INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA FLUMINENSE

PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS APLICADOS À ENGENHARIA E GESTÃO

Adriano de Oliveira Gonçalves

UM MÉTODO E UMA ARQUITETURA PARA INTEGRAÇÃO LINKED DATA DE SISTEMAS DE INFORMAÇÃO DISTINTOS

Campos dos Goytacazes / RJ

2020

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA FLUMINENSE

PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS APLICADOS À ENGENHARIA E GESTÃO

ADRIANO DE OLIVEIRA GONÇALVES

UM MÉTODO E UMA ARQUITETURA PARA INTEGRAÇÃO LINKED DATA DE SISTEMAS DE INFORMAÇÃO DISTINTOS

Mark Douglas de Azevedo Jacyntho

(Orientador)

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação do Instituto Federal de Educação, Ciência e Tecnologia Fluminense, no Curso de Mestrado Profissional em Sistemas Aplicados à Engenharia e Gestão (MPSAEG), como parte dos requisitos necessários à obtenção do título de Mestre em Sistemas Aplicados à Engenharia e Gestão.

Campos dos Goytacazes / RJ
2020

Biblioteca Anton Dakitsch
CIP - Catalogação na Publicação

G635m

Gonçalves, Adriano de Oliveira
Um Método e uma Arquitetura para Integração Linked Data de Sistemas de Informação Distintos / Adriano de Oliveira Gonçalves - 2020.
95 f.: il. color.

Orientador: Mark Douglas de Azevedo Jacyntho

Dissertação (mestrado) -- Instituto Federal de Educação, Ciência e Tecnologia Fluminense, Campus Campos Centro, Curso de Mestrado Profissional em Sistemas Aplicados à Engenharia e Gestão, Campos dos Goytacazes, RJ, 2020.
Referências: f. 84 a 89.

1. Integração de Dados. 2. Mapeamento Relacional para RDF. 3. Dados Ligados. 4. Ontologia. 5. Web Semântica. I. Jacyntho, Mark Douglas de Azevedo, orient. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da Biblioteca Anton Dakitsch do IFF com os dados fornecidos pelo(a) autor(a).

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA FLUMINENSE

PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS APLICADOS À ENGENHARIA E GESTÃO

Adriano de Oliveira Gonçalves

UM MÉTODO E UMA ARQUITETURA PARA INTEGRAÇÃO LINKED DATA DE SISTEMAS DE INFORMAÇÃO DISTINTOS

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação do Instituto Federal de Educação, Ciência e Tecnologia Fluminense, no Curso de Mestrado Profissional em Sistemas Aplicados à Engenharia e Gestão (MPSAEG), como parte dos requisitos necessários à obtenção do título de Mestre em Sistemas Aplicados à Engenharia e Gestão.

Aprovado(a) em 30 de setembro de 2020.

Banca Examinadora:

_____

Mark Douglas de Azevedo Jacyntho, D.Sc.
Instituto Federal de Educação, Ciência e tecnologia Fluminense (IFF)
(Orientador)

_____

Adriana Pereira de Medeiros, D.Sc.
Universidade Federal Fluminense (UFF)

_____

Aline Pires Vieira de Vasconcelos, D.Sc.
Instituto Federal de Educação, Ciência e tecnologia Fluminense (IFF)

_____

Thiago Ribeiro Nunes, D.Sc.
Instituto Federal de Educação, Ciência e tecnologia Fluminense (IFF)

AGRADECIMENTOS

A *Deus*, porque todas as coisas foram feitas por Ele, e sem Ele nada do que foi feito se fez. Creio que Ele é o autor das maiores soluções da vida, e que sobre elas todas as demais são construídas.

Ao meu pai, *Adailton Santana Gonçalves*, por sempre ter me ensinado com palavras, e me mostrado com exemplo, que a vida se constrói e se resolve com objetivo, trabalho, honestidade, humildade, criatividade e temor a Deus (e que se adicionar bom humor fica melhor). Por me estimular a não desistir quando as coisas dão errado e por despertar em mim a curiosidade e a paixão pela tecnologia.

À minha mãe *Sandra Lúcia de Oliveira Gonçalves*, por todo amor, carinho e sabedoria, sempre presentes, por me ensinar e mostrar que, através do estudo, da pesquisa e da busca pelo conhecimento, adquire-se incomensuráveis ferramentas para a vida. E por sempre me estimular a perseverar no bom caminho.

À minha esposa *Livia dos Santos da Rosa Gonçalves*, por sempre me incluir nas suas orações, por diversas vezes me tirar as dúvidas de língua portuguesa durante a escrita deste trabalho e por não permitir que eu me acomodasse nos momentos em que estive mais desanimado. Amo você.

Ao meu orientador *Mark Douglas Jacyntho*, com quem tive a honra de estudar neste curso e de trabalhar neste projeto de pesquisa, por todo o tempo, atenção, confiança, estímulo e conhecimento investidos em mim, e pelos exemplos que certamente levarei para toda a vida.

Aos professores do Mestrado em Sistemas Aplicados a Engenharia e Gestão (SAEG), do IFFluminense, por todo conhecimento transmitido, pelas oportunidades de tocar diferentes áreas de conhecimento e de explorar novos horizontes, e pelas ferramentas que adquiri para a vida durante este curso.

Aos integrantes da banca de Qualificação e de Defesa desta Dissertação, pelo tempo e atenção dedicados e por todas as valiosas contribuições.

Aos servidores administrativos do IFFluminense, pela atenção e orientações fornecidas.

Aos meus amigos *Adriano Souza* e *Patrick Belém*, atualmente meus superiores no setor de TIC da UFRJ, por, em especial, me incentivarem e me ajudarem muito, antes e

durante este curso, e por sempre proporcionarem condições favoráveis à minha dedicação aos estudos. Ao *Adriano Souza* por me apresentar, pela primeira vez, a Web Semântica.

A todos os amigos e colegas na Coordenação de TIC da UFRJ em Macaé-RJ, tanto na Equipe de Desenvolvimento quanto na Equipe de Infraestrutura, que, direta ou indiretamente, contribuíram para este trabalho. Agradeço a prontidão e disposição em ajudar sempre que precisei. Em especial ao *Helder Cosme*, que contribuiu de forma significativa na implementação e execução dos testes, essenciais aos resultados desta obra, e ao *João Paulo Glória*, que me apoiou nas demandas de trabalho nos momentos em que estive assoberbado com as atribuições do curso.

À coordenadora do SAEG, *Simone Vasconcelos*, por sempre me atender e me ajudar, com compreensão, nas dúvidas e necessidades que tive.

À Universidade Federal do Rio de Janeiro e ao Instituto Federal Fluminense, por me proporcionarem esta oportunidade de trabalho e pesquisa.

RESUMO

Os Sistemas de Informação têm ocupado, ao longo dos anos, um lugar de suma importância na rotina de empresas de todos os portes, seja no armazenamento e consulta de dados, assim como no gerenciamento de processos e tomadas de decisão. Diante disso, é comum encontrar, nas instituições, a existência de diversos sistemas diferentes que não se comunicam. A falta de um modelo padronizado para intercâmbio de informação entre tais sistemas culmina em problemas tais como duplicidade, inconsistências de dados e dificuldade de acesso integrado de informações. A Web Semântica provê um conjunto de padrões e tecnologias que visam apoiar o ser humano na busca, integração e processamento de informações, possibilitando o compartilhamento e reúso do conhecimento de forma automatizada. Este trabalho tem como objetivo propor um método e uma arquitetura para a integração de dados de sistemas de informação distintos, usando, como prova de conceito, dois estudos de caso reais, baseados em sistemas de uma grande universidade federal brasileira. Seguindo-se uma trajetória metodológica de pesquisa qualitativa e aplicada, a abordagem proposta foi implementada e testes foram realizados, diversificando-se os cenários. Diante dos resultados alcançados, nos quais observou-se viabilidade e desempenho adequados frente aos problemas apresentados, obteve-se êxito quanto aos objetivos estabelecidos. Estes sugerem que a Web Semântica surge como uma plataforma adequada à integração de dados dentro de uma instituição, sendo uma alternativa aos *data warehouses* convencionais. Espera-se que o método e arquitetura propostos sirvam como diretrizes para integração sistematizada de dados dentro da instituição em estudo, bem como em outras organizações, dado que se trata de um problema bastante recorrente nos dias atuais.

Palavras-chave: Integração Semântica de Dados, Mapeamento Relacional-RDF, Ontologia.

ABSTRACT

Information Systems, over the years, have a prime importance in the routine of companies of all sizes, whether in data storage and consultation, as well as in process management and decision making. In this picture, it is common to find, in institutions, the existence of several different systems that do not communicate. The lack of a standardized model for information exchange between such systems culminates in problems such as duplicity, data inconsistencies and difficulty of integrated access. The Semantic Web provides a set of standards and technologies that aims to support humans in the search, integration and processing of information, enabling the sharing and reuse of knowledge in an automated way. This work aims to propose a method and architecture for the data integration of distinct information systems, using, as proof of concept, two real case studies, based on systems of a large brazilian federal university. Following a methodological trajectory of qualitative and applied research, the proposed approach was implemented and tested, considering different sceneries. In view of the achieved results, in which adequate viability and performance were observed in the face of the exposed problems, the established objectives were reached. The results suggest that the Semantic Web raises as an adequate platform for data integration within an institution, being an alternative to conventional data warehouses. The proposed method and architecture are expected to serve as guidelines for systematized data integration within the institution under study, as well as in other organizations, as it is a very recurrent problem nowadays.

Keywords: *Semantic Data Integration, Relational-RDF Mapping, Ontology.*

LISTAS

Lista de ilustrações

Lista de tabelas

SUMÁRIO

1 INTRODUÇÃO

Os Sistemas de Informação (SI) têm ocupado, ao longo dos anos, um lugar de inegável importância na rotina de empresas de todos os portes, no armazenamento e consulta de dados, assim como no gerenciamento dos processos e na tomada de decisões. Diante disso, é comum encontrar, em grandes instituições, a existência de diversos sistemas diferentes que visam, cada um, atender a domínios específicos (Cverdelj-Fogaraši *et al.*, 2017). Contudo, a falta de um modelo padronizado de informação e de interconectividade entre tais sistemas, acarreta problemas como duplicidade, inconsistências de dados e dificuldade no acesso de informações integradas.

Tal realidade também pode ser encontrada em grandes universidades, que precisam contar com sistemas para gerenciamento de seus processos acadêmicos e administrativos, fragmentados por diversos departamentos e ligados a bases de dados relacionais heterogêneas, como, por exemplo: sistema para gerenciamento acadêmico, sistema para gerenciamento de recursos humanos (RH), sistemas para gerenciamento de portais, para gerenciamento de eventos e conferências, de avaliações institucionais, bolsas, monitorias, etc.

Neste cenário, surge, naturalmente, a crescente necessidade de integrar esses diversos sistemas de informação independentes dentro de uma organização, combinando recursos de múltiplos bancos de dados.

Tradicionalmente, essa integração de dados requer a construção de um *data warehouse*, um repositório centralizado de dados, com foco em otimização de consultas sem, no entanto, impactar nos bancos de dados originais. Tipicamente, a construção de um *data warehouse* envolve soluções proprietárias caras e demanda bastante esforço e tempo. Basicamente, esse esforço compreende processos de extração de dados dos repositórios originais, sua respectiva transformação, juntamente com alinhamento de diferentes esquemas de dados, e carregamento destes dados transformados no *data warehouse*, (Wood *et al.*, 2013). Este conjunto de processos é conhecido por *Extraction, Transformation and Load (ETL)*.

Com o objetivo de fornecer auxílio ao ser humano na busca, integração e processamento de informações, surge a Web Semântica, inicialmente proposta por Tim Berners-Lee, em Berners-Lee, Hendler e Lassila (2001), e fomentada pelo consórcio W3C (W3C, 2018). A Web Semântica é uma extensão da Web original, que consiste em um conjunto de tecnologias e padrões que viabiliza o compartilhamento e reúso de conhecimento,

em escala global, por meio de formatos de dados inteligíveis por máquinas, de forma que estas possam executar automaticamente, em larga escala, as tarefas que são executadas manualmente até então (Azevedo & Jacyntho, 2014). A utilização destes padrões tem se tornado uma tendência mundial, com considerável adesão nos meios científicos.

Os padrões e tecnologias da Web Semântica vêm ao encontro dessa necessidade de integração de dados dentro de uma empresa. Os mesmos princípios usados para integrar fontes de dados geograficamente distribuídas na Web, conhecidos como princípios *Linked Data* (Berners-Lee, 2009), podem ser reusados para combinar informações de banco de dados distintos dentro de uma organização, em um ambiente de Intranet.

A Web Semântica e os princípios *Linked Data* trazem quatro mecanismos inerentes que facilitam, sobremaneira, a integração de múltiplas fontes de dados, a saber (Wood *et al.*, 2013):

- Reúso dos mesmos endereços Web (URIs) para identificar recursos (objetos), sempre que possível;
- Reúso de vocabulários (ontologias) comuns, sempre que possível;
- Uso do predicado padrão *owl:sameAs* para indicar que um recurso particular é idêntico (ou quase idêntico) a outro recurso;
- Reaproveitamento da infraestrutura pré-existente do protocolo HTTP da Web original.

Trata-se de mecanismos padrão muito bem especificados e independentes de implementação proprietária, que fazem com que a Web Semântica apareça como alternativa aos *data warehouses* tradicionais, com grande potencial. Em contrastes com estes últimos, que possuem um custo alto e concentrado de implementação, as tecnologias *Linked Data* permitem que instituições sejam capazes de organizar a sua estrutura de acesso aos dados com esforço relativamente reduzido, podendo investir, passo a passo, no estabelecimento de *links* (ligações) de dados, vocabulários compartilhados e mapeamentos de origens de dados, permitindo, assim, uma integração mais aprofundada. (Heath & Bizer, 2011). Outra diferença que vale destacar é que, em processos tradicionais de ETL, o esforço da integração recai sobre o consumidor dos dados, uma vez que a inclusão de cada nova origem de dados envolve algum tipo de implementação para extração, transformação e carregamento. Com a utilização dos princípios e tecnologias da Web Semântica, o responsável por cada origem de dados pode contribuir simplesmente publicando-os em formato e vocabulários padronizados, sem alterar em nada as aplicações e bancos de dados originais. Desta forma, a integração de dados

acontece de forma automática, facilitando, assim, a utilização pelos consumidores de dados (Heath & Bizer, 2011).

1.1 Objetivo Principal

Como objetivo central, este trabalho visa propor um método e uma arquitetura para a integração semântica *Linked Data* de bases de dados de múltiplos sistemas de informação. Em outras palavras, integrar bases de dados independentes, transformando-as em um grafo de conhecimento semântico para facilitar e enriquecer os processos de tomada de decisão dentro de uma organização.

1.2 Objetivos Específicos

Para alcançar o objetivo geral acima apresentado, estabeleceu-se os objetivos específicos a seguir:

- Definir uma arquitetura e um método para integração de dados entre sistemas de informação independentes, utilizando as tecnologias e padrões da Web Semântica propostos pelo W3C, bem como seguindo os princípios *Linked Data* enunciados por Tim Berners-Lee;
- Propor ferramental de apoio para cada uma das etapas do método definido;
- Apresentar um estudo de caso real de instanciação da arquitetura e aplicação do método;
- Definir uma ontologia de domínio para integração dos dados do estudo de caso e mostrar como usá-la para o problema apresentado neste.

Tendo como alvo validar a solução proposta e examinar os potenciais benefícios que venham surgir da implementação de uma solução baseada em Web Semântica para o problema proposto, adotou-se, para este trabalho, o estudo de caso de sistemas corporativos em uma grande universidade federal brasileira. Pretende-se, com esta pesquisa, propor um modelo de dados, baseado em ontologias consagradas, para integração de dados entre diferentes sistemas corporativos da instituição em questão.

1.3 Contribuições

Além do método e arquitetura propostos para integração entre múltiplos bancos de dados independentes, que configuram o objetivo central deste trabalho, esta pesquisa traz ainda, como contribuições, um método para publicação de dados de bases relacionais como dados ligados (*Linked Data*) na Web, além de sugestões de ferramentas para cada um dos

passos dos métodos, dois estudos de caso reais como exemplos de sua aplicação, um modelo ontológico para o domínio de cada estudo de caso e estratégias/diretrizes para lidar com os desafios da implementação deste tipo de solução. O método para publicação de dados de bases relacionais na Web, embora faça parte do método de integração de dados, pode ser utilizado de forma independente por aquele que deseje apenas publicar dados relacionais como *Linked Data* na Web, consistindo assim, em mais uma relevante contribuição.

O modelo ontológico apresentado para integração de dados pode ser sistematicamente empregado para interligação de bases de sistemas corporativos dentro de universidades, por meio de um grafo semântico. Da mesma forma, o modelo ontológico apresentado para publicação de dados pode ser sistematicamente reusado para publicação e interligação de produções acadêmicas na Web de Dados. Espera-se que o modelo proposto neste trabalho sirva como um guia para integração de dados dentro e fora da instituição em estudo, figurando como uma alternativa interessante aos *data warehouses* convencionais e podendo ser expandido e atualizado de acordo com novas necessidades que venham a ser identificadas. E que o mesmo possa servir como referência para outras universidades ou, ainda, outros tipos de instituições, que pretendam modelar uma solução semântica para problemas similares, contribuindo assim para o aperfeiçoamento dos processos de trabalho, incrementado pela capacidade que a Web Semântica oferece às máquinas de interligar, compreender e descobrir informações.

1.4 Estrutura do documento

Inicialmente, no Capítulo 2, são apresentados os fundamentos da Web Semântica, *Linked Data* e integração de dados relacionais através de RDF. O Capítulo 3 apresenta a trajetória metodológica percorrida para a execução desta pesquisa. No Capítulo 4, são apresentados o método e arquitetura propostos, tanto para publicação *Linked Data* quanto para integração de dados entre sistemas, os quais configuram as principais contribuições desta pesquisa. No Capítulo 5, é descrito o processo de aplicação do método e arquitetura propostos nos estudos de caso reais. Em seguida, no Capítulo 6, discussões acerca dos resultados obtidos são abordadas. Já no Capítulo 7, são elencados alguns trabalhos relacionados e, por fim, o Capítulo 8 conclui este trabalho com as considerações finais e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Web Semântica

O termo Web Semântica designa uma extensão da Web atual, de forma a prover tecnologias e padrões que permitam a adição de significado formal explícito à informação publicada, que possa ser entendido por máquinas, viabilizando assim a execução automática de tarefas até então executadas manualmente, em larga escala (Yu, 2011). De acordo com Berners-Lee et al (2001), a Web Semântica não é uma Web separada, e sim uma extensão da atual, na qual a informação recebe um significado bem definido, de forma a permitir que computadores e pessoas trabalhem cooperativamente.

Conforme Jacyntho e Azevedo (2015):

> A Web Semântica baseia-se na simples idéia de reusar a arquitetura da Web original para compartilhar e interligar metadados (ou dados diretamente, no caso de *Linked Data*), assim como os recursos (entidades) descritos por eles, de forma homogênea e padrão, abstraindo das idiossincrasias tecnológicas internas de cada fonte (servidor) de dados (Jacyntho & Azevedo, 2015, p.3) .

Na Web Semântica, também conhecida como Web de Dados, entidades do mundo real (pessoas, locais, objetos, publicações, conceitos abstratos, etc.) são representadas por meio de recursos, que são identificados por um endereço Web único, um *Uniform Resource Indentifier* (URI). Ao acessar (dereferenciar) este endereço, retorna-se um documento contendo uma descrição do recurso solicitado, utilizando uma linguagem estruturada inteligível por máquina. Neste documento, o recurso é descrito utilizando-se *links* Web que o conecta, relacionando-o semanticamente, com representações de outros recursos, geograficamente distribuídos, identificados por outros URIs, usando, pois, a Web como um grafo de dados global que cresce a cada dia (Azevedo & Jacyntho, 2014; Heath & Bizer, 2011).

Segundo Antoniou *et al.* (2012), toda linguagem para intercâmbio de dados entre máquinas é definida por três pilares, a saber: modelo de dados, semântica e sintaxe. A seguir, são apresentados estes três componentes da Web Semântica.

2.1.1 *Resource Description Framework* (RDF): Modelo de dados

RDF é o modelo de dados padrão da Web Semântica. Trata-se de um modelo em grafo, que permite a descrição de recursos por meio de triplas {recurso – propriedade – valor} (Jacyntho, 2012). Desta forma, o modelo RDF adiciona semântica explícita à estrutura de *links* da Web ao nomear o *link* com o URI da correspondente propriedade de uma ontologia, assim como o seu nó de origem (URI do recurso) e seu nó de destino (URI de outro recurso ou valor literal). A utilização deste simples modelo permite que dados (semi)estruturados sejam combinados, publicados e compartilhados entre diferentes aplicações (*RDF Working Group*, 2014). Utilizando os autores deste trabalho, poder-se-ia afirmar que "Adriano conhece Mark", criando, por exemplo, a seguinte tripla RDF:

<*http://www.example.com/mestrado/saeg/alunos/adrianooliveiragoncalves*>
<*http://xmlns.com/foaf/0.1/knows*>
<*http://www.example.com/mestrado/saeg/professores/markdouglasjacyntho*> .

Neste exemplo de tripla, o nó de origem é o URI que identifica o aluno "Adriano de Oliveira Gonçalves", a aresta é o URI da propriedade "conhece" da ontologia para descrição de pessoas *FOAF* (Brickley & Miller, 2014) e o nó de destino é o URI que identifica o professor "Mark Douglas Jacyntho". Um conjunto de triplas forma um grafo RDF.

Segundo Jacyntho e Schwabe (2016):

> Em essência, um modelo de dados é apenas uma maneira de visualizar os dados. O modelo relacional estabelecido visualiza os dados por meio de relações e tuplas. O modelo em grafo RDF, baseado em triplas, é uma representação natural para vários tipos de aplicações (por exemplo, *Facebook*, *Twitter*, sistemas de recomendação, etc.), nas quais as entidades estão fortemente conectadas umas com as outras. Em contraste com o modelo relacional, essas aplicações consideram propriedades multivaloradas tão desejáveis na modelagem de dados reais, que trabalham com propriedades multivaloradas por padrão. Consultas de propriedades com valores múltiplos ou valor único são feitas exatamente do mesmo modo, sem preocupações com a necessidade de se associar a uma terceira tabela para modelar um relacionamento n-para-n. Além disso, o modelo RDF é mais conveniente se a aplicação tiver alta heterogeneidade em seu esquema ou necessidade frequente de adaptação de esquema. Os bancos de dados RDF simplificam o desenvolvimento de aplicações de dados ligados e também se alinham muito bem com numerosos algoritmos e técnicas estatísticas desenvolvidas para grafos (Jacyntho & Schwabe, 2016, p.1, tradução nossa).

É importante destacar que não há necessidade dos diferentes recursos envolvidos em ligações semânticas estarem publicados no mesmo servidor. Este é o cerne da Web de Dados: fontes de dados distintas com recursos interligados (*mashup* semântico), promovendo o reúso das informações e permitindo à máquina navegar de um recurso para o outro, e de uma fonte para outra, extraindo mais informações, independentemente da localização dos dados (Azevedo & Jacyntho, 2014).

O modelo RDF oferece uma linguagem de consulta padrão chamada *SPARQL Protocol and RDF Query Language* (SPARQL) (*The W3C SPARQL Working Group*, 2013), que desempenha um papel análogo ao da linguagem de consulta *Structured Query Language* (SQL) nos bancos de dados relacionais.

2.1.2 Ontologias: Semântica

Um dos componentes fundamentais da Web Semântica são modelos formais de representação de conhecimento, chamados ontologias (Berners-Lee *et al.*, 2001) que, no campo da Ciência da Computação, possuem a capacidade de promover o compartilhamento e a reutilização do conhecimento (Camilo & Silva, 2009). De acordo com Gruber (1995), uma ontologia é especificação de uma conceitualização. Conforme W3C OWL Working Group (2012), ontologias são vocabulários formalizados de termos, geralmente abrangendo um domínio específico e compartilhados por uma comunidade de usuários.

A utilização de ontologias traz uma série de vantagens, como, por exemplo, o fato de, diferentemente de como acontece com a representação textual dos documentos na Web, estas serem definidas em linguagem formal, não deixando espaço para as limitações semânticas e ambiguidades de entendimento da linguagem natural. Além disso, ontologias podem ser mapeadas entre diferentes linguagens computacionais, e especializadas para domínios de conhecimento mais específicos (Souza, 2016).

Na Web Semântica, as ontologias são criadas por meio das linguagens (metaontologias) *Web Ontology Language* (OWL) (Horrocks *et al.*, 2012) e RDF *Schema* (RDFS) (Brickley *et al.*, 2014), linguagens declarativas baseadas em lógica descritiva. Como ontologias OWL e RDFS são documentos RDF, as classes e propriedades que compõem seu vocabulário também são identificados por endereços Web (URIs), consistindo, desta forma, em recursos. Estes recursos também podem ser dereferenciados pelas aplicações para que, através das suas descrições RDF, a máquina possa compreender os tipos dos recursos

(classes), assim como os relacionamentos entre eles, inclusive sendo capaz de inferir novas triplas com base nas regras (axiomas) descritas na ontologia (Azevedo & Jacyntho, 2014).

Para que haja uma melhor comunicação entre aplicações, é fortemente recomendado o reúso de ontologias existentes para a descrição dos dados. Esta prática maximiza a probabilidade dos dados serem consumidos entre aplicações sem que seja necessário aplicar alguma modificação ou pré-processamento (Heath & Bizer, 2011). Ainda de acordo com Heath e Bizer (2011), a escolha de uma ontologia deve observar os seguintes critérios:

- Se o vocabulário é amplamente utilizado, de forma a facilitar a comunicação com aplicações *Linked Data* já existentes;
- Se o vocabulário sofre manutenção ativamente, e sob um processo claro de governança;
- Se o vocabulário cobre o suficiente dos dados que se deseja descrever, de forma que justifique a adoção dos seus termos e regras;
- Se o vocabulário é expressivo o suficiente para descrever os dados e o cenário desejado.

2.1.3 Sintaxes para publicação de arquivos RDF

Para que os dados do grafo abstrato RDF sejam efetivamente publicados na Web, foram criadas diversas sintaxes padrão para arquivos RDF, entre as quais: RDF/XML (Gandon & Schreiber, 2014), *Turtle* (Beckett *et al.*, 2014) e JSON-LD (Sporny *et al.*, 2014).

2.1.4 *Linked Data*

Um paradigma muito importante pertencente à Web Semântica é o conceito de *Linked Data* (Dados Ligados, em português). Trata-se de uma série de boas práticas para publicar e interligar dados estruturados na Web, provendo um caminho mais genérico e flexível para que os consumidores de dados possam descobrir e integrar dados de diferentes fontes de dados (Heath & Bizer, 2011). A aplicação desse paradigma viabiliza a chamada "Web de Dados Ligados" (*Web of Linked Data*, em inglês): uma Web de dados estruturados, totalmente compreensível por máquinas, tornando a busca por informações mais precisa e consistente (Jacyntho & Azevedo, 2015). A topologia da Web de Dados consiste em um enorme grafo global, interligando diversas fontes de dados abertos. O coração da Web de Dados é a fonte de dados DBpedia (Bizer *et al.*, 2009; DBpedia, 2017), que é uma versão *Linked Data* da Wikipedia. Na figura 1 é possível ter uma visão dessa topologia, onde é possível ver várias fontes de dados (círculos) interligadas por links RDF.

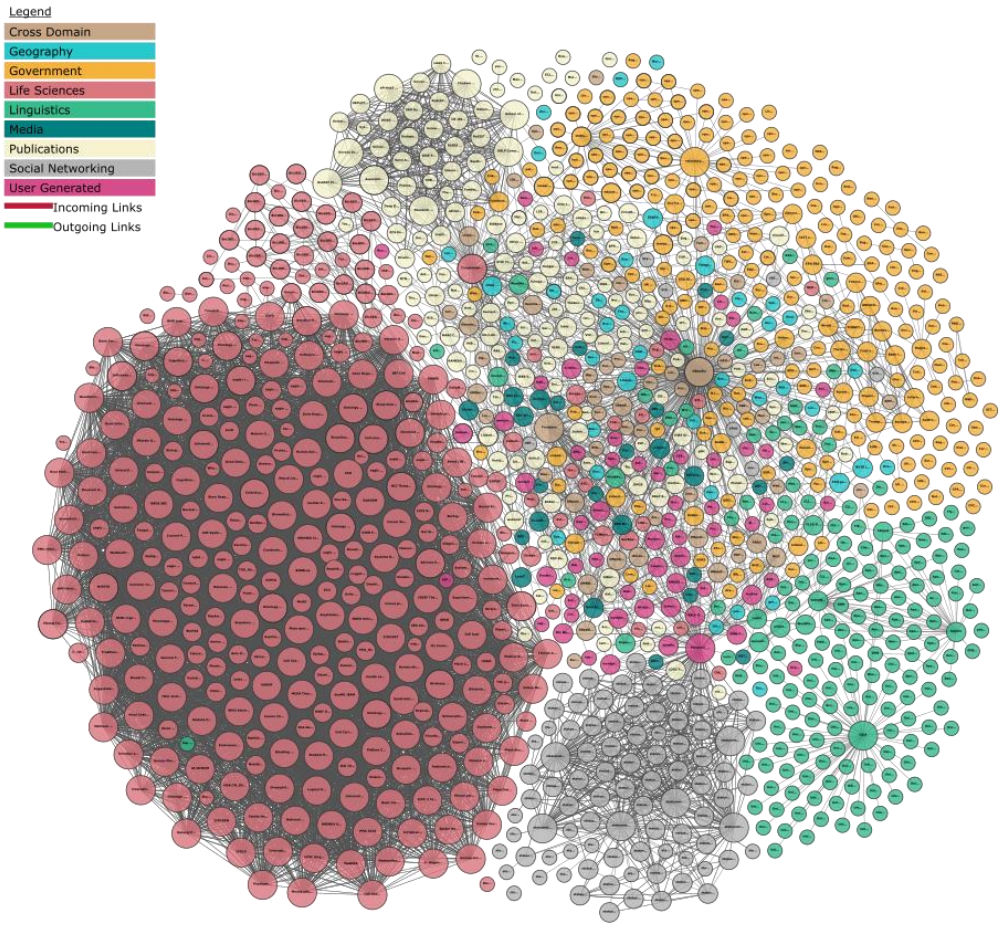

Figura 1 - LOD Cloud Diagram, Dezembro de 2017 (Abele *et al.*, 2017)

Em Berners-Lee (2009) são apresentados os quatro princípios *Linked Data* que devem nortear a publicação de dados semânticos na Web de Dados, a saber:

1. Use URIs para nomear recursos;
2. Use URIs *Hypertext Transfer Protocol* (HTTP), de forma que possamos acessar estes nomes;
3. Faça todos os URIs dereferenciáveis. Em outras palavras, quando um URI for acessado, retorne informações úteis, utilizando os padrões (RDF, SPARQL);
4. Inclua *links* para outros URIs, para que se possa descobrir (navegar para) mais recursos (*Linked Data mashup*).

O termo *mashup* denota técnicas utilizadas na Web que permitem combinar dados de múltiplas fontes em uma única aplicação, como por exemplo, reusar serviços de dados proprietários disponibilizados por gigantes da Web, como Google, Yahoo e eBay. *Linked Data mashup* consiste em utilizar as tecnologias da Web Semântica para integrar, de forma padronizada, dados estruturados de diferentes fontes, permitindo assim acesso através de

clientes genéricos, como navegadores RDF, máquinas de busca RDF e agentes de consulta na Web, superando, pois, as limitações dos *mashups* tradicionais (Bizer *et al.*, 2007). Este tipo de *mashup* semântico permite que as máquinas naveguem de um recurso para o outro, obtendo mais informações, independentemente da localização dos dados (Azevedo & Jacyntho, 2014).

2.1.5 O Modelo cinco estrelas para publicação de dados abertos

Com o objetivo de encorajar e orientar as pessoas na produção e publicação de dados abertos ligados, Tim Berners-Lee criou um modelo de classificação de maturidade de dados abertos em cinco níveis, ou cinco estrelas, descrito a seguir (Berners-Lee, 2009):

★ Disponível na Web (em qualquer formato, p. ex. PDF), mas com uma licença aberta, para serem dados abertos;

★★ A regra anterior, mais: dados estruturados inteligíveis por máquina (p. ex. uma planilha Excel em vez de uma imagem digitalizada de uma tabela);

★★★ Todas as regras anteriores, mais: formato não proprietário (por exemplo, CSV[1] em vez de Excel);

★★★★ Todas as regras anteriores, mais: usar os padrões do W3C (RDF e SPARQL) para identificar os recursos (entidades), de forma que as pessoas possam referenciá-los para reúso;

★★★★★ Todas as regras anteriores, mais: ligar os dados publicados a dados de outras pessoas para prover contexto (*Linked Data mashup*).

2.2 Integração de Dados entre sistemas utilizando a Web Semântica

Para integrar bancos de dados independentes utilizando as tecnologias e padrões da Web Semântica, um conceito que também pode ser chamado de integração semântica de dados, é necessário criar uma visão RDF para cada uma dessas bases, por meio de um processo de mapeamento, gerando, assim, um grafo de conhecimento integrado, utilizando, para tal, uma ou mais ontologias integradoras para a área de conhecimento em questão (Konstantinou &

[1] *Comma Separated Values* (CSV) - Arquivo texto que armazena dados tabulares, no qual cada linha é um registro e cada registro consiste de um ou mais campos separados por vírgulas.

Spanos, 2015). A seguir, é explicado, em linhas gerais, como funciona esse processo de mapeamento.

2.2.1 Mapeamento Relacional-RDF

Com o objetivo de viabilizar o reaproveitamento das informações estruturadas, armazenadas em Sistemas Gerenciadores de Bancos de Dados relacionais (SGBDs) para o formato de grafo RDF, existem ferramentas que permitem configurar, através de linguagens como R2RML (Das *et al.*, 2012) e *D2RQ Mapping Langua*ge (Cyganiak *et al.*, 2012), um mapeamento da estrutura de tabelas e *views* em ontologias, permitindo, inclusive, o acesso com a conversão em tempo real. A aplicação deste tipo de técnica pode fomentar uma série de benefícios, como (Konstantinou & Spanos, 2015):

- Anotação semântica de sítios Web;
- Integração de bases de dados heterogêneas;
- Acesso de dados baseado em ontologias;
- Geração em massa de dados em formato da Web Semântica;
- Elaboração semi-automática de ontologias de domínio;
- Descrição semântica de modelos de dados relacionais;
- Integração de bancos de dados com outras fontes de dados externas, publicadas na Web.

A figura 2 demonstra um exemplo de mapeamento, onde os dados de uma tabela **pessoa** com os campos **id**, **nome**, **email** e **data_nascimento** são mapeadas para as classes ***foaf:Person*** e ***schema:Person***, e as propriedades ***foaf:name***, ***foaf:mbox***, ***schema:email***, ***schema:birthDate***, das ontologias *FOAF*[2] e *Schema.org*[3]. Ao executar a ferramenta de mapeamento, cada registro de pessoa do banco de dados relacional será descrito como um recurso, associado ao respectivo URI HTTP, compondo triplas de acordo com as propriedades previamente definidas. Faz-se importante observar que é possível mapear os mesmos dados relacionais para mais de uma propriedade, inclusive combinando diferentes ontologias, e que o mapeamento não está restrito a tabelas, mas pode ser realizado através de *views* ou consultas SQL mais complexas.

[2] http://xmlns.com/foaf/spec/

[3] https://schema.org/

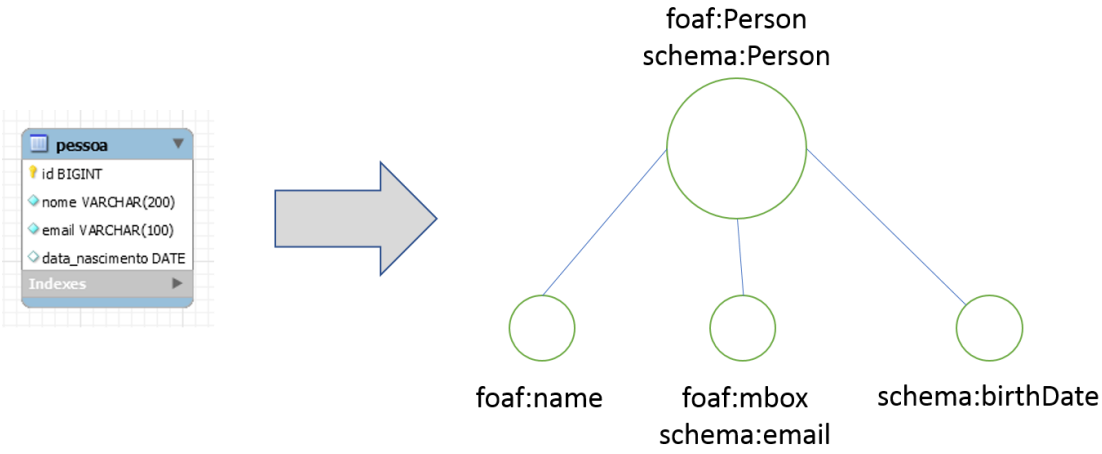

Figura 2 - Exemplo de mapemaneto Relacional – RDF. Elaborado pelos autores (2019)

.

3 TRAJETÓRIA METODOLÓGICA

A trajetória metodológica percorrida para o desenvolvimento deste trabalho, compreende as seguintes etapas:

1) Levantamento Bibliográfico

No decorrer desta pesquisa, foi realizado um estudo aprofundado sobre Web Semântica, bem como a análise de projetos já existentes. Com o objetivo de levantar o estado da arte dentro da área pesquisada, foram realizadas buscas por trabalhos relacionados e projetos existentes em bases de conhecimentos como *Scopus*®, *ScienceDirect* e *Web of Science*, além do *Google Scholar*. O estudo do material já existente permitiu observar diferentes estilos arquiteturais, analisando-se a topologia dos sistemas e dos bancos de dados utilizados. Alguns dos mais relevantes trabalhos encontrados são apresentados no Capítulo 7 deste documento.

2) Análise

Em uma segunda etapa, foi analisada uma série de ferramentas da Web Semântica, considerando suas funcionalidades, vantagens e desvantagens. Foram avaliadas ferramentas *Linked Data* de modelagem ontológica, mapeamento relacional-RDF e *mashup* automatizado, bem como as funcionalidades fornecidas pela linguagem SPARQL e pelos *triple stores*. Foi realizado um estudo detalhado na documentação de várias destas ferramentas, e testes foram conduzidos, visando entender melhor o funcionamento de cada uma e comparar as suas características. Diante das opções encontradas e testadas, bem como da análise de outros projetos, prosseguiu-se com a elaboração de uma arquitetura que pudesse abranger, da melhor forma possível, o *trade-off* entre flexibilidade e desempenho.

3) Construção do método e da arquitetura

Diante das conclusões obtidas na etapa de análise, e do aprendizado obtido, construiu-se então uma proposta de arquitetura, combinando as entradas e saídas das ferramentas selecionadas, bem como de um método para instanciá-la, evoluídos em duas etapas, cada uma sobre um estudo de caso diferente, descritas a seguir:

a) Publicação de dados relacionais em RDF

Esta pesquisa adotou uma abordagem *bottom-up*, partindo-se da solução de um problema específico, até a elaboração de uma solução genérica, aplicável a diferentes domínios. Em uma primeira etapa, foi elaborada uma solução para publicação de bases de

dados relacionais segundo os princípios *Linked Data*, validado por meio de um estudo de caso real de trabalhos acadêmicos. Esta etapa gerou um método, publicado em Gonçalves e Jacyntho (2020).

b) Integração de dados entre sistemas de informação

Diante dos resultados alcançados, e com base nos conteúdos estudados, o método, bem como a arquitetura, foram complementados, através de um processo iterativo e incremental, para atender o problema de integração de dados entre sistemas de informação distintos, dentro de um ambiente corporativo.

4) Validação do método e da arquitetura

Com o objetivo de corroborar a eficácia do método e arquitetura propostos, estes foram finalmente aplicados em um estudo de caso de sistemas de informação corporativos dentro de uma universidade pública brasileira. Este processo também incluiu a análise e experimentação de diferentes ferramentas *Linked Data*, visando avaliar sua adequação ao problema proposto, além de testes com dados reais, envolvendo a realização de consultas entre bases de dados integradas, visando analisar a viabilidade e desempenho do modelo, e as vantagens para o usuário final.

4 MÉTODO E ARQUITETURA PROPOSTOS

Este capítulo apresenta o método e arquitetura propostos para integração *Linked Data* entre bases de dados de sistemas de informação distintos, bem como um método para publicação de dados relacionais na Web. Embora o método de publicação de dados faça parte do método de integração de dados, ambos podem ser utilizados de forma independente, de acordo com a necessidade. Ressalta-se ainda que ambos os métodos e arquitetura se propõem a fornecer meios para se estabelecer uma visão RDF *read-only* (somente leitura) sobre os dados de sistemas de informação distintos, que pode ser acessada através de consultas SPARQL, por meio de bibliotecas de manipulação de RDF, disponíveis nas diversas linguagens de programação existentes no mercado, ou por meio de *softwares* especializados.

4.1 Arquitetura proposta

A figura 3 demonstra a arquitetura proposta para a integração de dados entre sistemas.

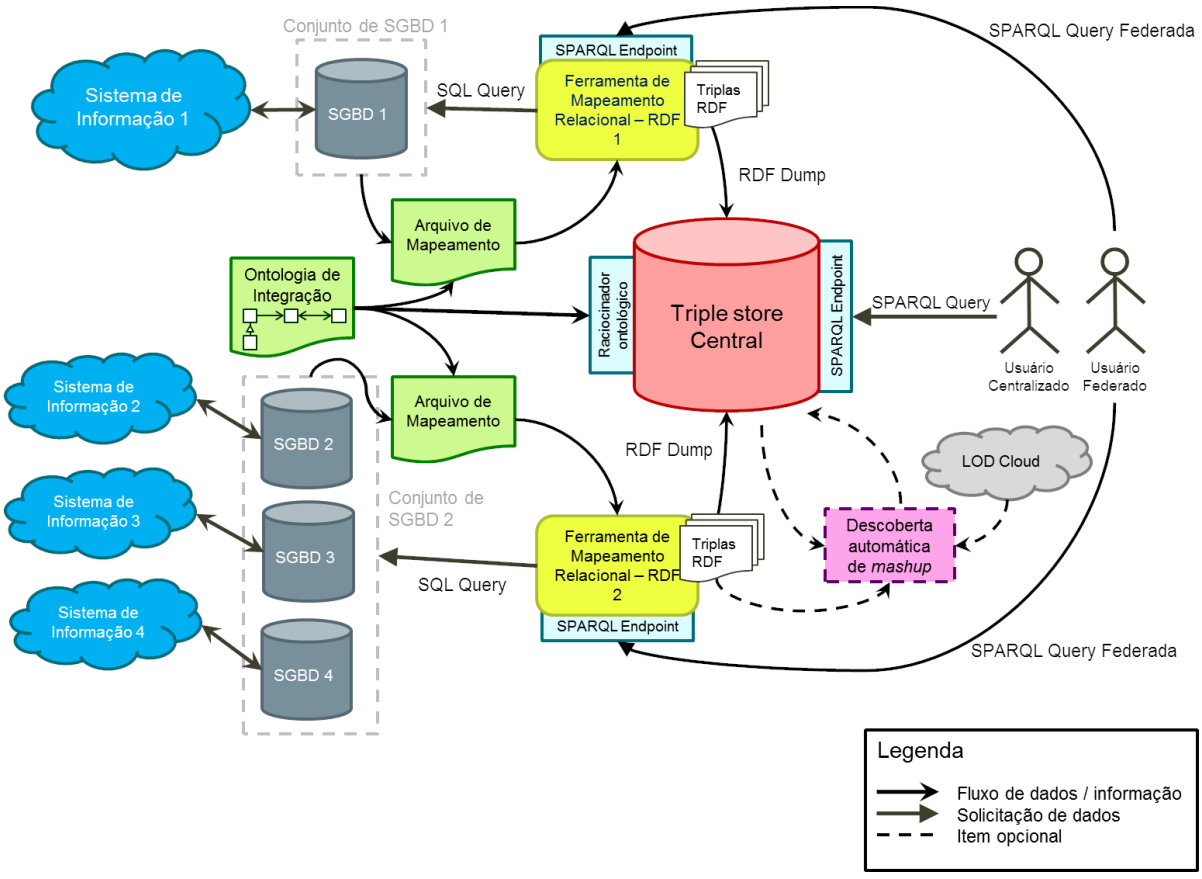

Figura 3 - Arquitetura proposta para implementação da integração de dados dos sistemas através de grafos RDF

As nuvens marcadas como “Sistema de Informação 1” a “Sistema de Informação 4” representam sistemas tradicionais ligados às suas respectivas bases de dados relacionais. Ao meio é possível ver duas ferramentas de mapeamento Relacional-RDF (*RDB2RDF*), que fazem a conversão de dados relacionais para um grafo RDF, assim como de consultas

SPARQL para SQL. O comportamento destas é definido nos arquivos de mapeamento, que traduzem as relações dos bancos relacionais para os conceitos da ontologia (vocabulário) de integração. As bases de dados relacionais, apresentadas no modelo, são separadas em conjuntos (Conjunto de SGBD 1 e 2) puramente para fins de ilustração, pois algumas ferramentas *RDB2RDF* permitem a conexão simultânea a múltiplas fontes de dados, em uma mesma instância, enquanto outras permitem apenas uma conexão por instância. O *triple store* (banco de dados RDF nativo) central, em vermelho, representa uma abordagem alternativa no acesso aos dados, de forma *offline*, ou seja: periodicamente extrai-se todas as triplas geradas pela ferramenta de mapeamento e carrega-se, em lote, em um *triple store*. Este segundo acesso visa proporcionar maior desempenho e disponibilidade na consulta aos dados, para clientes que não necessitem dos dados atualizados em tempo real, angariando assim, um menor impacto e dependência da utilização dos recursos computacionais dos servidores onde se encontram os SGBDs relacionais. Ligado ao *triple store*, é possível observar uma seta oriunda da ontologia de integração. Esta representa a importação da ontologia no *triple store* para que o seu raciocinador ontológico possa usá-la para fins de inferência sobre os dados. Tanto as ferramentas de mapeamento quanto o *triple store* fornecem *SPARQL enpoints*, através dos quais os dados podem ser consultados, e o *triple store*, inclusive, pode ser utilizando como ponto de central para processamento de consultas federadas.

Na parte direita do modelo é possível identificar um Usuário Centralizado e um Usuário Federado, representando sistemas ou pessoas, que podem acessar os grafos RDF mapeados, realizando consultas SPARQL centralizadas ou federadas, dependendo do seu perfil e necessidade, por meio dos respectivos *SPARQL endpoints*. Por fim, ilustrado através de um quadro com linhas pontilhadas, encontra-se um componente de descoberta automática de *mashup*. Trata-se de um recurso opcional, embora desejável, que visa estabelecer ligações, de forma automatizada, entre os recursos mapeados e outros recursos, podendo ser, inclusive, oriundos do grafo global de dados ligados (*LOD Cloud*), representado na figura como uma nuvem cinza. Uma explicação mais detalhada sobre a implementação desta arquitetura é fornecida nos tópicos a seguir.

4.2 Método para publicação *Linked Data* de dados relacionais

Nesta seção, é apresentado o método proposto para publicação de bases de dados relacionais como *Linked Data*, publicado na revista indexada Transinformação (Gonçalves & Jacyntho, 2020), apresentado no fluxograma da figura 4.

Figura 4 - Método proposto para disponibilização de dados relacionais em RDF.
Fonte: Elaborado pelos autores (2018).

A publicação na Web de Dados, segundo proposto neste método, decorre em um processo iterativo e incremental composto pelos seguintes passos:

1 – Definição das informações a publicar

A partir do banco de dados relacional do sistema utilizado, define-se quais informações são pertinentes à publicação *Linked Data*, considerando perguntas como:

- Quais informações são essenciais para o domínio desejado, expressando seus principais dados?
- Quais propriedades possuem ontologias conhecidas para modelar o domínio pretendido?
- Quais informações, por serem sensíveis ou privadas, inclusive por questões legais, não devem constar na publicação de dados, a ser disponibilizada abertamente na Web?

2 – Criar um modelo de classes conceitual com base na estrutura das relações

Com o objetivo de visualizar de forma mais simples e clara as informações escolhidas para serem publicadas, faz-se necessária a criação de um modelo de classes conceitual, de mais alto nível de abstração, baseado nas relações e seus campos, utilizando uma linguagem

de modelagem adequada, como, por exemplo, *Unified Model Language* (UML) (Object Management Group - OMG, [s.d.]), recomendada nesta proposta por ser um padrão largamente conhecido e adotado na área de computação.

3 – Criar um conjunto de *views* SQL com base no modelo de classes conceitual

Com o intuito de agregar as informações a serem publicadas, bem como não interferir na base de dados original, garantindo acesso seguro às informações pretendidas, recomenda-se o mapeamento semântico sobre *views* SQL, ou seja, consultas SQL pré-definidas, preferencialmente armazenadas na estrutura de um banco de dados auxiliar separado, expressando os dados a serem mapeados. Basicamente, deve ser criada uma *view* SQL para cada classe conceitual.

4 – Selecionar ontologias de acordo com o domínio de conhecimento em questão

Uma vez definidos os dados a publicar, conduz-se uma pesquisa minuciosa, visando encontrar ontologias capazes de descrever o domínio de conhecimento em questão, partindo-se das mais conhecidas e conceituadas, até outras mais específicas. A pesquisa pode ser realizada a partir de catálogos de ontologias como o *Linked Open Vocabularies* (LOV) (Vandenbussche *et al.*, 2016), além de sítios de busca e artigos acadêmicos. Nesta etapa, recomenda-se adotar os critérios de seleção de ontologias apresentados na seção 2.1.2. Com base nos resultados encontrados, define-se o mapeamento do modelo conceitual do domínio em questão para as classes e propriedades das ontologias selecionadas.

5 - Mapear o modelo de classes conceitual para as classes e propriedades das ontologias selecionadas

Esta etapa compreende o mapeamento das classes e propriedades (atributos e relacionamentos) do modelo de classes conceitual para classes e propriedades das ontologias, conforme ilustrado anteriormente na figura 2, o que pode ser documentado usando a mesma linguagem de modelagem escolhida no passo 2. O tópico 5.1.2, do estudo de caso real de uma conferência acadêmica, ilustra, por meio de estereótipos UML, como executar este passo mais detalhadamente.

6 - Gerar visão (grafo) RDF somente leitura sobre os dados relacionais a partir das *views* SQL criadas

Esta etapa diz respeito à implementação deste mapeamento no acesso à base de dados relacional, a partir das *views* SQL criadas no passo 3 do método. Para o mapeamento em tempo real do banco de dados relacional em uma visão RDF do mesmo, utiliza-se ferramentas

que permitem, por meio da especificação do mapeamento relacional-RDF a disponibilização das tuplas de um banco de dados relacional em triplas RDF, em tempo de execução (*on the fly*), sem a necessidade de armazená-las em um banco de dados RDF. Opções de ferramentas para a execução deste passo são apresentadas na seção 4.4 deste trabalho.

7 – Realizar *Mashup* automatizado com outras fontes de dados

Com o objetivo de alcançar as cinco estrelas propostas pelo modelo de Berners-Lee (2009), faz-se necessária a ligação de recursos da base a ser publicada com outros recursos na Web, utilizando-se fontes de dados e predicados pertinentes (*rdfs:seeAlso*, *owl:sameAs*, etc.). Para a pesquisa de fontes de dados, podem ser utilizados sites como o *The Linked Open Data Cloud* (LOD Cloud)[4].

Como é comum encontrar uma expressiva quantidade de dados em bancos relacionais já existentes, recomenda-se a utilização de técnicas de *mashup* automatizado para estabelecer a ligação entre recursos RDF extraídos pelo mapeamento e os recursos de outras bases. A seção 4.4 deste trabalho apresenta uma ferramenta para este fim.

8 – Publicar grafo RDF *Linked Data* na Web

A publicação dos dados na Web se dá através de *SPARQL endpoints* disponibilizados pela ferramenta de mapeamento escolhida, ou através do carregamento das triplas RDF em um banco de dados RDF centralizado. A seção a seguir, sobre a integração de dados de sistemas, aprofunda-se mais nas estratégias de acesso aos dados mapeados.

4.3 Método para integração de dados de sistemas de informação distintos

Como extensão do método apresentado na seção anterior, esta seção apresenta o método e arquitetura propostos para a integração de dados de sistemas distintos utilizando RDF, através de um processo iterativo e incremental, apresentado na figura 5, por meio de um diagrama de fluxo BPMN.

4 https://lod-cloud.net/

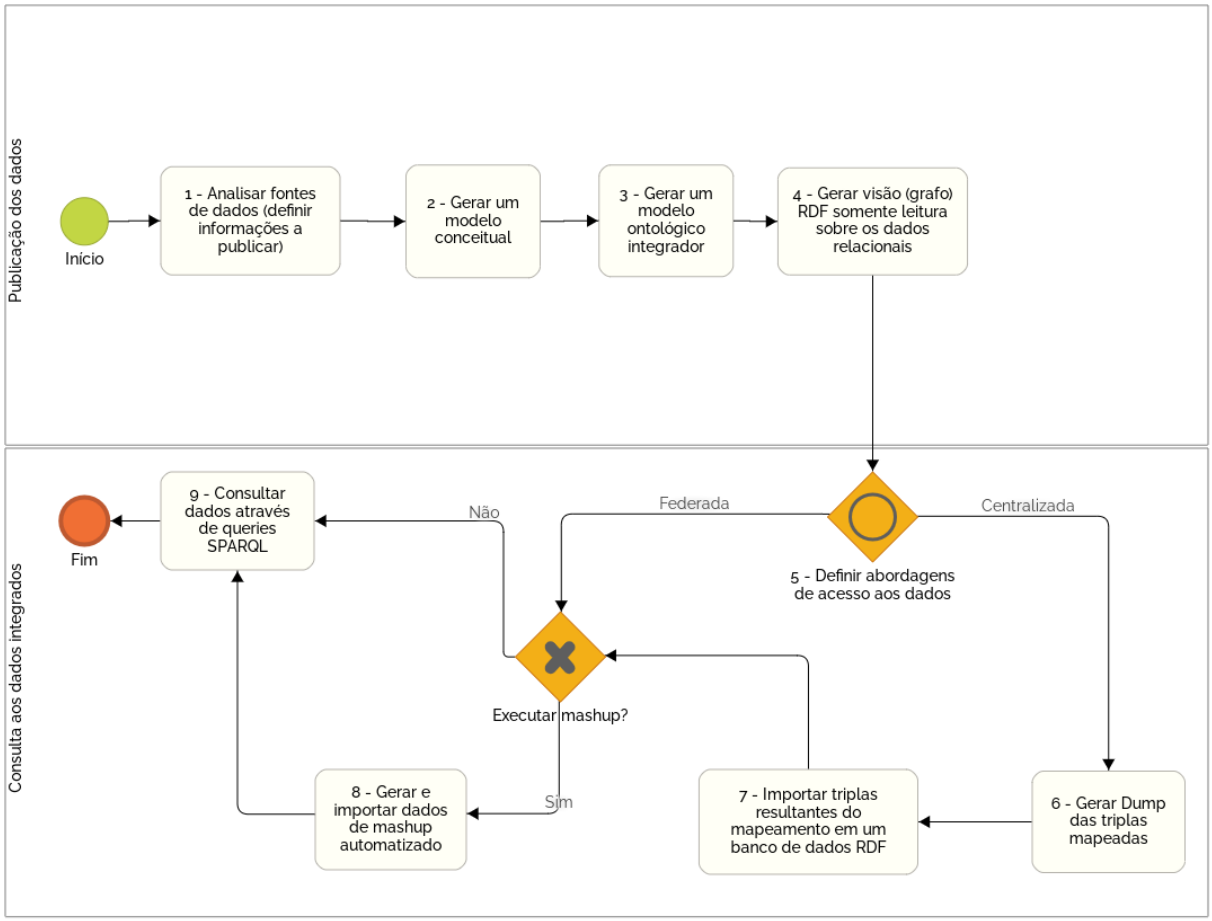

Figura 5 - Método proposto em notação BPMN. Fonte: Elaborado pelos autores (2019)

Como é possível ver na figura 5, o método foi dividido em duas etapas macro, ilustradas aqui como *swimlanes*: Publicação dos Dados e Consulta aos Dados Integrados. A primeira etapa macro, basicamente diz respeito ao método anteriormente apresentado, excluindo-se a parte do *mashup* automatizado, e acrescidas algumas particularidades inerentes ao cenário de integração de dados de sistemas. A segunda etapa aprofunda-se na técnica de disponibilização e consulta aos dados integrados por meio do mapeamento. A seguir, descreve-se detalhadamente cada passo do método proposto.

A) Publicação dos dados

1 - Analisar fontes de dados (definir informações a publicar)

Esta etapa corresponde ao passo 1 do método anteriormente apresentado. Nesta etapa são elencadas as necessidades da instituição, delimitando-se o escopo de trabalho. Para tal, lança-se mão de técnicas de levantamento de requisitos junto aos *stakeholders* (partes interessadas), como, por exemplo, reuniões e entrevistas. Durante o processo de levantamento de requisitos é necessário que sejam respondidas perguntas como:

- Quais informações são essenciais para o domínio desejado, expressando seus principais dados?
- Quais são os sistemas e bases relacionais a serem utilizadas como origem de dados?
- Quais informações os *stakeholders* necessitam obter a partir dos dados integrados?

2 - Gerar um modelo conceitual

Esta etapa é semelhante ao passo 2 do método anterior. Uma vez definidos os requisitos de dados para integração, elabora-se um modelo conceitual, representando as classes necessárias, através de uma linguagem de modelagem como, por exemplo, UML. O modelo deve refletir uma visão dos dados das origens selecionadas, conforme o interesse dos *stakeholders*. Sendo assim, a utilização de documentação já existente a respeito dos sistemas e seus bancos de dados ou, ainda, técnicas de engenharia reversa podem ser utilizadas como apoio neste processo.

Um ponto importante a observar, nesta etapa de modelagem, é a inclusão no projeto de atributos de valor único, que permitam identificar um mesmo recurso dentre os diversos sistemas da instituição alvo, como, por exemplo, números de documentos pessoais ou de matrículas, códigos oficiais e e-mails. Este cuidado pode facilitar consideravelmente no momento da realização de consultas que envolvam dados de diferentes sistemas.

3 - Gerar um modelo ontológico integrador

Este compreende os passos 4 e 5 do método anterior. Uma vez definido o modelo conceitual, prossegue-se com uma pesquisa minuciosa, visando encontrar ontologias capazes de descrever o domínio de conhecimento em questão, partindo-se das mais conhecidas e conceituadas, até outras mais específicas. Nesta etapa, realiza-se um estudo detalhado das ontologias, procurando entender a sua estrutura, através das documentações disponíveis, o que possibilita identificar como cada uma pode contribuir para a estrutura a ser desenvolvida. Como descrito no método anterior, a pesquisa de ontologias pode ser realizada a partir de catálogos de ontologias, além de sítios de busca e artigos acadêmicos, adotando-se os critérios de seleção de ontologias apresentados na seção 2.1.2.

Uma vez definidos os vocabulários a serem utilizados, desenvolve-se um modelo ontológico, conectando as classes e propriedades já existentes, baseando-se no modelo conceitual criado no passo anterior. Com o objetivo de melhor adequar o modelo ao domínio

desejado, pode ser necessário criar uma nova ontologia, com classes e propriedades extras. Para a modelagem de uma nova ontologia, o que pode ser considerado um fluxo alternativo interno a este passo do método, recomenda-se a utilização do processo descrito em Noy e McGuinness (2001). E para a sua implementação, utiliza-se as linguagens da Web Semântica para criação de ontologias: *RDF Schema* (RDFS) e *Ontology Web Language* (OWL).

4 - Gerar visão (grafo) RDF somente leitura sobre os dados relacionais

Este compreende os passos 3 e 6 do método anterior, dispostos aqui, propositalmente, em uma ordem diferente da anterior. Uma vez definido o modelo ontológico, criam-se *views* SQL com o objetivo de estabelecer uma camada de dados entre as tabelas relacionais originais e o mapeamento para grafo RDF. Através da definição de *views* SQL, é possível abstrair os detalhes de implementação de banco de dados de uma determinada aplicação, construindo-se uma interface que forneça os dados necessários para atender a demanda de integração, especificada no modelo.

Conforme descrito no método anteriormente apresentado, essas *views* SQL são utilizadas através de ferramentas capazes de, por meio da especificação do mapeamento Relacional-RDF, disponibilizar tuplas de um banco de dados relacional (ou de um conjunto de bancos de dados relacionais) em triplas RDF, em tempo de execução (*on the fly*), sem a necessidade de armazená-las em um banco de dados RDF. Opções de ferramentas para a execução deste passo (ferramentas *RDB2RDF*) são apresentadas na seção 4.4 deste trabalho.

É pertinente frisar a importância da padronização dos URIs utilizados para identificar os recursos nas triplas mapeadas. Para que seja facilitado o processo de integração, recomenda-se fortemente a utilização do mesmo padrão de URI para identificar um mesmo tipo de recurso, provenientes de diferentes origens de dados. Por exemplo, considerando-se um recurso que represente um aluno que tenha uma matrícula de número **123456**. Um exemplo plausível de URI HTTP poderia ser ***<http://www.exemplo.com/aluno/123456>***, utilizando-se a matrícula como parte do URI. Caso a única forma de identificar um recurso nas bases de dados relacionais seja algum dado sensível, pode ser utilizada no URI um *hash* para o mesmo, como, por exemplo, MD5 (Rivest, 1992) ou SHA1 (Eastlake & Jones, 2001). O mesmo exemplo acima, utilizando-se SHA1, poderia ficar da seguinte maneira: ***<http://www.exemplo.com/aluno/7c4a8d09ca3762af61e59520943dc26494f8941b>***. A utilização de dados sensíveis em URIs é desencorajada pelo Capítulo 7 da RFC-3986 (Berners-Lee *et al.*, 2005).

É importante ressaltar que, dependendo da ferramenta *RDB2RDF* adotada, a utilização (ou não) de atributos chave indexados, das tabelas relacionais, na composição dos URIs pode interferir drasticamente na performance das consultas *on-the-fly* (com conversão SPARQL–SQL). Sendo assim, recomenda-se a utilização de mais de um URI na descrição dos recursos, podendo o adicional ser incluído através do predicado ***owl:sameAs***.

B) Consulta aos dados integrados

5 – Definir abordagem de acesso aos dados (inclui os passos 6 e 7)

Uma vez gerado o grafo RDF sobre os dados relacionais, deve-se definir a abordagem de acesso aos dados integrados. Diante da arquitetura apresentada, existem basicamente duas estratégias, cada uma com suas vantagens e desvantagens: consulta centralizada e consulta federada.

a) Consulta federada

Esta abordagem consiste em disponibilizar um *SPARQL endpoint* para cada origem de dados (ou conjunto de origens de dados) que se deseja integrar, conforme ilustrado na arquitetura, através da figura 3. Para acessar os dados de forma integrada, como sendo um único grafo, executa-se uma consulta SPARQL federada, incluindo-se todos os *endpoints* necessários. Esta abordagem possui as seguintes vantagens:

- Acesso aos dados atualizados em tempo real, conforme registrado nas suas bases de dados relacionais;
- Modularização das fontes de dados mapeadas, sendo facilitada a incorporação imediata de novas bases ao grafo.

Contudo, faz-se necessário avaliar as seguintes desvantagens:

- Desempenho menor, devido a necessidade de processamento dos mapeamentos e conversão SPARQL/SQL e SQL/RDF, envolvendo a execução interna de junções SQL (JOIN), e do acesso simultâneo a múltiplos *endpoints*;
- Mais pontos de falha, uma vez que é necessário que todos os bancos de dados e ferramentas de mapeamento estejam funcionando e mantendo corretamente seus *endpoints*. A indisponibilidade de qualquer um dos bancos de dados ou dos *endpoints* já compromete a consulta;
- Inferência ontológica limitada ou inexistente, dependendo da ferramenta adotada para o mapeamento;

- Aumento da complexidade das consultas SPARQL, devido a necessidade da utilização de cláusulas inerentes a consultas federadas, além da declaração de predicados adicionais para lidar com as limitações de inferência;
- Dependendo da ferramenta *RDB2RDF* utilizada, pode não haver suporte completo para as cláusulas SPARQL;
- Utilização dos recursos computacionais dos bancos de dados relacionais a cada consulta.

b) Consulta centralizada

A abordagem centralizada de consulta aos dados integrados consiste em extrair todas as triplas geradas pelo mapeamento Relacional – RDF e importá-las em um banco de dados central RDF nativo (*triple store*), o que consiste nos passos 6 (Gerar *dump* das triplas mapeadas) e 7 (Importar triplas resultantes do mapeamento em um banco de dados RDF) deste método, conforme figura 5. Desta forma, as consultas são realizadas diretamente ao *SPARQL endpoint* do próprio banco RDF. A atualização desses dados é realizada periodicamente, através de importações em lote. É recomendável que essa atualização periódica seja feita de forma automática, podendo incluir, se necessário, uma etapa de descoberta de informações, como, por exemplo, a solução proposta em Halaç *et al.* (2013). Esta abordagem conta com as seguintes vantagens:

- Melhor desempenho, pois o *triple store* já possui mecanismos adequados ao acesso otimizado de dados já armazenados em grafo;
- Menor risco de indisponibilidade, pois, para a execução das consultas, é necessário manter funcionando apenas um único serviço;
- Inferência inerente ao módulo raciocinador do *triple store*. Por meio deste é possível utilizar inferência ontológica, ampliando, automaticamente, a base de conhecimento;
- Possibilidade de importação de inferências materializadas, ou seja, importar, no banco de dados, triplas resultantes do processamento de inferência ontológica;
- Uso dos recursos computacionais dos SGBDs relacionais apenas durante os processos de carga, que podem ser agendados para horários de menor impacto;
- Maior controle de acesso (segurança), devido a recursos avançados, disponibilizados pelos *triples stores*;
- Utilização de consultas SPARQL mais simples, pois todos os dados estão disponíveis em uma única fonte de dados;

- Possibilidade de organizar os dados em *named graphs*, um recurso que permite compartimentar tanto consultas quanto importação de dados.

Também é possível identificar, para esta abordagem, as seguintes desvantagens:

- Dado não atualizado em tempo real. O acesso se dá sempre aos dados atualizados na última importação;
- Acréscimo de uma etapa de carregamento de dados a ser executada periodicamente. A estratégia para esta etapa deve ser elaborada de acordo com a necessidade do modelo e os recursos disponíveis. Além disso, pode ser necessário atualizar a rotina de carregamento dos dados toda vez que se desejar incorporar uma nova fonte de dados ao grafo.

8 – Gerar e importar dados de *mashup* automatizado

Este passo opcional, mas muito desejável, corresponde ao passo 7 do método anterior. Trata-se de uma etapa em que os dados obtidos do mapeamento podem ser processados de forma a estabelecer, de forma automatizada, ligações entre os recursos mapeados e outros recursos, que podem ser oriundos das próprias origens de dados mapeadas ou ainda do grafo mundial de dados ligados (*mashup* interno e *mashup* externo), conforme ilustrado na arquitetura através da figura 3. Para este processo, recomenda-se a utilização de ferramental apropriado. No próximo tópico, apresenta-se uma sugestão de ferramenta para este passo.

Embora este passo possa ser adotado tanto na abordagem federada quanto na centralizada, faz-se necessária a importação das triplas oriundas do *mashup* para um *triple store*, mesmo que este seja adotado na arquitetura apenas para suprir esta demanda.

Ressalta-se que esse processo automatizado está sujeito a erros, o que pode ser crítico quando se trata de integração de dados entre sistemas corporativos. Sendo assim, recomenda-se atenção na elaboração dos *workflows* de processamento do *mashup*, utilizando-se critérios confiáveis de comparação, e a submissão dos resultados a algum processo de validação/verificação.

9 – Consulta de dados através de *queries* SPARQL

Conforme comentado no início deste capítulo, a visão *read-only* de dados de sistemas de informação distintos, conforme proposto neste modelo, é acessada através de consultas SPARQL, por meio de bibliotecas de manipulação de RDF, disponíveis nas diversas

linguagens de programação existentes no mercado, ou através de *softwares* apropriados, seguindo a abordagem definida no passo 5.

4.4 Apoio Ferramental

Este tópico descreve brevemente algumas ferramentas capazes de dar suporte ao método e arquitetura propostos, separadas por categoria. A maior parte delas foi utilizada nos estudos de caso deste trabalho.

4.4.1 Modelagem ontológica

O *Protegé*[5] é uma das mais conhecidas ferramentas para criação e edição de ontologias na linguagem OWL. Possui compatibilidade com os principais sistemas operacionais, além de incluir uma versão Web, para desenvolvimento compartilhado.

4.4.2 Mapeamento Relacional/RDF (*RDB2RDF*)

As ferramentas apresentadas a seguir são capazes de dar suporte ao passo 4 do método proposto (figura 5).

D2RQ[6]: trata-se de uma ferramenta *Open Source* que permite acessar bases relacionais como grafos RDF virtuais, somente leitura, sem que seja necessário replicá-los em um banco de dados nativo RDF. Através dele é possível (Bizer *et al.*, 2012):

- Realizar consultas SPARQL sobre SGBDs não RDF;
- Acessar o conteúdo de bancos de dados como *Linked Data* em formato Web;
- Criar *dumps* customizados do SGBD relacional em formato RDF, para serem carregados em um *triple store*;
- Acessar informações oriundas de SGBDs não RDF utilizando a API Apache Jena[7];
- Conectar a múltiplos SGBDs relacionais simultaneamente, consultando-se os dados como um único grafo.

Um mapeamento *D2RQ* é em si um documento RDF, escrito na sintaxe ***Turtle***. Já a linguagem de mapeamento – *D2RQ Mapping Language* (Cyganiak *et al.*, 2012) – é uma

[5] https://protege.stanford.edu/

[6] http://d2rq.org/

[7] http://incubator.apache.org/jena/

ontologia *RDF Schema* simples. Portanto, o mapeamento é expresso usando termos (classes e propriedades) desta ontologia.

Além do mencionado acima, *D2RQ* fornece uma interface Web que permite a navegação entre os recursos mapeados, provendo uma visão em RDF, para máquinas, e HTML, para humanos, baseando-se no conceito de dereferenciamento de URIs (Heath & Bizer, 2011). A aplicação pode ser executada através de linha de comando, trazendo embutida consigo um servidor Web próprio (*D2R-Server*), ou por meio de um servidor de aplicação Java.

A ferramenta é bem completa e madura, além de ainda possuir uma comunidade ativa que a utiliza, e de figurar em diversos artigos acadêmicos. Esta conta também com uma documentação bem completa, disponível em seu sítio Web, e o seu funcionamento atende perfeitamente a proposta apresentada neste trabalho.

D2RQ Mapper[8]: trata-se de uma ferramenta que permite criar e editar um arquivo de mapeamento para o *D2RQ* (Yamamoto & Katayama, 2015a, 2015b). Esta permite conectar diretamente aos SGBDs relacionais e criar o mapeamento *D2RQ*, por meio de um *frontend* Web amigável. Seus recursos incluem:

- Modelagem gráfica do mapeamento, permitindo conectar as relações e campos ao RDF;
- Possibilidade de testar o mapeamento no SGBD selecionado, através de consultas SPARQL *on-the-fly*;
- Exportação de *dump* das triplas mapeadas.

Uma vez concluído o processo de mapeamento, a ferramenta permite exportar o arquivo na *D2RQ Mapping Language*, para aplicação no *D2RQ* ou, ainda, na linguagem R2RML. Esta última é padronizada pela W3C e é compatível com outras aplicações, como, por exemplo, o *Ontop*.

A ferramenta pode ser utilizada diretamente no site do autor, ou instalada em um servidor local por meio de um container *Docker*. Vale comentar que *D2RQ Mapper* é, atualmente, compatível com os seguintes SGBDs: *MySQL*, *PostgreSQL* e *SQLite*.

[8] http://d2rq.dbcls.jp/

Apesar de não abranger todos os recursos e possibilidades do *D2RQ*, *D2RQ Mapper* constitui-se uma relevante contribuição, podendo ser utilizado para construir a base do mapeamento, podendo este ser ajustado manualmente, caso necessário, após a exportação.

Ontop[9]: assim como o *D2RQ*, o *Ontop* é capaz de exportar bases de dados relacionais em grafos virtuais de conhecimento, sem que seja necessário movê-los para um banco de dados nativo RDF. A ferramenta é capaz de expor um *SPARQL endpoint*, traduzindo as consultas realizadas para SQL nos SGBDs relacionais (Calvanese *et al.*, 2017, 2020). O mais relevante destaque desta ferramenta é a sua capacidade de realizar inferência ontológica *on-the-fly*, através da implementação que possui da API *Sesame Storage And Inference Layer (SAIL)*. Da sua suíte de recursos, destacam-se os seguintes:

- Disponibilização de *SPARQL endpoint*, conforme já citado anteriormente, incluído funções de agregação;
- Utilização de RDF 1.1 como modelo de dados em grafo;
- Suporte a Ontologias RDFS e OWL;
- Suporte à linguagem padrão R2RML, além da sua própria linguagem de mapeamento *Ontop Mappings*;
- Produz consultas SQL otimizadas, visando aumentar o desempenho das consultas;
- Capaz de materializar grafos virtuais em arquivos (*dump*), permitindo, inclusive, exportar triplas inferidas;
- *Plugin* para edição e teste do mapeamento no editor de ontologias *Protegé*.

Ontop pode ser utilizado através de um servidor de aplicação Java, por meio de um container Docker disponibilizado pelos desenvolvedores ou via linha de comando.

Além de possuir os recursos apresentados, a ferramenta conta com um grupo ativo de desenvolvedores, tendo apresentado versões recentes contendo aperfeiçoamentos ao projeto. É ainda pertinente citar que, em testes realizados em Chhaya *et al.* (2016), a ferramenta apresentou, em comparação com o *D2RQ*, melhor desempenho na execução de consultas com quantidades expressivas de dados, especialmente quando há envolvimento de relacionamentos, devido ao esforço do projeto na otimização das consultas SQL.

Apesar das vantagens supracitadas, *Ontop* ainda não possui uma interface para dereferenciamento dos URIs mapeados, nem a sua respectiva navegação para usuários

[9] https://ontop-vkg.org/

humanos (HTML), o que pode ser contornado com a criação de uma aplicação auxiliar. Percebeu-se ainda que a documentação disponível, analisada no decorrer desta pesquisa, encontra-se em processo de organização, dispondo, de forma clara e organizada, apenas os tópicos básicos, o que acaba prejudicando o processo de aprendizado na utilização da ferramenta. Os demais tópicos encontram-se espalhados em um repositório Git. Contudo, durante este período de trabalho, percebeu-se evoluções na organização da referida documentação, pela equipe responsável pelo projeto *Ontop*. Vale ainda ressaltar que é necessário conhecimento prévio em SQL para operar a ferramenta, e que o *Ontop* não permite conexão simultânea a mais de uma origem de dados.

R2RML Parser[10]: ferramenta *open source* que permite a exportação de dados de bases relacionais para RDF (Konstantinou *et al.*, 2014; Konstantinou & Spanos, 2015). O mapeamento é configurado por meio da linguagem R2RML. Apesar da ferramenta não permitir acesso dinâmico, *on-the-fly*, às bases relacionais, ela se propõe a gerar *dumps* incrementais, sendo capaz de detectar alterações realizadas nos dados originais.

Triplify[11]: como ferramenta *RDB2RDF*, *Triplify* faz o mapeamento totalmente através de consultas SQL. A ferramenta, desenvolvida na linguagem PHP, pode ser executada em ambiente Web ou via linha de comando. A aplicação permite ainda sua utilização com outros sistemas Web, como, por exemplo, gerenciadores de conteúdo. Embora o projeto tenha sido descontinuado e seu site encontre-se, no momento, inacessível, o download da aplicação pode ser realizado através do sítio <*https://sourceforge.net/p/triplify/wiki/Home/*>, e sua documentação visualizada no sítio *Internet Archive* (Jaenicke *et al.*, 2010). O *Triplify* não fornece *SPARQL endpoint*.

Tripliser[12]: esta não chega a ser uma ferramenta *RDB2RDF*, mas trata-se de uma ferramenta capaz de converter arquivos XML para triplas RDF, via linha de comando ou como biblioteca Java (Rogers, 2011). É apresentada neste tópico por ser uma opção de apoio a migrações e conversões de dados, dependendo das necessidades inerentes às origens de dados.

[10] https://github.com/nkons/r2rml-parser

[11] http://triplify.org/, https://sourceforge.net/p/triplify/wiki/Home/, https://web.archive.org/web/20150206005529/http://triplify.org/Documentation

[12] https://github.com/daverog/tripliser

4.4.3 *Mashup* automatizado

Silk Framework[13] é uma ferramenta capaz de realizar *mashup* automatizado, através de mecanismos para a criação de *workflows*, encadeando diversos tipos de processamento pré-definidos, para tratamentos e comparações entre os dados, aplicados a diversos tipos de origem semântica (arquivos RDF, *SPARQL endpoints*, entre outros). A partir do caminho desenvolvido nesses *workflows*, os dados obtidos entre as múltiplas fontes são comparados, e são geradas, automaticamente, novas triplas RDF, associando os recursos cujo resultado do processamento obedeça aos critérios definidos. O predicado a ser utilizado nessas novas triplas é definido junto ao *workflow* do *Silk*. Esta ferramenta é capaz de dar suporte ao passo 8 do método proposto.

4.4.4 *Triple store*

AllegroGraph[14], *GraphDB*[15] e *Stardog*[16] são recomendações de sistemas de bancos de dados RDF nativos, ou *triples stores*, capazes de atender à arquitetura proposta. Em Gonçalves e Jacynto (2020), parte desta pesquisa, o *GraphDB* apresentou resultados mais satisfatórios nos testes.

4.5 Vantagens da Arquitetura Proposta

O emprego de RDF como solução para o problema de integração de dados de sistemas, conforme proposto neste trabalho, traz uma série de vantagens, dentre as quais se destacam:

- O estabelecimento de uma ontologia integradora única (vocabulário), definida para publicação de dados, de forma que todas as bases, tanto as novas quanto antigas, sejam mapeadas para esta;
- A utilização de URIs comuns, o que permite que diferentes fontes de dados publiquem dados relacionados a um mesmo recurso, de forma inteligível por máquinas. Esses URIs podem ser baseados em dados de identificação únicos, como matrículas, CPF, e-mail, ISBN, etc.;
- A utilização da propriedade ***owl:sameAs*** para publicar dados ligados entre bases com URIs distintas, mas que se refiram a um mesmo recurso.

[13] http://silkframework.org/

[14] https://allegrograph.com/

[15] http://graphdb.ontotext.com/

[16] https://www.stardog.com/

Estas vantagens permitem que o processo de integração de dados seja realizado automaticamente, sem a necessidade da construção de *data warehouses* com processos trabalhosos de extração, transformação e carregamento de dados, e da dependência de ferramentas proprietárias.

Além das vantagens supracitadas, a utilização da tecnologia proposta traz ainda outras vantagens, a saber:

- A possibilidade de *mashup* automatizado por meio de algoritmos de comparação de *strings* (p.ex. ferramenta *Silk*[17]);
- *Mashup* automatizado por meio inferência ontológica (p.ex. propriedade inversa funcional e-mail);
- Possibilidade de descoberta de dados ocultos por meio de inferência ontológica;
- Consultas SPARQL distribuídas (algo que o SQL padrão não oferece);
- Eficiência, devido à ausência de *"joins"*, do ponto de vista do cliente. Propriedades RDF são multivaloradas e grafos RDF se unem naturalmente;
- Dados RDF seguem um padrão independente de fabricante, ou seja, diante da necessidade de migração entre sistemas de bancos de dados RDF, é possível fazê-lo sem nenhuma necessidade de transformação nos dados.

[17] http://silkframework.org/

5 ESTUDOS DE CASO

5.1 Estudo de Caso 1 – Publicações Acadêmicas

Esta seção descreve, em detalhes, a aplicação do método proposto para publicação *Linked Data* de dados relacionais (seção 4.2 - figura 4) ao primeiro estudo de caso desta pesquisa, ou seja, o mapeamento e publicação dos dados relacionais da base de publicações de uma conferência acadêmica como dados RDF *Linked Data*. Trata-se de uma explanação em alto nível de abstração para facilitar a compreensão. Não obstante, todos os arquivos de implementação (*views* SQL, especificação do mapeamento *D2RQ* das relações (classes) e seus campos (propriedades) em RDF, *workflows* de *mashup* do *Silk*, etc.) podem ser obtidos no repositório do projeto no GitHub[18].

5.1.1 Contextualização

Este estudo de caso consiste no mapeamento do banco de dados relacional convencional de gerenciamento de uma conferência acadêmica interna chamada **Semana de Integração Acadêmica**, de uma universidade pública federal brasileira, para o modelo RDF, contemplando: emprego de um conjunto de ontologias consagradas; estabelecimento automático de dados ligados com uma fonte bem conhecida/reusada de dados semânticos, e a seleção de softwares necessários para a publicação desses dados na Web de Dados Ligados.

A Semana de Integração Acadêmica é um evento anual, realizado pela universidade em questão, onde os seus alunos e docentes publicam resumos, expondo as suas pesquisas em curso. O evento é gerenciado através de um sistema de informações gerenciais, desenvolvido, pela instituição, especificamente para este fim, chamado **Sistema SIAC**. O sistema utiliza uma base de dados relacional como repositório para os dados.

5.1.2 Mapeamento do modelo de classes conceitual da conferência em ontologias

Os parágrafos seguintes apresentam uma breve descrição de cada uma das ontologias selecionadas para o estudo de caso. Em seguida, o mapeamento do modelo de classes conceitual da conferência para as mesmas, representado por meio de um diagrama de classes UML (figura 6).

Schema.org (Schema.org Community Group, [s.d.]): ontologia resultante do esforço conjunto das empresas Google, Microsoft, Yahoo e Yandex (Yandex, [s.d.]). Trata-se de um

18 https://github.com/adrianogoncalves/siac-ontologic-model

vocabulário para vários domínios de conhecimento (*crossdomain*), com o objetivo de inserir metadados estruturados em páginas Web, viabilizando retornos mais precisos pelas máquinas de busca. Esta ontologia foi utilizada como ponto de partida para estruturar o mapeamento proposto por este trabalho. Para abreviar URIs, o prefixo "schema" foi utilizado para referenciar o *namespace* da ontologia (*http://schema.org/*).

Dublin Core Metadata Iniative Metadata Terms (*DCTerms*) (DCMI Usage Board, 2012): uma das ontologias mais antigas e conhecidas na Web. Extensão da ontologia *Dublin Core*, trata-se de um vocabulário para descrever metadados genéricos. Para abreviar URIs, o prefixo "dcterms" foi utilizado para referenciar o *namespace* da ontologia (*http://purl.org/dc/terms/*).

DBpedia Ontology (“DBpedia Mappings”, [s.d.]): ontologia *crossdomain* definida pela equipe da DBpedia para a descrição dos seus dados provenientes da Wikipedia. Para abreviar URIs, o prefixo "dbo" foi utilizado para referenciar o *namespace* da ontologia (*http://dbpedia.org/ontology/*).

Semantic Web Conference (*SWC*) (Nuzzolese *et al.*, 2016): ontologia para descrever conferências científicas, baseada na ontologia SWRC, adicionando *links* para classes de ontologias conhecidas. Para abreviar URIs, o prefixo "swc" foi utilizado para referenciar o *namespace* da ontologia (*http://data.semanticweb.org/ns/swc/ontology/*).

Semantic Web for Research Communities (SWRC): ontologia que visa modelar conceitos relacionados a comunidades científicas (publicações, estudantes, universidades, entre outros) (Sure *et al.*, 2005). Esta disponibiliza uma série de classes e propriedades, porém não possui *links* para outras ontologias. Para abreviar URIs, o prefixo "swrc" foi utilizado para referenciar o *namespace* da ontologia (*http://swrc.ontoware.org/ontology#*).

The Bibliographic Ontology (BIBO): ontologia que provê conceitos e propriedades para descrever citações e referências bibliográficas (livros, artigos, etc.) (D’Arcus & Giasson, 2009). Para abreviar URIs, o prefixo "bibo" foi utilizado para referenciar o *namespace* da ontologia (*http://purl.org/ontology/bibo/)*.

Friend Of A Friend (FOAF) (Brickley & Miller, 2014): ontologia para descrever pessoas, suas atividades e relações com outras pessoas, grupos, entidades e documentos (Azevedo & Jacyntho, 2014). Para abreviar URIs, o prefixo "foaf" foi utilizado para referenciar o *namespace* da ontologia (*http://xmlns.com/foaf/0.1/*).

Simple Knowledge Organization System (*SKOS*) (Miles & Bechhofer, 2009): ontologia para modelar estruturas de organização de conhecimento (taxonomias, tesauros, folksonomias, etc.) (Azevedo & Jacyntho, 2014). Para abreviar URIs, o prefixo "skos" foi utilizado para referenciar o *namespace* da ontologia (*http://www.w3.org/2004/02/skos/core#*).

Academic Institution Internal Structure Ontology (*AIISO*) (Styles *et al.*, 2008): ontologia para descrever a estrutura interna de uma instituição acadêmica. Para abreviar URIs, o prefixo "aiiso" foi utilizado para referenciar o *namespace* da ontologia (*http://purl.org/vocab/aiiso/schema#*).

A figura 6 demonstra, por meio de um diagrama de classes UML, o mapeamento do modelo de classes conceitual, gerado a partir do modelo relacional do banco de dados da conferência, nas classes e propriedades das ontologias selecionadas.

O mapeamento de cada classe e suas propriedades (atributos e associações) é expresso por meio de estereótipos UML, representados por << >>. Por exemplo, a classe **AreaConhecimento** foi mapeada nas classes ***skos:Concept*** e ***swrc:ResearchTopic***; o atributo **nome** da **AreaConhecimento** foi mapeado nas propriedades ***rdfs:label*** e ***skos:prefLabel***; e a propriedade **areaConhecimento** da classe **Publicacao**, representado no diagrama pela associação entre **Publicacao** e **AreaConhecimento**, foi mapeado nas propriedades ***dcterms:subject*** e ***schema:genre***. Os outros mapeamentos seguem o mesmo raciocínio. A seleção dos termos utilizados da ontologia, para mapeamento, foi baseada no estudo aprofundado da documentação disponível das ontologias selecionadas, buscando-se escolher classes e propriedades cujas descrições e axiomas fossem capazes de refletir o quadro real do domínio selecionado.

É pertinente ressaltar que, tratando-se de um processo de publicação de dados abertamente na Web, é desejável que cada recurso seja descrito utilizando-se classes e propriedades de diferentes ontologias, o que permite que os dados sejam compreensíveis, com mais facilidade, por mais máquinas. Esta técnica foi empregada em quase todos os elementos do modelo criado e pode ser fortemente observada, por exemplo, na classe **Publicacao**, que foi mapeada para dez diferentes classes, como ***swc:Artefact, bibo:Article*** e ***foaf:Document***. Nesta mesma classe também é possível observar, no diagrama, a utilização de classes opcionais, como ***swc:Poster***, ***swc:TalkEvent*** e ***swc:WorkshopEvent***. Neste caso, nem todas as publicações são mapeadas para todas as classes, apenas as que se adequarem ao tipo de dados. No mundo real, algumas publicações são expostas como pôster, por exemplo, mas não têm apresentação oral, nem consistem em um *workshop*. Para dar suporte a estas classes

condicionais, utilizou-se, na fase de geração do grafo RDF, um recurso específico da ferramenta de mapeamento, que permite modelar as triplas geradas, seguindo condições predefinidas, baseadas nos dados retornados da base relacional.

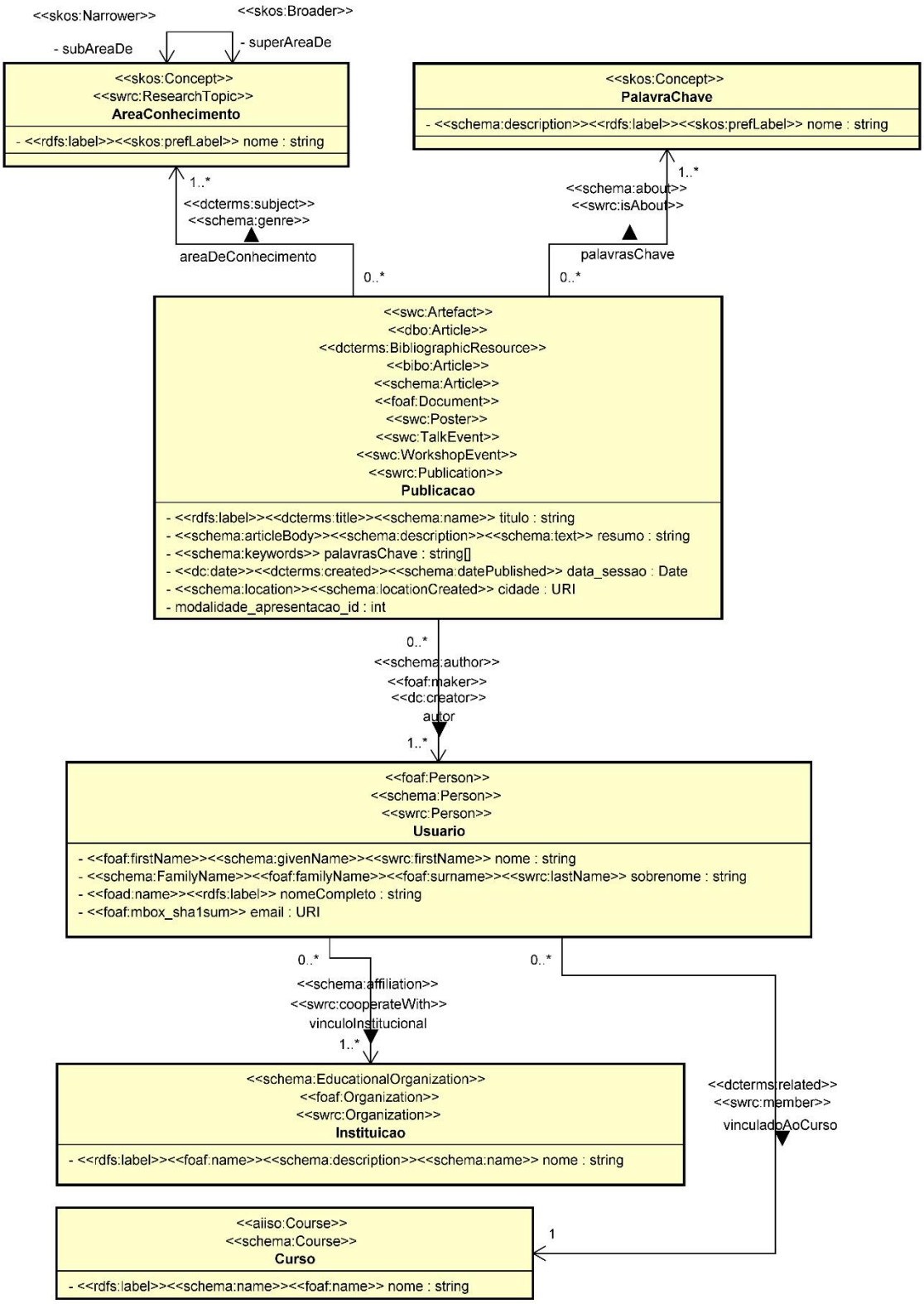

Figura 6 - Mapeamento do modelo conceitual da conferência para as ontologias selecionadas. Fonte: Elaborados pelos autores (2017).

5.1.3 Gerar visão (grafo) RDF somente de leitura sobre os dados relacionais

Todo este mapeamento representado em alto nível de abstração no diagrama da figura 6, foi especificado em um arquivo texto, por meio da linguagem declarativa de mapeamento relacional-RDF da plataforma *D2RQ* (*D2RQ Mapping Language*). Como apontado no passo 6 do método proposto (figura 4), o que, de fato, é mapeado são as *views* SQL criadas com base nas classes conceituais da figura 6. Conforme dito anteriormente, a linguagem de mapeamento *D2RQ* é uma ontologia *RDF Schema* simples e o arquivo de mapeamento é um documento RDF escrito na sintaxe *Turtle*. É com base nesse arquivo de mapeamento que a plataforma *D2RQ* possibilita acessar, *on the fly*, o banco de dados relacional como um grafo RDF virtual, apenas de leitura (*read-only*).

Para ilustrar, considerando-se novamente a classe **AreaConhecimento** (*view* **v_area_conhecimento**) do modelo apresentado na figura 6, esta foi mapeada para as classes ontológicas ***skos:Concept*** e ***swrc:ResearchTopic***, bem como seu atributo **nome** (campo **nome** da *view* **v_area_conhecimento**) para a propriedade ***rdfs:label***, por meio do seguinte trecho do arquivo RDF de mapeamento[19]:

```
map:area_conhecimento  d2rq:ClassMap ;
        d2rq:class skos:Concept, swrc:ResearchTopic ;
        d2rq:classDefinitionLabel "area_conhecimento" ;
        d2rq:dataStorage map:database_views ;
        d2rq:uriPattern "area_conhecimento/@@v_area_conhecimento.id@@" .

map:area_conhecimento__label  d2rq:PropertyBridge ;
            d2rq:belongsToClassMap map:area_conhecimento ;
            d2rq:column "v_area_conhecimento.nome" ;
            d2rq:property rdfs:label .
```

A partir desta especificação, o *D2RQ* cria um mapeamento de classe (*ClassMap*), nomeado como **map:area_conhecimento**, e uma ponte de propriedade (*PropertyBridge*), vinculada a este mapeamento de classe. Desta forma, todas as tuplas retornadas pela *view* **v_area_conhecimento** são mapeadas, na visão RDF, em recursos com a propriedade ***rdf:type*** ligada às classes ontológicas mencionadas, bem como os valores do campo **v_area_conhecimento.nome** são mapeados como valores da propriedade ***rdfs:label*** destes recursos. A propriedade ***d2rq:uriPattern*** serve para definir um padrão para formação dos URIs dos recursos gerados, assim como a propriedade ***d2rq:classDefinitionLabel*** permite

19 Considerando-se os prefixos de *namespace* das ontologias selecionadas e o prefixo d2rq: <http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/0.1#> da ontologia de mapeamento D2RQ.

definir uma etiqueta (*label*) usada no menu de acesso na representação HTML, voltada para humanos. Este mapeamento é capaz de gerar triplas como:

<*http://example.com/area_conhecimento/1234*>

<*http://www.w3.org/1999/02/22-rdf-syntax-ns#type*>

<*http://www.w3.org/2004/02/skos/core#Concept*> .

<*http://example.com/area_conhecimento/1234*>

<*http://www.w3.org/1999/02/22-rdf-syntax-ns#type*>

<*http://swrc.ontoware.org/ontology#ResearchTopic*> .

<*http://example.com/area_conhecimento/1234*>

<*http://www.w3.org/2000/01/rdf-schema#label*>

"*Ciência da Computação*" .

De forma análoga, são mapeadas todas as classes e propriedades do modelo conceitual para as classes e propriedades das ontologias selecionadas, utilizando-se as *views* criadas no banco de dados relacional, visando-se obter a visão em grafo RDF do modelo proposto.

5.1.4 *Linked Data Mashup*

Para geração do *mashup*, buscando-se encontrar ligações entre os recursos da conferência e os da DBpedia, foi utilizado o software *Silk Framework*, já apresentado anteriormente. Neste estudo de caso, realizou-se o processamento utilizando-se o *workflow* descrito na figura 7 e o predicado ***owl:sameAs***, que significa que dois recursos interligados por ele são, na verdade, o mesmo recurso identificado por URIs distintos, ou seja, representam a mesma entidade do mundo real.

Como é possível ver na figura 7, este processo compara os nomes das palavras-chave e áreas de conhecimento da conferência (mapeados na propridade ***skos:prefLabel***) com os nomes ou rótulos dos recursos da DBpedia (propriedade ***rdfs:label***), aplicando-se alguns filtros (retângulos verdes) e utilizando, para comparação de cadeias de caracteres, o algoritmo da Distância de *Levenshtein* (Bilenko *et al.*, 2003) (retângulo laranja), cuja implementação já se encontra disponível no *Silk*.

As triplas resultantes do *mashup* foram armazenadas em um banco de dados RDF separado dos dados relacionais originais.

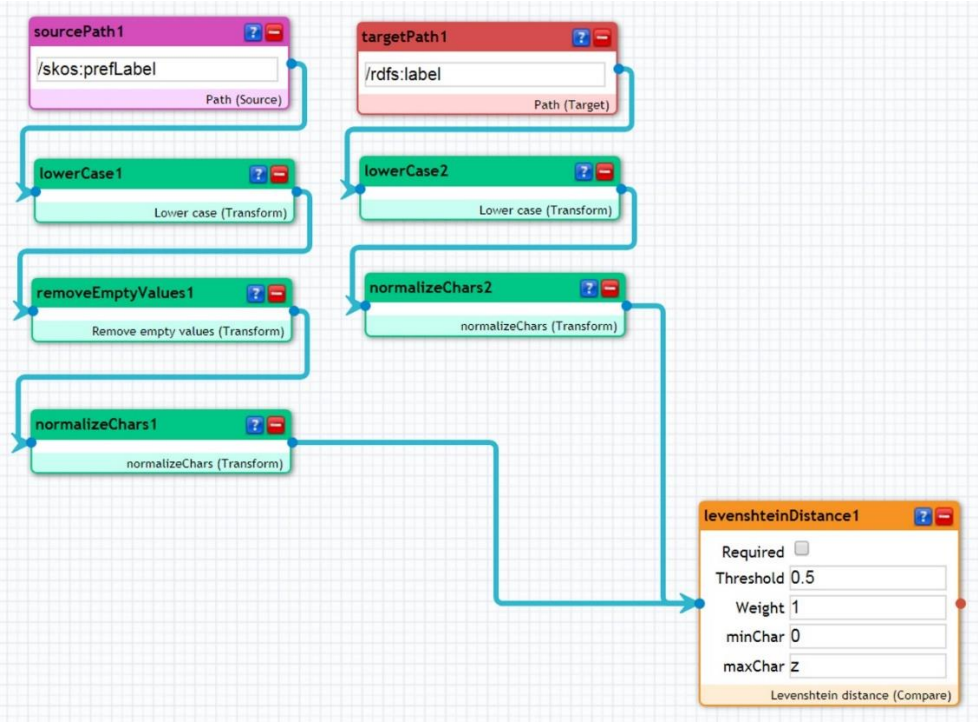

Figura 7 - Workflow do *Silk* para procurar ligações do tipo *owl:sameAs* entre os recursos da conferência e os recursos da DBpedia, com base nos respectivos nomes (labels).Fonte: Elaborados pelos autores (2017).

5.1.5 Publicação dos dados

Para publicar na Web os dados mapeados e ligados, foram usadas duas abordagens:

1. A disponibilização dos dois *SPARQL endpoints:* do *D2RQ* sobre os dados relacionais e do banco de dados RDF com o *mashup* com a DBpedia, permitindo a realização de consultas SPARQL federadas;
2. A exportação de todas as triplas geradas pelo *D2RQ* a partir dos dados relacionais originais, por meio do software ***dump-rdf***, distribuído junto à plataforma *D2RQ*, e a importação destas triplas e das triplas de *mashup* geradas pelo *Silk* em um terceiro banco de dados RDF centralizado independente, a ser atualizado em lote, de tempos em tempos, divulgando-se assim um único SPARQL *endpoint*. Mesmo neste caso, a interface Web do *D2RQ* precisou ser mantida, para que os URIs dos recursos pudessem ser dereferenciados (acessados).

Neste estudo de caso foram avaliados dois bancos de dados RDF, a saber: *Stardog* (Stardog Union, 2018) e *GraphDB* (Ontotext, 2018), ambos em suas versões gratuitas. Os dois

apresentaram resultados satisfatórios, do ponto de vista de viabilidade, desempenho e adequação ao método e arquitetura propostos.

5.1.6 Exemplo de mapeamento de uma publicação em RDF

Para ilustrar o mapeamento em RDF dos trabalhos acadêmicos da conferência, a figura 8 apresenta uma publicação (“Extração de regras em uma base de conhecimento por meio da navegação em um grafo”) em RDF (figura 8.a), representação para máquinas, e em HTML (figura 8.b), representação para humanos, ambas geradas pela plataforma *D2RQ*. As representações RDF e HTML de um autor são similares às representações da publicação, mudando apenas as propriedades e classes envolvidas.

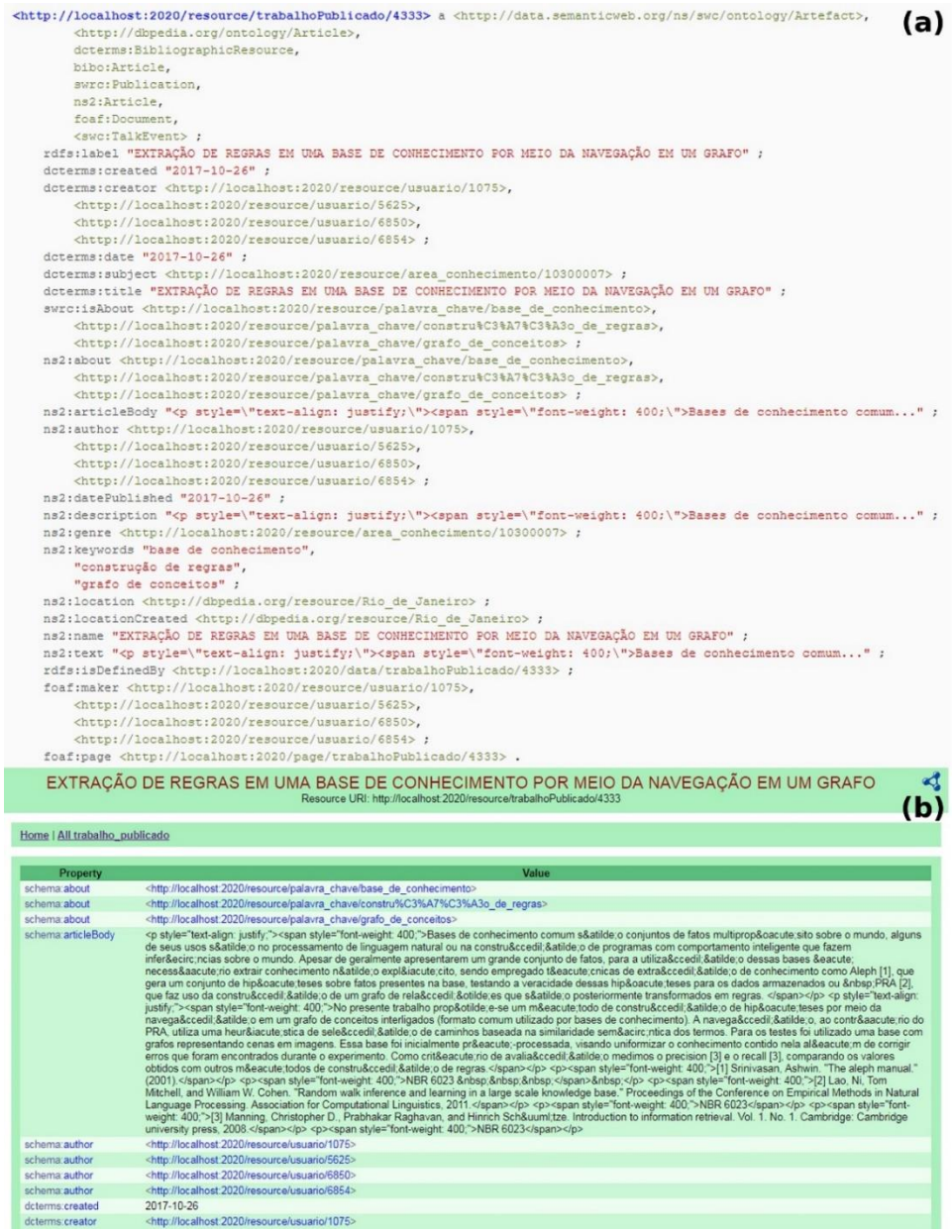

Figura 8 - Trecho da representação RDF (a) e da representação HTML (b) de uma publicação, geradas pela plataforma *D2RQ*. Fonte: Elaborados pelos autores (2018).

Para efeito de testes, foi utilizado o endereço local como URI base para o banco de dados semântico (*http://localhost:2020/*)[20]. Por razões de espaço, alguns valores textuais mais extensos foram abreviados com reticências na representação em RDF.

5.2 Estudo de Caso 2 – Integração de Dados de Sistemas de Informação

Esta seção descreve, em detalhes, o principal estudo de caso alvo deste trabalho, que consiste na aplicação da arquitetura e do método proposto, para a integração de dados de sistemas de informação, em uma situação real.

5.2.1 Contextualização

Este estudo de caso trata da integração de dados de sistemas de informação de uma grande universidade federal brasileira. Trata-se de uma instituição de grande porte, com dezenas de milhares de alunos ativos, e cuja administração é pulverizada em uma vasta gama de centros e unidades, embora todos respondam a uma única reitoria. Com isso, é comum encontrar dezenas de sistemas de informação e fontes de dados heterogêneas espalhadas pelas unidades, visando atender às constantes demandas gerenciais, no registro e uso das informações e no suporte aos processos de negócio. Também é muito comum encontrar redundância de dados entre esses sistemas, ou seja, uma mesma entidade, com seus dados relacionados, é cadastrada diversas vezes, sendo replicada em vários sistemas de informação, o que gera dificuldades na atualização e no acesso às informações de forma integrada.

De todos os sistemas de informação da instituição em questão, destacam-se dois principais, legados, que centralizam os seus mais importantes processos de negócio, inerentes a toda a instituição: o sistema de RH (SIRHU) e o acadêmico (Siga). O primeiro gerencia as funções de gerenciamento e processos relacionados a pessoal (Recursos Humanos), incluindo, por exemplo, contratações, desligamentos, benefícios e folhas de pagamento. É um sistema antigo e complexo, que vem sofrendo uma grande quantidade de mudanças ao longo dos anos, para atender a demandas gerenciais e mudanças legais. O Siga, também legado, é o sistema responsável por gerenciar os principais processos inerentes à atividade acadêmica, incluindo, por exemplo, o gerenciamento de cursos, alunos, professores e turmas, abrangendo uma infinidade de processos complexos, construídos ao longo dos anos. Os dois sistemas utilizam o mesmo SGBD relacional, compartilhando entre si as suas bases de dados. E estes, juntos,

20 Quando a aplicação for, de fato, posta em produção e publicada na Web, o endereço local http://localhost:2020 será substituído por um domínio real (*http://example.com*) ainda a ser definido.

constituem o mais importante banco de dados da instituição, contendo milhões de dados atualizados e acessados todos os dias por uma expressiva quantidade de usuários.

Dentre os demais diversos sistemas de informações gerenciais presentes na instituição, encontra-se o PLANID, que atende a uma demanda de alguns dos centros da universidade e de órgãos governamentais de fiscalização. Trata-se de uma aplicação onde os professores registram seus planos de trabalho para um determinado período, incluindo suas atividades de lecionação, pesquisa e extensão, a serem disponibilizados para a administração dos centros, como apoio à tomada de decisão, e para o público em geral, atendendo a necessidades legais de acesso à informação. No momento do levantamento dos requisitos para este estudo de caso, o sistema padecia da dificuldade de manter os dados acadêmicos referentes aos docentes sincronizados com as informações do sistema acadêmico da universidade (Siga). A alimentação era feita manualmente ou por meio de scripts de importação, a partir de dados solicitados pessoalmente à equipe de desenvolvimento responsável, processo este nem sempre acessível em tempo hábil. Sendo assim, esta limitação acabava por gerar constantes solicitações para atualização ou inserção de dados, o que diminuía a eficiência do uso do sistema, insatisfação de usuários finais e aumento das solicitações de suporte.

Outra demanda abordada por este estudo de caso diz respeito à necessidade de integração dos dados do Siga e SIRHU com os dados do Sistema SIAC, apresentado no estudo de caso anterior, visando atender a dois requisitos, solicitados pelos seus *stakeholders*:

1. Agilização do processo de preenchimento do formulário de inscrição no evento, uma vez que algumas das informações podem ser obtidas dos principais sistemas da universidade;
2. A geração de relatórios combinando informações acadêmicas com dados de publicações no evento.

Este trabalho pretende realizar um ensaio, atendendo a estas demandas específicas de integração de dados, envolvendo informações oriundas dos principais sistemas de informação da universidade, além do sistema PLANID e do Sistema SIAC, por meio da estrutura montada no primeiro estudo de caso. Este esforço contempla a instanciação do método e arquitetura propostos, envolvendo o emprego de um conjunto de ontologias consagradas, o estabelecimento de dados ligados dentro da instituição, no formato RDF, e a seleção de softwares necessários para a publicação, acesso e integração desses dados, dentro da universidade. Os tópicos a seguir apresentam a implementação dos passos do método proposto neste estudo de caso.

5.2.2 Analisar fontes de dados (definir informações a publicar)

Uma vez conhecidas as fontes de dados apresentadas na contextualização, identificou-se, em reunião com a equipe do sistema PLANID, a necessidade de ler e importar, a partir do Siga e SIRHU, as seguintes informações:

- Lista de professores, com seus dados pessoais e regime de contratação (20 horas, 40 horas, etc.);
- Lista de cursos aos quais os professores estão vinculados;
- Lista de disciplinas lecionadas pelos professores em um determinado período letivo.

Diante do requisitado, seguiu-se um processo de análise dos dados dos referidos sistemas, incluindo reuniões com equipes dos sistemas envolvidos. Este processo encontrou dificuldades devido à insuficiência de documentação disponível e do pouco contato entre equipes e deu-se, em grande parte, a partir da análise de *views* SQL já existentes na base de dados. Diante dos requisitos levantados, construiu-se um modelo conceitual, alvo do passo a seguir.

5.2.3 Gerar um modelo conceitual e do modelo ontológico integrador

Com base nos requisitos de integração, foi criado um modelo conceitual, por meio de um diagrama de classes UML, sobre o qual foi construído o modelo ontológico da solução. Com o objetivo de melhor visualização e descrição do domínio envolvido, promovendo melhor reúso da ontologia criada, foram adicionadas mais classes e propriedades ao modelo, além dos solicitados nos requisitos. O modelo ontológico final é apresentado na figura 9. Os parágrafos seguintes apresentam uma breve descrição de cada uma das ontologias selecionadas para o estudo de caso e, logo após, são explicadas as classes e propriedades apresentadas no modelo.

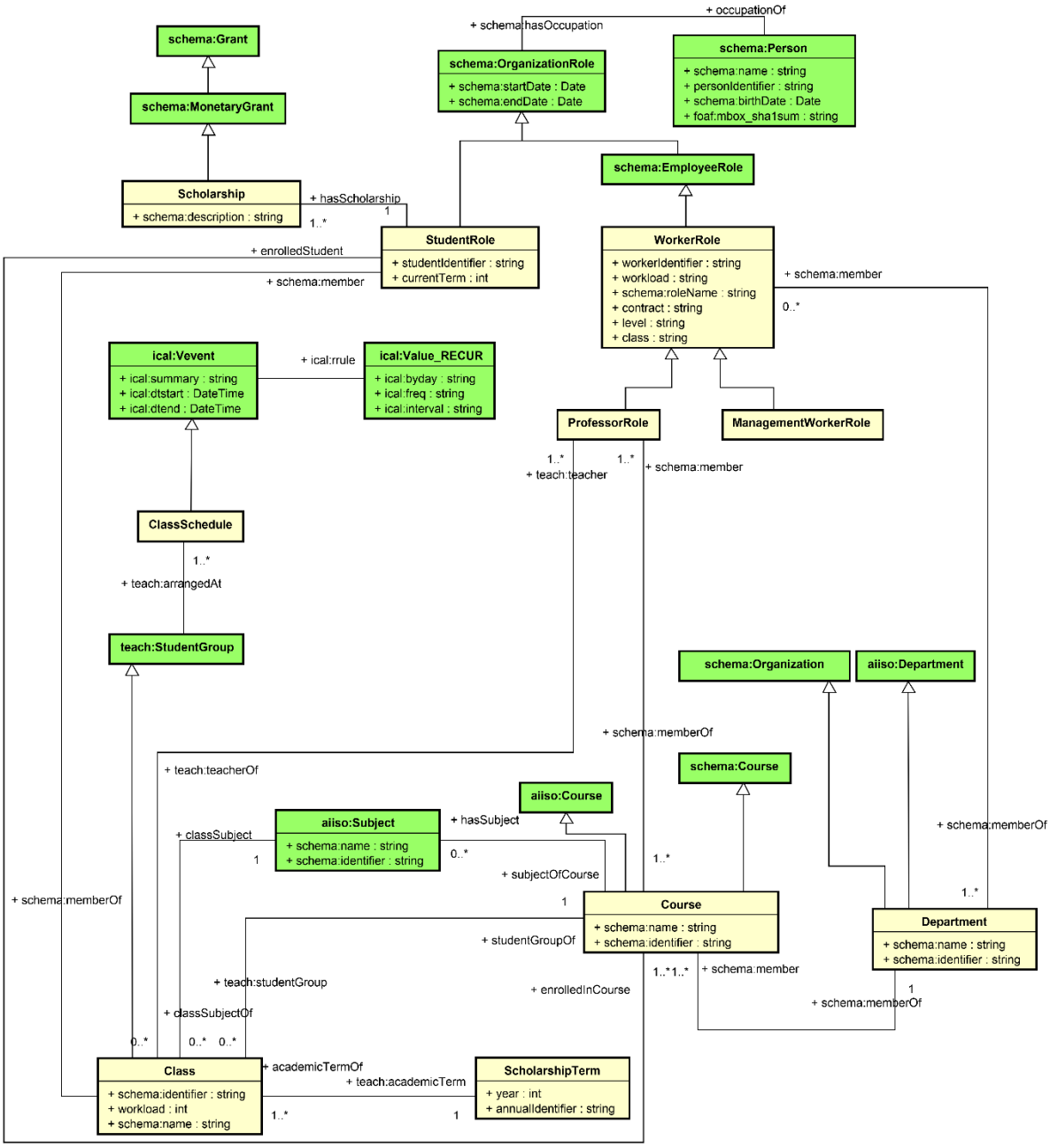

Figura 9 - Modelo ontológico para o domínio do estudo de caso, representado através de um diagrama de classes UML. Fonte: Elaborados pelos autores (2019).

Ontologias selecionadas

O modelo foi baseado em ontologias disponíveis na Web, obtidas por meio de pesquisas no catálogo *Linked Open Vocabularies* (LOV) (Vandenbussche *et al.*, 2016), além de sítios de busca e artigos acadêmicos, as quais são brevemente descritas a seguir.

Schema.org (Schema.org Community Group, [s.d.]): ontologia resultante de um empreendimento conjunto das empresas Google, Microsoft, Yahoo e Yandex (Yandex, [s.d.]). A ontologia provê um vocabulário para vários domínios de conhecimento (*crossdomain*), com o objetivo de inserir metadados estruturados em páginas *Web*, viabilizando resultados mais

precisos pelas máquinas de busca. Para abreviar URIs, o prefixo "schema" foi utilizado para referenciar o *namespace* da ontologia (*http://schema.org/*).

Academic Institution Internal Structure Ontology (*AIISO*) (Styles *et al.*, 2008): ontologia para descrever a estrutura interna de uma instituição acadêmica. Para abreviar URIs, o prefixo "aiiso" foi utilizado para referenciar o *namespace* da ontologia (*http://purl.org/vocab/aiiso/schema#*).

Teaching Core Vocabulary (Kauppinen *et al.*, 2012): vocabulário que provê termos para que professores possam descrever ligações entre elementos de seus cursos. Para abreviar URIs, o prefixo "teach" foi utilizado para referenciar o *namespace* da ontologia (*http://linkedscience.org/teach/ns#*).

RDF Calendar (Connolly & Miller, 2005): trata-se de um esforço para a migração do padrão iCalendar (RFC2554) para RDF, com o objetivo de integrar conteúdo da internet e dados de redes sociais com informações de calendário, através de Web Semântica. - Para abreviar URIs, o prefixo "ical" foi utilizado para referenciar o *namespace* da ontologia (*http://www.w3.org/2002/12/cal/icaltzd#*).

Friend Of A Friend (FOAF) (Brickley & Miller, 2014): ontologia para descrever pessoas, suas atividades e relações com outras pessoas, grupos, entidades e documentos (Azevedo & Jacyntho, 2014). Para abreviar URIs, o prefixo "foaf" foi utilizado para referenciar o *namespace* da ontologia (*http://xmlns.com/foaf/0.1/*).

Para descrever as novas classes e propriedades desenvolvidas para o domínio estudado neste trabalho, foi desenvolvida uma ontologia OWL, que foi nomeada *Public University Generic Ontology*, com *namespace http://purl.org/pugo/1.0#*. O processo de criação da ontologia foi baseado no método proposto em Noy e McGuinness (2001), e as suas classes e propriedades são apresentadas sem prefixo na figura 9 e nas demais referências a elas ao longo do texto.

Descrição das classes

A seguir, são descritas, em linhas gerais, as classes e propriedades criadas da ontologia *Public University Generic Ontology*. A tabela 1 descreve as classes da ontologia. A tabela 2 descreve o papel de algumas das classes de outras ontologias, reusadas neste modelo. A tabela 3 descreve algumas das propriedades utilizadas no modelo. A coluna “Classe no modelo” representa a classe em que as propriedades são representadas no diagrama da figura 9, cujas instâncias desempenharão o papel de sujeito nas triplas RDF. As propriedades criadas para

este modelo estão destacadas em negrito na tabela 3. As demais propriedades são autoexplicativas ou seguem o significado descrito nas documentações de suas respectivas ontologias.

Nome	**Superclasse**	**Descrição**
Class	*teach:StudentGroup*	Representa a turma de uma disciplina (*aiiso:Subject*), em um período letivo (*ScholarshipTerm*), tendo como membros alunos (*StudentRole*), e lecionada por um ou mais professores (*ProfessorRole*).
ClassSchedule	*ical:Vevent*	Descreve uma agenda de aulas para uma turma, através das regras estabelecidas em recursos do tipo *ical:Value_RECUR*.
Course	*aiiso:Course, schema:Course*	Descreve um curso oferecido pela universidade.
Department	*aiiso:Department, schema:Organization*	Descreve um departamento (unidade) da instituição.
ManagementWorkerRole	*WorkerRole*	Papel exercido por uma pessoa física como funcionário na universidade (principalmente servidores, no caso da universidade pública), atuante nas áreas administrativas.
ProfessorRole	*WorkerRole*	Papel exercido por uma pessoa física como docente na universidade.
Scholarship	*schema:MonetaryGrant*	Bolsa de estudos (auxílio financeiro).
ScholarshipTerm	*owl:Thing*	Período letivo.

StudentRole	*schema:OrganizationRole*	Representa a relação de uma pessoa com a universidade na condição de aluno.
WorkerRole	*schema:EmployeeRole*	Papel exercido por uma pessoa que realiza qualquer tipo de trabalho para a universidade, como funcionário ou servidor.

Tabela 1- Descrição das classes criadas para a ontologia Public University Generic Ontology

Nome	**Descrição**
schema:Person	Representa uma pessoa física, a desempenhar um ou mais papéis no modelo.
ical:Value_RECUR	Juntamente com classe *ClassSchedule*, descreve a frequência, datas e horários das aulas de uma turma (*Class*).
aiiso:Subject	Descreve uma disciplina, oferecida em um ou mais cursos (*Course*) e associada a uma ou mais turmas (*Class*).

Tabela 2 - Descrição do papel de classes importadas de outras ontologias

Classe no modelo	**Propriedade**	**Descrição**
Class	*schema:identifier*	Representa um identificador para a turma.
	classSubject	Disciplina (*aiiso:Subject*) da turma. Inversa à ***classSubjectOf***.
Course	***hasSubject***	Associa uma disciplina a um ou mais cursos (*Course*). Inversa à propriedade ***subjectOfCourse***.
ScholarshipTerm	***year***	Ano do período letivo.
	annualIdentifier	Identificador do período (“1”, por exemplo, para um período 01/2019);

StudentRole	***studentIdentifier***	Identificador (matrícula) do estudante. No caso da universidade em estudo, é um identificador numérico chamado DRE.
	currentTerm	Período letivo atual em que o aluno se encontra (xsd:int)
	enrolledInCourse	Associa um aluno (*StudentRole*) a um curso, no qual está matriculado.
WorkerRole	***class***	Classe do professor (referente à contratação)
	contract	Tipo de contrato do professor (por ex.: “DE”, “40h”)
	level	Nível do professor (referente à contratação)
	schema:roleName	Descrição do cargo do funcionário/servidor
	workerIdentifier	Identificador (matrícula) do funcionário. No caso da universidade, é um identificado numérico de servidor público federal chamado SIAPE
	workload	Carga horária do funcionário/servidor
schema:Person	*foaf:mbox_sha1sum*	Segue a especificação original da propriedade na ontologia *FOAF*: trata-se de uma forma de associar a pessoa a um endereço de e-mail, porém expondo apenas o seu hash. Destaca-se por ser tipo de propriedade que pode ser utilizada para encontrar *links* entre recursos representando a mesma pessoa em diferentes origens de dados.

	personIdentifier	Identificador de pessoa física, como identidade ou CPF. No caso da universidade em estudo, pode ser o CPF ou passaporte (no caso de estrangeiros);
aiiso:Subject	*schema:identifier*	Identificador (código) da disciplina.
	workload	Carga horária da disciplina
	schema:name	Nome de disciplina

Tabela 3 - Descrição das propriedades utilizadas no modelo ontológico

As descrições das demais classes e propriedades importadas de outras ontologias estão disponíveis nas suas respectivas documentações.

Exemplo de instanciação do modelo

A figura 10 apresenta um exemplo, com dados hipotéticos, de utilização do modelo na descrição de um professor que teria ministrado aulas para uma turma da disciplina “Web Semântica”, com aulas semanais às quintas-feiras, de 13 às 15hs, no período letivo 2019/2, utilizando-se a sintaxe *Turtle*.

```
###  http://grafo.universidade.br/cursos/1234-5678-9011-2222
<http://grafo.universidade.br/cursos/1234-5678-9011-2222> rdf:type pugo:Course ;
                                                          teach:studentGroup <http://grafo.universidade.br/turmas/8374-3934-0303-22J4> ;
                                                          schema:member <http://grafo.universidade.br/servidores/98765432> ;
                                                          schema:name "Ciência da Computação"^^xsd:string .


###  http://grafo.universidade.br/disciplinas/7777-4444-3333-2222
<http://grafo.universidade.br/disciplinas/7777-4444-3333-2222> rdf:type aiiso:Subject ;
                                                               schema:name "Web Semântica"^^xsd:string .


###  http://grafo.universidade.br/horariosTurmas/9393-848407575-6363
<http://grafo.universidade.br/horariosTurmas/9393-848407575-6363> rdf:type pugo:ClassSchedule ;
                                                                  ical:rrule [ rdf:type ical:Value_RECUR ;
                                                                               ical:byday "TH" ;
                                                                               ical:freq "WEEKLY" ;
                                                                               ical:interval "1"^^xsd:int
                                                                             ] ;
                                                                  ical:dtend "2019-01-01T15:00:00"^^xsd:dateTime ;
                                                                  ical:dtstart "2019-01-01T13:00:00"^^xsd:dateTime .


###  http://grafo.universidade.br/pessoas/7c4a8d09ca3762a
<http://grafo.universidade.br/pessoas/7c4a8d09ca3762a> rdf:type schema:Person ;
                                                       schema:hasOccupation <http://grafo.universidade.br/servidores/98765432> ;
                                                       pugo:personIdentifier "12345678900"^^xsd:string ;
                                                       schema:birthDate "1984-04-24"^^xsd:date ;
                                                       schema:name "Adriano de Oliveira Gonçalves" .


###  http://grafo.universidade.br/segmentacoes/2222-4444-5555-6789
<http://grafo.universidade.br/segmentacoes/2222-4444-5555-6789> rdf:type pugo:ScholarshipTerm ;
                                                                pugo:annualIdentifier "2"^^xsd:int ;
                                                                pugo:year "2019"^^xsd:int .


###  http://grafo.universidade.br/servidores/98765432
<http://grafo.universidade.br/servidores/98765432> rdf:type pugo:ProfessorRole ;
                                                   pugo:contract "DE" ;
                                                   pugo:workerIdentifier "98765432"^^xsd:string ;
                                                   schema:roleName "Professor do Magistério Superior" .


###  http://grafo.universidade.br/turmas/8374-3934-0303-22J4
<http://grafo.universidade.br/turmas/8374-3934-0303-22J4> rdf:type pugo:Class ;
                                                          teach:academicTerm <http://grafo.universidade.br/segmentacoes/2222-4444-5555-6789> ;
                                                          teach:arrangedAt <http://grafo.universidade.br/horariosTurmas/9393-848407575-6363> ;
                                                          teach:teacher <http://grafo.universidade.br/servidores/98765432> ;
                                                          pugo:classSubject <http://grafo.universidade.br/disciplinas/7777-4444-3333-2222> ;
                                                          pugo:workload "60"^^xsd:int .


###  http://grafo.universidade.br/unidades/36010500
<http://grafo.universidade.br/unidades/36010500> rdf:type pugo:Department ;
                                                 schema:member <http://grafo.universidade.br/servidores/98765432> ;
                                                 schema:identifier "36010500"^^xsd:string ;
                                                 schema:name "Departamento de Informática" .
```

Figura 10 - Exemplo de instanciação do modelo ontológico proposto, na sintaxe Turtle. Fonte: Elaborados pelos autores (2020).

Considerando os conjuntos de triplas na figura 10, pode-se observar:

- O primeiro conjunto descrevendo um curso de nome “Ciência da Computação” ao qual estão vinculados uma turma e um professor, identificados por seus respectivos URIs;
- O segundo conjunto, que descreve uma disciplina de nome “Web Semântica”;
- O terceiro conjunto descrevendo uma agenda de horários semanais, a ser utilizada por uma turma;
- O quarto conjunto trazendo a descrição de uma pessoa, de nome “Adriano de Oliveira Gonçalves”, com seu CPF e data de nascimento, e associado a uma instância de papel (ocupação), por seu respectivo URI;
- O quinto conjunto descrevendo um período letivo (02/2019);

- O sexto conjunto, que descreve o papel de professor referenciado pela pessoa descrita previamente, com sua matrícula de servidor, e mais alguns dados referentes à sua contratação;
- O sétimo conjunto, descrevendo uma turma associada, por URIs, a um período letivo, uma agenda, um professor (papel) e uma disciplina, além de sua carga horária;
- Por fim, o oitavo conjunto, trazendo a descrição de um departamento de nome "Departamento de Informática", ao que está vinculado o professor previamente descrito.

5.2.4 Gerar visão (grafo) RDF somente leitura sobre os dados relacionais

Nesta etapa foram criadas *views* SQL sobre os bancos de dados do Siga e SIRHU, de forma a fornecer uma interface de dados capaz de preencher as informações do modelo conceitual. Algumas *views* já existentes foram utilizadas como base para as *views* de integração, enquanto outras foram totalmente implementadas.

Como ferramenta *RDB2RDF*, adotou-se *Ontop* para mapear os dados dos bancos de dados dos sistemas Siga e SIRHU. A sua configuração foi realizada através do plugin fornecido com a ferramenta para o editor de ontologias *Protegé*, conforme demonstrado nas figuras 11 e 12.

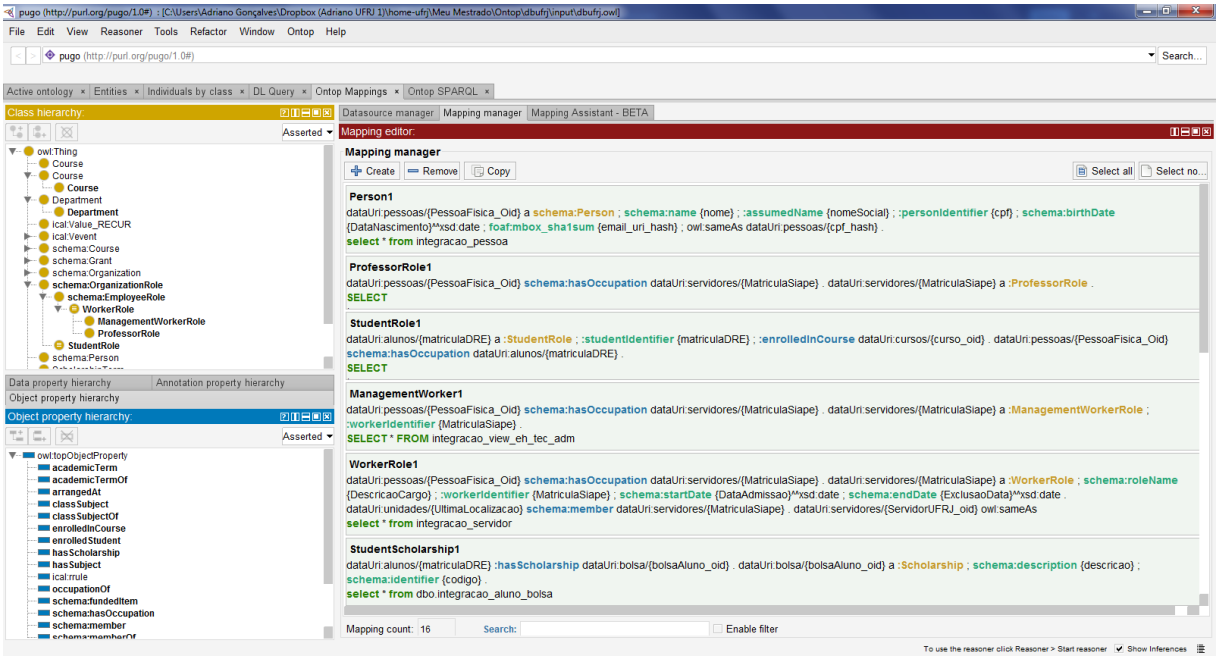

Figura 11 - Configuração do mapeamento Relacional – RDF, através do plugin oferecido pela ferramenta *Ontop*. Fonte: Elaborados pelos autores (2020).

A figura 11 apresenta a principal tela de mapeamento do plugin do *Ontop* para *Protegé*. Cada linha listada no painel à direita, representa um conjunto de mapeamentos, descritos na linguagem OBDA, a linguagem de mapeamentos do *Ontop*. Abrindo-se a edição de uma dessas linhas, visualiza-se a tela apresentada na figura 12.

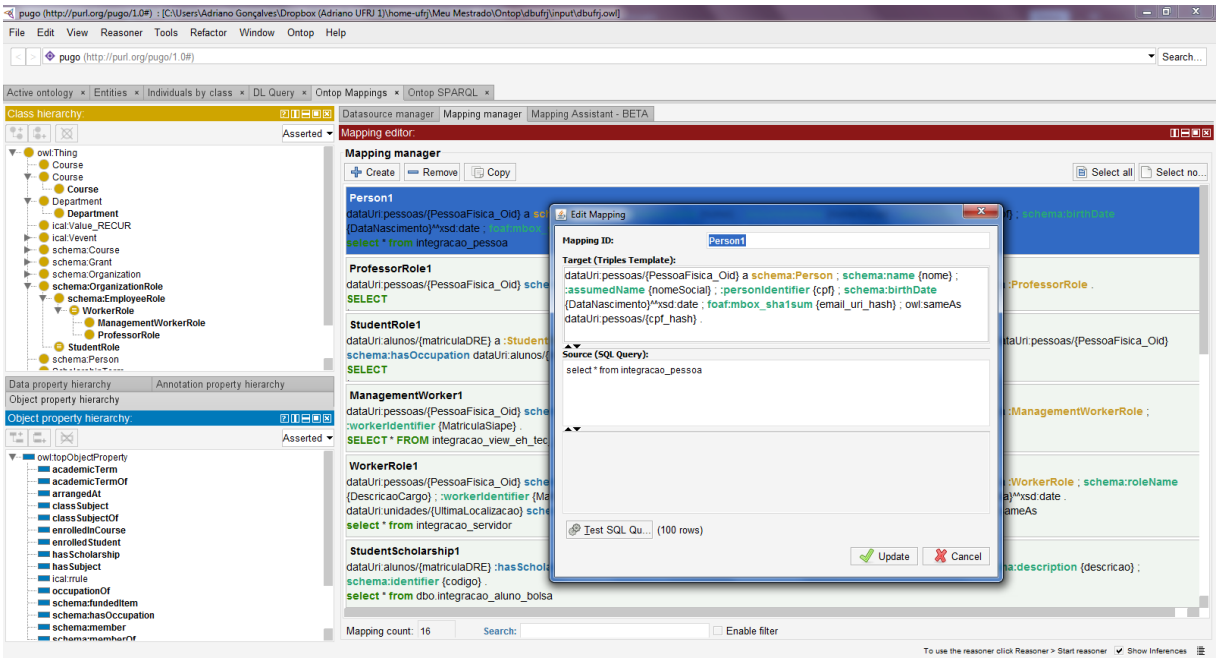

Figura 12 - Tela de configuração de um dos mapeamentos Relacional-RDF, através do plugin oferecido pela ferramenta *Ontop*. Fonte: Elaborados pelos autores (2020).

Extraindo-se como exemplo, a figura 12 apresenta um dos conjuntos de mapeamento, onde o campo *Source*, do formulário, descreve uma *query* SQL a ser utilizada como origem dos dados mapeados (**SELECT * FROM integracao_pessoa**), enquanto que o *Target* descreve as regras de mapeamento a serem aplicadas à origem. Neste exemplo, o mapeamento configurado foi o seguinte:

> *dataUri:pessoas/{PessoaFisica_Oid} a schema:Person ; schema:name {nome} ; :assumedName {nomeSocial} ; :personIdentifier {cpf} ; schema:birthDate {DataNascimento}^^xsd:date ; foaf:mbox_sha1sum {email_uri_hash} ; owl:sameAs dataUri:pessoas/{cpf_hash} .*

O código acima define que, como resultado do mapeamento dos dados, serão descritos recursos da classe ***schema:Person***, identificados por URIs seguindo o padrão ***dataUri:pessoas/{PessoaFisica_Oid}***, onde *dataUri* é um prefixo HTTP previamente definido, e **PessoaFisica_Oid** é um campo retornado pela *query* SQL. As informações seguintes descrevem, a cada ponto-e-vírgula, uma nova tripla que será gerada para cada um

desses recursos, baseados em propriedades das ontologias escolhidas e nos campos retornados pela *query*.

Configuração dos URIs

A configuração dos URIs deu-se da seguinte forma:

- Como prefixo, foi utilizado o URI base *http://grafo.universidade.br/*[21];
- Para os URIs dos recursos representando pessoas, cursos, turmas, bolsas de estudo e agenda de aulas, foram utilizados identificadores únicos do banco de dados (OIDs). O URI de uma pessoa, por exemplo, ficou como *<http://grafo.universidade.br/pessoas/1234-5678-AB33-CD44>*; e da mesma forma, os demais recursos;
- Para os URIs de papéis de aluno (*StudentRole*), professor (*ProfessorRole*), funcionário (*MangementWorkerRole*) foram utilizados os números de matrícula de aluno (DRE) ou de servidor público (SIAPE), como, por exemplo: *<http://grafo.universidade.br/alunos/123456 >*;
- Para os departamentos foi utilizado o código oficial do departamento na instituição.

Além dos URIs principais das pessoas, foi criado um URI adicional, baseado no *hash* SHA1 da sua identificação (CPF ou passaporte), adicionado ao mapeamento através do predicado ***owl:sameAs***. Este URI não foi utilizado como principal, neste estudo de caso, por questões de desempenho. A proposta é que este padrão possa ser utilizado para todos os sistemas onde não seja viável utilizá-lo como principal. Mais detalhes sobre esta técnica estão descritos na seção 4.3.

5.2.5 Consulta aos dados integrados

Para este estudo de caso, foram realizados testes em dois cenários:

1. Extração de dados pelos sistemas PLANID e SIAC
2. Relatórios cruzando dados entre sistemas Siga, SIRHU e SIAC

Para cada cenário, foi utilizada tanto a abordagem federada quanto a centralizada previstas na arquitetura.

[21] Este é um URI fictício. O URI original foi omitido devido a questões de tratamento de informações, pela unidade institucional envolvida neste trabalho.

Cenário 1 - extração de dados pelos sistemas PLANID e SIAC

O cenário **1** trata da solução para o problema da importação dos dados dos professores no sistema PLANID. Como é possível observar nas figuras 13 e 14, a ferramenta *Ontop* foi utilizada em ambas as abordagens, para mapear os dados relacionais do Siga e SIRHU para RDF. Como *triple store*, para a abordagem centralizada, foi utilizado o banco de dados *GraphDB*, devido ao seu desempenho positivo no estudo de caso anterior.

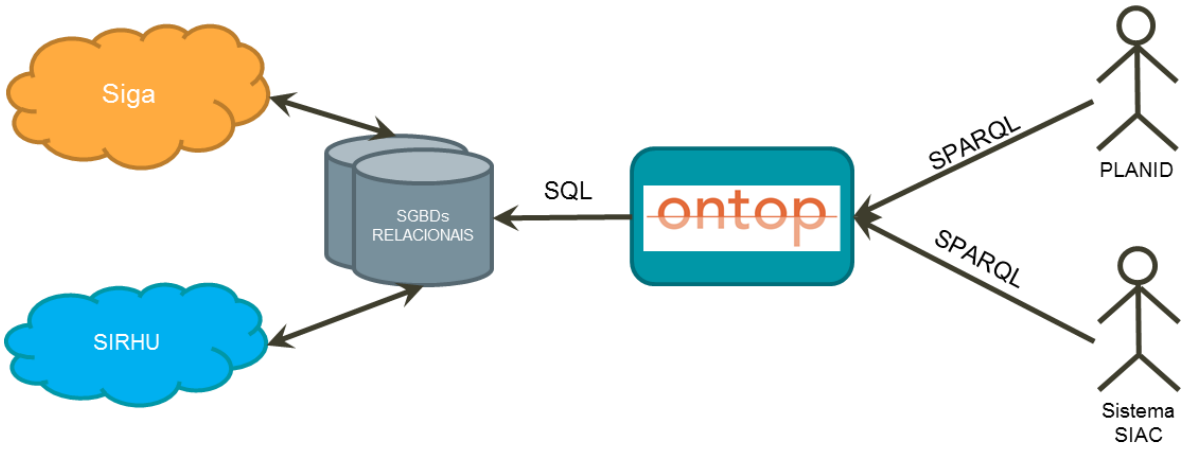

Figura 13 – Cenário de teste 1 utilizando consulta SPARQL com mapeamento *on-the-fly*. Fonte: Elaborado pelos autores (2020)

Na abordagem apresentada na figura 13, *Ontop* foi instanciado através de um container *Docker*, conectando-se diretamente aos bancos de dados dos sistemas Siga e SIRHU, e disponibilizando um *SPARQL endpoint* a ser consultado pelos sistemas PLANID e SIAC.

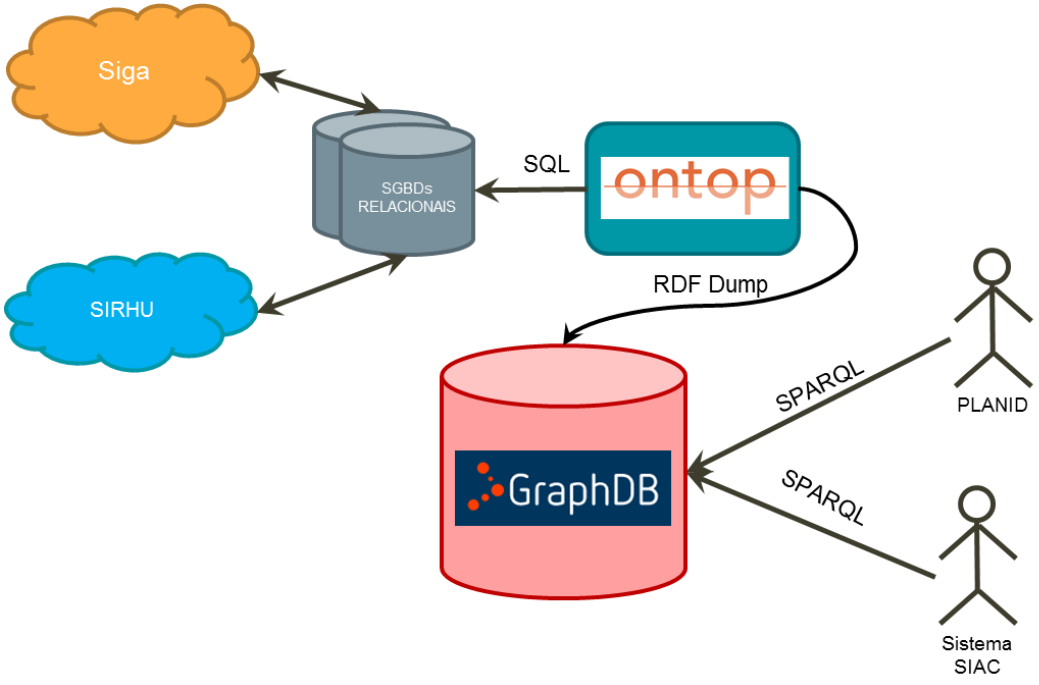

Figura 14 Cenário 1 utilizando a abordagem centralizada. Fonte: Elaborado pelos autores (2020)

Na arquitetura apresentada na figura 14, o *Ontop* foi utilizado, no modo cliente, para exportar um *dump* RDF das origens de dados, com base nos mapeamentos previamente configurados, o qual foi importado no *triple store GraphDB*. Com o objetivo de utilizar-se do recurso de inferência ontológica, a ontologia desenvolvida neste trabalho, na linguagem OWL, também foi importada no *triple store*, em um *named graph* diferente.

Seguindo a arquitetura apresentada, no sistema PLANID, desenvolvido na linguagem de programação Java, foi implementada uma rotina de importação dos dados dos professores utilizando-se a biblioteca *Jena*[22]. No Sistema SIAC, desenvolvido na linguagem PHP, foi utilizada a biblioteca *EasyRDF*[23] para extrair dados do *SPARQL endpoint*.

As figuras 15 e 16 demonstram, respectivamente, uma consulta realizada para obter dados dos professores, pelo sistema PLANID, e outra para obter os dados de um aluno, pelo Sistema SIAC. É pertinente comentar que, embora a importação dos cursos e disciplinas estivesse nos requisitos originais para o PLANID, e, embora estes tenham sido incluídos no modelo e no mapeamento dos dados, optou-se, por decisão de projeto, não os importar no momento desta implementação.

```
PREFIX schema: <http://schema.org/>
PREFIX pugo: <http://purl.org/pugo/1.0#>
SELECT DISTINCT ?cpf ?nome ?siape ?cargo ?nomeDepartamento ?codigoDepartamento ?regime ?nivel ?classe
{
    ?p pugo:personIdentifier ?cpf .
    ?p schema:name ?nome .
    ?p schema:hasOccupation ?r .
    ?r a pugo:ProfessorRole .
    ?r pugo:workerIdentifier ?siape .
    ?r schema:roleName ?cargo .
    ?dep a pugo:Department .
    ?dep schema:member ?r .
    ?dep schema:name ?nomeDepartamento .
    ?dep schema:identifier ?codigoDepartamento .
    VALUES ?codigoDepartamento  { "35240003" "38010200" } .
    ?r pugo:contract ?regime .
    ?r pugo:level ?nivel .
    ?r pugo:class ?classe .
}
```

Figura 15 - Consulta realizada pelo sistema PLANID para importação de dados dos professores de duas unidades da universidade. Fonte: Elaborado pelos autores (2020).

[22] https://jena.apache.org/

[23] http://www.easyrdf.org/

```
SELECT DISTINCT  ?nome ?bolsa ?codigoCurso ?nomeCurso ?periodoAluno ?dataNascimento
WHERE
{

    ?p schema:identifier '12345678900' .
    ?p schema:name ?nome .
    ?p schema:hasOccupation ?a .
    ?a a pugo:StudentRole .

    ?a pugo:enrolledInCourse ?c .
    ?c schema:name ?nomeCurso .
    ?c schema:courseCode ?codigoCurso .
    ?a pugo:currentTerm ?periodoAluno .
    ?p schema:birthDate ?dataNascimento .

    OPTIONAL {
        ?a pugo:hasScholarship ?bolsa .
    } .
}
```

Figura 16 - Consulta realizada pelo Sistema SIAC para obtenção dos dados de um aluno, por CPF. Fonte: Elaborado pelos autores (2020).

As figuras 17 e 18 demonstram, respectivamente, recortes das telas do sistema PLANID (importação de dados) e SIAC (formulário de cadastro), ao extrair dados do grafo mapeado.

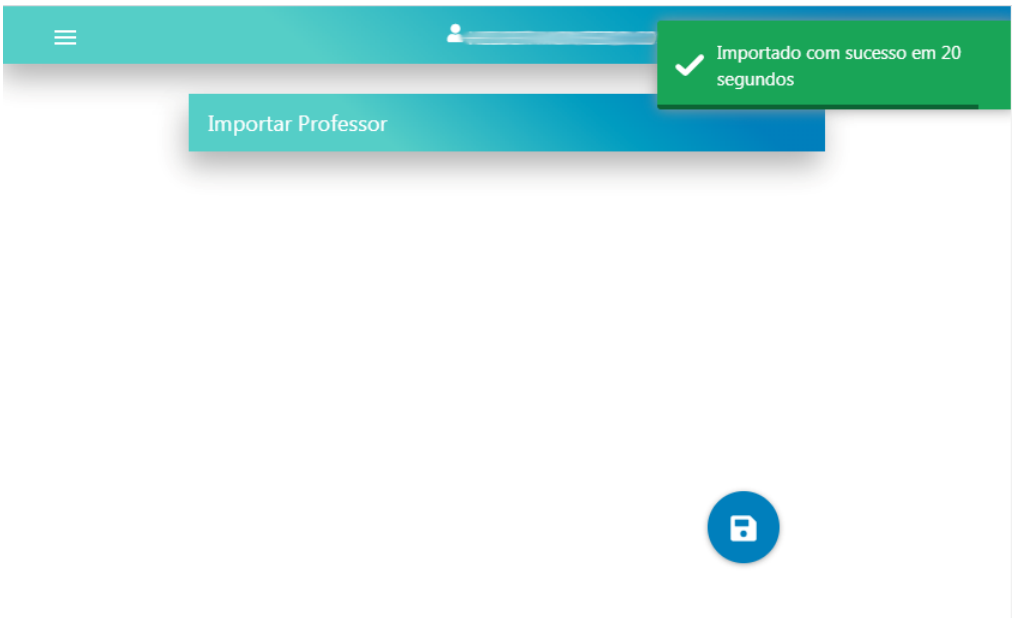

Figura 17 - Tela de importação de professores do sistema PLANID. Fonte: Elaborado pelos autores (2020).

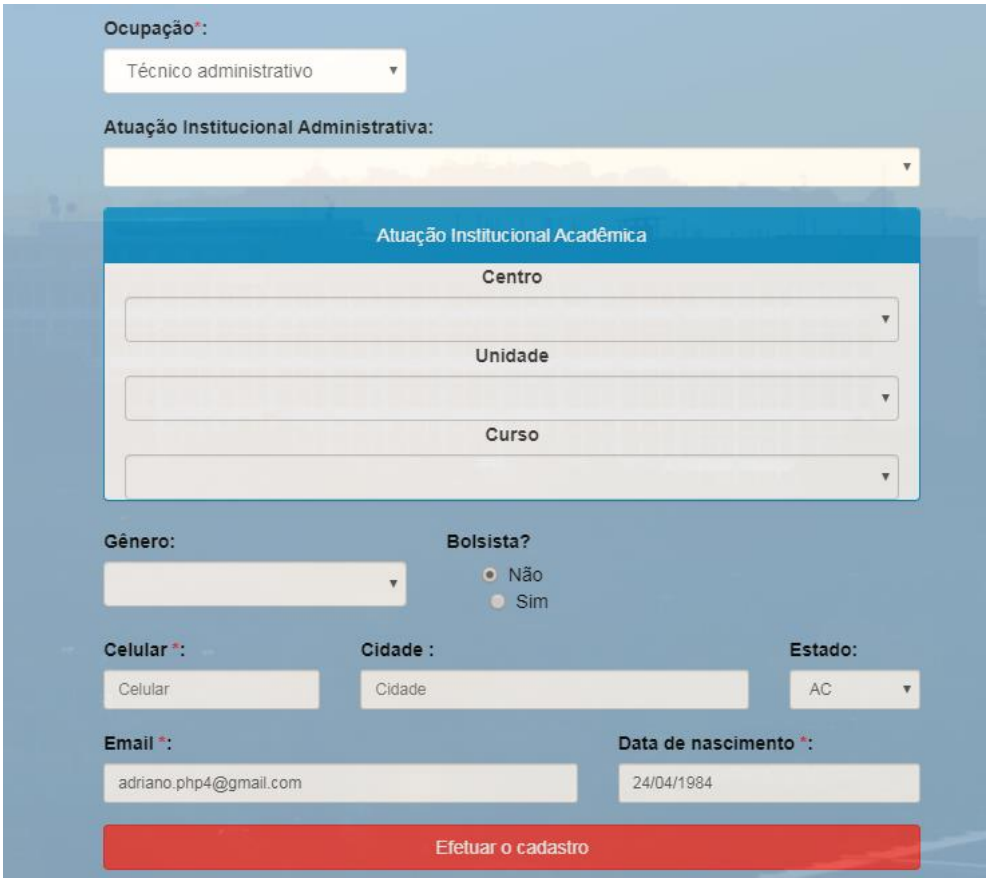

Figura 18 - Tela de cadastro do Sistema SIAC, preenchendo automaticamente alguns dos campos com informações obtidas do grafo RDF. Fonte: Elaborado pelos autores (2020).

As exportações e importações de *dumps* RDF no *triple store* foram realizadas manualmente, por razões de tempo. Possíveis alternativas para essa etapa são tratadas na conclusão deste trabalho, no capítulo 8, na parte de trabalhos futuros.

Cenário 2 - Relatórios cruzando dados entre os sistemas Siga, SIRHU e SIAC

O cenário 2, ilustrado nas figuras 19 e 20, trata do problema da geração de relatórios envolvendo dados dos sistemas Siga, SIRHU e SIAC. No tocante aos sistemas Siga e SIRHU, a instanciação da arquitetura deu-se da mesma forma que no cenário 1, explicado acima, utilizando-se *Ontop*.

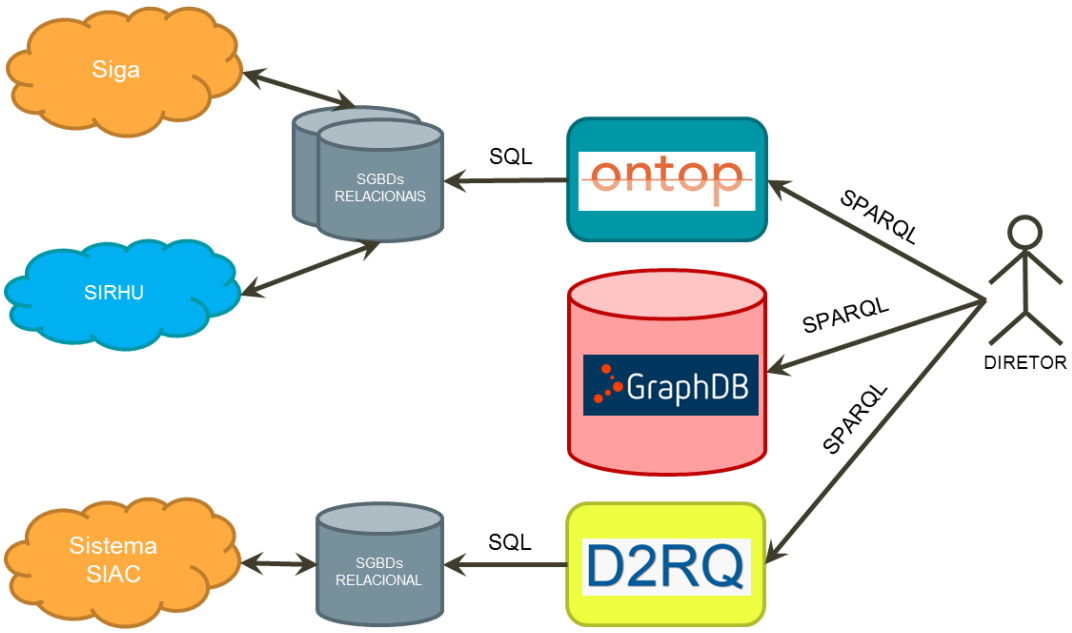

Figura 19 - Cenário 2 utilizando a abordagem federada. Fonte: Elaborado pelos autores (2020)

Perceba que, nestes casos, como é possível visualizar nas figuras 19 e 20, o Sistema SIAC não atua mais como ator, mas como fonte de dados. Para extração destes dados foi utilizada a estrutura elaborada no primeiro estudo de caso, de publicações acadêmicas, utilizando-se a ferramenta *D2RQ*, tanto para consultas *on-the-fly*, via *SPARQL endpoint* quanto para geração de *dumps* RDF, a serem importados no *GraphDB*. Com o objetivo de estabelecer ligação entre os dados mapeados das diferentes origens de dados, foi adicionado, ao modelo desenvolvido no primeiro estudo de caso, uma propriedade ***personIdentifier*** com o CPF/Passaporte, além de um URI alternativo, baseado no *hash* SHA1 da identificação, utilizando-se a propriedade ***owl:sameAs***. Na abordagem apresentada na figura 19, a ontologia criada neste trabalho foi importada no *triple store GraphDB*, que foi utilizado como *endpoint* para a execução das consultas federadas.

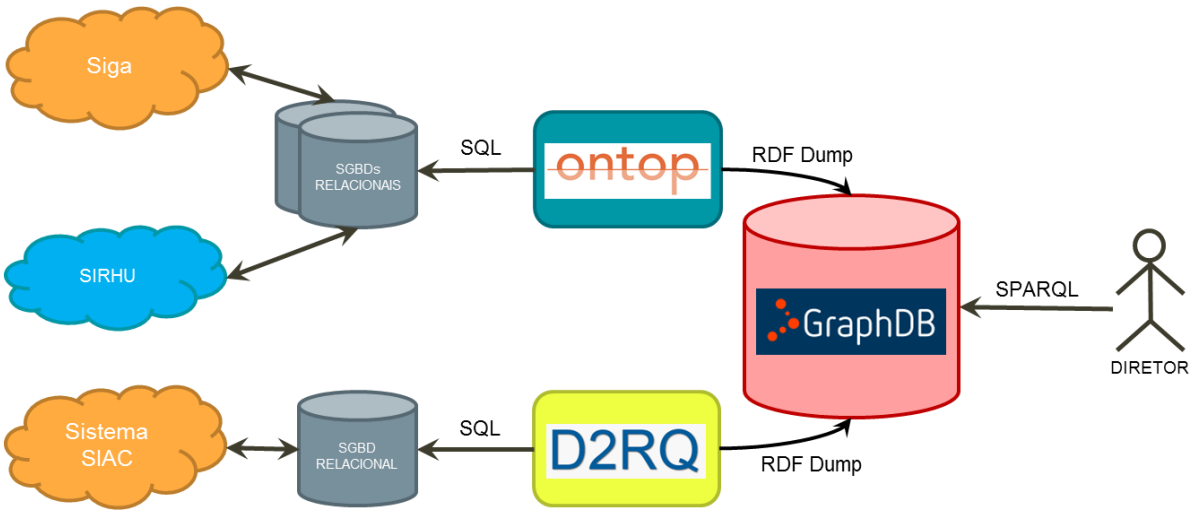

Figura 20 - Cenário 2 utilizando a abordagem centralizada. Fonte: Elaborado pelos autores (2020)

Na abordagem apresentada na figura 20, ambos os *dumps*, do *Ontop* e do *D2RQ*, foram importados no *GraphDB*, tal como a ontologia desenvolvida neste trabalho, cada um em um *named graph* diferente. Os relatórios finais foram testados na própria interface do *GraphDB*, na tela de consultas SPARQL. As figuras 21 e 22 apresentam as consultas SPARQL utilizadas para obtenção de um relatório hipotético, de alunos bolsistas que tenham trabalho publicado no evento gerenciado pelo Sistema SIAC, na abordagem centralizada e na abordagem federada, respectivamente. Perceba que na consulta da abordagem centralizada (figura 21), não é preciso usar o predicado *owl:sameAs* explicitamente para associar dois URIs distintos que representam o mesmo recurso, simplificando, sobremaneira, a consulta. Esta simplificação ocorre graças à inferência ontológica realizada pelo módulo raciocinador do *GraphDB*, com base nos axiomas de simetria e transitividade da propriedade *owl:sameAs*, fazendo com que as triplas relacionadas a primeiro URI sejam copiados no segundo e vice-versa.

```
SELECT DISTINCT ?nomePessoa ?nomeBolsa ?trabalho ?tituloTrabalho
WHERE
{
    ?p a schema:Person .
    ?p schema:name ?nomePessoa .
    ?p schema:hasOccupation ?o .
    ?o a pugo:StudentRole .
    ?o pugo:hasScholarship ?b .
    ?b schema:description ?nomeBolsa .
    ?trabalho a schema:Article .
    ?trabalho schema:author ?p .
    ?trabalho schema:name ?tituloTrabalho .
}
```

Figura 21 - Consulta SPARQL para obter alunos bolsistas que tenham trabalho publicado no evento gerenciado pelo Sistema SIAC, na abordagem centralizada. Fonte: Elaborado pelos autores (2020).

```
SELECT DISTINCT ?nomePessoa ?nomeBolsa ?trabalho ?tituloTrabalho
WHERE
{
    SERVICE <http://192.168.28.183:8080/sparql>{
        ?p a schema:Person .
        ?p schema:name ?nomePessoa .
        ?p schema:hasOccupation ?o .
        ?o a pugo:StudentRole .
        ?o pugo:hasScholarship ?b .
        ?b schema:description ?nomeBolsa .
        OPTIONAL{
            ?p owl:sameAs ?x .
        }
    }
    SERVICE <http://192.168.28.183:2020/sparql> {
      ?trabalho a schema:Article .
      ?trabalho schema:author ?p2 .
      ?trabalho schema:name ?tituloTrabalho .
        {
            {?p2 owl:sameAs ?x .}
            UNION
            {?p2 owl:sameAs ?p .}
            UNION
            {?trabalho schema:author ?p .}
        }
   }
}
```

Figura 22 - Consulta SPARQL para obter alunos bolsistas que tenham trabalho publicado no evento gerenciado pelo Sistema SIAC, na abordagem federada. Fonte: Elaborado pelos autores (2020).

6 RESULTADOS E DISCUSSÕES

Esta seção apresenta os resultados alcançados por esta pesquisa.

6.1 Publicação semântica *Linked Data* de bases de dados convencionais

Como é possível observar nos exemplos apresentados no tópico 5.1.5, diante dos ensaios realizados com o modelo ontológico proposto, constatou-se ser possível navegar com sucesso pelas informações da conferência, por meio da interface fornecida pelo *D2RQ*, tanto para humanos, em HTML, quanto para máquinas, em RDF e SPARQL, embora a interface tenha apresentado erros (exceções) durante testes com acessos simultâneos. Até este ponto, alcançou-se o nível quatro estrelas do modelo proposto por Berners-Lee (2009). Com o objetivo de alcançar o nível cinco estrelas, o processamento das triplas RDF por meio da ferramenta de *mashup Silk*, baseado no *workflow* apresentado na figura 7, conseguiu estabelecer *links* ***owl:sameAs*** entre as palavras-chave e áreas de conhecimento da conferência e recursos da DBpedia. De **10.784** palavras-chave e **1.335** áreas de conhecimento cadastradas no sistema, todos mapeados como instâncias da classe ***skos:Concept***, a ferramenta *Silk* foi capaz de encontrar, automaticamente, **6.923** *links*, realizando assim o *Linked Data mashup* de mais de 50% dos recursos processados.

Como exemplo de resultado de *mashup* obtido com o *Silk*, considerando ainda a publicação demonstrada na figura 8, dentre as suas três palavras chave (“base de conhecimento”, “construção de regras” e “grafos de conceitos”), mapeadas como instância de ***skos:Concept***, foi encontrada uma ligação automática para o termo “base de conhecimento” (*<http://localhost:2020/resource/palavra_chave/base_de_conhecimento>*) com recurso *<http://pt.dbpedia.org/resource/Base_de_conhecimento>* da DBpedia em Português, que, por sua vez, está ligado, dentro da própria base da DBpedia, através do predicado ***owl:sameAs***, ao recurso *<http://dbpedia.org/resource/Knowledge_base>* da DBpedia em Inglês. Este último recurso possui ainda mais informações úteis, sobre o respectivo conceito. Da mesma forma, a área de conhecimento da mesma publicação, “Ciência da Computação” (*<http://localhost:2020/resource/area_conhecimento/10300007>*) foi automaticamente associada ao recurso *<http://pt.dbpedia.org/resource/Ciência_da_computação>*, que, por sua vez, está ligado internamente, na DBpedia, ao recurso *<http://dbpedia.org/resource/Computer_science>*, que descreve as propriedades da área de conhecimento em questão.

Sobre os ensaios, com as duas abordagens de publicação de dados explicadas anteriormente, foram executadas consultas SPARQL nos dois cenários, experimentando os

dois bancos de dados RDF (*Stardog* e *GraphDB*), com seus respectivos raciocinadores de inferência ontológica habilitados. As consultas foram realizadas tanto usando explicitamente a propriedade *owl:sameAs* (figura 23.a), quanto usando inferência, com base nos axiomas de simetria e transitividade da propriedade *owl:sameAs* (figura 23.b). As figuras 23.a e 23.b apresentam os códigos das consultas SPARQL executadas, recuperando todos os artigos da conferência que estejam relacionados, seja por palavra-chave ou área de conhecimento, ao recurso *<http://pt.dbpedia.org/resource/Aprendizado>* da DBpedia. A figura 23.c apresenta o resultado obtido com ambas as consultas no *GraphDB*, demonstrando assim a aplicação bem-sucedida do conceito de *Linked Data* à base de dados da conferência. Este resultado promove a reutilização dos dados da DBpedia, disponibilizando dados inteligíveis por máquinas e os incorporando ao grafo mundial da Web de Dados Ligados. O alcance desta ligação permite, por exemplo, a associação dos artigos da conferência a outros de outras instituições de pesquisa, pelo mundo, que também possuam recursos de suas bases de dados ligados à DBpedia, fomentando, portanto, o crescimento e distribuição do conhecimento científico e o apoio à descoberta deste conhecimento.

(a)

```
SELECT ?publicacao ?titulo_publicacao
{
    ?k owl:sameAs <http://pt.dbpedia.org/resource/Aprendizado> .
    SERVICE <http://localhost:2020/sparql> {
      ?publicacao schema:name ?titulo_publicacao .
      {
        ?publicacao schema:about ?k
      }
      UNION {
        ?publicacao schema:genre ?k
      }
   }
}
```

(b)

```
SELECT DISTINCT ?publicacao ?titulo_publicacao
{
      ?publicacao a foaf:Document .
      ?publicacao schema:name ?titulo_publicacao .
      {
          ?publicacao schema:about <http://pt.dbpedia.org/resource/Aprendizado>
      }
      UNION {
          ?publicacao schema:genre <http://pt.dbpedia.org/resource/Aprendizado>
      }
}
```

(c)

	publicacao	titulo_publicacao
1	http://localhost:2020/resource/trabalhoPublicado/2506	AVALIAÇÃO DO APRENDIZADO DE CEFALOMETRIA COM ALUNOS DO CURSO DE GRADUAÇÃO EM ODONTOLOGIA
2	http://localhost:2020/resource/trabalhoPublicado/4955	RELATO DE EXPERIÊNCIA NA OFICINA ABATJOUR
3	http://localhost:2020/resource/trabalhoPublicado/4545	INTEGRAÇÃO UNIVERSIDADE-ESCOLA: UM ENCONTRO COM A CIÊNCIA

Figura 23 - (a) Consulta SPARQL federada (banco de dados RDF de mashup e endpoint *D2RQ*) e sem inferência. (b) Mesma consulta SPARQL, porém não federada (banco de dados RDF centralizado com todas as triplas) e com inferência. (c) Resultados de ambas consultas. Fonte: Elaborado pelso autores (2018).

Ambas as abordagens de publicação dos dados obtiveram os resultados esperados, sendo que a segunda abordagem, centralizada, sem a necessidade da realização de consulta federada, apresentou melhor desempenho e menor chance de erros de comunicação e de protocolos. É importante mencionar que o banco de dados RDF *Stardog*, diante da consulta demonstrada na figura 23.b, não foi capaz de inferir automaticamente a ligação direta, pelas propriedades *schema:about* e *schema:genre*, entre as publicações e os recursos da *DBpedia*, funcionando corretamente apenas com a consulta exposta na figura 23.a, com a propriedade *owl:sameAs* explícita. Não foram encontrados problemas nos demais cenários avaliados.

Foram realizados ainda testes de processamento buscando encontrar *links* semânticos entre a base de dados da conferência e a *DBpedia*, por meio de outro *workflow* do *Silk* que buscasse similaridades entre os conteúdos dos títulos e resumos dos recursos de ambas as bases de dados, estabelecendo ligações do tipo ***rdfs:seeAlso*** **(“veja também”, em Português).** Todavia, o processamento se mostrou extremamente custoso e demorado, e não foram encontrados *links* nesta abordagem. Pretende-se realizar uma análise mais aprofundada, visando uma reestruturação desse *workflow*, em trabalhos futuros.

6.2 Integração de dados de sistemas de informação distintos

Considerando-se os cenários apresentados nas figuras 13, 14, 19 e 20, a estrutura implementada foi capaz de atender as demandas solicitadas, permitindo a extração de dados a partir do grafo RDF, e a sua utilização nos sistemas de informação citados. Foram realizados testes utilizando as abordagens de consulta federada e centralizada. Para dar suporte à integração dos dados, uma nova ontologia, nomeada como *Public University Generic Ontology (PUGO)*, foi modelada e implementada, na linguagem OWL, e publicada na Web sobre o *namespace* *http://purl.org/pugo/1.0#*. Esta pode ser reutilizada por outros projetos que possuam necessidades semelhantes para o mesmo domínio.

No cenário 1 (figuras 13 e 14), observou-se, das consultas SPARQL utilizadas na elaboração da solução, apresentadas nas figuras 21 e 22, que a execução no *triple store* (*GraphDB*) levou, em média, cerca de 79% menos tempo da sua execução no *SPARQL enpoint* do *Ontop*. As consultas realizadas como teste, bem como os seus respectivos tempos de execução, tanto no *GraphDB* quanto no *Ontop*, podem ser obtidas no Apêndice I.

Com base nos resultados apresentados no Apêndice I, observa-se que o desempenho de uma abordagem centralizada no *triple store* é consideravelmente superior ao acesso realizado através do mapeamento *RDB2RDF on-the-fly*. Apesar disso, os tempos apresentados por ambas as abordagens, na maioria dos casos, podem ser considerados aceitáveis para as

finalidades a que se propõem neste estudo caso, uma vez que se consiga manter o desempenho de forma escalável, com alto fluxo de usuários simultâneos, o que não foi explorado nesta pesquisa. No caso da importação dos dados dos professores no sistema PLANID, a rotina levou, para analisar os 296 registros retornados e importar os faltantes, 20 segundos utilizando-se o *GraphDB*, e 36 segundos, utilizando os *endpoints* com mapeamento *on-the-fly*.

Acerca do cenário 2 (figuras 19 e 20), foi alcançado o objetivo de gerar relatórios envolvendo dados dos sistemas Siga e SIRHU com o Sistema SIAC. Contudo, diferentemente do cenário **1**, a distância entre o desempenho da abordagem centralizada e da abordagem federada é muito mais marcante. Em um dos casos, por exemplo, desistiu-se de esperar a conclusão da consulta federada depois de uma hora e meia de execução, enquanto a mesma consulta foi realizada em apenas 10 segundos na abordagem centralizada. As consultas testadas como exemplos de relatórios, bem como seus respectivos tempos de execução nas duas abordagens, podem ser obtidas no Apêndice II.

Sobre a abordagem apresentada na figura 20, foram importados no *triple store*, incluindo os dados exportados do Siga e SIRHU, Sistema SIAC e a ontologia desenvolvida, de acordo com as estatísticas do *GraphDB*, 15.388.551 triplas, gerando, com base nas regras importadas, mais 15.999.199 triplas inferidas, totalizando 31.387.750 triplas disponíveis no grafo.

A exportação dos dados mapeados do Siga e SIRHU pelo *Ontop*, utilizados em todos os cenários com a abordagem centralizada, levou cerca de 3 horas; e sua importação no *GraphDB* levou cerca de 26 minutos. Já os dados do sistema SIAC, com o *D2RQ*, levaram menos de 1 minuto para serem exportados, e 35s para importação no *GraphDB*.

Os resultados dos testes no cenário **2** sugerem que, embora seja viável a busca de recursos específicos (como, por exemplo, uma única pessoa e seus dados relacionados), a busca de relatórios completos mostra-se impraticável na abordagem federada, nos moldes adotados neste trabalho. Os testes realizados com a abordagem centralizada demonstraram bom desempenho nos resultados.

Este estudo de caso foi conduzido com o apoio da equipe de desenvolvimento da universidade (analistas de sistemas e programadores) e a proposta foi bem aceita pela equipe, que conseguiu implementá-la, de forma direta, sem se deparar com dificuldades significativas.

6.3 Dificuldades encontradas e lições aprendidas

Este tópico destaca alguns dos desafios enfrentados, na execução desta pesquisa, e lições aprendidas, algumas das quais já foram acrescidas ao método proposto.

Uma das dificuldades encontradas diz respeito à configuração geográfica e de rede das origens de dados utilizadas. Os bancos de dados do Siga e SIRHU e SIAC e PLANID, bem como seus respectivos servidores de aplicação (e também o servidor utilizado para executar o *Ontop* e o *D2RQ*), encontram-se em cidades diferentes, separados por cerca de 155km de distância, em linha reta. Além disso, não havia uma forma de acesso direto aos bancos de dados do Siga e SIRHU, de forma que este precisou ser adaptado através de túnel SSH. Estes detalhes técnicos podem causar impacto na velocidade da transmissão dos dados entre os bancos de dados e as ferramentas de mapeamento, mesmo a conexão de Internet disponível sendo de alta velocidade.

Embora não seja o foco desta pesquisa, vale comentar que outra dificuldade relevante foi a questão da segurança dos *SPARQL endpoints*. Embora a tecnologia forneça uma estrutura madura para a publicação de dados ligados na Web, quando se trata da disponibilização de dados restritos, é difícil encontrar os elementos necessários para implementar um controle de acesso mais elaborado. Embora seja possível encontrar um controle de usuário e senha em *triple stores* como o *GraphDB*, as implementações de cliente para as linguagens de programação nem sempre possuem suporte a consultas SPARQL utilizando autenticação. Quando possuem, a utilização deste recurso nem sempre é exposta na documentação de forma clara e, por vezes, requer desenvolvimento adicional de código. Além disso, durante a pesquisa, não foi encontrada solução padronizada para a realização de consultas federadas em *endpoints* autenticados. As ferramentas *RDB2RDF* testadas também não possuem o recurso de autenticação nos seus *SPARQL endpoints*.

Uma vez que procurou-se desenvolver este trabalho de integração de dados de sistemas fazendo o possível para que a ligação entre os recursos ocorra da maneira mais natural possível, sem a necessidade do advento de um componente de descoberta, outro desafio encontrado foi a queda abrupta de desempenho na criação, nos mapeamentos, de URIs baseadas em atributos não chave ou, ainda, em atributos processados, como é o caso do hash SHA1 do CPF/passaporte das pessoas. O objetivo era fazer com que todos os sistemas gerassem os mesmos URIs para recursos comuns, de forma que a ligação entre os dados se estabelecesse automaticamente. Sendo assim, pela questão de desempenho nestes casos, decidiu-se utilizar atributos chave (como um identificador único indexado, no caso) na

composição do URI padrão. O URI com atributo processado foi adicionado, então, através da propriedade ***owl:sameAs***. Esta solução é facilmente processada pelo *triple store* na abordagem centralizada, porém cria dificuldades na elaboração de consultas federadas. Em geral, isso não faz diferença quando o objetivo é buscar dados a partir de um recurso específico, pois as consultas podem ser realizadas utilizando-se uma propriedade identificadora padrão, como, por exemplo, a ***personIdentifier*** ou ***schema:indentifier***, dentro de todas as cláusulas SERVICE do SPARQL (como foi feito em um dos exemplos do Apêndice II); daí a importância de incluir este tipo de propriedade no modelo. Uma vez que todas as origens de dados utilizem o mesmo modelo ontológico, a ligação entre os recursos ocorre naturalmente para este tipo de consulta. Contudo, quando o objetivo é unir um conjunto maior de registros, esta mesma solução não é capaz de alcançar o objetivo, sendo necessário a construção de uma lógica adicional na consulta.

7 TRABALHOS RELACIONADOS

Halaç *et al.* (2013) utilizam a filosofia *Linked Data* para integrar dados distribuídos entre diferentes sistemas de uma universidade, propondo uma arquitetura que viabiliza a ligação entre as informações dos bancos de dados desses sistemas, além da criação de uma aplicação para auxiliar nessa integração de dados. O projeto utiliza a plataforma *D2RQ* (Bizer *et al.*, 2012) para mapear bancos de dados relacionais de alguns dos sistemas e a ferramenta *Silk Framework* (Isele *et al.*, [s.d.]) para a descoberta de *links* entre os recursos de diferentes fontes de dados. Com a evolução da pesquisa, optou-se pela alimentação periódica, em lote, de uma base RDF centralizada, a partir dos bancos de dados dos sistemas em questão, por meio de uma aplicação que realiza a extração de dados, conversão e carga de forma automática, com o auxílio de ferramentas como *D2RQ*, *Silk* e *Triplister* (Rogers, 2011), mantendo os dados ligados atualizados de forma dinâmica.

Segarra *et al.* (2016) propõem uma arquitetura de integração entre repositórios digitais distribuídos, em um modelo *Linked Data* baseado em ontologias consagradas, como *DCTerms*, *Bibo*, *Schema* e *FOAF*, utilizando um enfoque virtual através de consultas federadas SPARQL. O modelo proposto foi aplicado às bases de publicações acadêmicas de universidades equatorianas, obtendo escalabilidade e aplicabilidade na integração, assim como sua fácil expansão a outros sistemas de informação.

Em Santarém Segundo *et al.* (2017), é feita uma análise de como os padrões e tecnologias da Web Semântica contribuem no processo de construção de redes semânticas e na organização de informações, visando prover informações de mais relevância aos usuários. Para tal, é realizado um estudo na colaboração digital de publicações acadêmicas, com foco na plataforma *VIVO* (*Duraspace, Phoenix, Arizona*, Estados Unidos da América), que utiliza tecnologias da Web Semântica, como RDF e OWL, para descrever e relacionar recursos. Conclui-se, então, que a Web Semântica, com suas tecnologias, aumenta o campo de visão e de relacionamentos de conceitos, oferecendo aos usuários resultados mais ricos, do ponto de vista de relações semânticas, sendo a plataforma estudada (*VIVO*) um bom exemplo do uso dessas tecnologias.

Cverdelj-Fogaraši *et al.* (2017) propõem uma abordagem baseada em ontologias OWL para Sistemas de Gerenciamento de Documentos, com o objetivo de fomentar a sua flexibilização para diferentes domínios e a integração de dados entre diferentes sistemas de informação empresariais. Para tal, foi desenvolvida uma metaontologia genérica, baseada no padrão de metadados ebRIM (Oasis, 2012). A proposta foi validada através da sua

implementação em um estudo de caso de gerenciamento de documentos judiciais. Para isso, foram desenvolvidas ainda mais duas camadas ontológicas, sob a ontologia genérica proposta, descrevendo-se o domínio de gerenciamento de arquivos e o domínio judicial. Além da descrição de dados, a proposta do trabalho também incluiu o uso de ontologias para mapear correlações entre metadados de diferentes domínios, utilizando-se regras expressas em *Semantic Web Rule Language* (SWRL) (Horrocks *et al.*, 2004), promovendo assim um processo de tradução automatizada. Do ponto de vista do usuário final, a solução proposta permite a navegação entre diferentes visões dos dados automaticamente integrados, sob diferentes perspectivas de metadados.

Narvaez e Piedra (2018) propõem um modelo semântico para a integração *Linked Data* de dados entre diversos sistemas de uma universidade equatoriana. A abordagem proposta baseia-se em um processo de extração, transformação e carregamento (ETL) de dados de bases relacionais para um *triple store*, em formato RDF, utilizando um modelo ontológico composto de ontologias conhecidas, além de uma nova ontologia criada para o domínio em questão. Para a conversão dos dados de bases relacionais para RDF, foi desenvolvida uma aplicação específica para o caso. Para acesso aos dados, além do *SPARQL endpoint* fornecido pelo *triple store* utilizado (*Apache Marmotta* (The Apache Software Foundation, 2012)), foi construída uma API *RESTful*, capaz de fornecer os dados em diversos formatos conhecidos. Do ponto de vista de desempenho, foi realizada uma comparação entre os tempos de resposta do acesso aos dados utilizando-se a API semântica desenvolvida, JDBC e uma outra API já utilizada anteriormente pela instituição, tendo a primeira apresentado melhor desempenho do que a última, porém perdendo para o JDBC por uma pequena diferença.

Auer *et al.*(2012) apresentam uma arquitetura para publicação de dados ligados, baseando a sua proposta na implementação de uma solução sobre o *framework LOD2*, apresentado no artigo, em um processo iterativo e incremental. O *LOD2* seria uma suíte de ferramentas conhecidas da Web Semântica, organizadas de forma a interagirem entre si e interligadas por meio de uma API REST, viabilizando a publicação e consumo de dados ligados.

Zengenene *et al.* (2014) e Hidalgo-Delgado *et al.* (2018) propõem, em seus respectivos trabalhos, métodos para publicação de *Linked Data*. Zengenene *et al.* (2014) apresenta um método de 15 passos para publicação *Linked Data*, com foco em dados de bibliotecas. Assim como no anterior, a proposta de Hidalgo-Delgado *et al.* (2018) tem foco em dados

bibliográficos. Seu método, iterativo e incremental, conta com cinco passos, e propõe uma abordagem que envolve extração, processamento e carregamento dos dados em lote, para acesso *offline*.

Alguns dos trabalhos citados (Halaç *et al.* (2013), Segarra *et al.* (2016) e Narvaez e Piedra (2018)) conduzem o foco de suas obras no estudo de caso e na arquitetura construída para resolver o problema apresentado, mas não em estabelecer um método a ser utilizando de forma sistemática, englobando diferentes cenários. O estudo de Santarém Segundo *et al.* (2017) tem foco no estudo da plataforma *VIVO*, que permite catalogar, de forma semântica e interligada, dados acadêmicos, mas não tratando especificamente da integração de dados de sistemas já existentes. Cverdelj-Fogaraši *et al.* (2017) aponta o foco da sua pesquisa para a criação de uma metaontologia, genérica, capaz de ser especializada para atender a domínios específicos, podendo ser utilizada para a integração das informações de sistemas de informação empresariais. Contudo, também não tem foco em estabelecer um método e uma arquitetura para esta integração.

O foco de Auer *et al.*(2012) baseia-se na apresentação da suíte de ferramentas do *framework*, e em como cada ferramenta, e a relação entre elas, pode dar suporte a publicação, descoberta e consumo de dados ligados, contudo sem investir em um método sistematizado. Além disso, os sítios Web indicados para acesso a download e documentação do *framework* encontraram-se indisponíveis durante o decorrer dessa pesquisa. Embora Zengenene *et al.* (2014) apresente um método detalhado para publicação de dados ligados, voltado principalmente para bibliotecas, o mesmo mantém o enfoque na parte teórica, sem envolver um estudo de caso. Hidalgo-Delgado *et al.* (2018) também apresenta um método para publicação de dados ligados, a partir da integração de origens de dados não relacionais, de terceiros, e não bancos de dados internos a uma instituição. Além disso, o objetivo do método de ambos os trabalhos é publicar dados ligados abertos na Web, e não necessariamente a integração de dados de sistemas corporativos.

Em comparação com esses outros trabalhos, esta proposta diferencia-se por fornecer um método genérico e adaptável, apresentado em detalhes, para publicar e integrar, de forma sistemática, dados relacionais de sistemas de informações corporativos em grafo RDF, além de uma seleção de ontologias *Linked Data*, devidamente documentada, para o domínio de conhecimento de universidades. Destaca-se ainda por oferecer diferentes abordagens automatizadas de como publicar dados relacionais como *Linked Data*, variando entre consultas federadas e centralizadas, e acesso *on the fly* e com dados carregados em lote,

usando ferramentas bem conhecidas pela comunidade da Web Semântica, aplicadas a dois estudos de caso reais, com testes realizados com diferentes cenários. As ontologias e diretrizes/ferramentas usadas nas duas abordagens podem, perfeitamente, ser empregadas em outros projetos similares.

8 CONCLUSÃO

Neste trabalho foi proposto um método e uma arquitetura para publicação de dados relacionais nos padrões e tecnologias da Web Semântica, e para integração de dados de sistemas de informação, segundo os princípios *Linked Data*, validados por meio de dois estudos de caso reais. Foram adotados, como estudo de caso, o mapeamento de bases de dados relacionais de trabalhos acadêmicos e de sistemas de informação corporativos de uma universidade federal brasileira para o modelo de dados RDF, bem como a sua utilização, de forma útil, por outras aplicações, envolvendo a geração de relatórios combinando dados ligados.

Diante do observado nos testes realizados, os resultados foram satisfatórios, sugerindo que a Web Semântica fornece uma plataforma adequada à integração de dados dentro de uma instituição, sendo uma alternativa significativa aos *data warehouses* convencionais. Sendo assim, foram alcançados os objetivos propostos para esta pesquisa.

Para concluir esta dissertação, primeiro contribuições serão brevemente reiteradas. Em seguida, alguns trabalhos futuros serão enumerados.

8.1 Contribuições

As principais contribuição deste trabalho são o método e arquitetura propostos para publicação de dados relacionais nos padrões e tecnologias da Web Semântica, e o método para integração de dados de sistemas de informação, segundo os princípios *Linked Data.* Embora o método de publicação de dados esteja contido no método de integração de dados, os dois métodos poder ser utilizados de forma independente, de acordo com a necessidade, conforme demonstrado através dos experimentos realizados nos estudos de caso. Os passos para implementação de ambos os métodos e arquitetura propostos foram apresentados de forma detalhada, bem como foram sugeridas estratégias para lidar com os desafios encontrados. Sugestões de ferramentas existentes foram apresentadas para a viabilização da estrutura proposta.

Outra contribuição relevante foram os modelos ontológicos propostos para cada um dos estudos de caso, relacionados a diversas ontologias conhecidas. Estes podem ser sistematicamente empregados, respectivamente, à publicação e interligação de produções acadêmicas na Web de Dados e à interligação de bases de sistemas corporativos dentro de universidades, por meio de um grafo semântico.

Espera-se que o método e arquitetura propostos sirvam como diretrizes para integração sistematizada de dados dentro da instituição em estudo, bem como em outras organizações, dado que se trata de um problema bastante recorrente nos dias atuais.

Ao longo deste trabalho foi publicado um artigo, a saber:

- Gonçalves, A. O.; Jacyntho, M. D. A. Um método para publicação semântica Linked Data de bases de dados convencionais e um estudo de caso real de artigos acadêmicos. Transinformação, v. 32, n. e180051, 2020

8.2 Trabalhos Futuros

Procurando tornar a proposta ainda mais robusta, deve ser realizado um aprofundamento da pesquisa no concernente às questões de segurança, mais especificamente no controle de acesso às informações integradas, recurso ainda limitado dentro das tecnologias oferecidas pela Web Semântica. Além disso, seria de grande valia uma pesquisa mais detalhada no que diz respeito a técnicas que permitam otimizar o desempenho das consultas mapeadas, de forma a obter resultados mais eficientes em consultas federadas.

Neste trabalho, os processos de exportação de importação de dados foram realizados de forma manual. Uma proposta interessante seria o desenvolvimento de uma ferramenta, com interface amigável, que seja capaz de gerenciar o processo de integração entre os dados de diversos sistemas de uma instituição, através da utilização das tecnologias aqui apresentadas, incluindo recursos tais como: combinação de diferentes bancos de dados relacionais, mapeamento Relacional-RDF, *SPARQL endpoints*, componentes de descoberta, realização de dereferenciamento automático de URIs, cargas periódicas automáticas e emissão facilitada de relatórios.

Os ensaios realizados durante esta pesquisa tiveram como foco aplicar a arquitetura proposta sobre bases de dados relacionais como origem de dados, devido à imensa relevância e abrangência do modelo de dados relacional. Sendo assim, seria interessante a realização de experimentos com outros formatos de dados, como, por exemplo, JSON, XML e CSV. A ideia da arquitetura continua a mesma, bastando apenas utilizar ferramentas apropriadas para fazer o mapeamento do modelo de dados específico para o modelo de dados RDF.

Outro objetivo desejável seria a elaboração de um plano de treinamento voltado para profissionais de Tecnologia da Informação dentro de uma instituição, nas tecnologias e padrões da Web Semântica, envolvendo a elaboração de um manual didático sobre a

integração semântica de dados de sistemas, e o desenvolvimento de uma pesquisa sobre a eficácia desse plano.

Por fim, sobre a etapa de *mashup* automatizado apresentada neste trabalho, é importante aprofundar ainda mais a análise quali-quantitativa dos resultados encontrados, definindo mecanismos para medir a relevância e fidedignidade dos *links* RDF obtidos automaticamente com a ferramenta *Silk*.

9 REFERÊNCIAS

Abele, A. *et al.* Linking Open Data cloud diagram 2017. Disponível em: <http://lod-cloud.net/>. Acesso em: 13 dez. 2017.

Antoniou, G. *et al.* A Semantic Web Primer. 3rd. ed. [s.l.] The MIT Press, 2012.

Auer, S. *et al.* Managing the life-cycle of linked data with the LOD2 stack. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), v. 7650 LNCS, n. PART 2, p. 1–16, 2012.

Azevedo, R. S. N.; Jacyntho, M. D. A. Um Modelo Baseado Em Ontologias Linked Data Para Catalogação De Projetos de Software. Conferências Ibero-Americanas WWW/Internet e Computação Aplicada. Anais...Porto: 2014

Beckett, D. *et al.* RDF 1.1 Turtle. Disponível em: <https://www.w3.org/TR/turtle/>. Acesso em: 27 jul. 2018.

Berners-Lee, T. *et al.* The Semantic Web. Scientific American, v. 284, n. 5, p. 34–43, 2001.

Berners-Lee, T. *et al.* Uniform Resource Identifier (URI): Generic Syntax. Internet proposed standard RFC 3986. Disponível em: <https://tools.ietf.org/html/rfc3986>. Acesso em: 19 maio. 2020.

Berners-Lee, T. Linked Data. Disponível em: <https://www.w3.org/DesignIssues/LinkedData.html>. Acesso em: 29 nov. 2017.

Bilenko, M. *et al.* Adaptive name matching in information integration. IEEE Intelligent Systems, v. 18, n. 5, p. 16–23, set. 2003.

Bizer, C. *et al.* The RDF book mashup: From Web APIs to a Web of data. CEUR Workshop Proceedings. Anais...2007

Bizer, C. *et al.* Linked Data - The Story So Far. International Journal on Semantic Web and Information Systems, v. 5, n. 3, p. 1–22, jul. 2009.

Bizer, C. *et al.* D2RQ Platform - Accessing Relational Databases as Virtual RDF Graphs. Disponível em: <http://d2rq.org/>. Acesso em: 30 mar. 2018.

Brickley, D. *et al.* RDF Schema 1.1 - W3C Recommendation. Disponível em: <https://www.w3.org/TR/rdf-schema/>. Acesso em: 12 nov. 2018.

Brickley, D.; Miller, L. FOAF Vocabulary Specification 0.99. Disponível em: <http://xmlns.com/foaf/spec/>. Acesso em: 30 mar. 2018.

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

Calvanese, D. *et al.* Ontop Guide - Introduction. Disponível em: <https://ontop-vkg.org/guide/>. Acesso em: 20 mar. 2020.

Camilo, C. O.; Silva, J. C. Da. Um estudo sobre a interação entre Mineração de Dados e Ontologias. Disponível em: <http://www.inf.ufg.br/sites/default/files/uploads/relatorios-tecnicos/RT-INF_002-09.pdf>. Acesso em: 1 nov. 2018.

Chhaya, P. *et al.* Using D2RQ and Ontop to publish relational database as Linked Data. International Conference on Ubiquitous and Future Networks, ICUFN, v. 2016- Augus, p. 694–698, 2016.

Connolly, D.; Miller, L. RDF Calendar - an application of the Resource Description Framework to iCalendar Data. Disponível em: <https://www.w3.org/TR/rdfcal/>. Acesso em: 20 maio. 2019.

Cverdelj-Fogaraši, I. *et al.* Semantic integration of enterprise information systems using meta-metadata ontology. Information Systems and e-Business Management, v. 15, n. 2, p. 257–304, 2017.

Cyganiak, R. *et al.* The D2RQ Mapping Language. Disponível em: <http://d2rq.org/d2rq-language>. Acesso em: 29 nov. 2017.

Das, S. *et al.* R2RML: RDB to RDF Mapping Language. Disponível em: <https://www.w3.org/TR/r2rml/>. Acesso em: 24 set. 2019.

DBpedia. DBpedia - Towards a Public Data Infrastructure for a Large, Multilingual, Semantic Knowledge Graph. Disponível em: <http://dbpedia.org>. Acesso em: 16 jun. 2018.

DBpedia Mappings. Disponível em: <http://mappings.dbpedia.org/>. Acesso em: 30 mar. 2018.

DCMI Usage Board. Dublin Core Metadata Initiative. Disponível em: <http://dublincore.org/documents/dcmi-terms/>. Acesso em: 30 mar. 2018.

Eastlake, D.; Jones, P. RFC 3174 - US Secure Hash Algorithm 1 (SHA1). Disponível em: <https://tools.ietf.org/html/rfc3174>. Acesso em: 14 mar. 2020.

Gandon, F.; Schreiber, G. RDF 1.1 XML Syntax. Disponível em: <https://www.w3.org/TR/rdf-syntax-grammar/>. Acesso em: 27 mar. 2018.

Gonçalves, A. O.; Jacyntho, M. D. A. Um método para publicação semântica Linked Data de bases de dados convencionais e um estudo de caso real de artigos acadêmicos. Transinformação, v. 32, n. e180051, 2020.

Gruber, T. R. Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies, v. 43, p. 907–928, 1995.

Halaç, T. G. *et al.* Publishing and linking university data considering the dynamism of datasources. Proceedings of the 9th International Conference on Semantic Systems - I-SEMANTICS ’13. Anais...Graz: 2013

Heath, T.; Bizer, C. Linked Data: Evolving the Web into a Global Data Space. 1. ed. [s.l.] Morgan & Claypool, 2011. v. 1

Hidalgo-Delgado, Y. *et al.* Methodological guidelines for publishing library data as linked data. Proceedings - 2017 International Conference on Information Systems and Computer Science, INCISCOS 2017, v. 2017-Novem, n. April 2018, p. 241–246, 2018.

Horrocks, I. *et al.* SWRL: A Semantic Web Rule Language Combining OWL and RuleML. Disponível em: <https://www.w3.org/Submission/SWRL/>. Acesso em: 23 nov. 2017.

Horrocks, I. *et al.* Web Ontology Language (OWL). Disponível em: <https://www.w3.org/OWL/>. Acesso em: 10 nov. 2018.

Isele, R. *et al.* Silk - The Linked Data Integration Framework. Disponível em: <http://silkframework.org/>. Acesso em: 30 mar. 2018.

Jacyntho, M. D. A. Um Modelo de Bloqueio Multigranular para RDF. Tese de doutorado. Departamento de Informática, PUC-Rio, Rio de Janeiro, 2012, p. 277.

Jacyntho, M. D. A.; Azevedo, R. S. N. De. Uma Arquitetura Linked Data para Criação de Repositórios Semânticos Auto-Atualizáveis de Projetos de Software. 1º Encontro Interestadual de Engenharia de Produção. Anais...São João da Barra: 2015

Jacyntho, M. D.; Schwabe, D. A multigranularity locking model for RDF. Journal of Web Semantics, v. 39, p. 25–46, 2016.

Jaenicke, N. *et al.* Triplify - Documentation. Disponível em:

<https://web.archive.org/web/20150208025440/http://triplify.org:80/Documentation>. Acesso em: 11 nov. 2018.

Kauppinen, T. *et al.* Teaching Core Vocabulary. Disponível em: <http://linkedscience.org/teach/ns/>. Acesso em: 19 maio. 2019.

Konstantinou, N. *et al.* Exposing scholarly information as Linked Open Data: RDFizing DSpace contents. The Electronic Library, v. 32, n. 6, p. 834–851, 3 nov. 2014.

Konstantinou, N.; Spanos, D.-E. Materializing the Web of Linked Data. Cham: Springer International Publishing, 2015.

Miles, A.; Bechhofer, S. SKOS Simple Knowledge Organization System Namespace Document - HTML Variant. Disponível em: <https://www.w3.org/2009/08/skos-reference/skos.html>. Acesso em: 30 nov. 2017.

Narvaez, E.; Piedra, N. Un enfoque de Linked Data para garantizar la interoperabilidad semántica e integridad de datos académicos universitarios. CEUR Workshop Proceedings, v. 2096, p. 50–62, 2018.

Noy, N. F.; McGuinness, D. L. Ontology Development 101: A Guide to Creating Your First Ontology. Stantford Medical Informatics Technical Report, v. 32, p. 1–25, 2001.

Nuzzolese, A. G. *et al.* Semantic Web Conference Ontology - A Refactoring Solution. ESWC 2016. Anais...Heraklion: Springer, Cham, 2016

Oasis. OASIS ebXML RegRep Version 4.0. Disponível em: <http://docs.oasis-open.org/regrep/regrep-core/v4.0/regrep-core-rim-v4.0.html>. Acesso em: 15 nov. 2019.

Ontotext. GraphDB. Disponível em: <http://graphdb.ontotext.com/>. Acesso em: 30 mar. 2018.

RDF Working Group. Resource Description Framework (RDF). Disponível em: <https://www.w3.org/RDF/>. Acesso em: 12 nov. 2018.

Rivest, R. The MD5 Message-Digest Algorithm. Disponível em: <https://www.rfc-editor.org/info/rfc1321>. Acesso em: 1 ago. 2020.

Rogers, D. Tripliser. Disponível em: <http://daverog.github.io/tripliser/>. Acesso em: 30 mar. 2018.

Santarém Segundo, J. E. *et al.* Conceitos e tecnologias da Web semântica no contexto da colaboração acadêmico-científica: um estudo da plataforma Vivo. TransInformação, v. 29,

n. 3, p. 297–309, 2017.

Schema.org Community Group. schema.org. Disponível em: <http://schema.org/>. Acesso em: 30 mar. 2018.

Segarra, J. *et al.* Integration of digital repositories through federated queries using semantic technologies. 2016 XLII Latin American Computing Conference (CLEI). Anais...Valparaíso: IEEE, out. 2016Disponível em: <http://ieeexplore.ieee.org/document/7833406/>. Acesso em: 10 jun. 2018

Souza, A. N. De. A Web Semântica na Definição de um Modelo de Dados Estruturado para Apoiar Pesquisas no Lago Batata (Oriximiná/PA). Dissertação (Mestrado em Engenharia de Produção e Sistemas Computacionais). Universidade Federal Fluminense, Rio das Ostras - RJ, 2016.Sporny, M. *et al.* JSON-LD 1.0 - A JSON-based Serialization for Linked Data. Disponível em: <https://www.w3.org/TR/json-ld/>. Acesso em: 27 mar. 2018.

Stardog Union. Stardog. Disponível em: <https://www.stardog.com/>. Acesso em: 30 mar. 2018.

Styles, R. *et al.* Academic Institution Internal Structure Ontology (AIISO). Disponível em: <http://vocab.org/aiiso/>. Acesso em: 30 mar. 2018.

The Apache Software Foundation. Apache Marmotta. Disponível em: <https://marmotta.apache.org/>. Acesso em: 15 nov. 2019.

The W3C SPARQL Working Group. SPARQL 1.1 Overview. Disponível em: <https://www.w3.org/TR/sparql11-overview/>. Acesso em: 13 dez. 2017.

W3C. About W3C. Disponível em: <https://www.w3.org/Consortium/>. Acesso em: 30 mar. 2018.

W3C OWL Working Group. OWL 2 Web Ontology Language Document Overview (Second Edition). Disponível em: <https://www.w3.org/TR/owl2-overview/>. Acesso em: 23 nov. 2017.

Wood, D. *et al.* Linked Data: Structured data on the Web. 1. ed. [s.l.] Manning Publications, 2013.

Yamamoto, Y.; Katayama, T. D2RQ Mapper. CEUR Workshop Proceedings, v. 1546, p. 152–154, 2015a.

Yamamoto, Y.; Katayama, T. D2RQ Mapper. Disponível em: <http://d2rq.dbcls.jp/>.

Acesso em: 19 mar. 2020b.

Yandex. About Yandex. Disponível em: <https://yandex.com/company/>. Acesso em: 30 mar. 2018.

Yu, L. A Developer's Guide to the Semantic Web. [s.l.] Springer Science & Business Media, 2011.

Zengenene, D. *et al.* Towards a Methodology for Publishing Library Linked Data. Communications in Computer and Information Science, v. 385 CCIS, p. 81–92, 2014.

APÊNDICE I

Este apêndice apresenta uma tabela de consultas SPARQL realizadas nos testes que culminaram nos resultados apresentados no Capítulo 6, com seus respectivos tempos de execução na abordagem centralizada, com *GraphDB*, e Federada, com Ontop.

Descrição	**Consulta**	**Tempo no *GraphDB***	**Tempo no Ontop**

Dados de um aluno específico	SELECT DISTINCT
	?nome ?bolsa ?codigoCurso
	?nomeCurso ?periodoAluno
	?dataNascimento
	WHERE
	{
	?p schema:identifier '**12345678900'** .
	?p schema:name ?nome .
	?p schema:hasOccupation ?a .
	?a a pugo:StudentRole .
	?a pugo:enrolledInCourse ?c .
	?c schema:name ?nomeCurso .
	?c schema:courseCode ?codigoCurso .
	?a pugo:currentTerm ?periodoAluno .
	?p schema:birthDate ?dataNascimento .
	OPTIONAL {
	?p pugo:assumedName ?nomeSocial .
	?a pugo:hasScholarship ?bolsa .
	} .
	}
		0,2s	1,276s

Dados de todos os professores de duas unidades da universidade	SELECT DISTINCT ?cpf ?nome ?siape ?cargo ?nomeDepartamento ?codigoDepartamento ?regime ?nivel ?classe { ?p schema:identifier ?cpf . ?p schema:name ?nome . ?p schema:hasOccupation ?r . ?r a pugo:ProfessorRole . ?r pugo:workerIdentifier ?siape . ?r schema:roleName ?cargo .	6,5s	24,474s

	?dep a pugo:Department . ?dep schema:member ?r . ?dep schema:name ?nomeDepartamento . ?dep schema:identifier ?codigoDepartamento . VALUES ?codigoDepartamento { "35240003" "38010200" } . ?r pugo:contract ?regime . ?r pugo:level ?nivel . ?r pugo:class ?classe . }		

Dados de um professor, com todas as suas turmas	SELECT DISTINCT
	?n ?turma ?nomeTurma ?nomeDepartamento
	?codDepartamento
	{
	?p schema:name ?n .
	?p schema:hasOccupation ?r .
	?r a pugo:ProfessorRole .
	?turma teach:teacher ?r .
	?r pugo:workerIdentifier '9999999' .
	?turma teach:academicTerm ?periodo .
	?periodo pugo:year 2019 .
	?periodo pugo:annualIdentifier "2"^^xsd:int .
	?turma schema:name ?nomeTurma .
	?departamento schema:member ?r .
	?departamento schema:name ?nomeDepartamento .
	?departamento schema:identifier ?codDepartamento .
	}
		0,3s	1,822s

Dados de um servidor específico, por SIAPE	SELECT DISTINCT ?nomePessoa ?nomeDepartamento
	WHERE
	{
	?p schema:name ?nomePessoa .
	?p schema:hasOccupation ?r .
	?dep schema:member ?r .
	?dep schema:name ?nomeDepartamento .
	?r pugo:workerIdentifier '1234566'
	}
		0,3s	2,287s

Todas os vínculos da pessoa com a universidade, por	SELECT DISTINCT ?o WHERE { ?p a schema:Person .	0,2s	20,488s

CPF	?p schema:identifier '12345678900' . ?p schema:hasOccupation ?r . ?r a ?o . VALUES ?o { pugo:StudentRole pugo:ProfessorRole pugo:ManagementWorkerRole} }		

Tabela 4 - Consultas SPARQL testadas no cenário 1 do capítulo 5.2.5, com tempo de execução. Fonte: Elaborado pelos autores (2020).

APÊNDICE II

Este apêndice expõe uma tabela com exemplos de relatórios obtidos por meio de consultas SPARQL, com seus respectivos tempos de execução, em um dos cenários cujos resultados são apresentados no Capítulo 6.

Descrição	**Consulta**	**Tempo na abordagem centralizada (*GraphDB*)**	**Tempo na abordagem federada (Ontop + *D2RQ* + *GraphDB*)**

Uma pessoa específica e seus trabalhos publicados no evento gerenciado pelo Sistema SIAC	# Abordagem centralizada
	SELECT DISTINCT
	?nomePessoa ?tituloTrabalho ?ocupacao
	WHERE
	{
	?p a schema:Person .
	?p pugo:personIdentifier "12345678900" .
	?p schema:name ?nomePessoa .
	?p schema:hasOccupation ?o .
	?o a ?ocupacao .
	VALUES ?ocupacao { pugo:StudentRole pugo:ProfessorRole pugo:ManagementWorkerRole} .
	?trabalho a schema:Article .
	?trabalho schema:author ?p .
	?trabalho schema:name ?tituloTrabalho .
	}
		0,2s	2m:53s

	# Abordagem federada
	SELECT DISTINCT
	?nomePessoa ?tituloTrabalho ?ocupacao
	WHERE
	{
	SERVICE <http://192.168.28.183:8080/sparql>{
	?p a schema:Person .
	?p pugo:personIdentifier "12345678900" .
	?p schema:name ?nomePessoa .		

	?p schema:hasOccupation ?o .
	?o a ?ocupacao .
	VALUES ?ocupacao {
	pugo:StudentRole
	pugo:ProfessorRole
	pugo:ManagementWorkerRole
	} .
	}
	SERVICE <http://192.168.28.183:2020/sparql> {
	?p2 pugo:personIdentifier "12345678900" .
	?trabalho a schema:Article .
	?trabalho schema:author ?p2 .
	?trabalho schema:name ?tituloTrabalho .
	}
	}		

Todos os alunos bolsistas com algum trabalho publicado no evento gerenciado pelo Sistema SIAC	# Abordagem centralizada
	SELECT DISTINCT
	?nomePessoa ?nomeBolsa
	?trabalho ?tituloTrabalho
	WHERE
	{
	?p a schema:Person .
	?p schema:name ?nomePessoa .
	?p schema:hasOccupation ?o .
	?o a pugo:StudentRole .
	?o pugo:hasScholarship ?b .
	?b schema:description ?nomeBolsa .
	?trabalho a schema:Article .
	?trabalho schema:author ?p .
	?trabalho schema:name ?tituloTrabalho .
	}
		10s	Mais de 1h:30m

	# Abordagem federada
	SELECT DISTINCT
	?nomePessoa ?nomeBolsa
	?trabalho ?tituloTrabalho
	WHERE
	{
	SERVICE <http://192.168.28.183:8080/sparql>{
	?p a schema:Person .
	?p schema:name ?nomePessoa .		

	?p schema:hasOccupation ?o .
	?o a pugo:StudentRole .
	?o pugo:hasScholarship ?b .
	?b schema:description ?nomeBolsa .
	OPTIONAL{
	?p owl:sameAs ?x .
	}
	}
	SERVICE <http://192.168.28.183:2020/sparql> {
	?trabalho a schema:Article .
	?trabalho schema:author ?p2 .
	?trabalho schema:name ?tituloTrabalho .
	{
	{?p2 owl:sameAs ?x .}
	UNION
	{?p2 owl:sameAs ?p .}
	UNION
	{?trabalho schema:author ?p .}
	}
	}
	}		

Tabela 5 - Consultas SPARQL testadas no cenário 2 do capítulo 5.2.5, com tempo de execução.
Fonte: Elaborado pelos autores (2020).