UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE CIÊNCIAS

DEPARTAMENTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MATHEUS MAYRON LIMA DA CRUZ

UMA ABORDAGEM PARA CONSTRUÇÃO DE *MASHUPS* DE DADOS ESPECIFICADOS COMO UMA VISÃO SOBRE UM EKG

FORTALEZA

2021

MATHEUS MAYRON LIMA DA CRUZ

UMA ABORDAGEM PARA CONSTRUÇÃO DE *MASHUPS* DE DADOS ESPECIFICADOS COMO UMA VISÃO SOBRE UM EKG

Dissertação apresentada ao Curso de do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Orientadora: Profa. Dra. Vânia Maria Ponte Vidal

FORTALEZA

2021

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

C963a Cruz, Matheus Mayron Lima da.
Uma abordagem para construção de mashups de dados especificados como uma visão sobre um EKG / Matheus Mayron Lima da Cruz. – 2021.
60 f.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2021.
Orientação: Profa. Dra. Vânia Maria Ponte Vidal.

1. Mashup de Dados. 2. Integração Semântica. 3. Ontologias. I. Título.

CDD 005

MATHEUS MAYRON LIMA DA CRUZ

UMA ABORDAGEM PARA CONSTRUÇÃO DE *MASHUPS* DE DADOS ESPECIFICADOS COMO UMA VISÃO SOBRE UM EKG

Dissertação apresentada ao Curso de do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Aprovada em:

BANCA EXAMINADORA

Profa. Dra. Vânia Maria Ponte Vidal (Orientadora)
Universidade Federal do Ceará (UFC)

Prof. Dr. José Maria da Silva Monteiro Filho
Universidade Federal do Ceará (UFC)

Prof. Dr. Angelo Roncalli Alencar Brayner
Universidade Federal do Ceará (UFC)

Prof. Dr. José Gilvan Rodrigues Maia
Universidade Federal do Ceará (UFC)

A Deus, por me encher de coragem e esperança nos momentos de desespero. Aos meus pais, Ana e Carlos, pelo imenso amor dado a mim.

AGRADECIMENTOS

A Deus por sempre estar ao meu lado. Que Ele sempre me garanta a serenidade para aceitar o que não posso mudar, a coragem para mudar o que posso e a sabedoria necessária para distinguir.

Aos meus pais, Ana Célia e José Carlos, que sempre me deram amor, apoio e todos os meios necessários para meu desenvolvimento. Sem eles, eu certamente não estaria onde estou.

À minha namorada, Thaís, por me apoiar e por ser, muitas vezes, mais otimista do que eu. Seu otimismo foi fundamental em alguns momentos.

Aos meus queridíssimos colegas de laboratório que estiveram presentes e me acompanharam durante esses anos: Cristiano Melo, Narciso Arruda, Tiago Vinuto, Amanda Drielly, José Wellington, Caio Viktor, Túlio Vidal e tantos outros. Posso lhes dizer que o laboratório ARIDA foi, durante muito tempo, um segundo lar pra mim. Vocês fizeram parte do meu desenvolvimento não apenas como pesquisador, mas também como pessoa.

À professora Vânia Maria Ponte Vidal por me orientar e por sempre me incentivar na busca de uma solução diferente para os problemas. Aos professores José Maria da Silva Monteiro e Ângelo Roncalli Alencar Brayner pelos conselhos e considerações feitas durante o desenvolvimento deste trabalho.

Ao CNPq, por seu papel fundamental no incentivo a pesquisa brasileira. Eu não teria conseguido desenvolver minha pesquisa de mestrado sem o financiamento via bolsa de estudos. Peço ao CNPq que continue trabalhando neste sentido e que resista ao obscurantismo desses tempos.

"It is important to draw wisdom from many different places. If we take it from only one place, it becomes rigid and stale."

(Iroh)

RESUMO

Enterprise Knowledge Graphs (EKG) que aplicam as tecnologias da web semântica e princípios *linked data* têm se tornado soluções cada vez mais atraentes para empresas e organizações. Essa atenção se deve à capacidade desses *EKGs* em oferecer uma camada unificada e flexível sobre as fontes de dados. Nesse contexto, a camada semântica oferecida por um *EKG* pode ser aproveitada para dar suporte ao processo de construção de *mashups* de dados, que são visões materializadas construídas a partir da coleta, transformação e combinação de dados provenientes de diferentes fontes. Este trabalho apresenta uma abordagem para construção de *mashups* de dados especificados como uma visão sobre um *EKG*.

Palavras-chave: *mashup* de dados; integração semântica; ontologias.

ABSTRACT

Enterprise Knowledge Graphs (EKG) that apply Semantic Web technologies and linked data principles have become increasingly attractive solutions for companies and organizations. This attention is due to the ability of these EKGs to offer a unified and flexible layer over data sources. In this context, the semantic layer provided by an EKG can be used to support the process of building data mashups, which are materialized views built from the collection, transformation and combination of data from different sources . This work presents an approach for building data mashups as views over an EKG.

Keywords: data mashup; semantic integration; ontologies.

LISTA DE FIGURAS

LISTA DE ALGORITMOS

LISTA DE ABREVIATURAS E SIGLAS

EKG	*Enterprise Knowledge Graph*
ETL	*Extract-Transform-Load*
KG	*Knowledge Graph*
LD	*Linked Data*
OBDA	*Ontology-Based Data Access*
RDFS	*Resource Description Framework Schema*
RDF	*Resource Description Framework*
SPARQL	*SPARQL Protocol and RDF Query Language*
URI	*Universal Resource Identifier*
CNAE	Classificação Nacional de Atividades Econômicas
IBGE	Instituto Brasileiro de Geografia e Estatística
RFB	Receita Federal do Brasil
SIM	Sistema de Informação Sobre Mortalidade
SINASC	Sistema de Informações sobre Nascidos Vivos
SUS	Sistema Único de Saúde
VMD	Visão de Mashup de Dados

SUMÁRIO

1 INTRODUÇÃO

A aplicação de tecnologias da Web Semântica para realizar a integração de fontes de dados em ambientes de empresas e organizações tem sido um tópico de pesquisa relevante tanto na indústria quanto na academia. Nesse contexto, a utilização de *Enterprise Knowledge Graph*s (*EKG*s) que aplicam tecnologias da web semântica e adotam alguns dos princípios *Linked Data* em seu desenvolvimento tem se tornado soluções cada vez mais atrativas para fornecer um *dataspace* compreensivo para usuários e aplicações (GALKIN *et al.*, 2017).

De maneira geral, *EKG*s tem como principal objetivo prover uma camada de dados unificada, flexível e *human-friendly* sobre as fontes de dados importantes para uma organização, tornando transparente quaisquer desafios relacionados ao acesso ou a heterogeneidade dos dados presentes em diferentes fontes. Segundo Bishr (1998), a heterogeneidade dos dados pode ser sintática, esquemática ou semântica.

A heterogeneidade sintática ocorre quando as fontes de dados utilizam diferentes modelos para representar os dados. A heterogeneidade esquemática é decorrente de diferenças estruturais entre as fontes de dados. Por fim, a heterogeneidade semântica é resultante dos diferentes significados e interpretações dos dados em contextos diferentes. Para garantir a interoperabilidade de dados é necessário integrar semanticamente as fontes de dados.

Compreendemos como integração semântica o processo que utiliza uma representação conceitual dos dados e de seus relacionamentos para eliminar as possíveis heterogeneidades existentes entre os dados provenientes de diferentes fontes. Em um *EKG*, essa integração semântica é alcançada através da adoção de uma ontologia.

Ontologias são ideias para realizar esta representação conceitual, pois uma ontologia é, por definição, uma representação formal e explícita de uma conceitualização compartilhada (STUDER *et al.*, 1998). Nesse sentido, a ontologia atua como uma camada semântica em um *EKG*, enriquecendo as informações armazenadas nas fontes de dados com novas informações inferidas por meio de regras implementadas na própria ontologia.

Após ser implementado, um *EKG* são capazes de dar suporte a diversos *smart services*, como *chatbots*. Além desses serviços, a existência de um *EKG* também pode ser aproveitada para auxiliar no processo de criação de *mashups* de dados, que são visões integradas materializadas que possuem características interessantes para tarefas de análise de dados. Por integradas entende-se que essas visões combinam informações provenientes de diferentes fontes. Por materializadas entende-se que dados estão armazenados fisicamente em algum local.

1.1 Motivação

Em grandes organizações, com grandes volumes de dados heterogêneos, a eficácia da descoberta de conhecimento depende da construção de uma visão integrada contendo os dados relevantes para uma determinada aplicação, de forma a permitir uma melhor análise dos dados e, por conseguinte, uma melhor decisão. De acordo com Xiao *et al.* (2019), cientistas de dados gastam entre 80% a 95% do seu tempo selecionando e integrando os dados de forma *ad-hoc*. Além de ineficiente, uma integração *ad-hoc* pode introduzir sérios problemas de qualidade de dados e afetar análises realizadas sobre os dados integrados.

Segundo Vidal *et al.* (2015), a criação de uma visão integrada ou *mashup* de dados é uma tarefa complexa que envolve quatro desafios principais:

1. Seleção das fontes de dados relevantes para a aplicação;
2. Extração e tradução dos dados provenientes de fontes distintas e possivelmente heterogêneas para uma vocabulário comum;
3. Identificação de relacionamentos entre recursos de diferentes fontes;
4. Combinação e fusão de múltiplas representações de um mesmo objeto em uma representação concisa e unificada, e resolução das inconsistências existentes para melhorar a qualidade dos dados.

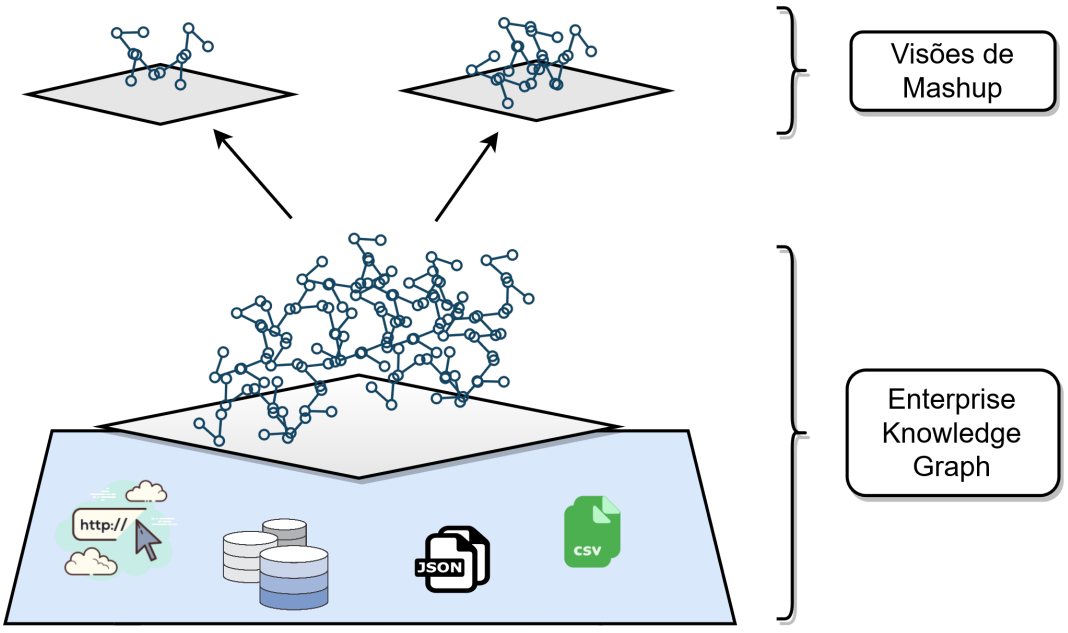
1.2 Descrição do Problema

Quando um *EKG* está disponível, os desafios associados à criação de um *mashup* de dados são um pouco diferentes:

- A seleção das fontes de dados relevantes em um *EKG* pode ser guiada pela busca de termos que fazem parte do vocabulário da ontologia de domínio;
- Problemas de heterogeneidade (sintática, esquemática e/ou semântica) existentes entre as fontes de dados que teriam de ser resolvidos por um processo de extração e tradução já foram resolvidos previamente durante a implementação do *EKG*;
- Os links que interligam os recursos existentes entre as fontes de dados já foram identificados e estão disponíveis para ser consultados no *EKG*.

Desta forma, podemos dizer que o principal desafio para construção de um *mashup* quando um *EKG* está disponível passa a ser como tirar proveito da infraestrutura oferecida por esse *EKG* para facilitar a construção de um *mashup* de dados. Na Figura 1, mostramos uma visão geral da abordagem proposta onde visões de *mashup* são definidas sobre um *EKG*.

Figura 1 – Visão geral da Abordagem Proposta



Fonte: O autor.

1.3 Contribuições

Neste trabalho, nós apresentamos uma abordagem semiautomática para construção de um *mashup* de dados como uma visão sobre um *EKG*. Nessa abordagem, o *EKG* é entendido como a implementação de uma visão semântica especificada através de um *framework* formal baseado em ontologias. Nesse *framework*, a visão semântica é formada por uma ontologia de domínio e conjunto de especificações de visões exportadas e de visões *linkset*.

A partir disso, nós formalizamos um processo onde o *mashup* é descrito como visão especificada através de consulta facetada. Essa consulta facetada é então utilizada na construção de um plano de consulta federada para recuperação dos sujeitos relevantes para a construção do *mashup*, e posteriormente, para recuperação das demais informações de cada um desses sujeitos.

Após recuperação dos dados, a construção do *mashup* é finalizada com a definição e aplicação das regras de fusão. Com exceção da especificação da consulta facetada que especifica a visão de *mashup* e a especificação das regras de fusão, o restante do processo pode ocorrer de forma automática.

Outra contribuição deste trabalho é o portal *SemanticSUS*. O *SemanticSUS* é um portal semântico criado para publicar os componentes da visão semântica construída sobre fontes de dados do SUS. A construção do *SemanticSUS* ocorreu no contexto dessa dissertação e foi fundamental para nos dar ideias sobre como estruturar o processo de construção de *mashup* a partir de uma visão semântica.

1.4 Organização da Dissertação

Essa dissertação tem 7 capítulos. O Capítulo 1 é este capítulo introdutório, os demais são:

- Capítulo 2 - Fundamentação Teórica - contém resumos dos assuntos que foram considerados relevantes para o compreensão dessa dissertação;
- Capítulo 3 - Trabalhos Relacionados - apresenta os trabalhos relacionados a construção de *mashups* de dados;
- Capítulo 4 - Especificando um *EKG* como uma visão semântica - apresenta uma abordagem onde o *EKG* é implementado a partir da definição de uma visão semântica;
- Capítulo 5 - Portal SemanticSUS - um portal semântico criado para publicação do resultado da integração semântica realizada sobre bases do SUS;
- Capítulo 6 - Processo para construção de um *mashup* sobre um *EKG* - propõe um processo semiautomático para construção de um *mashup* de dados, formalizando cada um dos passos realizados;
- Capítulo 7 - Conclusão e Trabalhos Futuros - conclui o trabalho e apresenta caminhos a serem seguidos a partir deste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, iremos apresentar alguns conceitos cujo entendimento consideramos essencial para o leitura dos capítulos seguintes, além de facilitar a compreensão do contexto em que esta dissertação está inserida.

2.1 Tecnologias da Web Semântica

Web Semântica e Linked Data

A Web Semântica é uma extensão da chamada Web tradicional, também referenciada como Web de Documentos. Essa nova web tem como proposta ser compreensível não apenas para humanos, como também para as máquinas, permitindo assim que humanos e computadores possam trabalhar de forma cooperativa (W3C, 2013) .

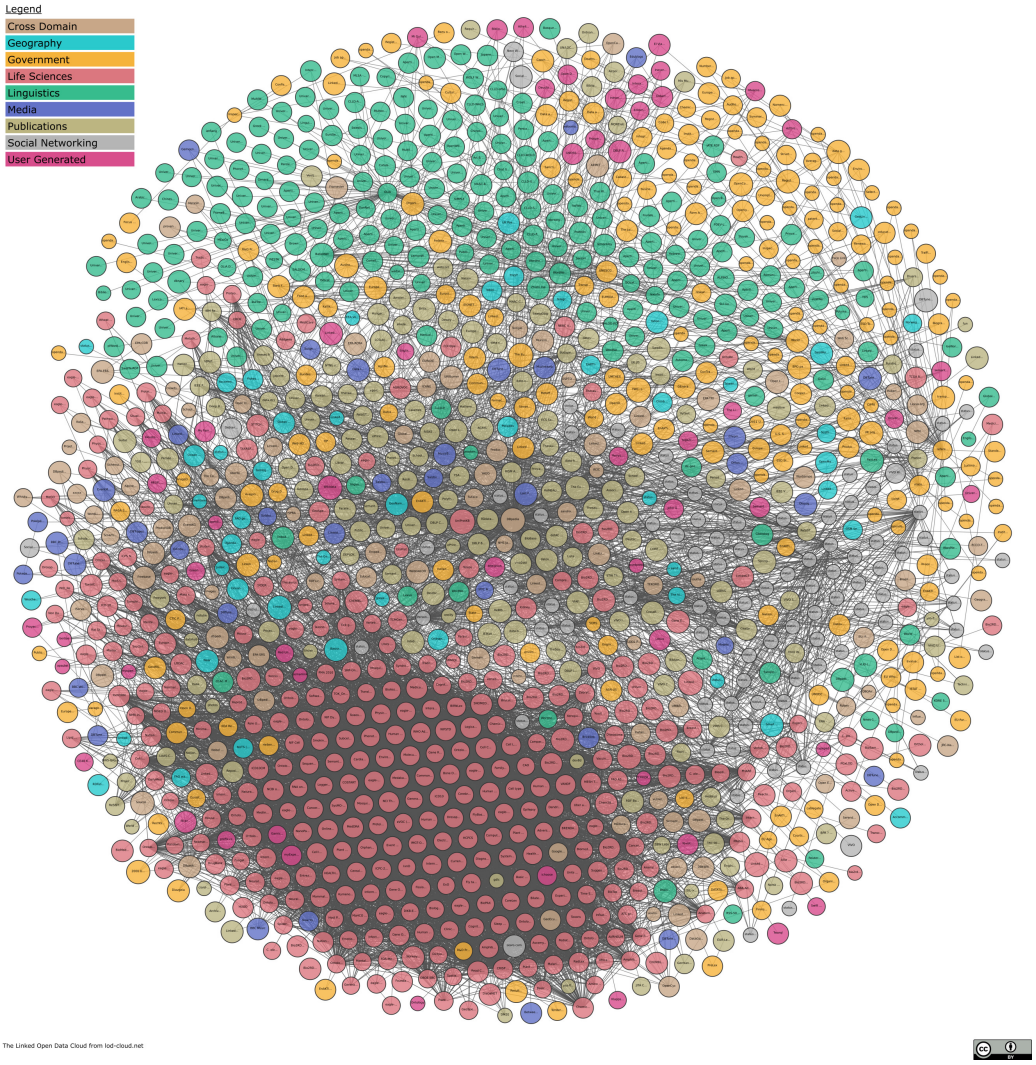
Nesse contexto nós temos *Linked Data* (*LD*): um conjuntos das melhores práticas para publicação e ligação de dados estruturados na Web. Essas melhores práticas foram introduzidas em Berners-Lee (2006) e são referenciados como os princípios *LD*. Esses princípios são:

1. Usar *Universal Resource Identifier* (*URI*) para nomear as "coisas", que podem ser objetos do mundo real ou conceitos abstratos;
2. Usar HTTP *URI* para permitir que as pessoas possam procurar o objeto desejado;
3. Quando alguém buscar por uma *URI*, prover informação útil;
4. Incluir links para outras *URI*s, permitindo que mais informação possa ser descoberta.

A ideia por trás desses princípios é permitir que os dados possam ser representados e acessados pela Web de forma padronizada, viabilizando também a ligação de diferentes fontes de dados por meio da utilização de *hyperlinks*.

De forma similar aos *hyperlinks* da web tradicional que ligam os documentos HTML em um espaço de informação global, os *hyperlinks* conectam os dados *Linked Data* existentes em um enorme grafo global. Atualmente, existe uma miríade de fontes de dados publicadas de forma aberta, utilizado os princípios *LD*. Várias dessas fontes podem ser encontradas na Nuvem *Linked Open Data* (Figura 2).

Figura 2 – Visão geral dos dados *Linked Open Data*



Fonte: https://lod-cloud.net/

Modelo RDF

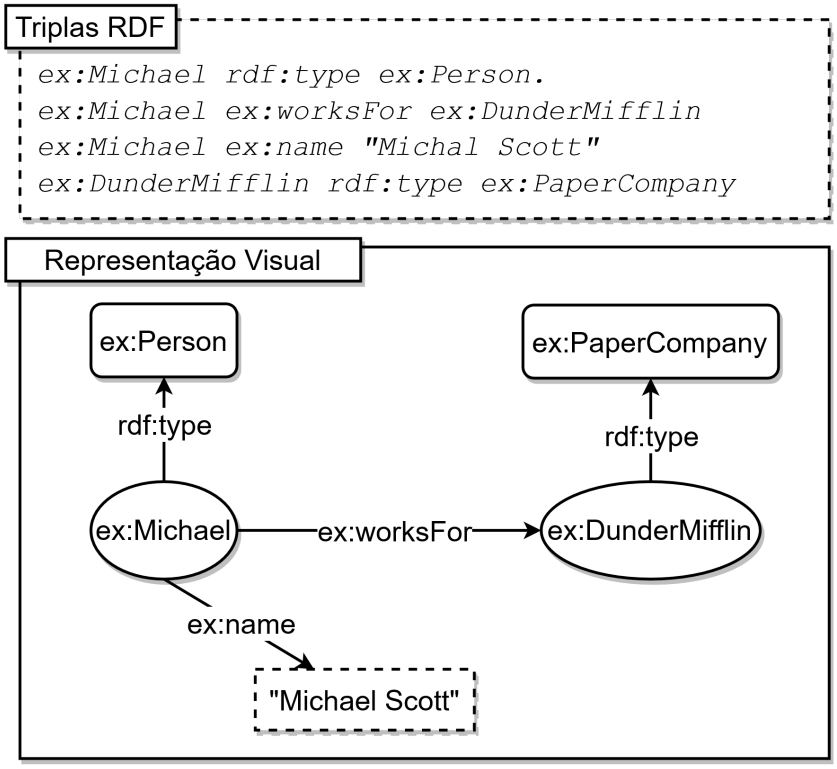
Resource Description Framework (*RDF*) (W3C, 2014a) é um modelo de dados baseado em grafos e sua utilização para publicação de dados na web é uma das recomendações W3C. A construção de um grafo *RDF* é feita a partir da utilização de triplas *RDF* compostas por: sujeito, predicado e objeto.

O sujeito de uma tripla representa o recurso sendo descrito, o predicado especifica uma propriedade que estabelece uma relação entre o sujeito e o objeto da tripla e, por fim, o objeto é o valor do predicado.

Definição 2.1.1 (Tripla *RDF*) *Seja U, B, L conjuntos disjuntos e infinitos de URIs, blank nodes, e literais, respectivamente. Uma tupla* $(s, p, o) \in (U \cup B) \times (U) \times (U \cup B \cup L)$ *denota uma tripla RDF, onde s é o sujeito, p é o predicado e o é o objeto. (ARENAS et al., 2009)*

Uma tripla *RDF* pode ser representada graficamente como um grafo direcionado, onde o sujeito e objeto representam os nós do grafo e o predicado é um arco nomeado que liga sujeito ao objeto. Na Figura 3, ilustramos um exemplo de grafo *RDF* com as representações em triplas e a representação visual do grafo formado por essas triplas.

Figura 3 – Exemplo de Grafo RDF



Fonte: O autor.

Ontologias e as linguagem RDFS

De acordo com Studer *et al.* (1998), uma ontologia é uma representação formal e explicita de uma conceitualização compartilhada. No campo da Web Semântica, a implementação de uma ontologia pode ser realizada utilizando a linguagem *Resource Description Framework Schema* (*RDFS*).

RDFS provê uma linguagem com a capacidade de adicionar semântica aos vocabulários criados pelo usuário e é uma das Recomendações W3C [1]. Essa linguagem amplia as capacidades expressivas do modelo *RDF*, permitindo que o significado dos objetos também possa ser descrito utilizando o mesmo modelo que descreve os dados. Com *RDFS* é possível realizar

[1] https://www.w3.org/wiki/RDFS

a modelagem de classes e propriedades especificando, por exemplo, relações de hierarquia entre classes e restrições sobre quais classes constituem domínio e o contradomínio de uma propriedade.

A linguagem SPARQL

SPARQL Protocol and RDF Query Language (*SPARQL*) é a recomendação W3C para consulta sobre fontes de dados *RDF* (W3C, 2014b). De forma similar à linguagem SQL para bancos de dados relacionas, a linguagem *SPARQL* permite a recuperação e manipulação de bases *RDF*.

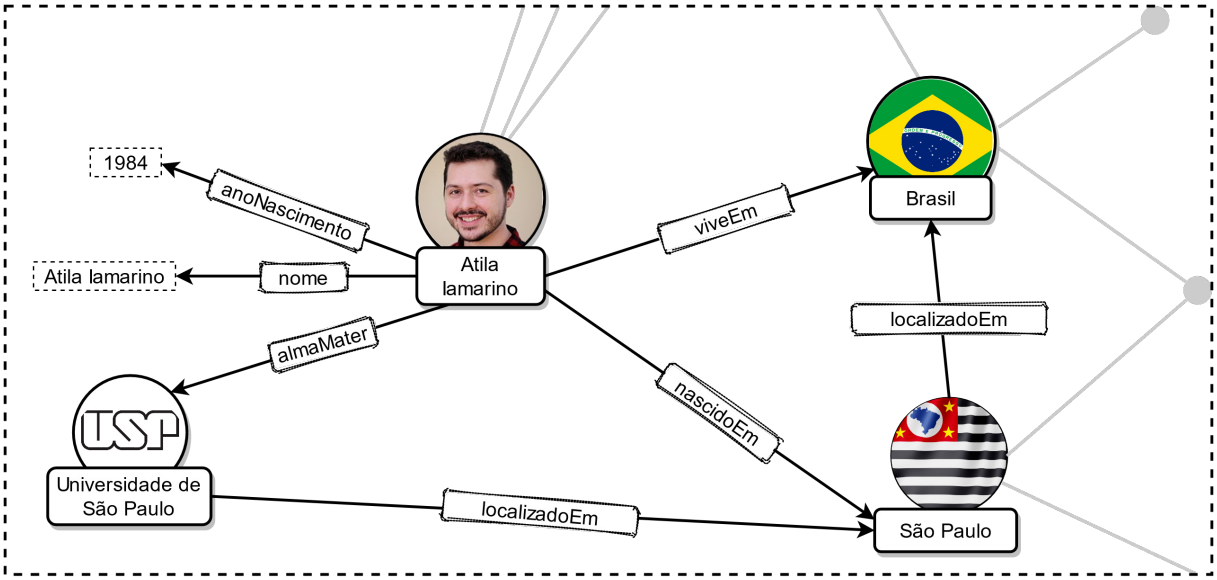
Uma consulta *SPARQL* consiste de um conjunto de padrões de tripla. Padrões de tripla são similares a triplas *RDF*, com a diferença que nos padrões de triplas sujeito, predicado e objeto podem ser variáveis. Em uma consulta, essas variáveis atuam como *placeholders* que são ligados a termos *RDF* para construção de uma solução. A linguagem *SPARQL* possui quatro formas de consulta:

- `SELECT`: Retorna os valores ligados a todas ou um subconjunto das variáveis nas soluções encontradas para consulta;
- `CONSTRUCT`: Retorna um grafo *RDF* construído através da substituição de variáveis em um conjunto de *templates* de tripla;
- `ASK`: Retorna um valor de verdadeiro ou falso indicando se um padrão de tripla possui algum resultado;
- `DESCRIBE`: Retorna um grafo *RDF* que descreve o recurso especificado pela consulta.

2.2 *Knowledge Graphs*

Em 2012, o termo *Knowledge Graph* (*KG*), ou grafo de conhecimento, foi popularizado pela empresa Google com a publicação em seu blog intitulada "*Introducing the Knowledge Graph: things, not strings*". Desde então, o conceito de *KG* começou a receber bastante atenção e a ser utilizado para se referir a uma miríade de coleções de dados baseadas em grafo. Apesar da popularidade do termo, não existe um consenso sobre qual deve ser a definição formal para grafo de conhecimento (EHRLINGER; WÖSS, 2016).

Figura 4 – Exemplo de Grafo de Conhecimento (KG)



Fonte: O autor.

Uma das definições existentes é apresentada por Paulheim (2016), que define que uma coleção de dados pode ser classificada como grafo de conhecimento se apresentar um conjunto mínimo de características. Essas características são descritas a seguir:

a) Descreve entidades do mundo real e seus relacionamentos, utilizando um grafo para isso;
b) Define possíveis classes e relacionamentos de entidades em um esquema;
c) Permite que as entidades se relacionem entre si de forma arbitrária;
d) Abrange vários domínios.

Essa definição de grafo de conhecimento não especifica qual modelo de dados deve ser utilizado na implementação de um *KG*. Entretanto, nesta dissertação, iremos nos focar em *KG*s construídos utilizando as tecnologias da Web Semântica e seguindo alguns do princípios *Linked Data*, i.e., grafos de conhecimento que adotam o modelo *RDF* como modelo de dados e

definem seu esquema utilizando a linguagem *RDFS*.

Enterprise Knowledge Graph

Um tipo especial de grafo de conhecimento são os grafos de conhecimento empresariais ou *EKG*s: grafos de conhecimentos construídos sobre as fontes de dados de uma empresa ou organização com a intenção de fornecer um acesso integrado às fontes de dados utilizadas por uma empresa. Segundo Hogan *et al.* (2020), os *EKG*s seguem algumas tendências:

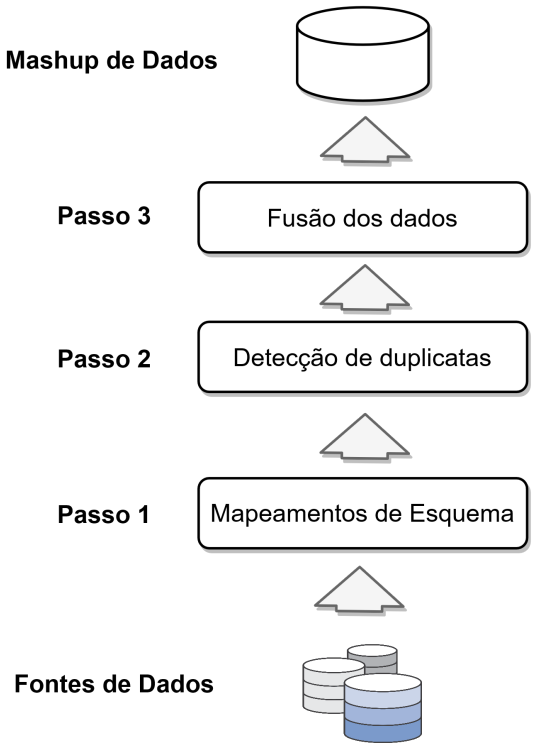
a) Os dados integrados pelo *EKG* costumam envolver dados heterogêneos e provenientes de fontes internas ou externas;
b) *EKG*s tendem a evolver uma grande quantidade de dados, com milhões de nós e arestas;
c) Normalmente, é necessário refinar o resultado *KG* inicial para melhorar a qualidade dos dados (e.g. adicionando links entre as fontes de dados, consolidados recursos duplicados);
d) Em geral, as ontologias utilizadas para representar os dados são *lightweight*.

Com esse tipo de integração, as empresas são capazes de prover recomendações melhores para os seus usuários, melhorar as suas ferramentas de buscas por informações, facilitar o processo de descoberta de conhecimento etc.

2.3 Mashup de Dados

Um *mashup* de dados é uma aplicação construída com o objetivo de coletar, transformar e combinar dados provenientes de diferentes fontes (VIDAL *et al.*, 2015). O processo de criação de uma *mashup* de dados é um processo de integração que tem como resultado uma visão integrada e materializada dos dados.

Figura 5 – Processo de criação de um *mashup* de dados



Fonte: Adaptado de Bleiholder e Naumann (2009)

De acordo com Bleiholder e Naumann (2009), o processo de construção de um *mashup* de dados pode ser dividido em três passos:

- **Passo 1:** No primeiro passo, são resolvidos os problemas de heterogeneidade sintática, de esquema e semântica. Isso pode ser obtido através da definição de um esquema para o *mashup* de dados, seguida pela especificação de mapeamentos que relacionam os termos do esquema das bases com termos do esquema adotado pelo *mashup* de dados;
- **Passo 2:** O segundo passo tem como objetivo realizar a detecção de duplicatas, i.e., a identificação de recursos representam o mesmo objeto no mundo real. Esse passo também pode ser referenciado na literatura como *record linkage*, *interlinking* ou *link discovery* (NENTWIG *et al.*, 2017). O resultado desse

processo é um identificador unificado capaz de descrever os indivíduos duplicados ou *links* que relacionam as diferentes representações. No modelo RDF, esses links normalmente são representados por meio da propriedade `owl:sameAs`;

- **Passo 3:** Nesse passo, é realizada a fusão dos dados. O processo de fusão consiste em combinar propriedades de dois ou mais recursos que representam o mesmo objeto no mundo real, com a finalidade de gerar uma representação unificada e concisa desse objeto. Nesse passo são resolvidos os conflitos nos valores das propriedades, além de possíveis irregularidades como a falta de valores, por exemplo.

Por se tratar de um enfoque materializado de integração, os dados coletados são fisicamente armazenados em um espaço separado de suas fontes originais. Por estarem centralizados, a execução de consultas sobre a visão integrada tendem a ser bastante eficientes. Além disso, esse processo também viabiliza que tarefas complexas como a limpeza e fusão dos dados possam ser feitas, garantindo, consequentemente, uma maior qualidade sobre os dados integrados.

Como desvantagem, esse enfoque apresenta a necessidade de um espaço em disco para armazenar os dados resultantes do processo de integração. Além disso, as abordagens que aplicam o enfoque materializado, em geral, seguem um processo *Extract-Transform-Load* (*ETL*) que precisa ser configurado pelo usuário. A configuração do processo *ETL* para criação de um *mashup* tende a exigir que o usuário possua não somente os conhecimentos técnicos para configurar as ferramentas utilizadas no processo de criação, como também um bom conhecimento sobre fontes de dados que devem ser utilizadas e como os dados provenientes dessas fontes se relacionam.

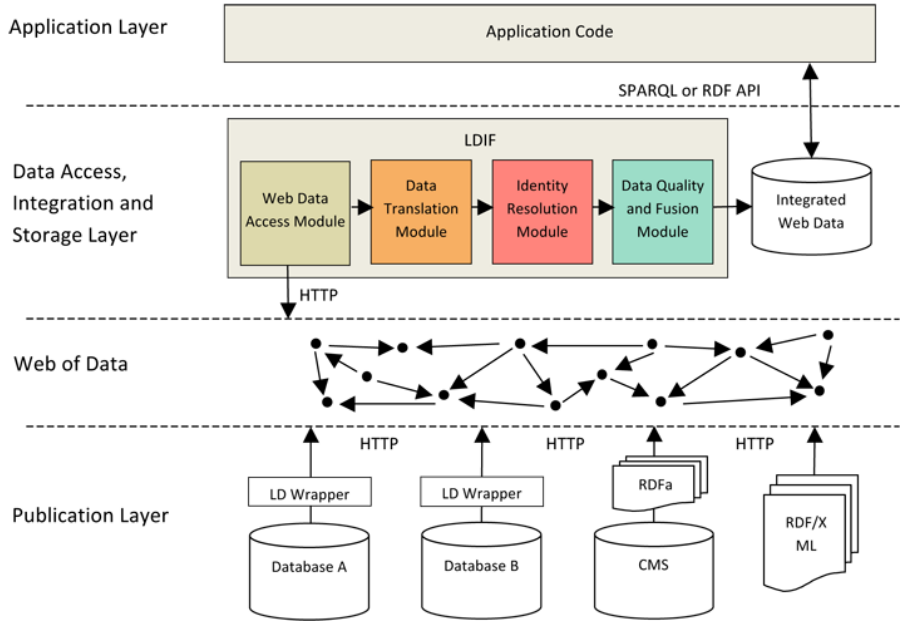
3 TRABALHOS RELACIONADOS

Nesta seção apresentamos alguns trabalhos existentes na literatura que apresentam abordagens para construção de *mashups* de dados. Após a apresentação dos trabalhos, nós concluímos o capítulo com uma breve discussão sobre as diferenças existentes os trabalhos apresentados e a solução proposta por esta dissertação.

3.1 LDIF - Linked Data Integration Framework

O LDIF (SCHULTZ *et al.*, 2011) se apresenta como um framework para integração de dados ligados. O LDIF é capaz de realizar todas as seguintes etapas do processo de integração: (i) coleta dos dados; (ii) tradução dos dados para um vocabulário comum; (iii) identificação dos links *owl:sameAs* existentes entre as fontes de dados integradas; (iv) avaliação de qualidade e fusão dos dados.

Figura 6 – Visão geral da arquitetura utilizada pelo LDIF



Fonte: http://ldif.wbsg.de/

A configuração de cada etapa é feita por meio de arquivos XML e executadas através de ferramentas especificas. O processo de descoberta de links *owl:sameAs* é feito por meio da ferramenta SILK (VOLZ *et al.*, 2009). O processo de avaliação de qualidade e fusão dos dados é feita pela ferramenta Sieve (MENDES *et al.*, 2012).

3.2 KARMA

KARMA é proposto em Knoblock *et al.* (2012) como um *framework* para integração de fontes de dados estruturadas (CSV, XML, JSON) e Web *API*s. KARMA provê uma interface gráfica para que o usuário defina os mapeamentos entre os esquemas das fontes de dados para uma ontologia OWL. Alguns desses mapeamentos são sugeridos através de algoritmos de aprendizagem de máquina e então refinados pelo usuário. Após definidos, os mapeamentos podem ser executados para geração do *mashup* dados.

3.3 MAURA

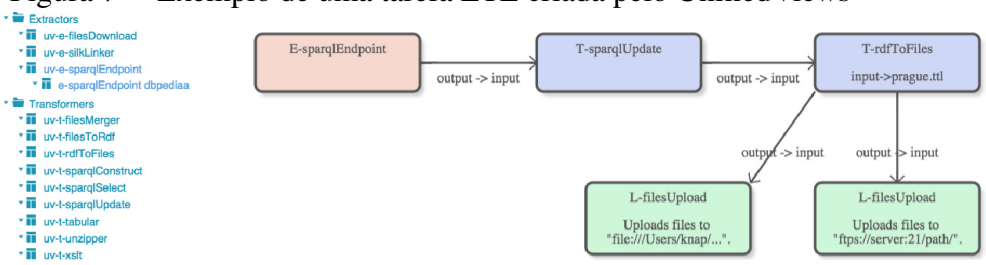
Em Cavalcante *et al.* (2017), MAURA é apresentado como um *framework* baseado em mediador semântico para facilitar a construção de *Linked Data Mashups*. MAURA mostra os passos inicias sobre como tirar proveito de uma camada semântica com a finalidade de facilitar a criação de um *mashup* de dados.

Para tanto, a partir de uma especificação de *mashup* definida pelo usuário, um recorte da visão semântica é gerado. Nesse recorte estão um conjunto de mapeamentos e regras de *linkage* que serão necessários para criação do *mashup*.

3.4 ODCleanStore e UnifiedViews

Em Knap *et al.* (2012) os autores apresentam ODCleanStore, um framework para realizar o gerenciamento de dados ligados na Web. O *workflow* seguido pelo ODCleanStore permite que os dados sejam limpos, ligados, transformados e passem por um processo de avaliação de qualidade. De forma similiar ao LDIF, ODCleanStore também utiliza o SILK para o processo de *linkage* na descoberta de links *owl:sameAs*.

Figura 7 – Exemplo de uma tarefa ETL criada pelo UnifiedViews



Fonte: (KNAP *et al.*, 2018)

Posteriormente, o ODCleanStore foi rebatizado como UnifiedViews (KNAP *et al.*, 2018). Essa nova versão passou a ter como foco prover um *framework* para criação e manutenção de processos ETL bastante customizáveis sobre dados RDF. Nesse *framework*, uma tarefa ETL é representada como uma *pipeline* composta por unidades de processamento de dados e ligações entre as unidades de processamento que definem o fluxo de dados dentro da *pipeline*. Segundo os autores, UnifiedViews apresenta melhorias relacionadas não apenas ao desempenho, como também uma interface gráfica mais agradável para o usuário.

3.5 Discussão

Todas as abordagens apresentadas, com exceção de MAURA, tem a proposta de realizar a construção de um *mashup* a partir da definição de um processo ETL. MAURA propõe algo similar a esta dissertação, buscando tomar proveitos da especificação de uma visão semântica sobre um conjunto de fontes para construção de mashups. A principal contribuição do nosso trabalho para o problema de construção de *mashups* é a proposta de uma abordagem para realizar a construção de mashups de dados utilizando o EKG e sua visão semântica.

4 ESPECIFICANDO UM EKG COMO UMA VISÃO SEMÂNTICA

Neste trabalho, iremos utilizar uma arquitetura de *EKG* baseada em um enfoque que combina ontologias e princípios *linked data* para enfrentar os desafios envolvidos no desenvolvimento de aplicações que precisam consumir dados integrados provenientes de fontes de dados heterogêneas. Essa abordagem permite que a construção do *EKG* seja conduzida de forma *pay-as-you-go* (Madhavan *et al.*, 2007), garantindo flexibilidade e extensibilidade para adição de novas fontes ao *EKG*.

4.1 Arquitetura do EKG

A arquitetura de *EKG* considerada por este trabalho é dividida em quatro camadas: Camada das Fontes de dados; Camada Semântica; Camada de Acesso e Integração de dados; e Camada de Aplicações. Cada camada representa um diferente nível de abstração, de forma que a camada das fontes de dados corresponde ao mais baixo nível de abstração e a camada de aplicações ao nível mais alto de abstração. Essas camadas são apresentada a seguir e uma visão geral dessa arquitetura é apresentada na Figura 8.

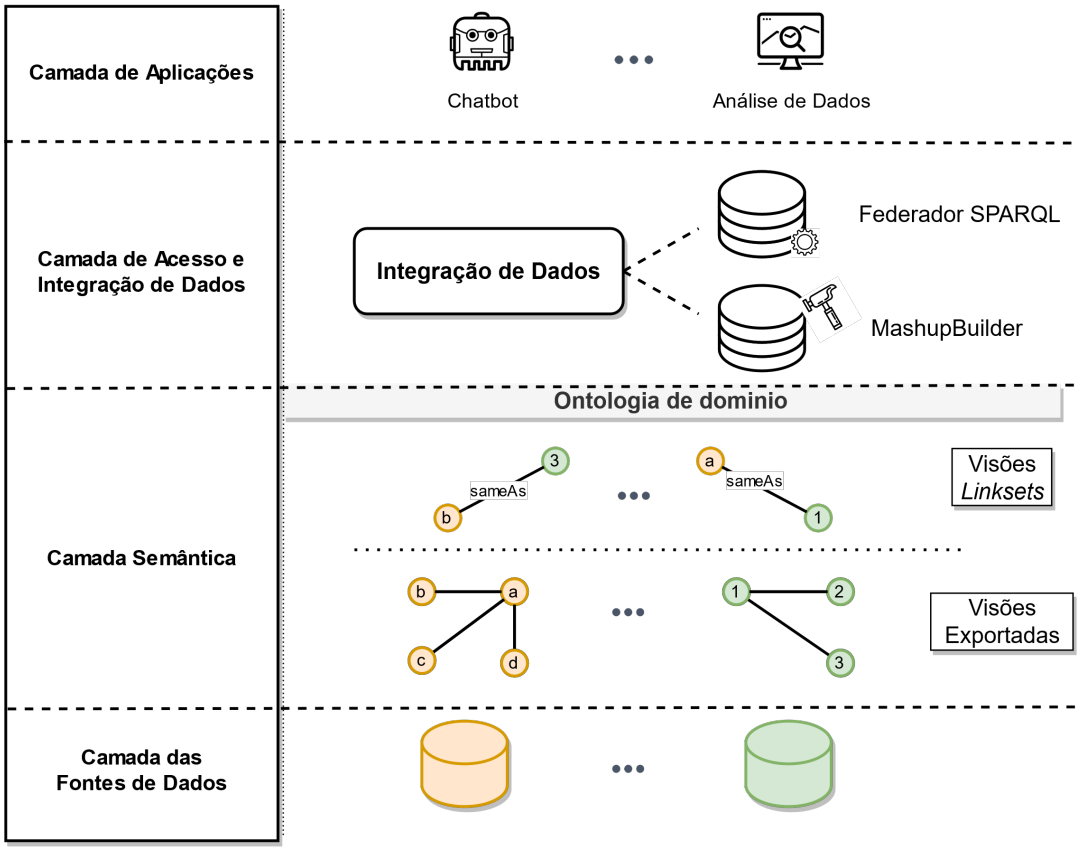
Camada das Fontes de Dados

A camada de dados é composta pelas fontes de dados selecionadas para serem integradas pelo *EKG*. De acordo com Bishr (1998), a heterogeneidade existente entre as fontes de dados pode ser sintática, esquemática ou semântica. A heterogeneidade sintática pode ser identificada quando as fontes utilizam diferentes modelos para representação dos dados, e.g., uma fonte adota o modelo relacional e outra o modelo orientado a documentos.

Já a heterogeneidade esquemática está relacionada a forma como as fontes de dados estruturam as suas informações, i.e, organizam seus esquemas. Esse tipo de heterogeneidade pode ser observado quando, e.g., duas bases relacionais usam esquemas organizados de formas diferentes para representar as mesmas informações.

Por fim, temos a heterogeneidade semântica que decorre das diferentes formas de expressar um determinado conceito ou ideia. É comum que fontes distintas utilizem nomenclaturas diferentes para se referir aos mesmo objetos no mundo real. Essas diferenças podem resultar na não identificação de informações que são correlacionadas ou em uma identificação de relacionamentos inexistentes entre termos que parecem similares.

Figura 8 – Arquitetura de um EKG implementado a partir de uma visão semântica



Fonte: O autor.

Camada Semântica

A camada semântica é responsável por resolver os problemas de interoperabilidade existentes entre as fontes de dados e explicitar as ligações existentes entre os recursos. Para tanto, nesta camada estão os seguintes componentes: ontologia de domínio, um conjunto de visões exportadas e um conjunto de visões *linkset*. A ontologia de domínio definida nesta camada é responsável por estabelecer um vocabulário comum entre as informações provenientes de diferentes fontes de dados, sendo assim uma peça fundamental para resolução dos problemas de interoperabilidade.

As visões exportadas são visões RDF definidas sobre as fontes de dados especificadas na camada anterior. Cada uma das fontes de dados pode ter suas informações publicadas por uma ou mais visões exportadas. Cada uma dessas visões pode ser interpretada como um grafo RDF gerado a partir das informações armazenadas na fonte de dados sobre a qual a visão exportada foi definida. O vocabulário utilizado por esse grafo RDF utiliza um subconjunto dos termos especificados pela ontologia de domínio. A publicação de cada uma das visões exportadas pode ser feita por meio de um enforque materializado ou virtual. Na literatura, o processo de

virtualização de uma fonte de dados utilizando uma ontologia é chamado de *Ontology-Based Data Access* (*OBDA*).

A decisão sobre qual enfoque deve ser utilizado para publicação de uma visão exportada pode levar em consideração alguns fatores que variam desde a necessidade de dados sempre atualizados até a indisponibilidade de ferramentas capazes de virtualizar alguns modelos de dados. Para fontes de dados que utilizam o modelo de dados relacional, existem alguns trabalhos que apresentam ferramentas e abordagens capazes de realizar essa publicação como, por exemplo, o Ontop (CALVANESE *et al.*, 2017). Independente da abordagem utilizada, a linguagem *SPARQL* pode ser utilizada para realizar consultas sobre as visões publicadas.

Ainda na camada semântica, temos as visões *linkset* que definem *links* existentes entre recursos de diferentes visões exportadas. Neste trabalho, vamos delimitar que esses visões estabeleçam apenas *links* `owl:sameAs`. Esses *links* têm a função de explicitar uma relação de equivalência entre dois recursos distintos, indicando que aqueles recursos se referem a um mesmo objeto no mundo real. Para realizar a descoberta dos *links* `owl:sameAs` entre recursos de visões *RDF*, existem algumas ferramentas como SILK (VOLZ *et al.*, 2009) e LIMES (NGOMO; AUER, 2011). Na abordagem adotada por este trabalho, cada visão *linkset* é materializada e armazenada em um banco de triplas e pode ser consultada por meio de uma consulta *SPARQL*.

Camada de Acesso e Integração de Dados

O objetivo desta camada é fornecer o acesso a uma visão integrada sobre visões publicadas pela camada semântica e, consequentemente, uma visão integrada sobre as fontes de dados. Nesta camada, essa visão integrada sobre as fontes de dados pode ser fornecida de duas formas: através de um federador de consultas *SPARQL* ou através da abordagem que chamamos de *MashupBuilder*. Apesar do processo de federação costumar ser suficiente para diversas aplicações, existem algumas aplicações que necessitam consumir uma visão integrada que possua um conjunto de características para garantir uma maior qualidade sobre os dados: a geração de uma representação unificada para os recursos que possuem múltiplas representações e a resolução dos conflitos de dados relacionadas as informações ligadas a esses recursos. No Capítulo 6, apresentamos a abordagem utilizada pelo *MashupBuilder* para criação de *mashups* definidos como uma visão sobre o *EKG*.

Camada de Aplicações

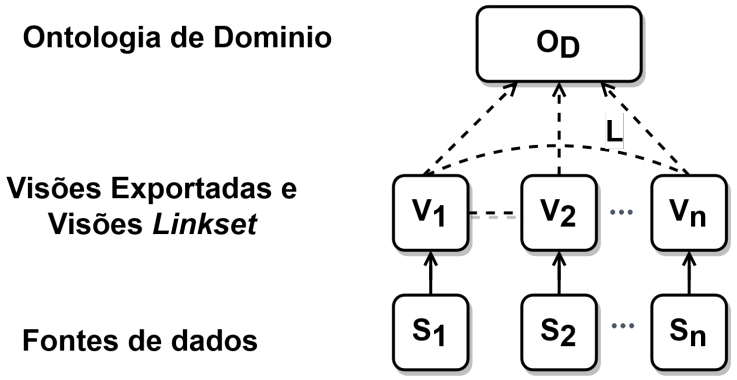
Por fim, temos a camada de aplicações. Nessa última camada, podemos ter aplicações de busca semântica, *chatbots*, sistemas de recomendação, ferramentas de análise de dados que realizam suas requisições de informação em termos da ontologia de domínio.

4.2 Especificando a Camada Semântica do EKG

Como já foi explicado anteriormente, a camada semântica do *EKG* é responsável por resolver as diferenças existentes entre fontes de dados distintas e, para tanto, são utilizadas um conjunto de visões *RDF*. Dessa forma, a especificação dessa camada é obtida como resultado da integração semântica das fontes de dados e a essa especificação damos o nome de visão semântica.

Consideramos como integração semântica o processo que utiliza uma representação conceitual dos dados e de seus relacionamentos para eliminar possíveis heterogeneidades existentes entre dados provenientes de diferentes fontes. Uma das formas de se realizar essa representação conceitual é através do uso de ontologias.

Figura 9 – *Framework* de três níveis para especificação de uma visão semântica



Fonte: O autor.

Dito isso, a partir do *framework* de três camadas proposto por Vidal *et al.* (2015) e ilustrado pela Figura 9, a visão semântica de um *EKG* pode ser formalizada. Nesse *framework*, a especificação da visão semântica resultante da integração semântica das fontes de dados $S_1, ...S_n$ é representada como uma tripla $\lambda = (O_D, V, L)$, onde:

- O_D representa a ontologia de domínio. O_D é responsável por estabelecer um vocabulário comum para descrever as fontes de dados;

- V representa um conjunto de especificações de visões exportadas $V_1, ..., V_n$, que descrevem as fontes $S_1, ..., S_n$ utilizando termos da ontologia O_D. A especificação de uma visão exportada V_i é uma tupla (M_{V_i}, O_{V_i}), onde:
  - M_{V_i} é um conjunto de mapeamentos que relacionam termos do vocabulário de O_D a termos do esquema da fonte S_i;
  - O_{V_i} é a ontologia da visão exportada. O vocabulário de O_{V_i} é o subconjunto do vocabulário de O_D utilizado por M_{V_i}.
- L representa um conjunto de especificações de visões de *linkset* $L_1, ..., L_m$, que descrevem relacionamentos `owl:sameAs` existentes entre recursos de classes semanticamente semelhantes. Uma visão *linkset* pode ser definida como uma tupla $(V_s, V_t, C_s, C_t, T, \mu)$ onde: V_s e V_t são visões exportadas em V; C_s e C_t são classes pertencentes aos vocabulários das visões V_s e V_t, respectivamente; T é um conjunto de propriedades que estão no vocabulário de V_s e V_t; μ é uma regra de *linkage* que utiliza as propriedade em T para definir se existe ou não uma ligação entre os recursos de C_s e C_t das visões V_s e V_t, respectivamente.

5 ESTUDO DE CASO: PORTAL SEMANTICSUS

O volume de dados públicos disponíveis que estão relacionadas ao domínio da saúde tem crescido consideravelmente nos últimos anos (VIACAVA *et al.*, 2018). No Brasil, vários desses dados estão armazenados em bases de dados ligadas ao Sistema Único de Saúde (SUS). Essas bases podem ser vistas como um grande espaço de dados heterogêneos que contém informações que podem ser tanto complementares quanto potencialmente conflitantes. A tarefa de análise e exploração sobre os dados do SUS é fundamental para o desenvolvimento de políticas públicas capazes de afetar diretamente o bem-estar da população.

A partir deste cenário, durante essa dissertação, trabalhamos na construção de uma visão semântica que tinha como objetivo inicial integrar semanticamente duas bases de dados: o Sistema de Informação Sobre Mortalidade (SIM) e o Sistema de Informações sobre Nascidos Vivos (SINASC). Essas duas fontes foram escolhidas com a finalidade de facilitar a condução de estudos que utilizam informações existentes nessas fontes de dados (e.g. estudos sobre mortalidade neonatal). Após construída, a visão semântica foi publicada no portal SemanticSUS (CRUZ *et al.*, 2019) e, posteriormente, foi utilizada para implementação da camada semântica de um *EKG* em Rolim *et al.* (2020).

Importante destacar que algumas das informações contidas nessas bases são informações sensíveis e o acesso a essas bases é restrito. Por isso gostaríamos ressaltar que o acesso a essas fontes de dados só foi possível por ter sido realizado no contexto do projeto GISSA (FREITAS *et al.*, 2017), que tinha como objetivo promover um sistema inteligente de governança para dar apoio à tomada de decisão em ambientes de saúde.

5.1 Construção de um *mashup* sobre as fontes de dados do SUS

Um *mashup* de dados é uma visão materializada construída através da transformação e integração de dados presentes em diferentes fontes de dados (VIDAL *et al.*, 2015). O processo de criação de *mashup* é um tarefa bastante complexa quando consideramos um cenário onde não existe uma visão semântica definida sobre as fontes de dados a serem consideradas. Isso é particularmente verdade quando consideramos contextos similares ao que pode ser encontrado nas bases do SUS. Nessas bases várias informações estão armazenadas como códigos cuja compreensão só é possível através da consulta de tabelas ou dicionários de dados externos as fontes de dados.

Além disso, cada base pode utilizar termos diferentes para se referir a uma mesma informação em seu esquema ou utilizar termos que parecem similares, mas que se referem a informações diferentes. Outro ponto importante é que alguns indivíduos podem ser encontrados em mais de uma base e é fundamental que as relações existentes entre as representações desses indivíduos possam ser devidamente identificadas. As informações que estão armazenadas nas fontes de dados sobre esses indivíduos podem ser conflitantes e esses conflitos também precisam ser resolvidos.

Em um cenário onde não existe uma visão semântica, todas essas questões levantadas precisam ser consideradas a cada novo *mashup* criado, resultando em um consumo de tempo e de recursos humanos bastante elevados (XIAO *et al.*, 2019). Além disso, quando a criação de um *mashup* é feita de forma *ad-hoc*, podem ocorrer problemas relacionados a qualidade de dados integrados, comprometendo assim a confiabilidade do *mashup* construído e, consequentemente, a confiabilidade das aplicações que consomem este *mashup*.

5.2 Construção da visão semântica publicada no SemanticSUS

5.2.1 Descrição das fontes de dados integradas pelo SemanticSUS

Sistema de Informações sobre Mortalidade (SIM)

Desenvolvido pelo Ministério da Saúde em 1975 e informatizado em 1979, o SIM [1] é resultado da unificação de mais de quarenta modelos de instrumentos utilizados para coletar dados sobre mortalidade no país. O SIM possui variáveis que podem ser aproveitadas para a construção de indicadores e para desenvolvimento de análises epidemiológicas .

Doze anos depois de sua informatização, com a implantação do SUS e sob a premissa da descentralização, a coleta de dados passou a ser uma atribuição dos Estados e Municípios, através das suas respectivas Secretarias de Saúde. Com a finalidade de reunir dados quantitativos e qualitativos sobre óbitos ocorridos no Brasil, o SIM é considerado uma importante ferramenta para gestão na área da saúde, sendo utilizada para embasar a tomada de decisão em diversas áreas da assistência à saúde.

[1] http://svs.aids.gov.br/dantps/cgiae/sim/

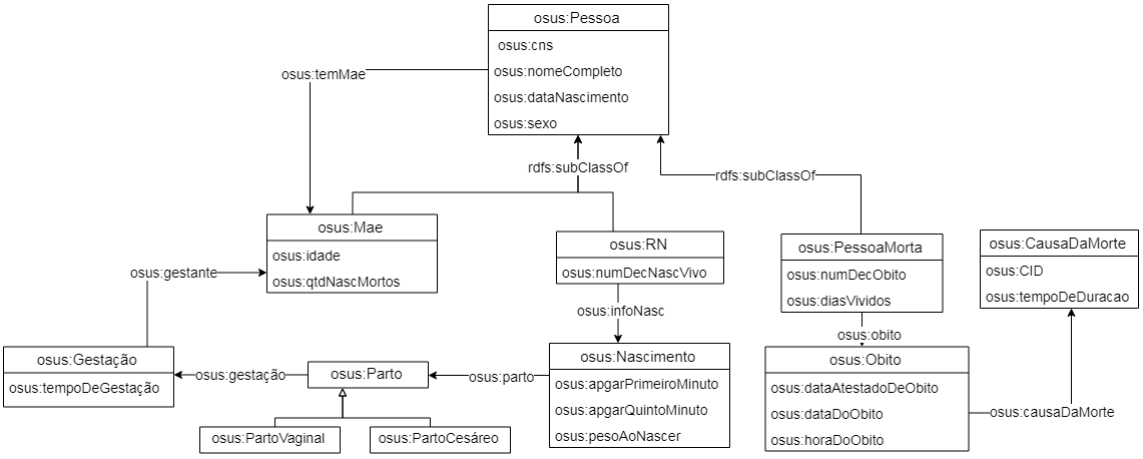
Sistema de Informações sobre Nascidos Vivos (SINASC)

Implantado oficialmente em 1990, o SINASC[2] é responsável pela coleta dos dados sobre nascimentos informados em todo Brasil e pelo fornecimento de dados de natalidade para todos os níveis do Sistema de Saúde. De maneira gradual, a implantação do SINASC ocorreu em todas as unidades da Federação. Em muitos municípios, desde o ano 1994, o SINASC tem apresentado um número maior de registros do que o publicado pelo IBGE, com base nos dados de Cartório de Registro Civil. A partir do SINASC, também é possível realizar a construção de indicadores úteis para planejar a gestão dos serviços de saúde.

5.2.2 Modelagem da ontologia de domínio

Para realizar a modelagem da ontologia de domínio, as fontes SIM e SINASC foram analisadas com a finalidade de entender quais informações poderiam ser extraídas de cada uma das fontes de dados. A compreensão obtida sobre as fontes a partir desta análise, deu suporte para o desenvolvimento da ontologia de domínio utilizada pela visão semântica. A Figura 10 ilustra uma representação visual da ontologia de domínio implementada.

Figura 10 – Representação visual da ontologia de domínio do SemanticSUS



Fonte: O autor.

5.2.3 Especificação das visões exportadas

Durante o processo de análise das fontes de dados, por vezes, foram encontrados campos preenchidos com códigos ou que utilizavam valores numéricos que também necessitavam de conhecimentos externos as fontes de dados (e.g., dicionários de dados disponibilizados pelo

[2] http://svs.aids.gov.br/dantps/cgiae/sinasc/

Ministério da Saúde). Essa compreensão obtida a partir desse processo foi fundamental para a definição dos mapeamentos que relacionam os termos utilizados pelos esquemas das fontes de dados aos termos da ontologia de domínio. A partir da aplicação desses mapeamentos, os valores que foram identificados como códigos são transformados em recursos e as descrições desses códigos são expressas através propriedades relacionadas aos recursos criados.

5.2.4 Especificação das visões linkset

Após a definição dos mapeamentos utilizados pelas visões exportadas, foram definidas as regras de *linkage* que fazem parte da especificação das visões *linkset*. Essa regras explicitam relacionamentos existentes entre os indivíduos do SIM e SINASC. Uma das regras definidas foi estabelecida sobre os recursos da classe `osus:RN` e `osus:PessoaMorta`, provenientes do SINASC e SIM, respectivamente. Essa regra especifica que um *link* `owl:sameAs` deve ser estabelecido entre os recursos que possuam o mesmo valor associado a propriedade declaração de nascido vivo (`osus:numDecNascVivo`). Os *links* gerados por essa regra são especialmente importantes para estudos sobre mortalidade infantil e neonatal.

5.3 Construção de um mashup a partir de visão semântica publicada no SemanticSUS

A visão semântica especificada sobre as fontes de dados SIM e SINASC pode ser utilizada para facilitar a construção de *mashups* de dados que necessitem combinar informações provenientes dessas fontes. Isso é possível porque as especificações definidas para construção da visão semântica podem ser aproveitadas no processo de especificação desses *mashups*.

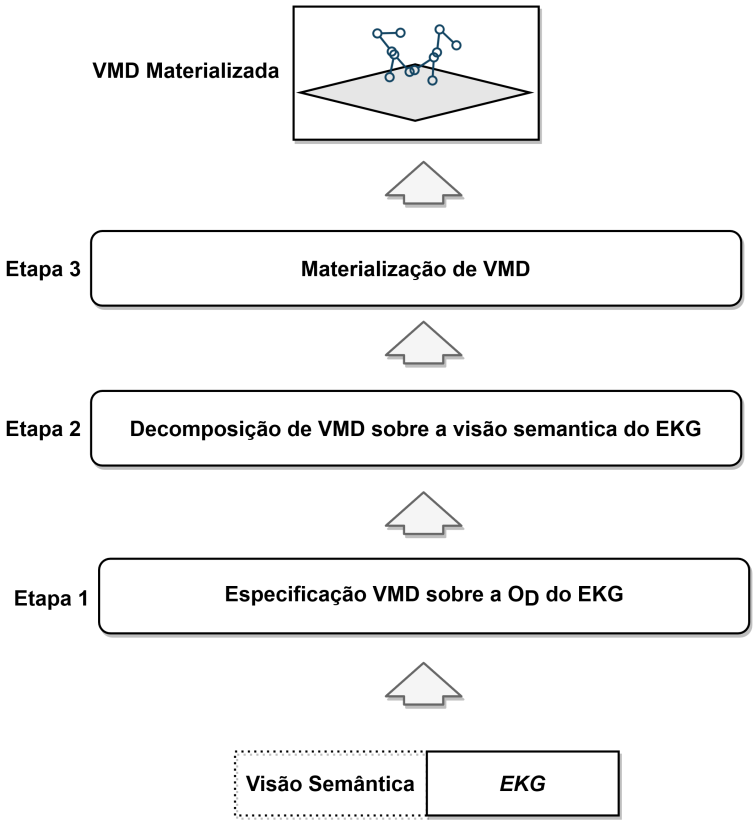
Os mapeamentos utilizados para construção do *mashup* podem ser obtidos a partir de uma seleção sobre os mapeamentos já estabelecidos nas especificações das visões exportadas que compõe o SemanticSUS. De maneira similar, as regras de *linkage* que utilizadas na identificação dos links `owl:sameAs` entre os recursos que irão compor o *mashup* também podem ser uma seleção das regras estabelecidas nas especificações de visões de *linkset* da visão semântica. Além dessas seleções, a construção do *mashup* também necessita que as regras de fusão responsáveis pela consolidação dos dados sejam especificadas. Apesar facilitar o processo de construção de *mashups*, essa abordagem possui limitações. A principal delas está relacionada a capacidade de aplicar filtros para delimitar quais recursos devem fazer parte do *mashup* de dados gerado, podendo resultar em uma quantidade massiva de dados materializados de forma desnecessária.

6 PROCESSO PARA CONSTRUÇÃO DE UM *MASHUP* SOBRE UM *EKG*

Neste capítulo, apresentamos a principal contribuição deste trabalho: uma abordagem semiautomática para construção de *mashups* quando um *EKG* está disponível e foi construído utilizando a metodologia discutida no Capítulo 4. A abordagem proposta é estruturada como um processo dividido em 3 Etapas e está ilustrado na Figura 11. Cada uma das etapas do processo é discutida separadamente em uma seção deste capítulo. Cada uma das etapas do processo é discutida separadamente em uma seção deste capítulo (Seções 6.2, 6.3 e 6.4).

Antes de iniciar a discussão sobre cada etapa, iremos utilizar a Seção 6.1 para introduzir um segundo estudo de caso, que será utilizado no decorrer deste capítulo para o desenvolvimento de um *running example*. Nesse novo estudo de caso, nos temos que a visão semântica foi especificada e é utilizada para implementar a camada semântica de um *EKG*. Esse estudo de caso, apesar de não se tratar de uma contribuição deste trabalho, foi incluído por se tratar de um cenário diferente do que foi apresentado no *SemanticSUS*.

Figura 11 – Visão geral do processo seguido para construção de VMD

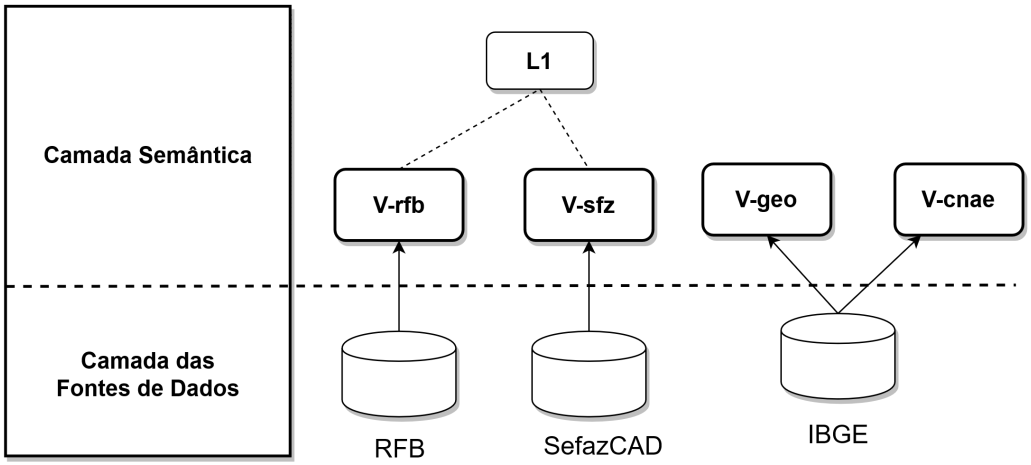


Fonte: O autor.

6.1 Estudo de Caso: EKG-SefazMA

O EKG-SefazMA foi criado para facilitar o acesso, análise e visualização de dados utilizados pela Secretaria da Fazenda do Estado do Maranhão. Seu principal objetivo é oferecer uma camada ontológica, conectada semanticamente aos dados, capaz de permitir um acesso integrado as múltiplas fontes de dados utilizadas. O EKG-SefazMA foi construído através da aplicação da abordagem apresentada no Capítulo 4. A Figura 12 mostra o recorte das informações contidas no EKG-SefazMA. É sobre esse recorte que iremos basear nosso *running example*.

Figura 12 – Recorte do EKG-SefazMA



Fonte: O autor.

6.1.1 Descrição das fontes de dados integradas pelo EKG-Sefaz

Receita Federal do Brasil

A Receita Federal do Brasil (RFB) é o órgão responsável por administrar os tributos dos tributos federais e realizar o controle alfandegário, atuando também para combater contravenções como evasão fiscal, contrabando e trafico de drogas. A RFB pode ser vista como a fonte de dados primária e com maior capacidade de fornecer uma informação confiável a respeito de CPFs e CNPJs. Se uma pessoa não possuir um CPF válido e ativo na RFB, para fins de concessão de inscrição, a participação como represente desta pessoa em empresas não será considerada válida junto aos estados. De forma similar, empresas que não possuem CNPJ podem até ser consideradas empresas de fato, mas não de direito. Se uma empresa está nessa condição, o exercício de suas atividades é considerado ilegal.

Cadastro Sefaz (SefazCAD)

O Cadastro de Dados de Contribuintes do Estado do Maranhão é uma base de dados que contém informações de Cadastro de Pessoas Jurídicas e Físicas, bem como dados referentes aos estabelecimentos, representantes legais, sócios, baixas e razões de desabilitação. No âmbito do Estado do Maranhão, esse cadastro é mantido pela Secretaria da Fazenda do Estado do Maranhão (SefazMA), com informações sobre todos os contribuintes de impostos. Estes são obrigados a inscrever seus estabelecimentos antes de iniciarem suas atividades e a comunicar quaisquer alterações dos dados declarados para sua inscrição.

IBGE - CNAE e Informações Geográficas

Desde 1934, o Instituto Brasileiro de Geografia e Estatística (IBGE) integra a administração federal brasileira como um instituto público cujas atribuições estão relacionadas às geociências e estatísticas sociais, demográficas e econômicas. Essas atribuições incluem a realização de censos e a organização das informações resultantes dos censos realizados para órgãos das esferas do governo, bem como para outras instituições e público geral.

Uma das informações geridas pelo IBGE é a Classificação Nacional de Atividades Econômicas (CNAE). A administração pública e o Sistema Estatístico Nacional adotam oficialmente essa classificação com o objetivo de identificar as atividades econômicas em cadastros e registros de pessoa jurídica. A CNAE[1] é divida em cinco níveis hierárquicos: seção, divisão, grupo, classe e subclasse. O quinto nível, o de subclasse, é definido para uso da administração pública.

Outra informação importante gerenciada pelo IBGE é a Tabela de Códigos de Municípios [2]. Essa tabela relaciona os municípios brasileiros a um código composto por 7 dígitos onde os dois primeiros dígitos de cada código se referem ao código da unidade da federação. Por exemplo, a cidade de Fortaleza, capital do Ceará, possui o código 2304400.

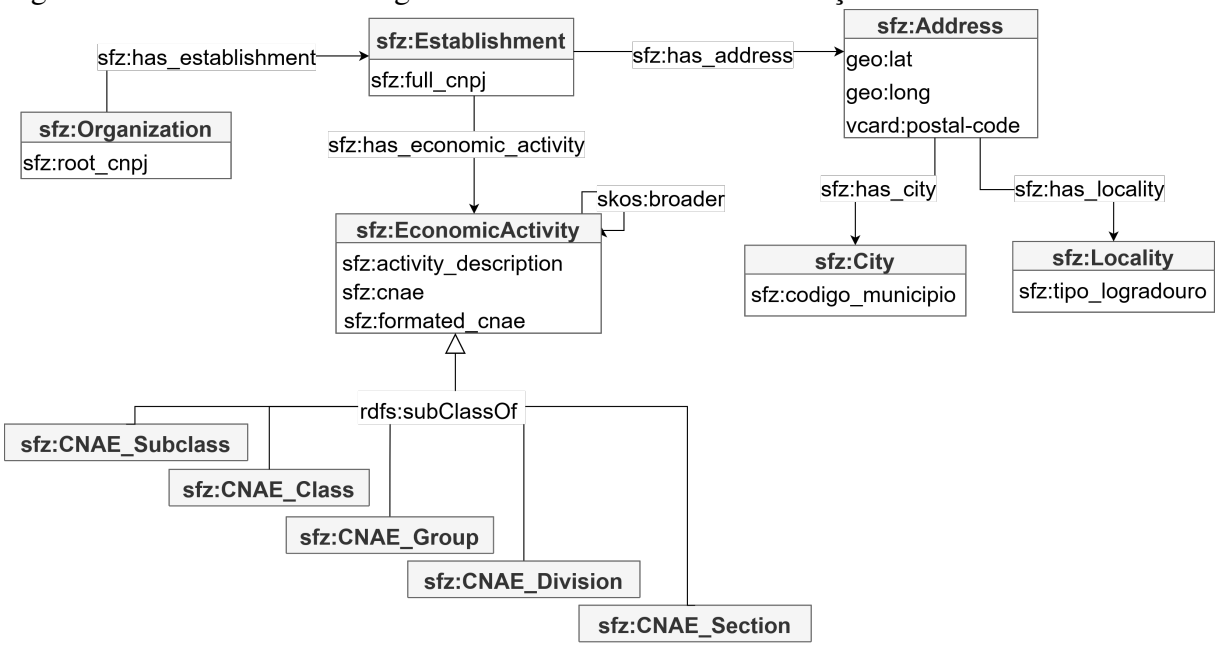
6.1.2 Descrição das visões exportadas e visões linkset

O recorte do EKG-SefazMA que vamos utilizar durante o texto contém as visões exportadas V-rfb, V-sfz, V-cnae e V-geo. As visões exportadas V-rfb e V-sfz publicam as

[1] https://concla.ibge.gov.br/busca-online-cnae.html
[2] https://www.ibge.gov.br/explica/codigos-dos-municipios.php

informações das fontes de dados da Receita Federal e do Cadastro SefazMA, respectivamente. A visão exportada V-cnae publica as informações estruturadas pela tabela CNAE. Por fim, a visão exportada V-geo publica informações precisas sobre endereços utilizando dados do IBGE (como a Tabela de Códigos de Municípios).

Figura 13 – Recorte da ontologia de domínio adotada na construção do EKG-SefazMA



Fonte: O autor.

A informações publicadas por cada uma dessas visões utiliza o vocabulário da ontologia ilustrada na Figura 13. Nós mostramos abaixo o vocabulário da ontologia exportada de cada visão:

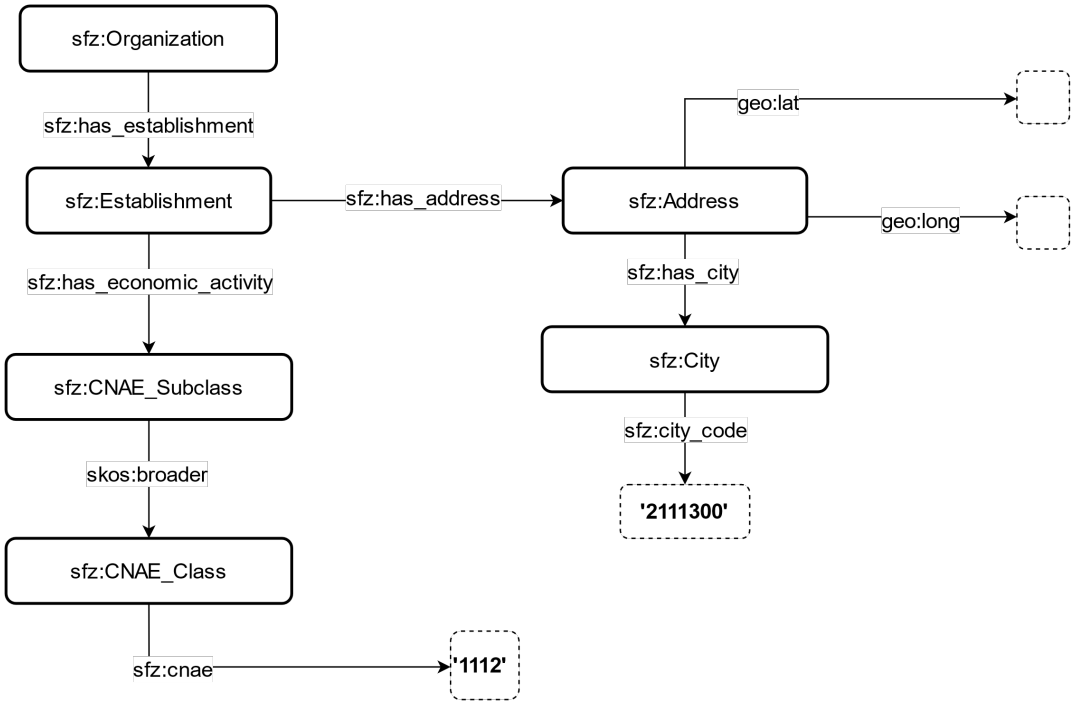
- $O_{rfb} = \{$ sfz:Organization, sfz:Establishment, sfz:CNAE_Subclass, sfz:has_establishment, sfz:has_economic_activity, sfz:full_cnpj, sfz:root_cnpj $\}$;
- $O_{sfz} = \{$ sfz:Organization, sfz:Establishment, sfz:Address, sfz:has_address, sfz:has_economic_activity, sfz:full_cnpj, sfz:root_cnpj $\}$;
- $O_{cnae} = \{$ sfz:CNAE_Subclass, sfz:CNAE_Class, sfz:CNAE_Group, sfz:CNAE_Division, sfz:CNAE_Section, skos:broader, sfz:activity_description, sfz:cnae, sfz:formated_cnae $\}$;
- $O_{geo} = \{$ sfz:Address, sfz:City, sfz:Locality, sfz:has_city, sfz:has_locality, geo:lat, geo:long, vcard:postal-code, sfz:city_code, sfz:tipo_logradouro $\}$.

Além das visões exportadas, o recorte do EKG-SefazMA também possui a visão *linkset* L1. A visão L1 liga os recursos `sfz:Establishment` existentes nas visões V-rfb e V-sfz. Os *links* `owl:sameAs` em L1 são criados a partir da comparação da propriedade `sfz:full_cnpj`

6.1.3 SefazVMD: uma visão de mashup de dados sobre EKG-SefazMA

Para iniciar nosso *running example*, vamos supor que exista a necessidade da construção de um *mashup* de dados para ser utilizado em um estudo sobre as organizações que possuem estabelecimentos cuja atividade econômica é caracterizada como 'Cultivo de Grão' (código CNAE 1112). Esse estudo tem o objetivo de avaliar a distribuição geográfica desses estabelecimentos dentro da cidade de São Luís (código IBGE 2111300). O *mashup* de dados necessário para esse estudo pode ser especificado através uma Visão de Mashup de Dados (VMD) sobre o EKG-SefazMA. Uma representação visual dessa VMD é apresentada na Figura 14.

Figura 14 – Representação visual da estrutura da *SefazVMD*



Fonte: O autor.

6.2 Etapa 1: Especificação da visão de *mashup* como uma consulta facetada

A utilização de interfaces de consulta facetada e de consultas visuais sobre *EKG*s são extremamente importantes para permitir que o usuário final possa expressar as suas necessidades de informação de forma mais autônoma. Considerando isso, nessa dissertação a especificação de uma VMD é feita por meio de uma consulta facetada α. A definição de consulta facetada adotada é apresentada pela Definição 6.2.1. A semântica de uma consulta facetada é estabelecida

pela Definição 6.2.2 utilizando lógica de primeira ordem. Essas definições foram baseadas no trabalho de Pankowski (2017).

Definição 6.2.1 *Uma consulta facetada sobre uma ontologia O é uma expressão* α *que está de acordo com a sintaxe:*

- $\alpha ::= t \mid t[\beta]$
- $\beta ::= b \mid \beta/\alpha \mid \beta \wedge \beta$
- $t ::= \wedge\{C_1, ..., C_n\}$

onde: (1) $C_1, ..., C_n$ *é um conjunto de predicados unários e todo* C_i *é uma classe na ontologia O; (2) b é um par* $(P, \textbf{any})$ *ou* $(P, \{a_1, ..., a_n\})$*, P é uma propriedade na ontologia O (com exceção de rdf:type),* ***any*** *denota qualquer constante e* $\{a_1, ..., a_n\}$ *é um conjunto de constantes. Essas constantes podem ser literais ou URIs .*

Definição 6.2.2 *A semântica de uma consulta facetada* α *pode ser definida como uma fórmula da lógica de primeira ordem, construída a partir de predicados e constantes que ocorrem em* α*, quantificadores existenciais, conetivos e variáveis. A semântica da consulta* α *é definida através de uma função semântica* $[\![\ ]\!]_x$*, onde x é uma variável e a função é definida como se segue:*

1. $[\![\wedge\{C_1, ..., C_n\}]\!]_x = C_1(x) \wedge ... \wedge C_n(x)$
2. $[\![\{a_1, ..., a_n\}]\!]_x = (x = a_1) \vee ... \vee (x = a_n)$
3. $[\![(R, any)]\!]_{x,y} = R(x, y)$
4. $[\![(R, \{a_1, ..., a_n\})]\!]_{x,y} = R(x, y) \wedge [\![\{a_1, ..., a_n\}]\!]_y$
5. $[\![t[b]]\!]_x = [\![t]\!]_x \wedge \exists y([\![b]\!]_{x,y})$
6. $[\![t[b/\alpha]]\!]_x = [\![t]\!]_x \wedge \exists y([\![b]\!]_{x,y} \wedge [\![\alpha]\!]_y)$
7. $[\![t[\beta_1 \wedge \beta_2]]\!]_x = [\![t[\beta_1]]\!]_x \wedge [\![t[\beta_2]]\!]_x$

As variáveis sobre quantificadores existenciais devem ser 'frescas', isto é, não devem ter sido utilizadas previamente.

As formalizações apresentadas são extremamente importantes para este trabalho, pois viabilizam que as demais etapas do processo sejam especificadas de maneira igualmente formal e independentes de implementação. Entretanto, acreditamos que a definição de consulta facetada adotada é abrangente o suficiente para suportar consultas equivalentes as definidas por ferramentas de consulta visual já existentes como a OptiqueVQS (SOYLU *et al.*, 2018), que utiliza a ontologia de uma fonte de dados *RDF* para fornecer uma interface de consulta visual para o usuário.

Exemplo 1 (*Running Example* - Etapa 1) *Utilizando a Definição 6.2.1, a visão de mashup SefazVMD pode ser pela seguinte consulta facetada Q*

$$Q = t_1[b_1/t_2[b_2/t_3[b_3/t_4[b_4]] \wedge b_5/t_5[b_6 \wedge b_7 \wedge b_8/t_6[b_9]]]]$$

onde:

t_1 ={sfz:Organization}; b_1 = (sfz:has_establishment, *any*);

t_2 = {sfz:Establishment}; b_2 = (sfz:has_economic_activity, *any*);

t_3 = {sfz:CNAE_Subclass}; b_3 = (skos:broader, *any*);

t_4 = {sfz:CNAE_Class}; b_4 = (sfz:cnae, {'1112'});

b_5 = (sfz:has_address, *any*); t_5 = {sfz:Address};

b_6 = (geo:lat, *any*); b_7 = (geo:long, *any*);

b_8 = (sfz:has_city, *any*); t_6 = {sfz:City};

b_9 = (sfz:city_code, {'2111300'});

Aplicando a Definição 6.2.2 sobre Q, nós obtemos a fórmula da lógica de primeira ordem $[\![Q]\!]_x$*, que estabelece a semântica de Q:*

$$
\begin{aligned}
[\![Q]\!]_x = & \text{sfz:Organization}(x) \\
& \wedge \exists_{y1}(\text{sfz:has\_establishment}(x, y1) \wedge \text{sfz:Establishment}(y1) \\
& \quad \wedge \exists_{y2}(\text{sfz:has\_economic\_activity}(y1, y2) \wedge \text{sfz:CNAE\_Subclass}(y2) \\
& \quad\quad \wedge \exists_{y3}(\text{skos:broader}(y2, y3) \wedge \text{sfz:CNAE\_Class}(y3) \\
& \quad\quad\quad \wedge \exists_{y4}(\text{sfz:cnae}(y3, y4) \wedge (y4 = \text{'1112'})))) \\
& \quad \wedge \exists_{y5}(\text{sfz:has\_address}(y3, y5) \wedge \text{sfz:Address}(y5) \\
& \quad\quad \wedge \exists_{y6}(\text{geo:lat}(y5, y6)) \wedge \exists_{y7}(\text{geo:long}(y5, y7)) \\
& \quad\quad \wedge \exists_{y8}(\text{sfz:has\_city}(y5, y8) \wedge \text{sfz:City}(y8) \\
& \quad\quad\quad \wedge \exists_{y9}(\text{sfz:city\_code}(y8, y9) \wedge (y9 = \text{'2111300'})))))
\end{aligned}
$$

6.3 Etapa 2: Decomposição da visão de *mashup* sobre a visão semântica do *EKG*

Como apresentado na Seção 4.2, uma visão semântica λ é definida como uma tupla (O_D, V, L) onde O_D é uma ontologia de domínio, V é um conjunto de especificações de visões exportadas e L é um conjunto de especificações de visões *linkset*. A decomposição de uma VMD sobre λ tem como objetivo explicitar a relação de relevância das visões especificadas por V e L para os elementos que compõem a *VMD*. O processo de decomposição foi especificado a partir das noções de relevância estabelecidas a seguir.

De maneira intuitiva, podemos dizer que uma visão exportada V_i é relevante para *VMD* caso exista alguma informação que possa ser encontrada em V_i e que precisa ser considerada para criação do *mashup* especificado pela *VMD*. Como explicado na seção anterior, nós delimitamos que a especificação de uma visão de *mashup* é feita através de uma consulta facetada α. Dessa forma, uma visão V_i é relevante para uma *VMD* se e somente se V_i é relevante para α que define aquela *VMD*.

A noção de relevância de V_i para α é baseada no resultado da aplicação da função $\mathrm{E}[\,]_x$ sobre α (Definição 6.2.1). A aplicação dessa função transforma uma consulta facetada α em um conjunto de triplas. A partir dessa representação em triplas, nós podemos dizer que V_i é relevante para α se e somente se V_i é relevante para alguma tripla (s, p, o) de $\mathrm{E}[\alpha]_x$. A Definição 6.3.2 define as condições para considerar uma visão exportada V_i relevante para uma tripla (s, p, o).

Definição 6.3.1 *A transformação de uma consulta facetada α em uma lista de triplas (s, p, o), pode ser definida por uma função $E[\,]_x$ onde x é uma variável e a função é definida como se segue:*

1. $E[\wedge\{C_1, ..., C_n\}]_x =$ *(x,* ***rdf:type****,* C_1*), ..., (x,* ***rdf:type****,* C_n*);*
2. $E[(R, any)]_{x,y} =$ *(x, R, y)*
3. $E[R, \{a_1, ..., a_n\}]_x =$ *(x, R,* $\{a_1, ..., a_n\}$*);*
4. $E[t[b]]_x = E[t]_x;\ E[b]_{x,y}$ $\quad [y \leftarrow newVar()]$
5. $E[t[b/\alpha]]_x = E[t]_x;\ E[b]_{x,y};\ E[\alpha]_y$ $\quad [y \leftarrow newVar()]$
6. $E[t[\beta_1 \wedge \beta_2]]_x = E[t[\beta_1]]_x;\ E[t[\beta_2]]_x$

De maneira similar, consideramos que uma visão *linkset* L_i é relevante para uma *VMD* se somente se L_i é relevante para consulta facetada α que define aquela *VMD*. Se L_i é

relevante para α, então L_i é relevante para pelo menos uma variável s existente em $E[\alpha]_x$. Essa noção de relevância é apresentada pela Definição 6.3.3.

Definição 6.3.2 *Seja (M_{V_i}, O_{V_i}) a especificação de uma visão exportada V_i e (s,p,o) uma tripla de um conjunto de triplas resultante da aplicação da função $E[\,]_x$ sobre uma consulta facetada α, definimos que V_i é relevante para (**s**,**p**,**o**) se:*

- ***p*** $=$ ***rdf:type*** *e* ***o*** *é uma classe C, tal que* $C \in O_{V_i}$
- ***p*** $\neq$ ***rdf:type*** *e* ***p*** *é uma propriedade, tal que* $\mathbf{p} \in O_{V_i}$

Definição 6.3.3 *Seja $(V_s, V_t, C_s, C_t, T, \mu)$ a especificação de uma visão linkset L_i, α uma consulta facetada e z uma variável em $E[\alpha]_x$. Consideramos que que L_i é relevante para variável z se somente se todas as seguintes afirmações forem verdadeiras:*

- *Existe uma tripla (z, rdf:type, C_s), tal que (z, rdf:type, C_s)* $\in E[\alpha]_x$
- *Existe uma tripla (z, rdf:type, C_t), tal que (z, rdf:type, C_t)* $\in E[\alpha]_x$

A partir das noções de relevância estabelecidas, o resultado desta etapa pode ser formalizado como se segue. Seja λ uma visão semântica e α a consulta facetada que especifica uma *VMD*, a decomposição dessa *VMD* sobre λ tem como resultado uma tupla (D, δ) onde:

- D é um conjunto de tuplas (s,p,o,v), tal que: $(s,p,o) \in E[\alpha]_x$ e v é uma visão exportada relevante para (s,p,o);
- δ é um conjunto de tuplas (s,l,v_s,v_t) tal que: s é uma variável em $E[\alpha]_x$, *linkset* l é uma visão linkset relevante para s, e v_s e v_t são as visões exportadas ligadas pelos *links* definidos por l.

Exemplo 2 (*Running Example* - Etapa 2) *Aplicando a função definida pela Definição 6.3.1 sobre a consulta Q que define SefazVMD, nós obtemos o resultado que se segue:*

$$
\begin{aligned}
E[Q]_x = [&(x, \textbf{rdf:type}, \text{sfz:Organization}), (x, \text{sfz:has\_establishment}, y1),\\
&(y1, \textbf{rdf:type}, \text{sfz:Establishment}), (y1, \text{sfz:has\_economic\_activity}, y2),\\
&(y2, \textbf{rdf:type}, \text{sfz:CNAE\_Subclass}), (y2, \text{skos:broader}, y3),\\
&(y3, \textbf{rdf:type}, \text{sfz:CNAE\_Class}), (y3, \text{sfz:cnae}, \{\text{`1112'}\}),\\
&(y1, \text{sfz:has\_address}, y4), (y4, \textbf{rdf:type}, \text{sfz:Address}),\\
&(y4, \text{geo:lat}, y5), (y4, \text{geo:long}, y6), (y4, \text{geo:has\_city}, y7),\\
&(y7, \textbf{rdf:type}, \text{sfz:City}), (y7, \text{geo:city\_code}, \{\text{`2111300'}\})]
\end{aligned}
$$

A partir do conjunto de triplas produzidos por $E[Q]_x$, a decomposição de SefazVMD sobre a visão semântica de EKG-Sefaz pode ser feita, obtendo como resultado os conjuntos D_Q e δ_Q mostrados abaixo:

$$
\begin{aligned}
D_Q = [&(x, \textbf{rdf:type}, \text{sfz:Organization}, \text{V-sfz}), (x, \textbf{rdf:type}, \text{sfz:Organization}, \text{V-rfb}),\\
&(x, \text{sfz:has\_establishment}, y1, \text{V-sfz}), (x, \text{sfz:has\_establishment}, y1, \text{V-rfb})\\
&(y1, \textbf{rdf:type}, \text{sfz:Establishment}, \text{V-rfb}), (y1, \textbf{rdf:type}, \text{sfz:Establishment}, \text{V-sfz}),\\
&(y1, \text{sfz:has\_economic\_activity}, y2, \text{V-rfb}), (y1, \text{sfz:has\_address}, y4, \text{V-sfz}),\\
&(y2, \textbf{rdf:type}, \text{sfz:CNAE\_Subclass}, \text{V-sfz}), (y2, \textbf{rdf:type}, \text{sfz:CNAE\_Subclass}, \text{V-cnae}),\\
&(y2, \text{skos:broader}, y3, \text{V-cnae}), (y3, \textbf{rdf:type}, \text{sfz:CNAE\_Class}, \text{V-cnae}),\\
&(y3, \text{sfz:cnae}, \{\text{`1112'}\}, \text{V-cnae}), (y4, \textbf{rdf:type}, \text{sfz:Address}, \text{V-sfz}),\\
&(y4, \textbf{rdf:type}, \text{sfz:Address}, \text{V-geo},)(y4, \text{geo:lat}, y5, \text{V-geo}), (y4, \text{geo:long}, y6, \text{V-geo}),\\
&(y4, \text{geo:has\_city, y7, V-geo}), (y7, \textbf{rdf:type}, \text{sfz:City}, \text{V-geo}),\\
&(y7, \text{geo:city\_code}, \{\text{`2111300'}\}, \text{V-geo})]
\end{aligned}
$$

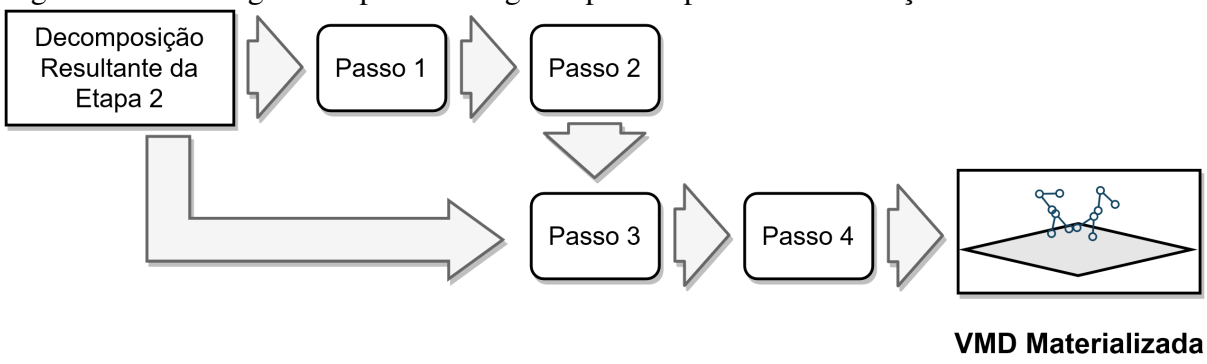
$$\delta_Q = \{(y1, L_1, \text{V-sfz}, \text{v-rfb})\}$$

6.4 Etapa 3: Materialização da visão de *mashup*

Esta última etapa finaliza o processo de construção do *mashup* com a materialização da *VMD*. Esse processo de materialização será baseado no conceito de sujeitos relevantes que explicaremos a seguir. Consideramos como sujeitos relevantes os recursos que devem fazer parte do *mashup* de dados e que tem pelo menos uma propriedade que o descreve em *VMD*, i.e., são sujeitos de alguma propriedade. De maneira similar, iremos utilizar o termo variáveis relevantes para nós referir as variáveis que ocupam a posição de sujeito em alguma das triplas de $E[\alpha]_x$.

Como ilustrado na Figura 15, o processo de materialização foi subdividido em 4 passos. No primeiro passo, é construído um plano de consulta para recuperar pelo menos uma das representações dos sujeitos relevantes para *VMD*. Em outras palavras, mesmo que múltiplas representações dos objetos considerados como sujeitos relevantes existam no EKG, se essas representações estiverem ligadas por *links* `owl:sameAs` nas visões de *linkset* relevantes para *VMD*, o plano construído irá conseguir recuperar pelo menos uma das representações desses sujeitos. As variáveis projetadas por esse plano são as variáveis relevantes.

Figura 15 – Visão geral do processo seguido pela etapa de materialização



Fonte: O autor.

O segundo passo é responsável pela execução desse plano de consulta, tendo como resultado um conjunto de soluções. Nessas soluções, cada variável está ligada a uma representação de um sujeito relevante para VMD. No terceiro passo, é feita a extração das propriedades dos sujeitos relevantes que devem estar na VMD. Essa extração é feita com base nas representações recuperadas pelo passo dois, levando em conta as demais representações definidas pelos links `owl:sameAs` existentes nas visões *linkset*. Por fim, no quarto passo, são definidas e aplicadas as regras de fusão.

Passo 1: Construção do plano de consulta para recuperação dos sujeitos relevantes para VMD

O principal elemento deste passo é o Algoritmo 2, BUILDQUERYPLAN. Esse algoritmo recebe como entrada decomposição D e δ, resultantes da Etapa 2, e tem como resultado um plano de consulta. Para tanto, ele utiliza os algoritmos EXCLUSIVETRIPLESQUERYPLAN (Algoritmo 1) e SHAREDTRIPLESQUERYPLAN (Algoritmo 3) de forma auxiliar para geração de sub-planos. EXCLUSIVETRIPLESQUERYPLAN é responsável por gerar sub-planos das triplas cujo padrão podem ser encontrados em apenas uma visão exportada. De maneira similar, SHAREDTRIPLESQUERYPLAN é responsável pela geração dos sub-planos associados as triplas cujo padrão podem ser encontrados em mais de uma visão exportada.

Outros algoritmos também utilizados de forma auxiliar são TOBGP, NEWFRESHVAR e JOINCONNECTEDPLANS. Esses algoritmos não tem sua implementação em pseudo-código mostrada nessa dissertação, mas são descritos a seguir:

- TOBGP: realiza a transformação de um conjunto de triplas em um conjunto de padrões de tripla *SPARQL*;
- NEWFRESHVAR: retorna uma nova variável que não foi utilizada ainda;

- JOINCONNECTEDPLANS: recebe uma lista de sub-planos de consulta e tem como resultado um plano de consulta que conecta os sub-planos que compartilham pelo menos uma variável através de operações `JOIN`.

Passo 2: Execução da consulta para recuperação dos sujeitos relevantes

A execução do plano de consulta gerado pelo Passo 1 obtém como resultado uma lista de soluções, onde cada solução é composta por um conjunto de valorações que relacionam cada variável projetada a termos RDF (W3C, 2014b) . Cada valoração pode ser vista como uma tupla (x, v_i^x) onde x é uma variável projetada pela consulta e v_i^x é um termo RDF associado a x na solução μ_i. Vamos nos referir a esse conjunto de soluções como conjunto de soluções γ.

Algoritmo 1: EXCLUSIVETRIPLESQUERYPLAN

Entrada: $x, triples, v, linksets$

```
1  início
2      qNode ←SERVICE(v,TOBGP(triples))
3      linksets ≠ ∅ newVar ←NEWFRESHVAR()
4      sameasBGP ←TOBGP({(newVar, owl:sameAs, x)})
5      sameAsNode ← ∅
6      para cada l ∈ linksets hacer
7          sameAsNode ∈ ∅ sameAsNode ←SERVICE(l, sameasBGP)
           sameAsNode ←UNION(sameAsNode,SERVICE(l, sameasBGP))
8      fin
9      auxT ← {(newVar, p, o) | (s, p, o) ∈ triples}
10     auxNode ←SERVICE(v,TOBGP(auxT))
11     sameAsNode ←JOIN(auxNode, sameAsNode)
12     qNode ←UNION(qNode, sameAsNode)
13     qNode
14 fim
```

Algoritmo 2: BUILDQUERYPLAN

Entrada: D, δ

início

$nodeList \leftarrow \{\}$

$subjectVars \leftarrow \{s \mid (s,p,o,v) \in D\}$

para cada $x \in subjectVars$ **hacer**

$subplan \leftarrow \emptyset$

$exclTriples \leftarrow \{(s,p,o,v) \mid (s,p,o,v) \in D \wedge s = x \wedge \nexists_w((s,p,o,w) \in D \wedge w \neq v)\}$

$shrTriples \leftarrow \{(s,p,o,v) \mid (s,p,o,v) \in D \wedge s = x \wedge \exists_w((s,p,o,w) \in D \wedge w \neq v)\}$

para cada $v \in \{v \mid (s,p,o,v) \in exclTriples\}$ **hacer**

$triples \leftarrow \{(s,p,o) \mid (s,p,o,w) \in exclTriples \wedge w = v\}$

$linksets \leftarrow \{l \mid (y,l,v_s,v_t) \in \delta \wedge y = x \wedge (v_s = v \vee v_t = v)\}$

$qNode \leftarrow$ EXCLUSIVETRIPLESQUERYPLAN$(x, triples, v, linksets)$

$subplan = \emptyset\ subplan \leftarrow qNode \quad subplan \leftarrow$ `JOIN`$(subplan, qNode)$

fin

para cada $(s,p,o) \in \{(s,p,o) \mid (s,p,o,v) \in shrTriples\}$ **hacer**

$triple \leftarrow \{(s,p,o)\}$

$Vset \leftarrow \{v \mid (s1,p1,o1,v) \in shrTriples \wedge s = s1 \wedge p = p1 \wedge o = o1\}$

$linksets \leftarrow \{l \mid (y,l,v_s,v_t) \in \delta \wedge y = x \wedge (v_s \in Vset \vee v_t \in Vset)\}$

$qNode \leftarrow$ SHAREDTRIPLESQUERYPLAN$(x, triple, Vset, linksets)$

$subplan = \emptyset\ subplan \leftarrow qNode \quad subplan \leftarrow$ `JOIN`$(subplan, qNode)$

fin

$objExclTriples \leftarrow \{o \mid (s,p,o,w) \in exclTriples\} \cap subjectVars$

$objShrTriples \leftarrow \{o \mid (s,p,o,w) \in shrTriples\} \cap subjectVars$

$subjectVarsUsed \leftarrow \{x\} \cup objExclTriples \cup objShrTriples$

$nodeList \leftarrow nodeList \cup (starVarsUsed, subplan)$

fin

$queryPlan \leftarrow$ JOINCONNECTEDPLANS$(nodeList, \gamma)$

$queryPlan$

fim

Algoritmo 3: SHAREDTRIPLESQUERYPLAN

Entrada: $x, triple, Vset, linksets$

início

$qNode \leftarrow \emptyset$

para cada $v \in Vset$ **hacer**

$qNode = \emptyset\ qNode \leftarrow$ `SERVICE`$(v,$TOBGP$(triple))$

$qNode \leftarrow$ `UNION`$(qNode,$`SERVICE`$(v,$TOBGP$(triple))$

fin

$linksets \neq \emptyset$

$newVar \leftarrow newFreshVar()$

$auxT \leftarrow \{(newVar, p, o) \mid (s, p, o) \in triple\}$

$auxNode \leftarrow \emptyset$

para cada $v \in Vset$ **hacer**

$auxNode = \emptyset\ auxNode \leftarrow$ `SERVICE`$(v,$TOBGP$(auxT))$

$auxNode \leftarrow$ `UNION`$(auxNode,$`SERVICE`$(v,$TOBGP$(auxT))$

fin

$sameasBGP \leftarrow$TOBGP$(\{(newVar,$ **owl:sameAs**$, x)\})$

$sameAsNode \leftarrow \emptyset$

para cada $l \in linksets$ **hacer**

$sameAsNode \in \emptyset\ sameAsNode \leftarrow$ `SERVICE`$(l, sameasBGP)$

$sameAsNode \leftarrow$ `UNION`$(sameAsNode,$`SERVICE`$(l, sameasBGP))$

fin

$sameAsNode \leftarrow$ `JOIN`$(auxNode, sameAsNode)$

$qNode \leftarrow$ `UNION`$(qNode, sameAsNode)$

qNode

fim

Passo 3: Construção e execução das consultas para extração das propriedades dos sujeitos relevantes

O processo de extração dos dados para materialização de *VMD* é feito através de consultas SPARQL CONSTRUCT. Para tanto, são utilizadas o conjunto de quadras *D*, resultante da Etapa 2. Cada consulta é gerada em torno de uma das variáveis relevantes com o objetivo de extrair um 'pedaço' do grafo publicado por cada visão exportada. Em outras palavras, cada consulta tem apenas uma das variáveis relevantes como sujeito de seus padrões de tripla, focando buscar as propriedades associadas aos objetos ligados aquela variável. Outras variáveis relevantes podem aparecer na posição de objeto dos padrões de triplas.

Antes da execução de cada consulta, um conjunto de valores é associado as variáveis relevantes daquela consulta. Esse conjunto de valores é construído a partir do conjunto de soluções em γ, resultado do Passo 2. O conjunto δ também é utilizado para complementar as valorações possíveis para variável relevante que ocupa a posição de sujeito de cada consulta.

Passo 4: Definição e aplicação das regras de fusão

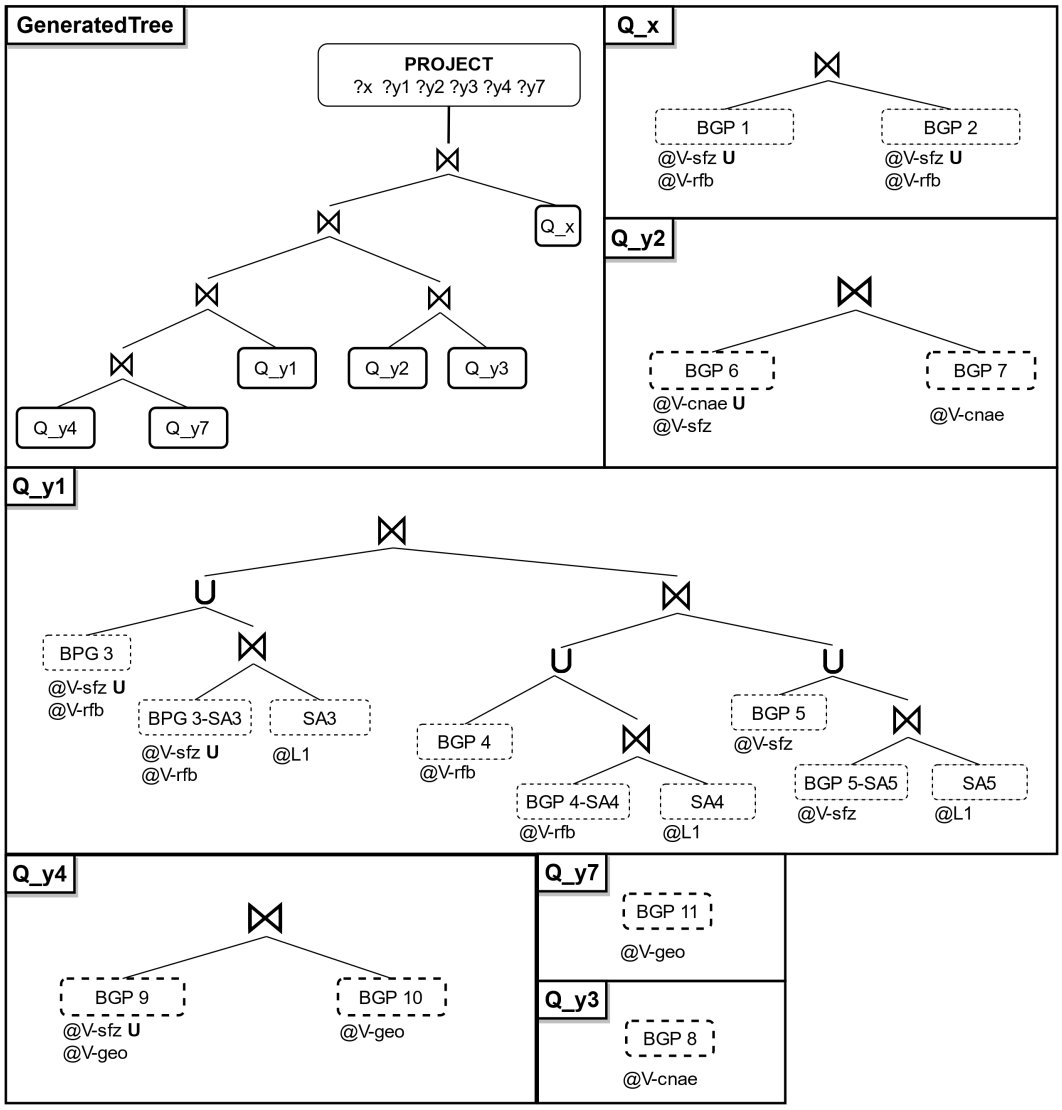
Nessa etapa, são definidas e aplicadas as regras de fusão sobre os sujeitos relevantes. Essas regras tem dois objetivos principais: (1) definir como deve ser a representação unificada para os sujeitos relevantes com múltiplas representações; (2) definir qual tratamento adequado para as propriedades que podem ter seus valores provenientes de diferentes visões exportadas (com exceção da propriedade **rdf:type**).

Exemplo 3 (*Running Example* - Etapa 3) ***Passo 1:*** *Utilizando como entrada D_Q e δ_Q (resultantes da Etapa 2), a execução do algoritmo* BUILDQUERYPLAN *terá como resultado a árvore* ***GeneratedTree*** *apresentada na figura abaixo.*

BGP 1 {?x rdf:type sfz:Organization }
BGP 2 {?x sfz:has_establishment ?y1 .}
BGP 3 {?y1 rdf:type sfz:Establishment .}
BGP 4 {?y1 sfz:has_economic_activity ?y2}
BGP 5 {?y1 sfz:has_address ?y4.}
BGP 6 {?y2 rdf:type sfz:CNAE_Subclass.}
BGP 7 {?y2 skos:broader ?y3. }
BGP 8 {
?y3 rdf:type sfz:CNAE_Class.
?y3 sfz:cnae ?y3_cnae.
VALUES ?y3_cnae {'1112'} .
}
BGP 9 {?y4 rdf:type sfz:Address.}
BGP 10 {
?y4 geo:lat ?y5 .
?y4 geo:long ?y6.
?y4 geo:has_city ?y7.
}

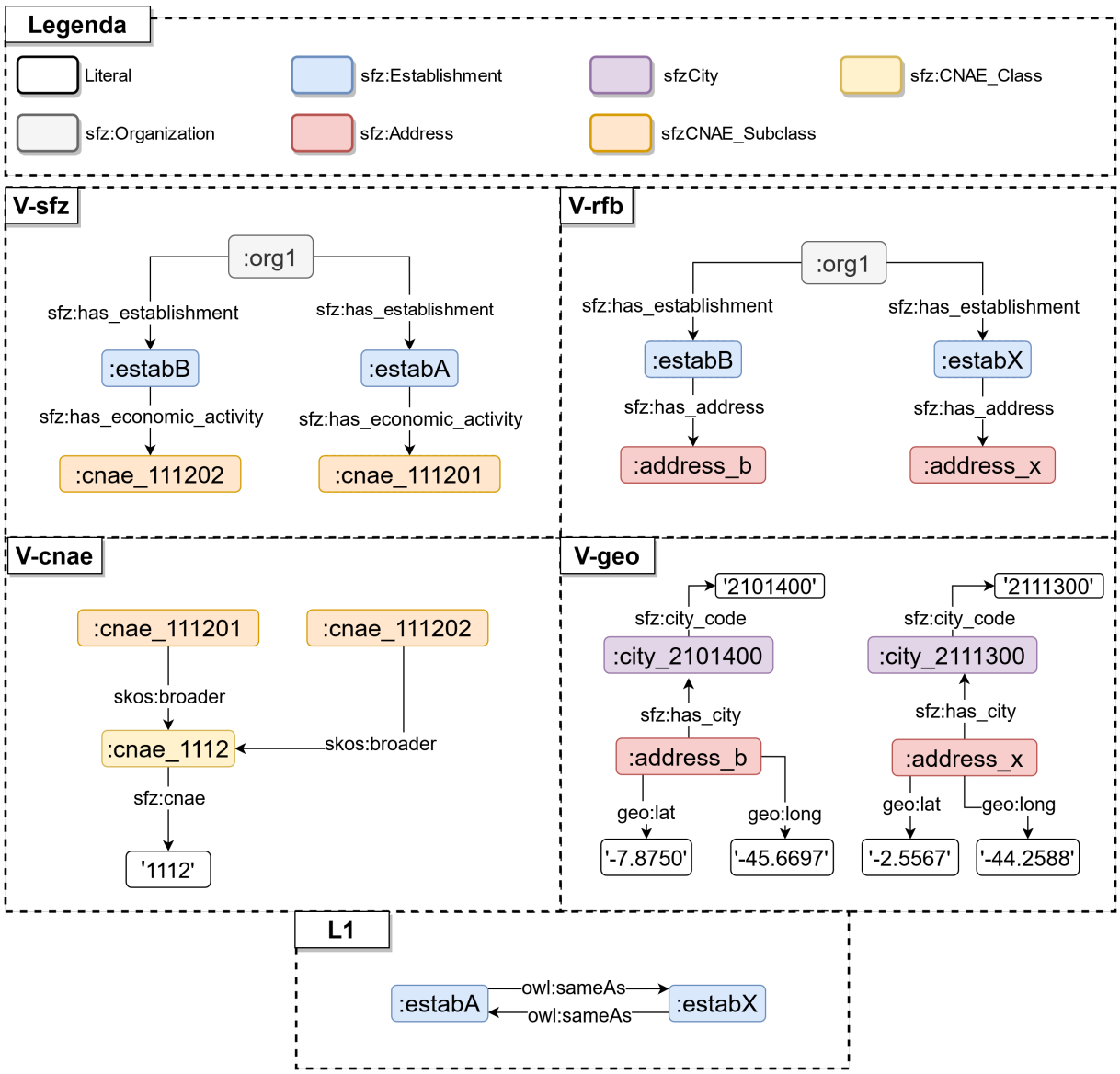
BGP 11 {
?y7 rdf:type sfz:City.
?y7 geo:city_code ?y7_city_code .
VALUES ?y7_city_code {'2111300'}
}

SA3 {?SA3 owl:sameAs ?y1}
SA4 {?SA4 owl:sameAs ?y1}
SA5 {?SA5 owl:sameAs ?y1}
BGP 3-SA3 {?SA3 rdf:type sfz:Establishment .}
BGP 4-SA4 {?SA4 sfz:has_economic_activity ?y2}
BGP 5-SA5 {?SA5 sfz:has_address ?y4.}



A árvore ***GeneratedTree*** *é formada pela junção das subárvores Q_x, Q_y1, Q_y2, Q_y3, Q_y4 e Q_y7. Cada uma dessas subárvores é gerada dentro do algoritmo* BUILDQUERYPLAN, *Linhas 4-35, e conectadas por operações de junção pelo algoritmo* JOINCONNECTEDPLANS *(Linha 36).*

Passo 2: *Para exemplificar o resultado deste passo, e dos passos seguintes, vamos assumir a seguinte configuração para visão linkset L1 e para visões exportadas V-sfz, V-rfb, V-cnae e V-geo:*



A partir dessa configuração, a execução da árvore ***GeneratedTree*** *sobre as visões do EKG tem como resultado o conjunto* $\gamma = \{\mu_1\}$*, onde :*

$$\mu_1 = \{(x,:org1),(y1,:estabA),(y2,:cnae\_111201),(y3,:cnae\_1112),$$
$$(y4,:address\_x),(y7,:city\_2111300)\}$$

Passo 3: *Utilizando* D_Q *são construídas as consultas SPARQL CONSTRUCT para extração. Vamos exemplificar mostrando apenas as consultas construídas tendo como foco a variável* $y1$*:*

Qy1-rfb

```
CONSTRUCT {
    ?y1 rdf:type ?y1_type .
    ?y1 sfz:has_economic_activity ?y2.
}
WHERE {
    OPTIONAL {
        ?y1 rdf:type ?y1_type.
        VALUES ?y1_type {sfz:Establishment}
     }
    ?y1 sfz:has_economic_activity ?y2 .
}
```

Qy1-sfz

```
CONSTRUCT {
    ?y1 rdf:type ?y1_type.
    ?y1 sfz:has_address ?y4 .
}
WHERE {
    OPTIONAL {
        ?y1 rdf:type ?y1_type.
        VALUES ?y1_type {sfz:Establishment}
    }
    ?y1 sfz:has_address ?y4.
}
```

Após a construção dessas consultas, o conjunto de valores que serão ligados as variáveis relevantes de cada consulta é construído a partir de γ*:*

$$y1_{rfb} = \{\{(y1,:estabA),(y2,:cnae\_111201)\}\}$$
$$y1_{sfz} = \{\{(y1,:estabA),(y4,:address\_x)\}\}$$

Como existem visões linkset relevantes para $y1$ *em* δ_D*, os link* `owl:sameAs` *relacionados aos valores de y1 são extraídos de suas visões linkset e utilizados para 'completar' os conjuntos* $y1_{rfb}$ *e* $y1_{sfz}$ *:*

$$y1_{rfb} = \{\{(y1,:estabA),(y2,:cnae\_111201)\},\{(y1,:estabX),(y2,:cnae\_111201)\}\}$$
$$y1_{sfz} = \{\{(y1,:estabA),(y4,:address\_x)\},\{(y1,:estabX),(y4,:address\_x)\}\}$$

Durante a execução das consultas sobre as visões exportadas, as variáveis relevantes nas consultas Qy1-rfb e Qy1-sfz tem seus valores ligados as soluções de $y1_{rfb}$ *e* $y1_{sfz}$*, respectivamente. Essas consultas ficam da seguinte forma:*

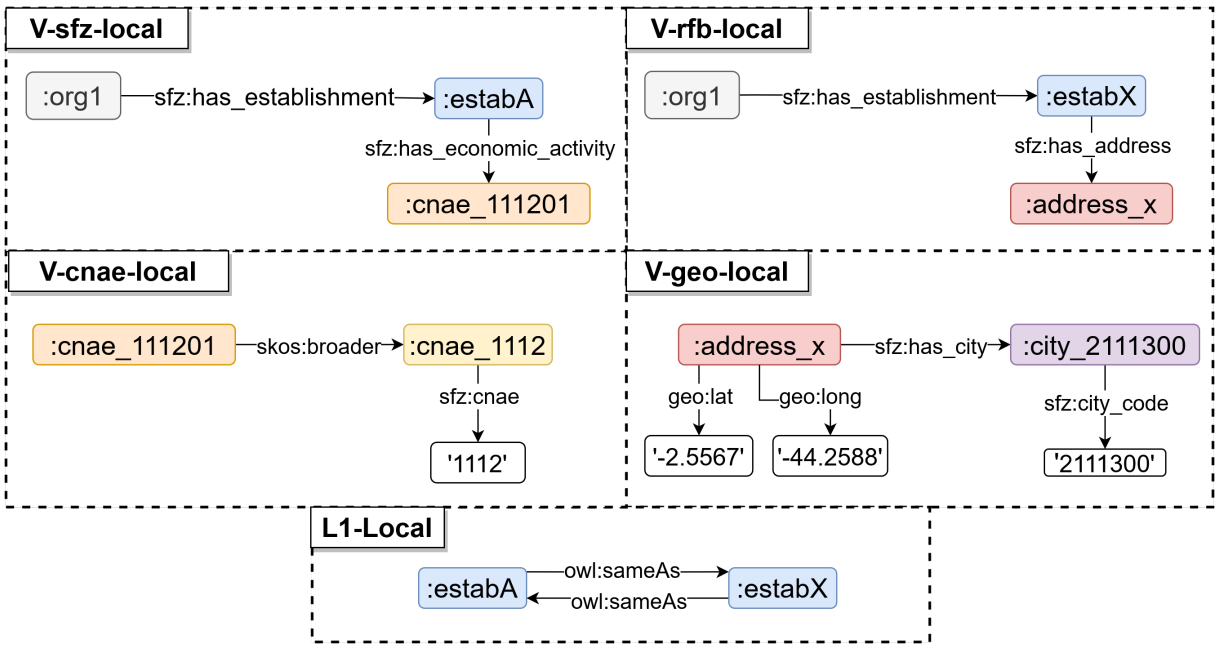
Qy1-rfb com valores de $y1_{rfb}$

```
CONSTRUCT {
    ?y1 rdf:type ?y1_type.
    ?y1 sfz:has_economic_activity ?y2.
}
WHERE {
    OPTIONAL {
        ?y1 rdf:type ?y1_type.
        VALUES ?y1_type {sfz:Establishment}
     }
    ?y1 sfz:has_economic_activity ?y2.

    VALUES (?y1 ?y2){
        (:estabA :cnae_111201)
        (:estabX :cnae_111201)
    }
}
```

Qy1-sfz com valores de $y1_{sfz}$

```
CONSTRUCT {
    ?y1 rdf:type ?y1_type.
    ?y1 sfz:has_address ?y4.
}
WHERE {
    OPTIONAL {
        ?y1 rdf:type ?y1_type.
        VALUES ?y1_type {sfz:Establishment}
    }
    ?y1 sfz:has_address ?y4.

    VALUES (?y1 ?y4){
        (:estabA :address_x)
        (:estabX :address_x)
    }
}
```

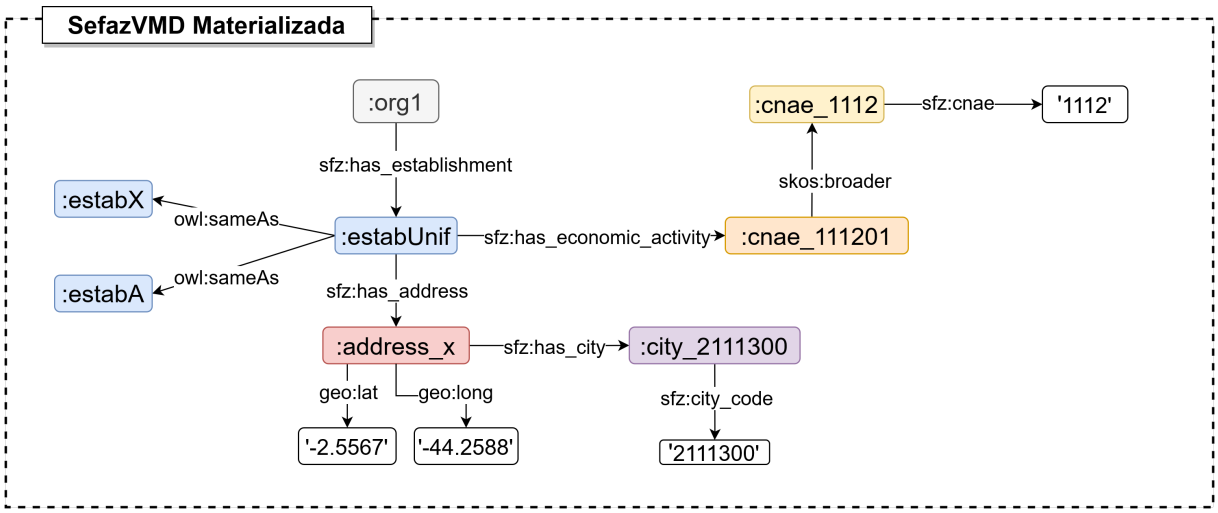
A construção e execução das consultas de extração para as outras variáveis relevantes é bem similar ao exemplificado para variável $y1$*. Como resultado, o Passo 3 tem como resultado os seguintes grafos locais:*



Passo 4: *No passo final, são definidas as regras de fusão para se conseguir uma representação unificada sobre os recursos com múltiplas representações e resolver possíveis inconsistências existentes nos valores das propriedades provenientes de fontes distintas. Analisando D_Q e δ_Q, é possível dizer que são necessárias duas regras de fusão:*

- *A primeira é definida sobre* sfz:has_establishment*, uma propriedade que pode ter seus valores obtidos das visões exportadas* V-sfz *e* V-rfb*. Para este caso, adotamos uma regra que apenas mantém todos os valores existente para aquela propriedade;*
- *A segunda regra é definida sobre os recursos ligados por links* `owl:sameAs` *em L1-local. Essas representações são substituídas por uma nova representação gerada e são criados links* `owl:sameAs` *que ligam a representação gerada as representações originais.*

A aplicação dessas regras finaliza o processo de materialização, resultando no grafo abaixo:



7 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho, propomos uma abordagem para construção de *mashup* de dados como uma visão sobre um *EKG*. Nesta abordagem, consideramos que o *EKG* foi implementado a partir de uma visão semântica composta por uma ontologia de domínio e um conjunto de especificações de visões exportadas e *linkset*. Considerando isso, a construção do *mashup* se inicia pela especificação de uma visão de *mashup* definida como uma consulta facetada. Essa consulta facetada é construída utilizando os termos da ontologia de domínio do *EKG*.

Com exceção da definição das regras de fusão, todo o processo de extração dos recursos e consolidação do *mashup* pode ser feito de forma automática através do processo apresentado por esta dissertação. O resultado final deste processo é um *mashup* de dados, que pode ser entendido como um *KG*, que possui a estrutura definida pela visão de *mashup*.

Nesta dissertação, tivemos como foco a formalização das etapas necessárias para execução da abordagem proposta. A partir do que foi apresentado, acreditamos que os seguintes trabalhos futuros sejam possíveis:

- Implementação da abordagem semiautomática proposta para construção de *mashups* formalizada por esta dissertação;
- Implementação de uma interface que utilize o conceito de consulta facetada para facilitar a especificação de uma Visão de *Mashup*;
- Extensão da abordagem para suportar a definição de visões mais complexas.

REFERÊNCIAS

ARENAS, M.; GUTIERREZ, C.; PÉREZ, J. Foundations of RDF databases. In: **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**. [*S. l.*: *s. n.*], 2009. v. 5689 LNCS, p. 158–204. ISBN 3642037534.

BERNERS-LEE, T. **Linked Data - Design Issues**. [*S. l.*], 2006. Disponível em: http://www.w3.org/DesignIssues/LinkedData.html. Acesso em: 20 jan. 2021.

BISHR, Y. Overcoming the semantic and other barriers to GIS interoperability. **International Journal of Geographical Information Science**, v. 12, n. 4, p. 299–314, jun 1998. ISSN 1365-8816.

BLEIHOLDER, J.; NAUMANN, F. Data Fusion. **ACM Computing Surveys**, v. 41, n. 1, p. 1–41, 2009. ISSN 15577341.

CALVANESE, D. *et al.* Ontop: Answering SPARQL queries over relational databases. **Semantic Web**, v. 8, n. 3, p. 471–487, 2017. ISSN 22104968.

CAVALCANTE, G. M. L. *et al.* **MAURA: Um framework baseado em Mediador Semântico para Construção Eficiente de Linked Data Mashups**. Dissertação (Mestrado) – Instituto Federal de Educação, Ciência e Tecnologia do Ceará, 2017.

CRUZ, M. M. L. *et al.* Semanticsus: Um portal semântico baseado em ontologias e dados interligados para acesso, integração e visualização de dados do sus. In: **Anais Estendidos do XIX Simpósio Brasileiro de Computação Aplicada à Saúde**. Porto Alegre, RS, Brasil: SBC, 2019. p. 13–18. ISSN 0000-0000.

EHRLINGER, L.; WÖSS, W. Towards a definition of knowledge graphs. **CEUR Workshop Proceedings**, v. 1695, 2016. ISSN 16130073.

FREITAS, R. *et al.* Using linked data in the data integration for maternal and infant death risk of the sus in the gissa project. In: **Proceedings of the 23rd Brazillian Symposium on Multimedia and the Web**. New York, NY, USA: Association for Computing Machinery, 2017. (WebMedia '17), p. 193–196. ISBN 9781450350969.

GALKIN, M.; AUER, S.; SCERRI, S. Enterprise Knowledge Graphs: A Backbone of Linked Enterprise Data. **Proceedings - 2016 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2016**, p. 497–502, 2017.

HOGAN, A. *et al.* Knowledge graphs. **arXiv**, 2020. ISSN 23318422.

KNAP, T. *et al.* Odcleanstore: a framework for managing and providing integrated linked data on the web. In: SPRINGER. **International Conference on Web Information Systems Engineering**. [*S. l.*], 2012. p. 815–816.

KNAP, T. *et al.* UnifiedViews: An ETL Tool for RDF Data Management. **Semantic Web**, v. 0, n. 0, p. 1–16, 2018. ISSN 00319007.

KNOBLOCK, C. A. *et al.* Semi-automatically mapping structured sources into the semantic web. In: **The Semantic Web: Research and Applications**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 375–390. ISBN 978-3-642-30284-8.

Madhavan, J. *et al.* Web-scale data integration: You can only afford to pay as you go. In: **CIDR**. [*S. l.*: *s. n.*], 2007. p. 342–350.

MENDES, P. N.; MÜHLEISEN, H.; BIZER, C. Sieve: Linked Data quality assessment and fusion. **ACM International Conference Proceeding Series**, p. 116–123, 2012.

NENTWIG, M. *et al.* A survey of current Link Discovery frameworks. **Semantic Web**, v. 8, n. 3, p. 419–436, 2017. ISSN 22104968.

NGOMO, A. C. N.; AUER, S. LIMES - A time-efficient approach for large-scale link discovery on the web of data. **IJCAI International Joint Conference on Artificial Intelligence**, n. January, p. 2312–2317, 2011. ISSN 10450823.

PANKOWSKI, T. Rewriting and Executing Faceted Queries over Ontology-Enhanced Databases. **Procedia Computer Science**, Elsevier B.V., v. 112, p. 137–146, 2017. ISSN 18770509.

PAULHEIM, H. Knowledge graph refinement: A survey of approaches and evaluation methods. **Semantic Web**, v. 8, n. 3, p. 489–508, dec 2016. ISSN 22104968.

ROLIM, T. *et al.* Um enfoque incremental para construção do grafo de conhecimento do SUS. In: **Anais do XX Simpósio Brasileiro de Computação Aplicada à Saúde**. Porto Alegre, RS, Brasil: SBC, 2020. p. 72–83. ISSN 0000-0000.

SCHULTZ, A. *et al.* LDIF -Linked Data Integration Framework. **CEUR Workshop Proceedings**, v. 782, p. 1–6, 2011. ISSN 16130073.

SOYLU, A. *et al.* OptiqueVQS: A visual query system over ontologies for industry. **Semantic Web**, v. 9, n. 5, p. 627–660, 2018. ISSN 15700844.

STUDER, R.; BENJAMINS, V.; FENSEL, D. Knowledge engineering: Principles and methods. **Data & Knowledge Engineering**, v. 25, n. 1-2, p. 161–197, mar 1998. ISSN 0169023X.

VIACAVA, F. *et al.* SUS: oferta, acesso e utilização de serviços de saúde nos últimos 30 anos. **Ciência & Saúde Coletiva**, v. 23, n. 6, p. 1751–1762, jun 2018. ISSN 1678-4561.

VIDAL, V. M. P. *et al.* Specification and incremental maintenance of linked data mashup views. In: **Advanced Information Systems Engineering**. Cham: Springer International Publishing, 2015. p. 214–229. ISBN 978-3-319-19069-3.

VOLZ, J. *et al.* Silk - A Link Discovery Framework for the Web of Data. **CEUR Workshop Proceedings**, v. 538, 2009. ISSN 16130073.

W3C. **Semantic Web**. 2013. Disponível em: https://www.w3.org/2001/sw/. Acesso em: 20 jan. 2021.

W3C. **RDF 1.1 Concepts and Abstract Syntax**. 2014. Disponível em: https://www.w3.org/TR/rdf11-concepts/. Acesso em: 20 jan. 2021.

W3C. **RDF 1.1 Concepts and Abstract Syntax**. 2014. Disponível em: https://www.w3.org/TR/rdf11-concepts/. Acesso em: 18 jan. 2021.

XIAO, G. *et al.* Virtual Knowledge Graphs: An Overview of Systems and Use Cases. **Data Intelligence**, v. 1, n. 3, p. 201–223, 2019.