos da proposta é permitir que usuários finais, com poucos conhecimentos em computação e Web Semântica, sejam capazes de desenvolver suas próprias *Visões de Aplicações de Mashups*[1] sobre informações distribuídas. Nesse *framework*, um *LDM* é formalmente especificado utilizando a metodologia de (VIDAL et al., 2015). A partir dessa especificação, usuários finais utilizam parâmetros para construir suas próprias visões de aplicações por meio de uma *interface* gráfica, de forma fácil, rápida e intuitiva. Um dos diferenciais

[1] Nessa dissertação, o termo *visões de aplicações de mashups* será abreviado em apenas *visões de aplicação*

dessa abordagem é que o processo de integração semântica é realizado uma única vez, e a partir de então, a especificação gerada será reutilizada para a criação automática de *mashups* posteriores. Para isso, o Mediador Semântico realiza um processo de reescrita de especificação. Nesse processo, o mediador aplica os parâmetros do usuário na especificação de *mashup*, gerando uma nova especificação. Essa nova especificação pode ser utilizada para construção de aplicações ou auxiliar no desenvolvimento de *data warehouses*. Nesse capítulo também são demonstrados casos de uso de aplicações criadas com o mediador.

## 5.2 MAURA - Mediador Semântico

### 5.2.1 Visão geral

Como discutido no Capítulo 4, seguindo os conceitos de (VIDAL et al., 2015), um *Linked Data Mashup* pode ser formalmente especificado em 4 etapas: (i) modelagem de uma ontologia de domínio; (ii) especificação das ontologias exportadas, definindo mapeamentos das ontologias-fonte para a ontologia de domínio; (iii) defininção das heurísticas para descobertas de *links owl:sameAs* e (iv) definição das regras de fusão.

Um mediador convencional é um *software* que reescreve uma consulta em subconsultas, as executa sobre as diversas bases de dados, trata os dados recebidos e retorna uma visão materializada ao usuário. No mediador semântico proposto nesta dissertação, o processo de reescrita é realizado sobre a especificação de um *mashup*. O *framework* descrito no Capítulo 4, funciona como um módulo do *framework* mediador semântico, responsável por gerar a especificação de *mashup*. O mediador então utiliza essa especificação para construir *Visões de Aplicações de Mashups*[2] em tempo de execução. Isto é, o *mashup* é **virtual** e só é materializado quando requisitado pelo usuário.

Para realizar o processo de construção de uma visão de aplicação em tempo de execução, o mediador armazena a especificação previamente criada em um formato próprio. Além disso, o usuário especifica os parâmetros da visão de aplicação a ser criada. A especificação dos parâmetros pode variar de acordo com a implementação do mediador, podendo ser feita por meio de uma *interface* gráfica. Os parâmetros do usuário são: uma ontologia, cujo vocabulário é um subconjunto do vocabulário da ontologia de domínio e um conjunto de filtros. O mediador aplica os parâmetros do usuário à especificação armazenada, a fim de gerar uma nova especificação que corresponda às necessidades do usuário. Essa especificação é então materializada e retornada ao usuário como uma visão de aplicação de *mashup*. Em resumo, o mediador constrói uma visão de aplicação seguindo os passos:

1. Especifica-se um *Linked Data Mashup* sobre as fontes heterogêneas;

[2] Esse termo será abreviado em "visões de aplicações"durante essa dissertação

2. Usuário define os parâmetros para construção da visão de aplicação;
3. O mediador aplica os parâmetros na especificação existente, gerando uma nova;
4. O mediador materializa a especificação gerada e retorna ao usuário.

Esta abordagem que uma especificação de *mashup* pode ser utilizada por terceiros e incrementada, caso necessário.

O conceito de reutilização de especificação, criado pelo MAURA, determina uma abordagem *pay-as-you-go*. Isto é, novas fontes de dados podem ser integradas à especificação sempre que necessário, sem interferir na criação de aplicações que não necessitem de tais fontes. Além disso, essa especificação de *mashup* é armazenada no mediador e pode ser disponibilizada para terceiros, economizando tempo no processo de integração e possibilitando estudos sobre as fontes integradas. A arquitetura do mediador semântico é detalhada nas subseções seguintes.

### 5.2.2 Arquitetura 4-Camadas

A arquitetura do mediador é resumida na Figura 31. Na **Camada de Integração Semântica**, $M$ é um *LDM*, especificado utilizando o *framework* discutido no Capítulo 4. Também é nessa camada em que é realizado o processo de reescrita da especificação e materialização do *mashup*.

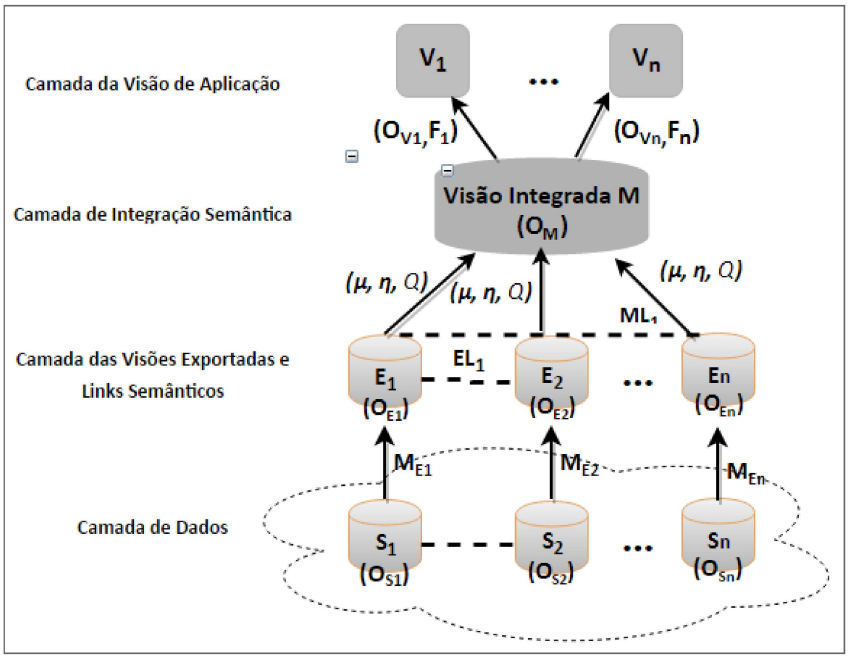



Figura 31 – Arquitetura 4 Camadas do Mediador Semântico

Na **Camada de Visão de Aplicação**, os usuários podem especificar os parâmetros, $(O_{Vi}, F_i)$, para construção da visão de aplicação, por meio de uma *interface* gráfica. A **Ontologia de Aplicação** $O_{Vi}$, cujo vocabulário deve ser um subconjunto do vocabulário de $O_M$, representa os conceitos de interesse do usuário; enquanto $F_i$ representa um conjunto de filtros (e.g. cidade ou ano ) que serão aplicados sobre os dados. Como será discutido na subseção 5.2.3, para a materialização da visão $V_i$, é necessário aplicar os parâmetros $(O_{Vi}, F_i)$ na especificação do *mashup* $M$, gerando uma especificação de $V$. $V$ representa uma visão de aplicação sobre o *mashup* $M$. A visão de aplicação $V$ é então materializada como definido em 4.2.6. As camadas de **Visões Exportadas e Links Semânticos** e de dados são análogas ao *framework* do Capítulo 4.

### 5.2.3 Construção de uma Visão de Aplicação

Para a construção de uma visão de aplicação de *mashup*, denotada por $V$ na Figura 31, são necessárias 3 etapas: (i) geração da especificação de $V$ sobre $M$; (ii) geração da especificação de $V$ sobre as fontes de dados e (iii) materialização de $V$. Essas etapas são descritas a seguir.

#### 5.2.3.1 Geração da Especificação de $V$ sobre $M$

A especificação de V sobre M é uma tupla $\delta = (O_V, F_V)$, onde:

- $O_V$ é uma *ontologia de aplicação*, tal que o vocabulário de $O_V$ seja um subconjunto do vocabulário de $O_M$;
- $F_V$ é um conjunto de filtros definidos sobre os conceitos de $O_V$, na forma $C_S/P$ $\mu$, tal que $C_S \in O_V$; $P$ é uma propriedade de $C_S$ e $\mu$ uma condição, e.g. $P > 50$.

A ontologia de aplicação deve representar os conceitos de interesse do usuário. Para definição dos parâmetros, pode ser utilizada uma *interface* gráfica, caso a implementação do mediador dê suporte.

### 5.2.4 Geração da Especificação de $V$ sobre as Fontes de Dados

Nessa etapa, a especificação de $V$ sobre as fontes de dados $S_1, ..., S_n$ é gerada por meio da combinação dos parâmetros do usuário com a especificação do *mashup* $M$. Sejam:

$\delta = (O_V, F_V)$, os parâmetros do usuário;

$\lambda_m = \{M, O_M, E_M, EL_M, \mu_m, Q\}$, a especificação de $M$;

E $\lambda_v$, especificação de $V$, definida pelo resultado da combinação de $\delta$ e $\lambda_m$, tal que:

$\lambda_v = \{V, O_V, E_V, EL_V, \mu_v, Q\}$, onde:

- $V$ é o nome da visão de aplicação;

- $O_V$ é a ontologia de domínio da especificação de $V$;
- $E_V$ é o conjunto de visões exportadas, onde cada $E_{Vi} \in E_V$ é uma tupla $E_{Vi} = \{EV_i, S_i, O_{Si}, O_{EVi}, M_{EVi}\}$, definida da seguinte forma:
  $\forall E_{Vi} \in E_V$,
  $O_{EVi} = O_V \cap O_{EMi}$, tal que $O_{EMi} \in E_{Mi}$ e
  Considerando: $\forall \tau_{mj} \in M_{EMi}$, $\tau_{mj} = C_S(\phi) \rightarrow C_T$, onde $C_S$ é uma classe fonte; $\phi$ é uma condição e $C_T$, uma classe target. $\phi$ denota uma condição para que $C_S$ seja mapeado em $C_T$.
  Assim, $\tau_m \in M_{EVi}$ sse $C_S(\phi) \in O_S$ e $C_T \in O_{EV}$. Além disso, $\forall \tau_m \in M_{EVi}$ que mapeia $C_S(\phi) \rightarrow C_T$, se $C_T \equiv C_i$, tal que $F_i = C_i/P\mu_i \in F$, então a condição $\mu_i$ deve ser adicionada à condição $\phi$ de $C_S$.
- $\mu_v$ são as regras de fusão, definidas na forma:
  $\forall \alpha_i \in \mu_m$, onde $\alpha_i$ é uma regra para fusão de duas entidades, $C_u$ e $C_v$, $\alpha_i \in \mu_v$ sse $C_u \in O_{EVi}$ e $C_v \in O_{EVj}$.

$Q$ permanece inalterado de $\lambda_m$ para $\lambda_v$, pois as fontes de dados são as mesmas. A especificação dos *links* semânticos também permanece inalterada. Isso porque o vocabulário de $O_V$ é um subconjunto do vocabulário de $O_M$, assim, quaisquer regras $EL_i \in EL_M$ de $\lambda_m$ também serão válidas para $\lambda_v$. A materialização de $V$ é realizada como discutido em 4.2.6.

Entre as vantagens dessa abordagem, pode-se destacar:

- *Pay-as-you-go*. O processo de integrar semanticamente as fontes de dados é realizado uma única vez e, a partir de então, pode ser incrementado sempre que necessário.
- *Mashup* parametrizado. Um *mashup* pode ser criado sem a necessidade de alterar ou criar uma especificação de *mashup*. Além disso, apenas é materializado o que for especificado pelo usuário;
- Não há necessidades de conhecimentos específicos em Web Semântica. O uso de parâmetros também possibilita que usuários de propósito geral criem *mashups* de forma transparente, i.e. o usuário não precisa nem conhecer o processo de integração semântica nem como o mediador materializa os dados;
- Impulsionar a comunidade científica. Uma das propostas do *framework* mediador é que uma especificação de *mashup* possa ser disponibilizada na *Linked Open Data*, permitindo que outros pesquisadores e desenvolvedores a utilizem. Por exemplo, um pesquisador pode utilizar um *mashup*, criado por outra equipe, para aplicar algoritmos de mineração de dados e inferir fatos antes escondidos;
- *Mashups* podem ser criados de forma automática para fontes com modelos de dados similares. A especificação de um *mashup* concilia os modelos heterogêneos das fontes de dados. Assim, fontes de dados com modelos de dados já integrados

podem ser utilizadas pelo mediador para a construção automática de *mashups*. Por exemplo, os dados da fonte SINASC diferem em cada município brasileiro, porém seu modelo é idêntico. Se um usuário de outro município precisar gerar um *mashup*, basta aplicar novos parâmetros correspondentes. Esse caso é discutido na subseção 5.3.3.

## 5.3 Casos de Uso

Nessa Seção, o uso do mediador é demonstrado com o auxílio de casos de uso. Nas abordagens para integração de dados citadas até aqui, a criação de aplicações se dá com a materialização do *mashup* completo, independente se o usuário precisa ou não de todas as informações. Por exemplo, para criar determinada aplicação, um gestor de saúde pode não precisar de todas as informações de *Datasus_HUB* (Cáp. 4, Seção 4.3), mas apenas uma *visão*, i.e. uma porção dos dados. Vale notar que se o *mashup* for muito maior que a necessidade do usuário, diversos dados não-relevantes serão materializados. Porém, com a abordagem parametrizada, utilizada pelo mediador proposto, o usuário especifica apenas o que é relevante para a sua aplicação. Outra característica do mediador é a extensibilidade. Novas fontes podem ser adicionadas ao *mashup* sempre que necessário, sem afetar os *mashups* já criados. Finalmente, o mediador também propõe impulsionar a comunidade científica. A seguir, tais características são demonstradas com o auxílio de casos de uso. Em todos os exemplos, é utilizada o *mashup Datasus_HUB*.

### 5.3.1 Caso de uso 1 : SOS:Gestacao

Aqui, o mediador será utilizado para recriar a aplicação SOS:Gestacao (Seção 5.3.1). Na abordagem utilizada em 5.3.1, o *mashup Datasus_HUB* foi inteiramente materializada. A aplicação SOS:Gestacao, por sua vez, não requer todas informações do *mashup*, mas apenas um *recorte*. Nessa aplicação, um gestor de saúde da cidade de Tauá/CE quer correlacionar o uso de drogas, tabaco e álcool, durante a gestação, com a malformação em recém-nascidos. Pela ontologia de domínio de *Datasus_HUB* (Figura 28), é possível notar que há mais informações do que o gestor necessita[3]. Dessa forma, o gestor definiria os parâmetros $\delta = (O_V, F_V)$ como se segue:

- $O_V =$ gissa:Fumante, gissa:UsuarioAlcool, gissa:UsuarioDrogas, gissa:Mae, gissa:Gestacao, foaf:Person, gissa:Municipio, gissa:Nascimento e gissa:AnomaliaCongenita;
- $F =$ { gissa:Municipio/rdfs:label = "Tauá"}.

A seguir, a especificação $\lambda_v$ é gerada automaticamente pelo mediador, combinando $\delta$ e $\lambda_m$, como definido em 5.2.4.

[3] Esse é um exemplo fictício e, portanto, o foco não é discutir quais informações são relevantes ou não para a análise de partos prematuros, mas sim demonstrar a aplicabilidade do mediador

- $\lambda_v = \{SOS{:}Gestacao, O_V, E'_{sinasc}, E'_{esus}, EL_m, \mu_m, Q\}$, onde,

- $E'_{sinasc} = \{Sinasc\_EV'; S_{sinasc}; O_{sinasc}; O_{E'_{sinasc}}; M_{E'_{sinasc}}\}$, tal que,
  $O_{E'_{sinasc}}$: *{gissa:Mae, gissa:Gestacao, gissa:Nascimento, gissa:AnomaliaCongenita}*;
- $E'_{esus} = \{Esus\_EV'; S_{esus}; O_{esus}; O_{E'_{esus}}; M_{E'_{esus}}\}$, tal que, $O_{E'_{esus}}$: *{foaf:Person, gissa:Municipio, gissa:UsuarioAlcool, gissa:Fumante, gissa:UsuarioDrogas}*.

Aqui, também foi utilizado o formalismo especificado em (VIDAL et al., 2014) para mapeamentos relacional em RDF. Um trecho dos mapeamentos de *Esus_EV'* é representado na tabela 7.

Tabela 6 – Mapeamentos *Esus_EV'*

| | |
|---|---|
| **ACC1** | gissa:UsuarioAlcool ≡ TB_Detalhe_Pessoa[CO_UNICO] [alcool = '1'] |
| **ACC2** | gissa:PortadorDoencaCardiaca ≡ TB_Detalhe_Pessoa[CO_UNICO] [doenca_cardiaca = '1'] |
| **ACC3** | gissa:Cancer ≡ TB_Detalhe_Pessoa[CO_UNICO] [cancer = '1'] |
| **ACC4** | gissa:Diabetes ≡ TB_Detalhe_Pessoa[CO_UNICO] [diabetes = '1'] |
| **ACC5** | gissa:Hipertenso ≡ TB_Detalhe_Pessoa[CO_UNICO] [hipertenso = '1'] |

Finalmente, as regras de fusão são:

- $FPA_1$: *gissa:nomeCompleto[foaf:Person]* ≡ *KVBR/gissa:nomeCompleto*;
- $FPA_2$: *gissa:dataNascimento[foaf:Person* ≡ *KVBR/gissa:dataNascimento*;

Como discutido em 5.2.4, as regras de descoberta dos *links* semânticos, $EL_m$, bem como as regras de qualidade, $Q$, se mantém de $\lambda_m$ para $\lambda_v$.

A materialização de $V$ permite a criação de aplicações, e.g. gráficos e *dashboards*, que auxiliem um gestor na tomada de decisão. Sempre que um novo *mashup* precisar ser criado, basta que o usuário utilize novos parâmetros, sem que haja a necessidade de modificar a especificação do *mashup*.

### 5.3.2 Caso de Uso 2: Integração com DBPedia

Para demonstrar a abordagem *pay-as-you-go*, inerente ao mediador proposto, será adicionada a fonte de dados DBPedia[4] à especificação $\lambda_m$. Para esse exemplo, suponha dois gestores: A e B. O gestor A quer utilizar os dados da *DBPedia*, relacionados ao Código Internacional de Doenças 10 (CID-10), para enriquecer as informações de seu *mashup*. Na *DBPedia*, cada entidade doença (*dbo:Disease*[5]) possui um código CID-10 (*dbo:icd10*) e um resumo (*dbo:abstract*), que dá detalhes sobre a doença, algo que não existe nas fontes já integradas. Para isso, novas regras têm que ser adicionadas à especificação $\lambda_m$.

Seja $E_{dbpedia} = \{DBPedia\_EV, S_{dbpedia}, O_{dbpedia}, O_{E_{dbpedia}}, M_{dbpedia}\}$, tal que:

[4] http://dbpedia.org
[5] dbo é um prefixo cuja URI é *http://dbpedia.org/ontology/*

**Regras de mapeamento**:

Tabela 7 – Mapeamentos *DBPedia_EV*

| | |
|---|---|
| **ACC1** | gissa:AnomaliaCongenita(x) ← dbo:Disease(y) |
| **ACC2** | dbo:abstract(t, x) ← dbo:Disease(y); dbo:abstract(y, x) |
| **ACC3** | dbo:icd10(t, x) ← dbo:Disease(y); dbo:abstract(y, x) |

**Regras de Links Semânticos**:

- Seja $t \in E_{sinasc}$ e $u \in DBPedia_EV$ instâncias de objetos, tal que existam as triplas ($t$, *rdf:type*, *gissa:AnomaliaCongenita*) e ($u$,*rdf:type*, *dbo:Disease*). Seja também $c_1$ e $c_2$ objetos das triplas ($t$,*dbo:icd10*, $c_1$) e ($u$,*dbo:icd10*,$c_2$), respectivamente. Assim,
  $\exists(t, owl : sameAs, u)$, sse $\sigma(c_1, c_2) > \alpha$, onde $\sigma$ = distância de Levenshtein (YUJIAN; BO, 2007).

**Regras de Fusão**:

- $FPA_1$: *dbo:abstract*[*gissa:AnomaliaCongenita*] $\equiv$ *KVBR/dbo:abstract*;

Após integrar a nova fonte ao *mashup* já existente, o gestor A pode criar uma visão de aplicação, incorporando a DBPedia em sua ontologia de aplicação $O_V$. O gestor B não precisa dessa nova integração com a DBPedia, assim, basta não incorporá-la à sua ontologia de aplicação. Caso a materialização do *mashup* não fosse parametrizada, todo o *mashup*, incluindo os dados da DBPedia, seria materializado.

### 5.3.3 Reutilização de Especificações para Impsulsionar Estudos

O conceito de reutilização de uma especificação de *Linked Data Mashup*, criado pelo *framework*, denota segue uma abordagem *pay-as-you-go*. Esta abordagem determina que uma especificação de *mashup* pode ser utilizada por terceiros e incrementada, caso necessário. Nesta subseção, é discutido como o MAURA pode ser utilizado para impulsionar a comunidade científica. Nessa discussão, será utilizada a especificação de *mashup* *Datasus_Hub*

O mediador MAURA utiliza uma especificação de *mashup* para criar visões de aplicações. Essa especificação determina que os *modelos* heterogêneos das fontes de dados foram conciliados semanticamente. Todas as etapas da especificação são regras criadas sobre os modelos, nunca sobre as instâncias. A especificação de um *link semântico*, subseção 5.3.2, por exemplo, é definida sobre as ontologias exportadas de duas fontes, não sobre seus dados. O mesmo vale para as regras de fusão e mapeamentos. Seguindo essa abordagem, fontes com dados distintos, mas descritas sobre um mesmo modelo, podem utilizar especificações já existentes para criar *mashups* e aplicações de forma automática, evitando o re-trabalho.

Com isso, uma equipe que reutilize uma especificação de *mashup* pode focar seu esforço numa outra tarefa, como na mineração de dados ou análises.

Por exemplo, suponha uma implementação do mediador semântico que dê suporte à conexão com determinada fonte de dados na LOD. Nessa implementação, um usuário pode depositar uma especificação na LOD, bem como também pode buscar por uma especificação, como em uma lista.

Suponha também que uma equipe de tecnologia de determinada empresa de, por exemplo, Fortaleza/CE, seja dividida em duas diretrizes: integração de dados e mineração de dados. A equipe de integração está criando um *data warehouse* para analisar casos de óbitos maternos. No Brasil, as informações sobre óbitos estão no Sistema de Informações sobre Mortalidade (SIM)[6], enquanto informações sobre mãe estão distribuídas no e-SUS e SINASC. Note que se tal equipe tiver a possibilidade de utilizar o mediador para acessar a fonte de dados na LOD que contém o *mashup Datasus_Hub*, seu trabalho será bastante encurtado. A equipe de integração terá que adicionar uma nova fonte (SIM) à especificação de *Datasus_Hub* para criar os *mashups* que a empresa necessita. Também note que, por conta do menor esforço requirido para a integração das fontes, parte dessa equipe pode ser redirecionada para, por exemplo, a mineração de dados. Além disso, se essa equipe, após a integração da fonte SIM à especificação já existente, disponibilizar essa nova especificação na fonte LOD, um terceiro grupo pode utilizá-la e assim por diante. O conceito de reutilização de especificações criado nesta dissertação pode ser o primeiro passo para uma *Web de Dados Integrada*.

## 5.4 Conclusão

Nesse Capítulo foi apresentado o MAURA, um *framework* baseado em Mediador Semântico para construção de *Linked Data Mashups*. Um dos módulos do mediador é o *framework* apresentado no Capítulo 4, responsável por especificar um *mashup* em *Linked Data*. O mediador utiliza essa especificação de *mashup* para permitir que usuários criem visões de aplicações de forma fácil, intuitiva e de forma transparente, i.e. sem conhecer o processo de integração semântica ou como a visão de aplicação é gerada. MAURA cria o conceito de reutilização de especificação. Isso permite que uma especificação de *mashup* possa ser reutilizada por terceiros, economizando tempo na tarefa de integração de dados e possibilitando estudos sobre esses dados. Dentre os objetivos que espera-se alcançar com o uso da abordagem proposta, podem ser destacados: (i) permitir que usuários sem conhecimentos em integração de dados criem *mashups*; (ii) impulsionar a comunidade de integração de dados em *Linked Data* e (iii) apoiar a tomada de decisão de gestores.

[6] http://www2.datasus.gov.br/DATASUS/index.php?area=060701

# 6 A implementação de um protótipo para o MAURA

## 6.1 Introdução

O tema "Integração de Dados"é alvo de pesquisas desde a década de 80, com diversos trabalhos sobre integração de visões; mapeamentos, dentre outros. Até há pouco tempo, a grande maioria dos trabalhos nessa área se focava em bancos de dados relacionais, onde os dados são descritos na forma de tabelas, com pouca semântica sobre as informações. A Web Semântica, porém, propôs uma nova forma de visualização dos dados e, portanto, representa um novo paradigma para integração de dados.

No Capítulo 5 foi proposto o MAURA, um *framework* baseado em mediador semântico, descrito com o auxílio de regras formais e casos de uso. Uma das características desse *framework* é a de possibilitar que um usuário sem conhecimentos específicos em Web Semântica possa criar seu próprio *mashup*. Esse *framework* também propõe a reutilização de especificações e, com isso: novas fontes podem ser adicionadas à integração; grupos de estudo e/ou desenvolvimento podem reutilizar especificações criadas por terceiros para conduzir estudos; impulsionar pesquisas, dentre outros.

Este capítulo apresenta uma guia para a implementação do *framework* MAURA. Para isso, são apresentados um diagrama conceitual e os principais algoritmos necessários no mediador. Além disso, é apresentado um protótipo, abordando as tecnologias utilizadas e os desafios encontrados. Por fim, o uso da ferramenta é demonstrado com o auxílio de um estudo de caso.

## 6.2 Modelagem de MAURA

### 6.2.1 Modelo Conceitual

Para auxiliar no entendimento da abordagem e para guiar o processo de implementação, foi desenvolvido um modelo conceitual, representado na Figura 32.

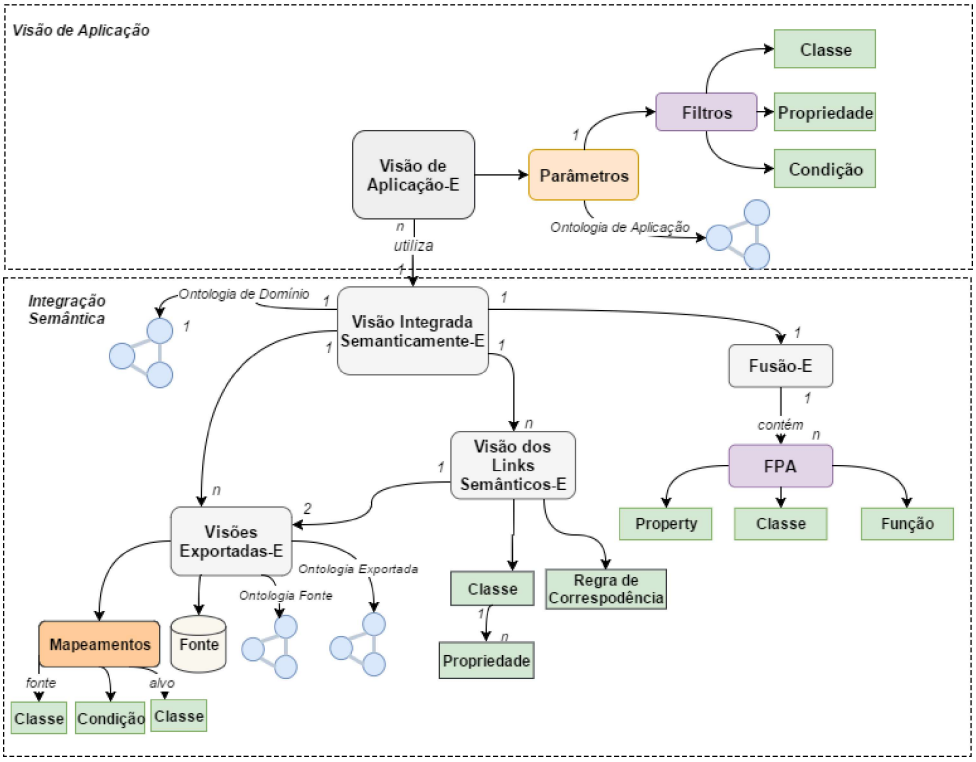



Figura 32 – Modelo Conceitual Mediador Semântico

Nessa figura, todas as visões são virtuais, portanto, *especificações*. Ao final de cada visão, há um caractere para defini-lo como especificação ou materialização. O caractere *"E"* determina que uma visão é uma *especificação*, enquanto *"M"* determina que é uma *materialização*. Nessa figura, as duas principais visões são *Visão Integrada Semanticamente-E* e *Visão de Aplicação-E*, descritas a seguir.

#### 6.2.1.1 Visão Integrada Semanticamente

Após o processo de especificação de um *mashup*, o resultado é uma visão integrada das fontes, representada na Figura por *Visão Intregada Semanticamente-E*. Essa visão contém quatro componentes: (i) um conjunto de *Visões Exportadas-E*; (ii) um conjunto de *Visões de Links Semânticos-E*; (iii) uma especificação de fusão, denotada por *Fusão-E* e (iv) uma *Ontologia de Domínio*.

**(i) Visão Exportada-E**. Cada especificação de visão exportada contém um conjunto de mapeamentos (*Mapeamentos*); uma *Ontologia Exportada* e uma *Fonte de Dados*. Um mapeamento contém uma regra que mapeia uma entidade fonte para uma entidade alvo, tal que uma entidade pode ser um objeto ou um relacionamento. Uma fonte de dados contém uma identificação da fonte, que pode ser uma URI ou os dados materializados. A visão exportada também contém uma *Ontologia Fonte*, descrevendo sua fonte correspondente.

**(ii)Visão de Links Semânticos-E**. Um link semântico determina que duas instâncias em fontes distintas referenciam um mesmo objeto do mundo real. Cada visão de *link* semântico contém duas *Visões Exportadas-E*; uma classe, que deve estar presente em ambas as visões exportadas; um conjunto de propriedades, que devem pertencer à classe e, finalmente, uma *Regra de Correspondência*. Essa regra define critérios para criação do *link*, denotando a similaridade entre objetos das visões exportadas.

**(iii)Fusão-E**. A especificação de uma fusão é constituída por um conjunto de *FPAs*. Cada FPA contém uma classe, uma propriedade e uma função de fusão.

**(iv)Ontologia de Domínio**. Finalmente, a ontologia de domínio descreve os conceitos conciliados semanticamente das fontes.

#### 6.2.1.2 Visão de Aplicação

A aplicação a ser criada pelo o usuário é representada pela entidade *Visão de Aplicação-E*. Essa visão contém um conjunto de *Parâmetros*. Cada parâmetro é composto por uma *Ontologia de Aplicação* e um conjunto de *Filtros*, onde cada filtro é representado por uma classe, uma propriedade e uma condição. Essa ontologia representa os conceitos de interesse do usuário.

### 6.2.2 Diagrama de Fluxo

O funcionamento do mediador pode ser dividido em 4 etapas: (i) especificação de uma integração semântica; (ii) definição dos parâmetros do usuário; (iii) aplicação dos parâmetros na especificação e (iv) materialização do *mashup*. Há etapas que só serão executadas corretamente caso seja seguido uma sequência. Por exemplo, na etapa de integração semântica, os *links* semânticos não podem ser especificados antes das visões exportadas. Além disso, tais etapas também seguir um fluxo, por exemplo, não se pode materializar o *mashup*, etapa (iv), antes da etapa de integração semântica, (i).

## 6.3 Mediador Semântico: Implementação

Como discutido na Seção 5.2 do Capítulo 5, o principal processo do mediador é o de reescrita. Nesse processo, os parâmetros do usuário são aplicados na especificação de *mashup*, gerando uma nova especificação. Os principais algoritmos necessários nesse processo são descritos a seguir.

### 6.3.1 Reescrita da especificação

A reescrita de uma especificação é o processo em que o mediador combina os parâmetros do usuário com uma especificação já existente, gerando uma nova. Essa nova especificação é

materializada e resulta no *mashup* requisitado pelo usuário. O processo de reescrita pode ser dividido em 3 etapas: (i) interseção da ontologia de aplicação com as ontologias exportadas; (ii) adição dos filtros aos mapeamentos e (iii) definição das novas regras de fusão. Para cada etapa, foi desenvolvido um algoritmo baseado no modelo conceitual representado na Figura 32. O algoritmo 1 descreve o processo de reescrita de uma especificação.

**Algoritmo 1** Reescrita de Especificação

**Exige:** $\lambda_m$, especificação de uma integração semântica
**Exige:** $\delta$, parâmetro do usuário, na forma $\delta = (O_V, F)$
1: Inicialize a especificação $\lambda_v = \lambda_m$.
2: *Sejam $E_M$ um conjunto de visões exportadas em $\lambda_m$ e $E_V$ um conjunto de visões exportadas em $\lambda_v$*
3: **para todo** $E_{Mi} \in E_M$ **faça**
4: *Sejam $O_{Vi} \in E_{Vi}$ e $O_{Mi} \in E_{Mi}$:*
5: $O_{Vi} \leftarrow$ *devolveIntercecao*$(O_{Mi}, O_V)$
6: **fim para**
7: *aplicaFiltros*$(M_V, F)$
*Sejam $F_m$ e $F_v$ conjuntos de regras de fusão de $\lambda_m$ e $\lambda_v$, respectivamente*
8: $F_v \leftarrow$*removeRegrasFusao*$(E_V, F_m)$
9:
**devolve** $\lambda_v$

Nesse algoritmo, o usuário entra com dois parâmetros: (i) $\lambda_m$, especificação de uma integração semântica como definido no modelo conceitual (Figura 32) e (ii) $\delta$, parâmetros do usuário na forma $\delta = (O_V, F)$. Na linha 1, as ontologias exportadas da especificação do usuário ($\lambda_v$) serão definidas. Para isso, na linha 1 é feito uma chamada ao método *devolveIntersecao*. Esse método, como o nome propõe, retorna uma interseção entre as duas ontologias: ontologia exportada da i-ésima visão exportada de $\lambda_m$ e a ontologia de aplicação ($O_V$). O resultado retornado pela função será armazenado na ontologia exportada da i-ésima visão exportada da especificação do usuário ($\lambda_v$). Na linha 1, os filtros são aplicados nos mapeamentos da especificação do usuário $\lambda_v$. Para isso, é utilizado o método *aplicaFiltros*. Finalmente, as regras de fusão da nova especificação são definidas na linha 1. Os métodos *devolveIntersecao, aplicaFiltros* e *regrasFusao*, bem como suas etapas correspondentes são detalhados nas subseções seguintes.

#### 6.3.1.1 Interseção entre as Ontologias

Segundo definido formalmente no Capítulo 5, cada ontologia exportada da nova especificação deve ser uma interseção da ontologia de aplicação com a ontologia exportada da visão integrada. Par aisso, foi desenvolvido o algoritmo descrito a seguir.

**Algoritmo 2** Interseção de Ontologias

**Exige:** $O_1$ e $O_2$, duas ontologias
1: Inicialize a lista de declarações $A, BeC$ como vazias.
2: $A \leftarrow$*pegaDeclaracoes($O_1$)*
3: $B \leftarrow$*pegaDeclaracoes($O_2$)*
4: **para todo** $s \in A$ **faça**
5: **se** $s \in B$ **então**
6: $C \leftarrow s$
7: **fim se**
8: **fim paradevolve** $C$
**Garante:** $C \equiv A \cap B$

Para esse algoritmo, é definido que uma ontologia pode ser representada por um conjunto de declarações (*statements*, do inglês). Uma declaração é toda e qualquer afirmação feita por uma ontologia, e.g. *ex:Gabriel rdf:type ex:Pessoa* e *ex:propriedade rdf:type rdfs:Property*, desde definições de propriedades até axiomas. Sendo assim, os dois parâmetros de entrada são as ontologias $O_1$ e $O_2$. Nas linhas 2 e 2 é utilizado um método genérico chamado *pegaDeclaracoes*, responsável por transformar uma ontologia em uma lista de declarações. Dito isso, a linha 2 itera sobre as declarações da ontologia $O_1$. Caso essa declaração exista na ontologia $O_2$ (linha 2), essa declaração será armazenada em $C$ (linha 2).

#### 6.3.1.2 Adição dos Filtros aos Mapeamentos

No algoritmo a seguir, os filtros passados por parâmetro pelo usuário são adicionados aos mapeamentos das visões exportadas do *mashup*.

**Algoritmo 3** Adiciona Filtros aos Mapeamentos

**Exige:** M, um conjunto de mapeamentos definidos na forma $C_S(\phi) \rightarrow C_T$
**Exige:** F, conjunto de filtros definidos na forma $C/P\ \mu$
1: **para todo** $m \in M$ **faça**
2: **para todo** $f \in F$ **faça**
3: **se** $C_T \in m \equiv C \in f$ **então**
4: *concatenarCondicoes* $\phi$ de $C_S \in m$ com $\mu \in f$
5: **fim se**
6: **fim para**
7: **fim paradevolve** $M$

Esse algoritmo recebe um conjunto de mapeamentos, onde cada mapeamento contém uma classe fonte ($C_S$); uma condição($\phi$) e uma classe alvo($C_T$). O outro parâmetro de entrada é um conjunto de filtros, denotado por F, onde cada filtro contém uma classe ($C$), uma propriedade ($P$) e uma condição $\mu$. Para verificar se determinado mapeamento deve receber um filtro, o algoritmo faz uma iteração $nxm$, onde n e m representam os tamanhos dos conjuntos $M$ e $F$ respectivamente (linhas 3 e 3). Na linha 3 é realizada uma

comparação entre a classe alvo ($C_T$) de um mapeamento $m$ com a classe $C$ de cada filtro $f \in F$. Caso a equivalência seja *verdadeira*, a condição $\mu$ do filtro $f$, pertencente a $F$, deve ser concatenada com a condição $\phi$ do mapeamento $m \in M$.

#### 6.3.1.3 Novas Regras de Fusão

Nessa etapa, as visões exportadas da especificação do usuário já foram definidas. Agora, deve ser feito um processo para remover as regras de fusão que não tenham uma classe correspondente nas ontologias exportadas. Esse processo é descrito no seguinte algoritmo.

**Algoritmo 4** Remove Regras de Fusão

**Exige:** E, conjunto de visões exportadas
**Exige:** F, conjunto de regras de fusão, na forma $FPA : P[C] \equiv f/Q$.
*Inicialize um conjunto de regras de fusão* $F' = 0$
1: **para todo** $E_i \in E$ **faça**
2: *Seja* $O_e$ *uma ontologia exportada de* $E_i$
3: *Seja* $C(O_e)$ *o conjunto de todas as classes da ontologia* $O_e$
4: **para todo** $c \in C(O_e)$ **faça**
5: **para todo** $f \in F$ **faça**
6: **se** $c \equiv C \in f$ **então**
7: $F' \leftarrow f$
8: **fim se**
9: **fim para**
10: **fim para**
11: **fim para**
**devolve** $F'$

Para esse algoritmo, os parâmetros de entrada são: E, um conjunto de visões exportadas e F, um conjunto de regras de fusão, tal que cada regra $f \in F$ é descrita na forma $FPA : P[C] \equiv f/Q$. Na linha 4, uma lista $F'$ de regras de fusão é inicializada como vazia. Essa lista irá conter apenas as regras que contém uma classe nas visões exportadas correspondentes. Na linha 4, todas as visões do parâmetro passado são verificadas. Cada visão exportada contém uma ontologia exportada, que, por sua vez, contém um conjunto de classes. A verificação de cada classe de cada ontologia exportada é realizada na linha 4. Na linha 4, verifica-se se alguma classe de uma ontologia exportada faz parte de alguma regra de fusão. Em caso verdadeiro, essa regra de fusão tem uma classe correspondente em alguma visão exportada e, portanto, deve ser adicionada em $F'$ (linha 4).

## 6.4 Mediador Semântico: Protótipo

Nessa Seção, são discutidos os detalhes técnicos da implementação do protótipo: as tecnologias e padrões utilizados e os desafios encontrados.

### 6.4.1 Tecnologias

Durante a implementação, as principais decisões de projetos foram:

- Qual API RDF/OWL utilizar?
- Qual linguagem implementar?
- Padrões de projetos a serem utilizados?

Tais questões são discutidas nas subseções a seguir.

#### 6.4.1.1 API RDF e OWL

Existem vários *frameworks* para auxiliar no desenvolvimento com ontologias e arquivos RDF. Dentre os mais conhecidos, podem ser citados Jena e OWL API. As características de cada tecnologia são discutidas no Capítulo 2, na subseção 2.3.6.

Ambas APIs poderiam ser utilizadas pela implementação proposta. Durante uma análise sobre as APIs, ficou claro que a OWL API tem um foco para manipulação de ontologias: inferências e definição de axiomas. Enquanto o Jena é um manipulador de RDF com uma *interface* bem simples e entendível. Por praticidade, a API utilizada nesta implementação foi o Jena.

Porém, um dos problemas encontrados durante o desenvolvimento é que o Jena, dependendo da quantidade de operações, pode apresentar tempos de respostas inviáveis, caso o modo depuração esteja ativado.

#### 6.4.1.2 Linguagem

Scala é uma das principais opções em implementações na Web Semântica. O LDIF, por exemplo, utiliza Scala na maior parte de seu projeto. Segundo os autores, o Scala, como a linguagem propõe, por ser altamente escalável em ambientes distribuídos, foi a escolha ideal para o projeto LDIF.

Porém, o Java também é bastante utilizado em implementações para Web Semântica, sendo utilizado em diversas APIs, e.g. Jena, OWL Api, Fuseki. Para a prova de conceito do Mediador, foi utilizado o Java pelos seguintes motivos:

- Suporte. Diversas APIs da Web Semântica dão suporte a Java, como: Jena, OWL API, Fuseki, Virtuoso, dentre outros;
- Facilidade. O Java é bem dinfundindo na *Web* e possui grande suporte de diversas IDEs.

#### 6.4.1.3 Padrões de Projeto

Esse protótipo foi desenvolvido de forma a ser *extensível* ao invés de *modificável*. Esse princípio assegura que o código do *software* não deve ser alterado em caso de, por exemplo,

uma API de manipulação de Ontologias seja modificada. Para isso, foi utilizado o conceito de *Independência de API*. A Figura 33 representa como essa abordagem foi utilizada no protótipo para a API RDF ser abstraída.

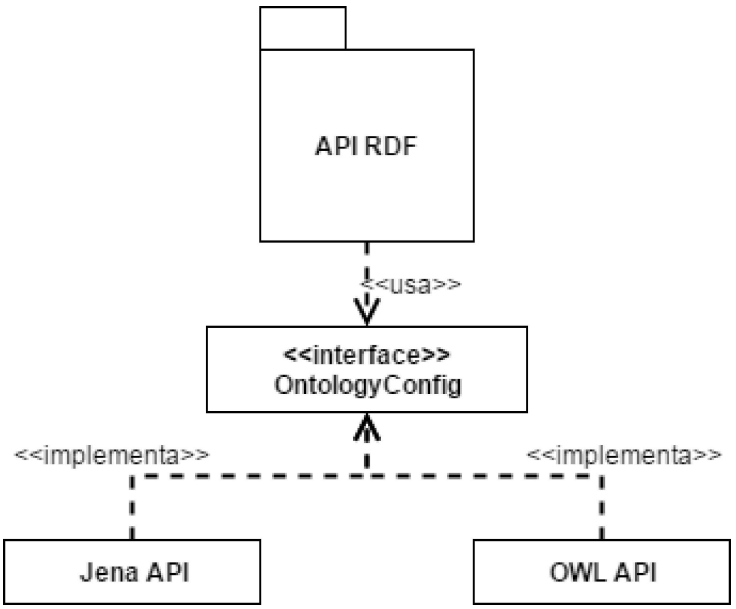



Figura 33 – Exemplo de independência de API

### 6.4.2 Visão Integrada - Mashup

Para o desenvolvimento da visão integrada (*mashup*) foram desenvolvidos diversos módulos. Os dois principais são *"core"* e *"mediador_semântico"*. O módulo *mediador_semântico* contém cinco módulos: *visão_integrada, visões_exportadas, visões_links* e *onto_domínio*. Cada um desses módulos representa uma entidade do modelo conceitual (Figura 32). Já o módulo *core* contém as principais entidades que são comuns aos demais módulos.

Nessa implementação, uma especificação de *mashup* é definida por um conjunto de diretórios e um arquivo XML. Nesse arquivo, são especificados os diretórios de cada etapa da especificação: visões exportadas, ontologia de domínio, *links* semânticos e fusão. Além disso, também é definido no arquivo XML, o diretório que contém as visões de aplicações do usuário. Um arquivo exemplo de configuração de uma visão integrada pode ser visto a seguir.

```
<SemanticMediator>

   <IntegratedView dir = "semantic_integrated_view_dir">
      <name>Mediador</name>
      <DomainOntology>domain_onto</DomainOntology>
      <ExportedViews>exported_views</ExportedViews>
      <LinksetViews>linkset_views</LinksetViews>
```

```
      <FusionRules>fusion_rules</FusionRules>
   </IntegratedView>

   <AppViews dir = "app_views">
      <AppView>
         <name>esus_sinasc</name>
         <AppOntology>app_ontology</AppOntology>
         <Filters>filters</Filters>
      </AppView>
   </AppViews>

</SemanticMediator>
```

Cada um dos diretórios definidos nesse arquivo XML contém suas próprias regras de especificação, definidas também na forma de arquivos. Cada diretório é brevemente discutido a seguir.

**Diretório de Visões Exportadas**. Esse diretório contém um conjunto de diretórios, onde cada diretório corresponde a uma visão exportada. Em cada diretório, são armazenados dois arquivos: um arquivo de mapeamentos R2RML e um arquivo no formato *Turtle*, descrevendo a ontologia fonte.

**Diretório de Links Semânticos**. Nesse diretório, as heurísticas para descoberta dos *links* são especificadas em arquivos XML. Para cada duas visões exportadas, um arquivo XML deve ser criado. Nesse arquivo, são definidos os diretórios de cada visão exportada; a classe. Um exemplo do arquivo de configuração XML pode ser visto abaixo.

```
<LinksetView>
   <Prefixes>
      <Prefix key="gissa" uri="https://www.atlantico.com.br#"/>
      <Prefix key="owl" uri="http://www.w3.org/2002/07/owl#" />
   </Prefixes>
    <Linksets>
        <Linkset id="persons">
            <LinkType>owl:sameAs</LinkType>

            <ExportedViews>
                <ExportedView>e-sus</ExportedView>
                <ExportedView>sinasc</ExportedView>
            </ExportedViews>

            <MatchPredicate metric="3-gram">
                <Class>gissa:Pessoa</Class>
```

```
            <Property>gissa:nomeCompleto</Property>
        </MatchPredicate>

    </Linkset>
  </Linksets>
</LinksetView>
```

**Diretório Fusão**. A fusão, assim como na especificação, é definida com o auxílio de FPAs (Cáp. 4, subseção 4.2.5). As FPAs são descritas num arquivo .fpa e são definidas como discutido no Cáp. 4, Seção 4.3.

**Diretório Ontologia de Domínio**. Nesse diretório, a ontologia de domínio é descrita num arquivo OWL. O trecho a seguir, mostra uma declaração pertencente a ontologia *gissa.owl*. Essa declaração define a propriedade *gissa:temMalFormacao*.

```
gissa:temMalFormacao rdf:type owl:ObjectProperty ;

   rdfs:domain gissa:Nascimento;

     rdfs:range gissa:MalFormacao;

     rdfs:label "Nascimento com mal formacao"@pt .
```

### 6.4.3 Visão de Aplicação

No protótipo desenvolvido, o usuário define os parâmetros por meio de arquivos. É importante notar que essa é apenas uma das possíveis implementações do mediador descrito no Capítulo 5. Esse protótipo pode ser estendido e dar suporte à *interfaces* gráficas, possibilitando que os usuários definam seus parâmetros de forma fácil e intuitiva.

Cada visão de aplicação é definida por dois arquivo XML: uma ontologia de aplicação e um conjunto de filtros. A materialização da visão de aplicação ocorre no diretório corrente. A ontologia de aplicação é definida, assim como a ontologia de domínio, num arquivo OWL. Um exemplo de filtro é mostrado a seguir.

```
FILTER = gissa:Pessoa/idadeReal > 20
```

### 6.4.4 Reescrita de Especificação

Para a implementação do Algoritmo 2, foi utilizado uma facilidade do Jena API. Essa API permite listar todas as declarações de uma ontologia. Assim, facilmente as declarações de duas ontologias puderam ser comparadas.

Nesse protótipo, os mapeamentos são definidos utilizando a linguagem R2RML. Como abordado no Capítulo 2, mapeamentos R2RML são descritos na forma de *TripleMap*. Um dos campos de um *TripleMap* é o campo *LogicalTable*, que contém uma consulta SQL e identifica quais dados serão buscados no banco relacional. Assim, o Algoritmo 3 foi implementado adicionando a cláusula *WHERE* na consulta SQL de cada *TripleMap* em que o *SubjectMap* possua um sujeito correspondente no filtro.

Por fim, para a implementação do Algoritmo 4, foi realizada uma varredura sobre todas as FPAs contidas no arquivo de configuração das regras de fusão. Cada FPA é lida como um *string*, é quebrada (*split*) e tem seus objetos armazenados na forma de *propriedade / Classe = Função*. Assim, foi feita uma verificação sobre a entidade *"Classe"* de cada FPA.

### 6.4.5 Materialização

A materialização de cada etapa da especificação é um processo complexo e está fora do escopo dessa dissertação. Por exemplo, para materializar os mapeamentos é necessária uma *engine* que os processe e que seja capaz de gerar o grafo RDF. Já na materialização dos *links* semânticos, é necessário um processo de mineração de dados que identifique a similaridade das entidades. Por isso, foram utilizadas ferramentas do *Linked Data Integration Framework* (LDIF) nessa etapa. Assim, a materialização deste protótipo gera arquivos de configuração para as ferramentas do LDIF, como SILK e SIEVE.

## 6.5 Conclusão

Nesse Capítulo foi descrito um guia de como o *framework* Mediador Semântico deve ser implementado. Para isso, foram desenvolvidos um modelo conceitual e algoritmos, descrevendo as principais funcionalidades do *framework*. Além disso, também foi apresentado um protótipo do mediador. Durante a etapa de desenvolvimento desse protótipo, foram encontrados diversos desafios e decisões de projetos, ambas discutidas nesse Capítulo. Esse protótipo é uma primeira versão que tem como objetivo demonstrar a aplicabilidade da abordagem. Nas etapas seguintes, será estendido, dando suporte à *interfaces* gráficas, que facilitará o uso pelo usuário, bem como suportará à requisição de serviços pela *web* e comunicação com fontes da *Linked Open Data*.

# 7 Conclusão

## 7.1 Considerações Finais

Essa dissertação apresentou o MAURA, um *framework* para construção eficiente de *Linked Data Mashups* que possibilita usuários de propósito geral construirem seus próprios *mashups*, sem a necessidade de conhecimentos específicos em Web Semântica ou integração de dados.

O MAURA cria o conceito de reutilização de especificações de *Linked Data Mashups*, o que permite construir *mashups* de forma automática em fontes de dados com modelos identicos, porém com dados distintos. Por exemplo, as fontes de dados utilizadas nesta dissertação, SINASC e e-SUS, embora sejam de Tauá, utilizam um modelo de dados padronizados em todo Brasil. Assim, um gestor de qualquer município brasileiro pode criar um *Linked Data Mashup* de forma automática utilizando o MAURA.

O estudo de caso sobre análise dos fatores de riscos de óbitos-infantis e partos prematuros, desenvolvido nessa dissertação, é aplicado ao sistema GISSA, um projeto de pesquisa e de desenvolvimento, cujo objetivo é auxiliar os diversos atores da área de saúde nos diversos processos de tomadas de decisão envolvidos no contexto do programa Rede Cegonha do Ministério da Saúde - MS. O trabalho desta dissertação auxiliou o sistema GISSA a resolver o desafio da heterogeneidade semântica, haja vista que a heterogeneidade sintática tem sido tratada pelo Departamento de Informática do MS (DATASUS) com tecnologias clássicas (e.g. barramento SOA).

Do ponto de vista científico, o conceito de reutilização de especificações *Linked Data Mashups*, criado nesta dissertação, ambiciona impulsionar estudos, pois permite grupos distintos reutilizem integrações previamente criadas e foquem seus esforços em outras áreas. Nesse contexto, o MAURA pode representar o primeiro passo na construção de um portal *web*, onde grupos de pesquisa poderão incrementar *mashups* já existentes, enquanto grupos de desenvolvimento poderão utilizar tais *mashups* para criar aplicações cada vez mais ricas em conteúdo. Um primeiro passo para a *Web Integrada de Dados*.

Do ponto de vista de desenvolvimento, a expectativa é que o GISSA passe a consumir dados integrados disponibilizados pelo MAURA. Em assim procedendo, espera-se que os resultados hoje obtidos pelo GISSA possam ser refinados pela abordagem proposta nesta dissertação, inviáveis com seus mecanismos de *data warehouses*. A abordagem desta dissertação vai além do contexto da Rede Cegonha, podendo atingir outras áreas do conhecimento.

Finalmente, é muito estimulante perceber que esse trabalho vai ao encontro de uma

problemática real da sociedade, permitindo contribuir com uma das melhores formas que um sistema computacional pode fazer: salvar vidas.

## 7.2 Trabalhos Futuros

Os principais trabalhos futuros a serem listados são:

- Criar uma fonte de dados abertos na *Linked Open Data* (LOD) para depósito e consulta de especificações de *Linked Data Mashups*.
- Extender o protótipo do mediador para uma plataforma *web*, onde usuários poderão criar *mashups* utilizando especificações depositadas na LOD;
- Permitir uma abordagem orientada à serviços, onde um cliente pode requisitar por dados integrados;
- Incorporar o mediador como módulo de inteligencia do GISSA, permitindo análises integradas sobre riscos em óbitos-infantis, partos prematuros, dentre outros;
- Utilizar fontes abertas da *Linked Open Data*, e.g. DBPedia, para enriquecer o *Linked Data Mashup* criado nesta dissertação. Além disso, incorporar o Sistema de Informações sobre Mortalidade (SIM) ao *mashup*, tornando possível analisar, também, os casos de óbitos maternos.

# Referências

AGHAEI, S.; ALI, M.; KHOSRAVI, H. Evolution of the world wide web: From web 1.0 to web 4.0. *International Journal of Web & Semantic Technology*, v. 3, n. 1, jan 2012. Citado na página 25.

AN, Y.; BORGIDA, A.; MYLOPOULOS, J. Discovering the semantics of relational tables through mappings. *Journal of Data Semantics*, v. 7, p. 1–32, 2006. Citado na página 43.

BASILI, V. R.; CALDIERA, G.; ROMBACH, H. D. The goal question metric approach. In: *Encyclopedia of Software Engineering*. [S.l.]: Wiley, 1994. Citado na página 50.

BATINI, C.; LENZERINI, M.; NAVATHE, S. B. A comparative analysis of methodologies for database schema integration. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 18, n. 4, p. 323–364, dez. 1986. ISSN 0360-0300. Disponível em: <http://doi.acm.org/10.1145/27633.27634>. Citado na página 37.

BELLEAU, F. et al. Bio2rdf: Towards a mashup to build bioinformatics knowledge systems. *J. of Biomedical Informatics*, Elsevier Science, San Diego, USA, v. 41, n. 5, p. 706–716, out. 2008. ISSN 1532-0464. Disponível em: <http://dx.doi.org/10.1016/j.jbi.2008.03.004>. Citado na página 47.

BERNERS-LEE, T. et al. World-wide web: The information universe. *Electronic Networking: Research, Applications and Policy*, v. 1, n. 2, p. 74–82, 1992. Citado na página 22.

BERNERS-LEE, T. et al. The semantic web. *Scientific american*, New York, NY, USA:, v. 284, n. 5, p. 28–37, 2001. Citado na página 25.

BIZER, C. D2rq - treating non-rdf databases as virtual rdf graphs. In: *In Proceedings of the 3rd International Semantic Web Conference (ISWC2004*. [S.l.: s.n.], 2004. Citado na página 60.

BIZER, C.; HEATH, T.; Berners-Lee, T. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, v. 5, n. 3, p. 1–22, 2009. Citado 2 vezes nas páginas 17 e 45.

BIZER, C. et al. Silk - a link discovery framework for the web of data. In: *18th International World Wide Web Conference*. [s.n.], 2009. Disponível em: <http://www2009.eprints.org/227/>. Citado 3 vezes nas páginas 59, 66 e 70.

BORAN, A. et al. A smart campus prototype for demonstrating the semantic integration of heterogeneous data. In: _____. *Web Reasoning and Rule Systems: 5th International Conference, RR 2011, Galway, Ireland, August 29-30, 2011. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 238–243. ISBN 978-3-642-23580-1. Disponível em: <http://dx.doi.org/10.1007/978-3-642-23580-1_18>. Citado na página 60.

BRAY, T. et al. *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. 2008. World Wide Web Consortium, Recommendation REC-xml-20081126. Citado na página 23.

BRICKLEY, D.; MILLER, L. *FOAF Vocabulary Specification 0.97*. [S.l.], 2010. Disponível em: <http://xmlns.com/foaf/spec/20100101.html>. Citado 2 vezes nas páginas 29 e 31.

CAROTHERS, G.; PRUD'HOMMEAUX, E. W3C Recommendation, *RDF 1.1 Turtle*. 2014. Disponível em: <http://www.w3.org/TR/2014/REC-turtle-20140225/>. Citado na página 29.

CARROLL, J. J. et al. Jena: Implementing the semantic web recommendations. In: *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers &Amp; Posters*. New York, NY, USA: ACM, 2004. (WWW Alt. '04), p. 74–83. ISBN 1-58113-912-8. Disponível em: <http://doi.acm.org/10.1145/1013367.1013381>. Citado na página 36.

CERI G. PELAGATTI, G. B. S. Structured methodology for designing static and dynamic aspects of data base applications. v. 6, p. 31–45, 1981. Citado na página 36.

CHAMPIN, P.-A. Rdf-rest: A unifying framework for web apis and linked data. In: VERBORGH, R. et al. (Ed.). *SALAD@ESWC*. CEUR-WS.org, 2013. (CEUR Workshop Proceedings, v. 1056), p. 10–19. Disponível em: <http://dblp.uni-trier.de/db/conf/esws/salad2013.html#Champin13>. Citado na página 59.

CHANDRASEKARAN, B.; JOSEPHSON, J. R.; BENJAMINS, V. R. What are ontologies, and why do we need them? *IEEE Intelligent Systems*, IEEE Computer Society, Los Alamitos, CA, USA, v. 14, n. 1, p. 20–26, 1999. ISSN 1094-7167. Citado na página 26.

CHAU, M.; CHEN, H. A machine learning approach to web page filtering using content and structure analysis. *Decis. Support Syst.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 44, n. 2, p. 482–494, jan 2008. ISSN 0167-9236. Disponível em: <http://dx.doi.org/10.1016/j.dss.2007.06.002>. Citado na página 25.

COLBY, L. S. et al. Algorithms for deferred view maintenance. In: *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*. New York, NY, USA: ACM, 1996. (SIGMOD '96), p. 469–480. ISBN 0-89791-794-4. Disponível em: <http://doi.acm.org/10.1145/233269.233364>. Citado na página 42.

D'AQUIN, M. et al. Characterizing knowledge on the semantic web with watson. In: GARCIA-CASTRO, R. et al. (Ed.). *EON*. CEUR-WS.org, 2007. (CEUR Workshop Proceedings, v. 329), p. 1–10. Disponível em: <http://dblp.uni-trier.de/db/conf/eon/eon2007.html#dAquinBGASM07>. Citado na página 59.

DB-ENGINES. *DB-Engines Ranking*. 2017. Acessado em 17/02/2017 às 14:51. Disponível em: <http://db-engines.com/en/ranking>. Citado na página 16.

DING, Y.; SUN, Y.; SINGHI, M. Muzk mesh: Interlinking semantic music data. *2010 IEEE/ACM International Conference on Web Intelligence-Intelligent Agent Technology (WI-IAT)*, IEEE Computer Society, Los Alamitos, CA, USA, v. 01, p. 699–702, 2010. Citado na página 47.

FIELDING, R. et al. *Hypertext Transfer Protocol – HTTP/1.1*. United States: RFC Editor, 1999. Citado na página 22.

FOX, P. et al. A volcano erupts: Semantically mediated integration of heterogeneous volcanic and atmospheric data. In: *Proceedings of the ACM First Workshop on CyberInfrastructure: Information Management in eScience*. New York, NY, USA: ACM, 2007. (CIMS '07), p. 1–6. ISBN 978-1-59593-831-2. Disponível em: <http://doi.acm.org/10.1145/1317353.1317355>. Citado 2 vezes nas páginas 59 e 61.

GISSA. *GISSA - Governança Inteligente de Sistemas de Saúde*. 2015. Disponível em: <http://www.taua.ce.gov.br/noticias/projeto-gissa-prefeitura-de-taua-apresenta-projeto-inovador-que-servira-como-referencia-na-rede-publ
Citado na página 18.

GRACIA, J.; D'AQUIN, M.; MENA, E. Large scale integration of senses for the semantic web. In: *Proceedings of the 18th International Conference on World Wide Web*. New York, NY, USA: ACM, 2009. (WWW '09), p. 611–620. ISBN 978-1-60558-487-4. Disponível em: <http://doi.acm.org/10.1145/1526709.1526792>. Citado 3 vezes nas páginas 59, 60 e 61.

GRANT, J.; BECKET, D. *RDF Test Cases - N-Triples*. 2004. Disponível em: <http://www.w3.org/TR/rdf-testcases>. Citado na página 29.

GRAY, A. J. et al. Applying linked data approaches to pharmacology: Architectural decisions and implementation. *Semantic Web Journal*, v. 5, n. 2, p. 101–113, 2012. Citado na página 47.

GREEN, J. et al. Creating a semantic integration system using spatial data. In: *Proceedings of the 2007 International Conference on Posters and Demonstrations - Volume 401*. Aachen, Germany, Germany: CEUR-WS.org, 2008. (ISWC-PD'08), p. 70–71. Disponível em: <http://dl.acm.org/citation.cfm?id=2889529.2889564>. Citado na página 60.

GRUBER, T. R. A translation approach to portable ontology specifications. *Knowl. Acquis.*, Academic Press Ltd., London, UK, UK, v. 5, n. 2, p. 199–220, jun. 1993. ISSN 1042-8143. Disponível em: <http://dx.doi.org/10.1006/knac.1993.1008>. Citado 2 vezes nas páginas 17 e 26.

GUPTA, A.; MUMICK, I. S. Materialized views. In: GUPTA, A.; MUMICK, I. S. (Ed.). Cambridge, MA, USA: MIT Press, 1999. cap. Maintenance of Materialized Views: Problems, Techniques, and Applications, p. 145–157. ISBN 0-262-57122-6. Disponível em: <http://dl.acm.org/citation.cfm?id=310709.310737>. Citado na página 42.

HAARSLEV, V.; MÖLLER, R. Racer system description. In: _____. *Automated Reasoning: First International Joint Conference, IJCAR 2001 Siena, Italy, June 18–22, 2001 Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001. p. 701–705. ISBN 978-3-540-45744-2. Disponível em: <http://dx.doi.org/10.1007/3-540-45744-5_59>. Citado na página 36.

HAMMER, J.; MCLEOD, D. An approach to resolving semantic heterogeneity in a federation of autonomous, heterogeneous database systems. *International Journal of Cooperative Information Systems*, v. 02, n. 01, p. 51–83, 1993. Citado 2 vezes nas páginas 39 e 63.

HARTH, A. et al. On-the-fly integration of static and dynamic linked data. In: *Proceedings of the Fourth International Conference on Consuming Linked Data - Volume 1034*.

Aachen, Germany, Germany: CEUR-WS.org, 2013. (COLD'13), p. 1–12. Disponível em: <http://dl.acm.org/citation.cfm?id=2874359.2874361>. Citado 3 vezes nas páginas 59, 61 e 62.

HEATH, T.; BIZER, C. *Linked Data: Evolving the Web into a Global Data Space*. 1st. ed. Morgan & Claypool, 2011. ISBN 9781608454303. Disponível em: <http://linkeddatabook.com/>. Citado 3 vezes nas páginas 17, 18 e 45.

HEFLIN, J. et al. An introduction to the owl web ontology language. *Lehigh University. National Science Foundation (NSF)*, 2007. Citado na página 33.

HOANG, H. H. et al. Semantic informatintegration with linked data mashups approaches. *International Journal of Distributed Sensor Networks*, Taylor & Francis, Inc., Bristol, PA, USA, v. 2015, p. 248:248–248:248, jan 2014. ISSN 1550-1329. Disponível em: <https://doi.org/10.1155/2015/431342>. Citado 3 vezes nas páginas 47, 50 e 51.

HORI, M.; EUZENAT, J.; PATEL-SCHNEIDER, P. *OWL Web Ontology Language*. [S.l.], 2004. Published online on June 11th, 2003 at <https://www.w3.org/TR/2003/NOTE-owl-xmlsyntax-20030611/>. Citado na página 32.

HORRIDGE, M.; BECHHOFER, S. The owl api: A java api for owl ontologies. *Semant. web*, IOS Press, Amsterdam, The Netherlands, The Netherlands, v. 2, n. 1, p. 11–21, jan. 2011. ISSN 1570-0844. Disponível em: <http://dl.acm.org/citation.cfm?id=2019470.2019471>. Citado na página 36.

HORROCKS, I. et al. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. [S.l.], 2004. Disponível em: <http://www.w3.org/Submission/SWRL>. Citado na página 60.

HULL, R. Managing semantic heterogeneity in databases: A theoretical prospective. In: *Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. New York, NY, USA: ACM, 1997. (PODS '97), p. 51–61. ISBN 0-89791-910-6. Disponível em: <http://doi.acm.org/10.1145/263661.263668>. Citado 2 vezes nas páginas 16 e 39.

HULL, R.; JACOBS, D. Language constructs for programming active databases. In: *Proceedings of the 17th International Conference on Very Large Data Bases*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1991. (VLDB '91), p. 455–467. ISBN 1-55860-150-3. Disponível em: <http://dl.acm.org/citation.cfm?id=645917.672162>. Citado na página 42.

INMON, W. H.; KELLEY, C. *Rdb - VMS: Developing a Data Warehouse*. New York, NY, USA: John Wiley & Sons, Inc., 1993. ISBN 0894354299. Citado na página 42.

JENTZSCH, A. et al. Enabling Tailored Therapeutics with Linked Data. In: *Proceedings of the WWW2009 workshop on Linked Data on the Web (LDOW2009)*. [S.l.: s.n.], 2009. Citado na página 47.

JULA, A.; SUNDARARAJAN, E.; OTHMAN, Z. Review: Cloud computing service composition: A systematic literature review. *Expert Syst. Appl.*, Pergamon Press, Inc., Tarrytown, NY, USA, v. 41, n. 8, p. 3809–3824, jun 2014. ISSN 0957-4174. Disponível em: <http://dx.doi.org/10.1016/j.eswa.2013.12.017>. Citado na página 53.

KÄMPGEN, B. et al. Accepting the xbrl challenge with linked data for financial data integration. In: _____. *The Semantic Web: Trends and Challenges: 11th International Conference, ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014. Proceedings*. Cham: Springer International Publishing, 2014. p. 595–610. ISBN 978-3-319-07443-6. Disponível em: <http://dx.doi.org/10.1007/978-3-319-07443-6_40>. Citado 2 vezes nas páginas 60 e 61.

KIMBALL, R.; ROSS, M. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. 2nd. ed. New York, NY, USA: John Wiley & Sons, Inc., 2002. ISBN 0471200247, 9780471200246. Citado na página 43.

KITCHENHAM, B. *Procedures for Performing Systematic Reviews*. Department of Computer Science, Keele University, UK, 2004. Citado na página 49.

KITCHENHAM, B.; CHARTERS, S. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. [S.l.], 2007. Disponível em: <http://www.dur.ac.uk/ebse/resources/Systematic-reviews-5-8.pdf>. Citado na página 49.

KITCHENHAM, B.; MENDES, E.; TRAVASSOS, G. A systematic review of cross vs within company cost estimation studies. In: *Proceedings EASE 2006*. [S.l.]: BCS, 2006. p. 89–98. Citado na página 53.

KONDRAK, G. N-gram similarity and distance. In: *Proceedings of the 12th International Conference on String Processing and Information Retrieval*. Berlin, Heidelberg: Springer-Verlag, 2005. (SPIRE'05), p. 115–126. ISBN 3-540-29740-5, 978-3-540-29740-6. Disponível em: <http://dx.doi.org/10.1007/11575832_13>. Citado na página 69.

KOUKOURIKOS, A.; VOUROS, G. A.; KARKALETSIS, V. Towards enriching linked open data via open information extraction. In: *KNOW-A-LOD @ ESWC*. [S.l.: s.n.], 2012. Citado na página 47.

KOZáK, J. et al. Linked open data for healthcare professionals. In: *Proceedings of International Conference on Information Integration and Web-based Applications & Services*. New York, NY, USA: ACM, 2013. (IIWAS '13), p. 400:400–400:409. ISBN 978-1-4503-2113-6. Disponível em: <http://doi.acm.org/10.1145/2539150.2539195>. Citado na página 47.

LANGEGGER, A.; WöSS, W.; BLöCHL, M. A semantic web middleware for virtual data integration on the web. In: *Proceedings of the 5th European Semantic Web Conference on The Semantic Web: Research and Applications*. Berlin, Heidelberg: Springer-Verlag, 2008. (ESWC'08), p. 493–507. ISBN 3-540-68233-3, 978-3-540-68233-2. Disponível em: <http://dl.acm.org/citation.cfm?id=1789394.1789441>. Citado 2 vezes nas páginas 60 e 61.

LASSILA, O.; SWICK, R. R. *Resource Description Framework (RDF) Model and Syntax Specification*. [S.l.], 1999. Disponível em: <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>. Citado na página 27.

LE-PHUOC, D. et al. Rapid prototyping of semantic mash-ups through semantic web pipes. In: *Proceedings of the 18th International Conference on World Wide Web*. New York, NY, USA: ACM, 2009. (WWW '09), p. 581–590. ISBN 978-1-60558-487-4. Disponível em: <http://doi.acm.org/10.1145/1526709.1526788>. Citado 3 vezes nas páginas 56, 60 e 61.

LEE, T. B. *Realising the Full Potential of the Web*. 1997. Disponível em: <https://www.w3.org/1998/02/Potential.html>. Citado na página 24.

LEE, T. B. *The World Wide Web: A very short personal history*. 1998. Disponível em: <https://www.w3.org/People/Berners-Lee/ShortHistory.html>. Citado na página 22.

LEE, T. B. *Is your Linked Open Data 5 Star?* 2009. Disponível em: <https://www.w3.org/DesignIssues/LinkedData.html>. Citado na página 46.

LENZERINI, M. Data integration: A theoretical perspective. In: *Proceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*. New York, NY, USA: ACM, 2002. (PODS '02), p. 233–246. ISBN 1-58113-507-6. Disponível em: <http://doi.acm.org/10.1145/543613.543644>. Citado 3 vezes nas páginas 16, 36 e 38.

LOPES, G.; VIDAL, V.; OLIVEIRA, M. A framework for creation of linked data mashups: A case study on healthcare. In: *Proceedings of the 22Nd Brazilian Symposium on Multimedia and the Web*. New York, NY, USA: ACM, 2016. (Webmedia '16), p. 327–330. ISBN 978-1-4503-4512-5. Disponível em: <http://doi.acm.org/10.1145/2976796.2988213>. Citado 2 vezes nas páginas 50 e 57.

MAHDAVI-HEZAVEHI, S.; GALSTER, M.; AVGERIOU, P. Variability in quality attributes of service-based software systems: A systematic literature review. *Inf. Softw. Technol.*, Butterworth-Heinemann, Newton, MA, USA, v. 55, n. 2, p. 320–343, feb 2013. ISSN 0950-5849. Disponível em: <http://dx.doi.org/10.1016/j.infsof.2012.08.010>. Citado na página 53.

MALLETT, R. et al. The benefits and challenges of using systematic reviews in international development research. *Journal of Development Effectiveness*, v. 4, n. 3, p. 445–455, 2012. Disponível em: <http://EconPapers.repec.org/RePEc:taf:jdevef:v:4:y:2012:i:3:p:445-455>. Citado na página 49.

MANCHESTER, T. U. of. *OWL - List of Reasoners*. 2016. Disponível em: <http://owl.cs.manchester.ac.uk/tools/list-of-reasoners/>. Citado na página 32.

MANOLA, F.; MILLER, E. *RDF Primer*. [S.l.], 2004. Published online on February 10th, 2004 at <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>. Citado na página 17.

MATA, F.; PIMENTEL, A.; ZEPEDA, S. Integration of heterogeneous data models: A mashup for electronic commerce. In: *2010 IEEE Electronics, Robotics and Automotive Mechanics Conference*. [S.l.: s.n.], 2010. p. 40–44. Citado na página 47.

MCBRIDE, B. *RDF Vocabulary Description Language 1.0: RDF Schema*. 2004. (W3C Recommendation). Disponível em: <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>. Citado na página 30.

MCGUINNESS, D. L.; HARMELEN, F. van. *OWL Web Ontology Language Overview*. [S.l.], 2004. Citado na página 32.

MéDINI, L. et al. Towards semantic resource mashups. In: MALESHKOVA, M. et al. (Ed.). *SALAD@ESWC*. CEUR-WS.org, 2014. (CEUR Workshop Proceedings, v. 1165), p. 6–9. Disponível em: <http://dblp.uni-trier.de/db/conf/esws/salad2014.html#MediniCMC14>. Citado 2 vezes nas páginas 59 e 61.

MENDES, P. N.; MüHLEISEN, H.; BIZER, C. Sieve: Linked Data Quality Assessment and Fusion. In: *2nd International Workshop on Linked Web Data Management (LWDM 2012) at the 15th International Conference on Extending Database Technology, EDBT 2012.* [s.n.], 2012. p. to appear. Disponível em: <http://www.wiwiss.fu-berlin.de/en/institute/pwo/bizer/research/publications/Mendes-Muehleisen-Bizer-Sieve-LWDM2012.pdf>. Citado 2 vezes nas páginas 66 e 71.

NOY, N. F.; MUSEN, M. A. Prompt: Algorithm and tool for automated ontology merging and alignment. In: *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*. AAAI Press, 2000. p. 450–455. ISBN 0-262-51112-6. Disponível em: <http://dl.acm.org/citation.cfm?id=647288.721118>. Citado na página 57.

PARSIA, B.; SIRIN, E. *Pellet: An OWL DL Reasoner*. [S.l.], 2003. Citado na página 36.

PAULHEIM, H. Exploiting linked open data as background knowledge in data mining. In: *Proceedings of the International Workshop on Data Mining on Linked Data, with Linked Data Mining Challenge collocated with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD 2013), Prague, Czech Republic, September 23, 2013.* [s.n.], 2013. Disponível em: <http://ceur-ws.org/Vol-1082/extendedAbstract.pdf>. Citado na página 47.

PENG, Z. et al. Semantic-based mobile mashup platform. In: *Proceedings of the 2010 International Conference on Posters & Demonstrations Track - Volume 658*. Aachen, Germany, Germany: CEUR-WS.org, 2010. (ISWC-PD'10), p. 109–112. Disponível em: <http://dl.acm.org/citation.cfm?id=2878399.2878427>. Citado na página 59.

PIPINO, L. L.; LEE, Y. W.; WANG, R. Y. Data quality assessment. *Commun. ACM*, ACM, New York, NY, USA, v. 45, n. 4, p. 211–218, abr. 2002. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/505248.506010>. Citado na página 69.

PRUD'HOMMEAUX, E.; HARRIS, S.; SEABORNE, A. (Ed.). *SPARQL 1.1 Query Language*. [S.l.], 2013. Disponível em: <http://www.w3.org/TR/sparql11-query>. Citado na página 35.

PRUETT, M. *Yahoo! Pipes*. First. [S.l.]: O'Reilly, 2007. ISBN 9780596514532. Citado na página 60.

REITER, R. Towards a logical reconstruction of relational database theory. In: _____. *On Conceptual Modelling: Perspectives from Artificial Intelligence, Databases, and Programming Languages*. New York, NY: Springer New York, 1984. p. 191–238. ISBN 978-1-4612-5196-5. Disponível em: <http://dx.doi.org/10.1007/978-1-4612-5196-5_8>. Citado na página 39.

SCHULTZ, A. et al. Ldif : Linked data integration framework. In: . [s.n.], 2011. Disponível em: <http://iswc2011.semanticweb.org/fileadmin/iswc/Papers/PostersDemos/iswc11pd_submission_49.pdf>. Citado 4 vezes nas páginas 50, 59, 61 e 62.

SEVERAL. *Automated Generation of RDF Views over Relational Data Sources with Virtuoso*. 2009. Disponível em: <http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VOSSQL2RDF>. Citado na página 34.

SHEARER, R.; MOTIK, B.; HORROCKS, I. Hermit: A highly-efficient owl reasoner. In: DOLBEAR, C.; RUTTENBERG, A.; SATTLER, U. (Ed.). *OWLED*. CEUR-WS.org, 2008. (CEUR Workshop Proceedings, v. 432). Disponível em: <http://dblp.uni-trier.de/db/conf/owled/owled2008.html#ShearerMH08>. Citado na página 36.

SIRIN, E. et al. Pellet: A practical owl-dl reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, v. 5, n. 2, p. 51 – 53, 2007. ISSN 1570-8268. Software Engineering and the Semantic Web. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1570826807000169>. Citado na página 32.

SMITH, M. K.; WELTY, C.; MCGUINNESS, D. L. *OWL Web Ontology Language Guide*. [S.l.], 2004. Published online on February 10th, 2004 at <https://www.w3.org/TR/owl-guide/>. Citado na página 33.

TOUMA, R.; ROMERO, O.; JOVANOVIC, P. Supporting data integration tasks with semi-automatic ontology construction. In: *Proceedings of the ACM Eighteenth International Workshop on Data Warehousing and OLAP*. New York, NY, USA: ACM, 2015. (DOLAP '15), p. 89–98. ISBN 978-1-4503-3785-4. Disponível em: <http://doi.acm.org/10.1145/2811222.2811228>. Citado na página 57.

TRAN, T. N. et al. Linked data mashups: A review on technologies, applications and challenges. In: *Intelligent Information and Database Systems - 6th Asian Conference, ACIIDS 2014, Bangkok, Thailand, April 7-9, 2014, Proceedings, Part II*. [s.n.], 2014. p. 253–262. Disponível em: <http://dx.doi.org/10.1007/978-3-319-05458-2_27>. Citado na página 50.

TSARKOV, D.; HORROCKS, I. Fact++ description logic reasoner: System description. In: _____. *Automated Reasoning: Third International Joint Conference, IJCAR 2006, Seattle, WA, USA, August 17-20, 2006. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 292–297. ISBN 978-3-540-37188-5. Disponível em: <http://dx.doi.org/10.1007/11814771_26>. Citado na página 36.

ULLMAN, J. D. Information integration using logical views. *Theoretical Computer Science*, v. 239, n. 2, p. 189 – 210, 2000. ISSN 0304-3975. Disponível em: <//www.sciencedirect.com/science/article/pii/S0304397599002194>. Citado na página 39.

VIDAL, V. M. P. et al. Advanced information systems engineering: 27th international conference, caise 2015, stockholm, sweden, june 8-12, 2015, proceedings. In: _____. Cham: Springer International Publishing, 2015. cap. Specification and Incremental Maintenance of Linked Data Mashup Views, p. 214–229. ISBN 978-3-319-19069-3. Disponível em: <http://dx.doi.org/10.1007/978-3-319-19069-3_14>. Citado 6 vezes nas páginas 18, 50, 57, 63, 73 e 74.

VIDAL, V. M. P. et al. A semi-automatic approach for generating customized r2rml mappings. In: *Proceedings of the 29th Annual ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2014. (SAC '14), p. 316–322. ISBN 978-1-4503-2469-4. Disponível em: <http://doi.acm.org/10.1145/2554850.2554933>. Citado 2 vezes nas páginas 68 e 79.

W3C. *Semantic Web - XML2000, slide 10*. 2000. Disponível em: <https://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html>. Citado na página 26.

W3C. *Resource Description Framework (RDF)*. 2004. Disponível em: <https://www.w3.org/RDF/>. Citado 2 vezes nas páginas 27 e 45.

W3C. *R2RML RDB to RDF Mapping Language*. [S.l.], 2016. Available at https://www.w3.org/TR/r2rml/. Citado 2 vezes nas páginas 35 e 70.

WANG, T. D.; PARSIA, B.; HENDLER, J. A survey of the web ontology landscape. In: *Proc. of the ISWC 2006*. [s.n.], 2006. Disponível em: <http://www.mindswap.org/papers/2006/survey.pdf>. Citado na página 34.

WEGELER, T. et al. Evaluating the benefits of using domain-specific modeling languages: An experience report. In: *Proceedings of the 2013 ACM Workshop on Domain-specific Modeling*. New York, NY, USA: ACM, 2013. (DSM '13), p. 7–12. ISBN 978-1-4503-2600-1. Disponível em: <http://doi.acm.org/10.1145/2541928.2541930>. Citado na página 53.

WIDOM, J. Research problems in data warehousing. In: *Proceedings of the Fourth International Conference on Information and Knowledge Management*. New York, NY, USA: ACM, 1995. (CIKM '95), p. 25–30. ISBN 0-89791-812-6. Disponível em: <http://doi.acm.org/10.1145/221270.221319>. Citado na página 43.

WIEDERHOLD, G. Mediators in the architecture of future information systems. *Computer*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 25, n. 3, p. 38–49, mar. 1992. ISSN 0018-9162. Disponível em: <http://dx.doi.org/10.1109/2.121508>. Citado na página 41.

WISHART, D. S. et al. Drugbank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Research*, v. 36, n. Database-Issue, p. 901–906, 2008. Disponível em: <http://dblp.uni-trier.de/db/journals/nar/nar36.html#WishartKGCSTGH08>. Citado na página 47.

YUJIAN, L.; BO, L. A normalized levenshtein distance metric. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 29, n. 6, p. 1091–1095, jun. 2007. ISSN 0162-8828. Disponível em: <http://dx.doi.org/10.1109/TPAMI.2007.1078>. Citado na página 80.

ZHOU, G.; HULL, R.; KING, R. Generating data integration mediators that use materialization. *Journal of Intelligent Information Systems*, v. 6, n. 2, p. 199–221, 1996. ISSN 1573-7675. Disponível em: <http://dx.doi.org/10.1007/BF00122128>. Citado na página 42.

ZHOU, Z.; MASHUQ, M. *Web Content Extraction Through Machine Learning*. [S.l.], 2013. Citado na página 25.

ZHUGE, Y. et al. View maintenance in a warehousing environment. In: *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*. New York, NY, USA: ACM, 1995. (SIGMOD '95), p. 316–327. ISBN 0-89791-731-6. Disponível em: <http://doi.acm.org/10.1145/223784.223848>. Citado na página 42.

ZIEGLER, P.; DITTRICH, K. R. *Data Integration – Problems, Approaches, and Perspectives*. 2007. Citado 2 vezes nas páginas 37 e 44.

# Apêndices

# Anexos