PGCOMP - Programa de Pós-Graduação em Ciência da Computação
Universidade Federal da Bahia (UFBA)
Av. Adhemar de Barros, s/n - Ondina
Salvador, BA, Brasil, 40170-110

http://pgcomp.dcc.ufba.br
pgcomp@ufba.br

As informações que trafegam em redes sociais nem sempre podem ter a origem verificada, os caminhos descritos e as modificações registradas. Questões como essas geram incerteza e desconfiança aos processos e interações em Redes Sociais Online (do inglês, Online Social Networks (OSNs)). Com o surgimento das Redes Sociais Online Descentralizadas (do inglês, Decentralized Online Social Networks (DOSNs)) e o aumento da quantidade de usuários ativos nessas redes, torna-se importante desenvolver soluções efetivas referentes à proveniência dos dados descentralizados. A proveniência é um fator de importância considerável, pois, por meio de seus resultados, é possível avaliar a autenticidade, confiabilidade e relevância das informações. Rastreamento e captura de proveniência são tarefas fundamentais para DOSNs, pois produzem respostas sobre as etapas percorridas pelas informações. O grande volume de dados, a velocidade de geração e compartilhamento de informações e a estratégia descentralizada de armazenamento torna a proveniência em DOSNs uma tarefa não trivial. Dessa forma, este trabalho propõe o modelo ontológico de proveniência DOSN-PROV, que é um modelo específico para o domínio de DOSNs, baseado na especificação PROV-O do World Wide Web Consortium (W3C). Além disso, este trabalho também propõe serviços, baseados no DOSN-PROV, para fornecer suporte à captura e ao rastreamento de proveniência em DOSNs. O DOSN-PROV foi avaliado em duas etapas para demonstrar a conformidade com o domínio proposto. Por fim, os serviços passaram por uma avaliação de desempenho e seus resultados indicaram tempos de resposta aceitáveis para as tarefas de captura e rastreamento.

Palavras-chave: Redes Sociais Descentralizadas, Proveniência, Ontologia, Serviços de Proveniência, Captura de Proveniência, Rastreamento de Proveniência.

MSC | 127 | 2021

DOSN-PROV: Modelo e Serviços para Proveniência de Dados em Redes Sociais Descentralizadas

Cíntia da Costa Souza

UFBA

DOSN-PROV: Modelo e Serviços para Proveniência de Dados em Redes Sociais Descentralizadas

Cíntia da Costa Souza

Dissertação de Mestrado

Universidade Federal da Bahia

Programa de Pós-Graduação em Ciência da Computação

Dezembro | 2021

Universidade Federal da Bahia
Instituto de Computação

Programa de Pós-Graduação em Ciência da Computação

DOSN-PROV: MODELO E SERVIÇOS PARA PROVENIÊNCIA DE DADOS EM REDES SOCIAIS DESCENTRALIZADAS

Cíntia da Costa Souza

DISSERTAÇÃO DE MESTRADO

Salvador
10 de dezembro de 2021

CÍNTIA DA COSTA SOUZA

DOSN-PROV: MODELO E SERVIÇOS PARA PROVENIÊNCIA DE DADOS EM REDES SOCIAIS DESCENTRALIZADAS

Esta Dissertação de Mestrado foi apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Cássio Vinícius Serafim Prazeres

Salvador
10 de dezembro de 2021

Ficha catalográfica elaborada pela Biblioteca Universitária de Ciências e Tecnologias Prof. Omar Catunda, SIBI – UFBA.

S729 Souza, Cíntia da Costa

DOSN-PROV: modelo e serviços para proveniência de dados em redes sociais descentralizadas / Cíntia da Costa Souza. – Salvador, 2021.

67 f.

Orientador: Prof. Dr. Cássio Vinícius Serafim Prazeres

Dissertação (Mestrado) – Universidade Federal da Bahia. Instituto de Computação, 2021.

1. Redes Sociais Online. 2. Ontologia. 3.. Engenharia de Software. I. Prazeres, Cássio Vinícius Serafim. II. Universidade Federal da Bahia. III. Título.

CDU 004.43

"*DOSN-PROV: Modelo e Serviços para Proveniência de Dados em Redes Sociais Descentralizadas*"

Cíntia da Costa Souza

Dissertação apresentada ao Colegiado do Programa de Pós-Graduação em Ciência da Computação na Universidade Federal da Bahia, como requisito parcial para obtenção do Título de Mestre em Ciência da Computação.

Banca Examinadora

Prof. Dr.Cassio Vinicius Serafim Prazeres
(Orientador PGCOMP)

Prof. Dr. Mario Vieira dos Santos e Souza
(UFBA)

Profª. Drª. Rita Cristina Galarraga Berardi
(UTFPR)

Dedico este trabalho à minha mãe.

AGRADECIMENTOS

Agradeço primeiramente a Deus por me permitir mais essa grande conquista, mesmo em meio a tantas lutas e dificuldades.

Agradeço a minha família, em especial à minha linda mãe por todo amor e apoio incondicional durante todo esse processo, essa conquista não teria o mesmo significado sem a sua presença e nada disso seria possível sem a sua ajuda; à minha avó Joaquina, por toda preocupação enquanto estive longe e por todo carinho quando voltava à Brasília.

Ao meu namorado e acima de tudo melhor amigo Júnior Leles, por dividir comigo todos os momentos felizes e também os momentos de agonia desde o início de nossa vida acadêmica, sem você ao meu lado seria muito mais difícil.

Agradeço a Izabel Ornellas Simabuko e Marco Antônio Simabuko, pelo grande incentivo, ensinamentos e toda ajuda no começo da minha vida acadêmica. Vocês são minha família do coração.

Agradeço ao meu sogro José Ronaldo, pela confiança e por todo o apoio em todos os momentos, especialmente durante o período do mestrado.

Ao meu orientador, Prof. Dr. Cássio Prazeres, agradeço por todos os ensinamentos, por toda paciência nas reuniões de orientação e por toda a humanidade com que trata seus orientandos.

Agradeço aos membros da banca examinadora, Profa. Dra. Rita Berardi e Prof. Dr. Marlo Souza, pela disponibilidade em participar e contribuir com esta dissertação.

As queridas amigas de Brasília, Mayara Aguiar, Iara Farias e Aline França, que mesmo com toda distância se fizeram presente em minha vida, entenderam com todo carinho do mundo minha ausência constante.

As amigas do Goiás, Janaína Barbosa e Gisele Carneiro , por todo incentivo e compartilhamento de experiências, sempre em meio a boas risadas.

Aos amigos "da prédia", Renato Araújo e Marcos Vinícius Ferreira, pelos momentos divertidos e pelos inúmeros compartilhamentos de conhecimento. Agradeço as meninas da turma 2016.1, Dhennya Campos e Beatriz Brito, por toda ajuda em minha primeira chegada a Salvador, pela disposição em sempre ajudar e pelos momentos de descontração no começo do mestrado.

Por fim, agradeço a CAPES pelo apoio financeiro concedido a esta pesquisa.

"Quanto mais aumenta nosso conhecimento,
mais evidente fica nossa ignorância"
—JOHN F. KENNEDY (1917-1963)

RESUMO

As informações que trafegam em redes sociais nem sempre podem ter a origem verificada, os caminhos descritos e as modificações registradas. Questões como essas geram incerteza e desconfiança aos processos e interações em Redes Sociais Online (do inglês, *Online Social Networks* (OSNs)). Com o surgimento das Redes Sociais Online Descentralizadas (do inglês, *Decentralized Online Social Networks* (DOSNs)) e o aumento da quantidade de usuários ativos nessas redes, torna-se importante desenvolver soluções efetivas referentes à proveniência dos dados descentralizados. A proveniência é um fator de importância considerável, pois, por meio de seus resultados, é possível avaliar a autenticidade, confiabilidade e relevância das informações. Rastreamento e captura de proveniência são tarefas fundamentais para DOSNs, pois produzem respostas sobre as etapas percorridas pelas informações. O grande volume de dados, a velocidade de geração e compartilhamento de informações e a estratégia descentralizada de armazenamento torna a proveniência em DOSNs uma tarefa não trivial. Dessa forma, este trabalho propõe o modelo ontológico de proveniência DOSN-PROV, que é um modelo específico para o domínio de DOSNs, baseado na especificação PROV-O do *World Wide Web Consortium* (W3C). Além disso, este trabalho também propõe serviços, baseados no DOSN-PROV, para fornecer suporte à captura e ao rastreamento de proveniência em DOSNs. O DOSN-PROV foi avaliado em duas etapas para demonstrar a conformidade com o domínio proposto. Por fim, os serviços passaram por uma avaliação de desempenho e seus resultados indicaram tempos de resposta aceitáveis para as tarefas de captura e rastreamento.

Palavras-chave: Redes Sociais Descentralizadas, Proveniência, Ontologia, Serviços de Proveniência, Captura de Proveniência, Rastreamento de Proveniência.

ABSTRACT

The origin, traveled paths and modification occurred along the way of information on social networks cannot always be verified. Issues such as these generate uncertainty and distrust in processes and interactions in OSNs. With the emergence of DOSNs and the increase of the number of active users in these networks, it becomes essential to develop effective solutions regarding the provenance of decentralized data. The provenance is a factor of considerable importance because it is possible to evaluate the information's authenticity, reliability, and relevance through its results. Tracing and capture of provenance are critical tasks for DOSNs since they produce answers about the steps go through by the information. The large volume of data, the speed of generation and sharing of information, and the decentralized storage strategy make the provenance of DOSNs a non-trivial task. Thus, this work proposes the PROV-DOSN provenance ontological model, a specific model for the DOSNs domain based on the PROV-O specification from W3C. In addition, this work also proposes services based on the DOSN-PROV to support the capture and tracking of provenance in DOSNs. We evaluated DOSN-PROV in two steps to demonstrate compliance with the proposed domain. Finally, the services underwent a performance evaluation, and their results indicated acceptable response times for the capture and tracking tasks.

Keywords: Decentralized Social Networks, Provenance, Ontology, Provenance Services, Capture of Provenance, Provenance Tracking.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

ACL	*Access Control List* 13
CSS	*Cascading Style Sheets* 34
DHTs	*Distribuited Hash Tables* 7
DOSNs	*Decentralized Online Social Networks* 2
FOAF	*Friend of a Friend* 14
HTML	*HyperText Markup Language* 34
HTTP	*Hypertext Transfer Protocol* 8
IRI	*Internationalized Resource Identifier* 14
LOD	*Linking Open Data* 11
OSNs	*Online Social Networks* 1
OWL2	*OWL2 Web Ontology Language* 19
P2P	*Peer-to-Peer* 7
POD	*Personal Online Datastore* 9
PROV-O	*Prov Ontology* 18
RDF	*Resource Description Framework* 10
SIOC	*Semantically Interlinked Online Communities* 22
SKOS	*Simple Knowledge Organization System* 23
SMTP	*Simple Mail Transfer Protocol* 8
SOLID	*Social Linked Data* 13
SPARQL	*Simple Protocol and RDF Query Language* 11
SWEO	*Semantic Web Education and Outreach Interest Group* 11
TCP	*Transmission Control Protocol* 8
UDP	*User Datagram Protocol* 8
URI	*Uniform Resource Identifier* 11
W3C	*World Wide Web Consortium* 10

Capítulo 1

INTRODUÇÃO

O termo proveniência se refere a origem ou fonte de algo (MOREAU, 2010). Trazendo essa definição para o contexto dos dados, proveniência diz respeito a registros históricos de derivação dos dados, permitindo reprodutibilidade, interpretação e diagnóstico de problemas (LIM et al., 2010). Em uma outra definição, Herschel, Diestelkämper e Lahmar (2017) afirmam que dados de proveniência são metadados que descrevem os dados e todo seu processo de produção, podendo ser coletado por técnicas de proveniência de dados.

Segundo Moreau (2010), dados de proveniência têm causado significativo interesse na comunidade de pesquisa em Ciência da Computação, por permitirem analisar, inferir qualidade e verificar processos de decisão de confiança das informações. Além disso, de acordo com Magliacane (2012), proveniência de dados permite analisar, verificar e inferir qualidade de processos de confiança dos dados; é fundamental para uma série de aplicações; e têm sido pesquisada sob diferentes perspectivas.

A proveniência de dados é um fator importante a ser considerado ao avaliar a relevância e a confiabilidade das informações, podendo responder a vários tipos de perguntas sobre os dados: como foram gerados, quais caminhos percorreram e se sofreram alterações até chegar ao estado atual. Essas respostas levaram Taxidou et al. (2018) a afirmar que prover esses tipos de dados representam desafios de pesquisa na área de Redes Sociais Online (do inglês, *Online Social Networks* (OSNs).

Conforme afirma Recuero (2002), as relações sociais entre agentes humanos deram origem às redes sociais e quando essas interações são intermediadas por plataformas digitais elas são chamadas de OSNs. Assim, Wang et al. (2015) definem as OSNs como estruturas sociais constituídas por meio de um conjunto de atores sociais e um conjunto de relacionamentos entre esses atores. Logo, Stantic (2017) considera que esses relacionamentos e atores passam por uma gradativa evolução e que a geração e compartilhamento das informações de usuários crescem em proporções únicas.

De acordo com Barbier et al. (2013), conhecer sobre a proveniência das informações postadas e compartilhadas nas OSNs é fundamental para determinar a autenticidade,

confiabilidade e resolução de conflitos de dados. Relevante parte das interações sociais acontecem através de OSNs proprietárias como Facebook[1], Instagram[2] e Twitter[3]. (TAXIDOU et al., 2018).

As OSNs proprietárias possuem arquitetura centralizada, em que o controle das informações dos usuários é mantido pelo provedor de serviços. Logo, segundo Salve, Guidi e Mori (2018), a arquitetura centralizada gera problemas referentes à privacidade dos usuários, alto grau de confiança no provedor de serviços e a formação de silos de informações.

Como alternativa para solução desses problemas de OSNs proprietárias, foram criadas iniciativas a partir de pesquisas acadêmicas como PeerSoN (BUCHEGGER et al., 2009) e SafeBook (STRUFE, 2009), bem como projetos *open-source* como Diáspora[4], OpenSocial e NoseRub. Essas iniciativas são redes sociais com arquitetura descentralizada (SHARMA; DATTA, 2012), conhecidas como Redes Sociais Online Descentralizadas (do inglês, *Decentralized Online Social Networks* (DOSNs)) (BAHRI; CARMINATI; FERRARI, 2018).

DOSNs são OSNs implementadas de forma descentralizada, ou seja, com arquitetura par-a-par (do inglês, *Peer-to-Peer* (P2P)). De acordo com Salve, Guidi e Mori (2018), P2P é considerada por cientistas a arquitetura adequada para implementar as DOSNs. Desse modo, Yeung et al. (2009) afirmam que por ser implementada por uma arquitetura descentralizada, seus usuários não se vinculam a nenhum serviço de rede social específico, mas determinam servidores confiáveis para armazenar suas informações, gerando aos usuários autonomia sobre seus próprios dados.

Como exemplos de DOSNs que possuem sua base em redes de servidores gerenciados pelos próprios usuários, temos Diáspora e Friendica[5], que possuem mais de 669.000 usuários. Assim como estas, existem várias outras DOSNs, que se diferenciam por estratégia de privacidade, técnica de replicação ou infraestrutura de armazenamento (SALVE; GUIDI; MORI, 2018). Portanto, com o surgimento de alternativas descentralizadas e a quantidade de usuários ativos nestas redes sociais, a proveniência de DOSNs requer soluções eficientes referentes aos dados que trafegam em suas redes.

1.1 MOTIVAÇÃO

As OSNs e DOSNs possibilitam mudanças na forma como as informações são criadas, compartilhadas e utilizadas. Com isso, a proveniência dessas informações desempenhará um papel determinante nesta mudança, permitindo garantir confiabilidade, integridade, disponibilidade e autenticidade aos dados (CHENEY et al., 2009). Entretanto, Trivedi, Bindu e Thilagam (2018) afirmam que entender os processos dos dados de proveniência não é uma tarefa trivial. Além disso, identificar origem, fontes de propagação e rastreio de informação em mídias sociais também se configuram desafios a serem solucionados.

De acordo com Taxidou et al. (2018), conhecer sobre os dados de proveniência é

[1] <https://www.facebook.com/>
[2] <https://www.instagram.com>
[3] <https://twitter.com>
[4] <https://joindiaspora.com>
[5] <https://friendi.ca/>

importante para entender a relevância e confiabilidade das informações. Porém, apesar de ser um tema de pesquisa ativo nas áreas de banco de dados e fluxos de trabalho, Taxidou et al. (2015) afirmam que a proveniência em redes sociais recebeu pouca atenção, se comparada às demais áreas de pesquisa de disseminação de informação. Além disso, ainda não existem mecanismos suficientes para geração e rastreamento de dados de proveniência no contexto das redes sociais (TAXIDOU et al., 2015).

Dessa forma, investigar problemas referentes à captura e rastreamento de proveniência é de fato importante, visando assegurar confiança, qualidade, integridade e autenticidade aos dados que trafegam em DOSNs.

1.2 DESCRIÇÃO DO PROBLEMA

O movimento de descentralização da *Web* tem gerado expectativa quanto à possibilidade dos usuários deterem o poder sobre seus dados (XIA et al., 2019). A partir desse movimento surgiram as DOSNs, que, de acordo com Yeung et al. (2009), propõem permitir um ambiente favorável aos usuários de redes sociais, no qual os utilizadores possuem o controle sobre seus dados, sobre a privacidade e disseminação de suas informações. Logo, nessas redes, os usuários detêm o poder total sobre seus dados, incluindo autonomia para criar, atualizar e deletar qualquer informação que tenha gerado. Seguramente, esta é uma das principais vantagens propostas pelas DOSNs: devolver aos usuários o controle sobre seus dados, incluindo a decisão de local de armazenamento e permissão de acesso aos dados.

No entanto, no contexto de redes sociais, o poder sobre os dados pode gerar uma lacuna de inconsistência e desconfiança causada pela possibilidade do usuário excluir informações e eliminar rastros de possíveis práticas nocivas, violações ou infrações cometidas dentro dessas redes (e.g., discurso de ódio, pedofilia, racismo e disseminação de notícia falsa). Esse poder irrestrito dos usuários sobre os seus dados provoca desconfiança às DOSNs quanto à integridade, autenticidade e irretratabilidade das informações que trafegam nestas redes.

De acordo com Gundecha et al. (2013), as informações de DOSNs nem sempre podem ter suas origens verificadas, sua motivação determinada ou finalidades associadas definidas. Dessa forma, essas questões trazem incerteza e desconfiança aos processos de DOSNs, sendo necessário desenvolver soluções de proveniência para garantir a confiança de processos internos referentes aos dados que trafegam nestas redes.

A velocidade de geração e compartilhamento das informações, aliado ao grande volume de dados nas redes sociais, segundo Duong et al. (2017), tornam a atividade de coleta, rastreamento, registro de origem e de movimentação de dados entre as diferentes bases de armazenamento uma tarefa não trivial. Nesse sentido, a utilização de um modelo ontológico específico e a criação de serviços de coleta e rastreamento de proveniência de dados podem garantir a confiança dos processos internos de DOSNs.

1.3 HIPÓTESE

Um modelo ontológico específico para proveniência baseado em PROV-O do *World Wide Web Consortium* (W3C) para o domínio de DOSNs pode garantir uma estrutura capaz de representar de forma eficaz as necessidades referentes aos processos de DOSNs. Adicionalmente, utilizar serviços de coleta e rastreamento de proveniência modelados através de uma ontologia específica para DOSNs pode garantir confiança aos processos internos de DOSNs.

1.4 OBJETIVOS

Esta dissertação tem como objetivo principal desenvolver um modelo ontológico, específico para proveniência, e serviços para suporte à proveniência, referentes à coleta e ao rastreamento de dados em redes sociais descentralizadas. Esse objetivo principal pode ser dividido em quatro objetivos específicos listados a seguir:

- Desenvolver um modelo ontológico específico para proveniência em redes sociais descentralizadas;
- Implementar os serviços referentes ao suporte a proveniência: (i) serviço de captura e (ii) serviço de rastreamento;
- Avaliar o modelo de proveniência em duas etapas: (i) verificação a partir de casos de aplicação e recuperação da dados e (ii) validação utilizando uma ferramenta de identificação de falhas e inconsistências;
- Avaliar os serviços de proveniência quanto ao tempo de resposta dos serviços, considerando variações no número de triplas e quantidade de usuários.

1.5 CONTRIBUIÇÕES

O desenvolvimento de um modelo específico para proveniência de dados em DOSNs, tendo como base a ontologia PROV-O, permite a representação e o uso de proveniência de forma eficiente em DOSNs. Adicionalmente, a criação de serviços para utilização do modelo em DOSNs, a fim de dar suporte à captura e ao rastreamento dos dados, tornam essas redes mais confiáveis.

Com a utilização da arquitetura de proveniência proposta nesta dissertação de mestrado, espera-se auxiliar DOSNs a manter sua estrutura confiável e os dados que transitam nestas redes passíveis de verificação e prova de autoria. As principais contribuições resultantes desta dissertação são:

- Uma arquitetura de proveniência para DOSNs,
- Modelo de proveniência para DOSNs,
- Definição de requisitos de proveniência para DOSNs,
- Serviços de proveniência referentes à captura e ao rastreamento de dados.

1.6 ESTRUTURA DO TRABALHO

O restante do texto está estruturado da seguinte forma: no Capítulo 2 é apresentada a fundamentação teórica contendo os principais conceitos e tecnologias que fundamentam esta pesquisa. O Capítulo 3 descreve o modelo DOSN-PROV, os serviços para suporte a proveniência em DOSNs e os trabalhos relacionados a esta pesquisa encontrados na literatura. A avaliação do modelo DOSN-PROV e dos serviços de proveniência, bem como os resultados da avaliação, são abordados no Capítulo 4. Por fim, o Capítulo 5 apresenta a conclusão, incluindo contribuições, limitações e trabalhos futuros.

Capítulo 2

FUNDAMENTAÇÃO TEÓRICA

Este capítulo descreve os principais conceitos que fundamentam esta pesquisa e as tecnologias envolvidas na elaboração deste trabalho. Dentre estes conceitos apresentados estão DOSNs, proveniência de dados, *Linked Data*, ecossistema SOLID, ontologia e metodologias para modelagem de ontologias. Por fim, a ontologia PROV-O e os vocabulários FOAF, SIOC e vCard.

2.1 REDES SOCIAIS ONLINE DESCENTRALIZADAS

Segundo Salve et al. (2016), as DOSNs surgiram da necessidade de controle dos usuários de OSNs sobre seus dados. Essas redes são implementadas em plataforma distribuída de gerenciamento de informações possuindo uma estrutura baseada em arquitetura P2P, como tabelas *hash* distribuídas (do inglês, *Distribuited Hash Tables* (DHTs)) ou redes de servidores confiáveis (DATTA et al., 2010).

A formação dessas redes de servidores confiáveis ou redes *Peer-to-Peer* (P2P) é composta por vários pares distribuídos, autônomos, heterogêneos e dinâmicos, onde cada participante da rede pode compartilhar recursos como: processamento, espaço de armazenamento, software e conteúdos de arquivos (SALVE et al., 2016).

Nas DOSNs os dados e informações utilizados pelas plataformas de redes sociais não são armazenados nos servidores das plataformas, o que permite aos usuários terem a liberdade de escolha do local de armazenamento de seus dados. Assim, os usuários podem optar por um ou mais servidores diferentes, o que de acordo com Koll, Li e Fu (2017) não acontecia nas OSNs, onde os dados dos usuários eram armazenados em *datacenters* interconectados e controlados por entidades centralizadas.

Para Yeung et al. (2009), a estrutura descentralizada das DOSNs permite aos usuários controlarem seus próprios dados, dando a eles autonomia sobre suas informações e os tornando ativos e cientes principalmente de fatores como:

- **Privacidade**: os próprios usuários das DOSNs decidem quem tem acesso às informações e quais restrições são impostas sobre os dados.

- **Propriedade**: os usuários têm propriedade total sobre os dados, visto que as informações são armazenadas em servidores confiáveis ou servidores em seu próprio computador.
- **Divulgação**: os usuários decidem como são divulgadas as informações, de acordo com suas relações de amizade e preferências.

As DOSNs não são todas iguais, elas diferem quanto a recursos fornecidos, ambiente de desenvolvimento e forma de armazenamento de dados. No entanto, foram notadas semelhanças na arquitetura dessas DOSNs e para dar uma visão mais clara sobre as abordagens dessas redes, Datta et al. (2010) propuseram a arquitetura de referência mostrada na Figura 2.1, composta por 6 camadas. Essa arquitetura de referência permite uma abstração de variadas abordagens relacionadas a DOSNs na literatura de pesquisa.

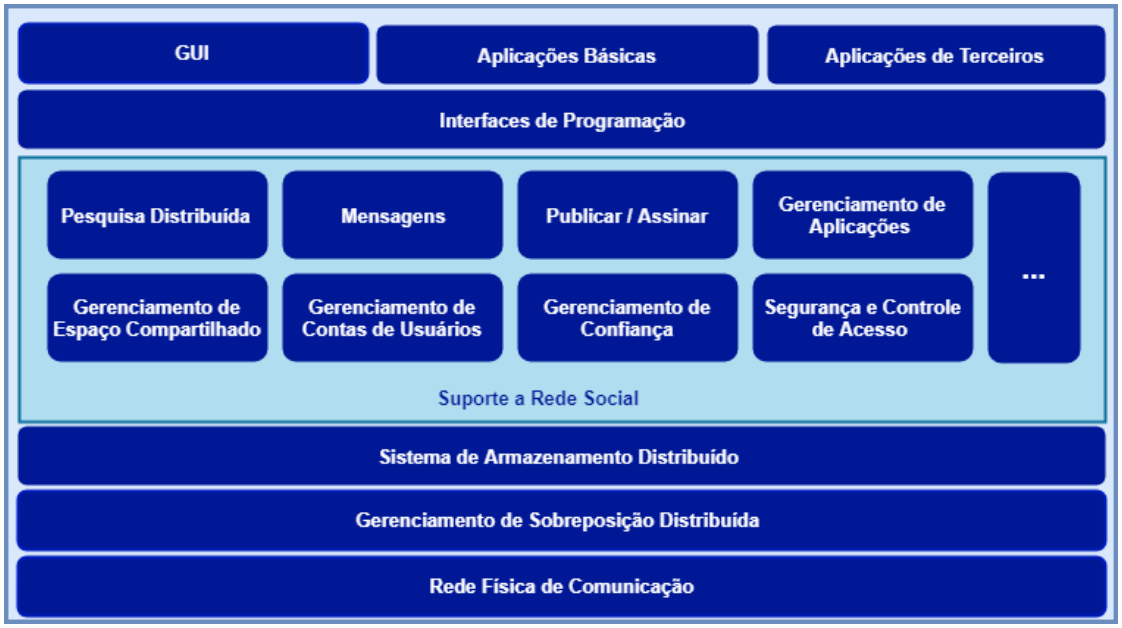


Figura 2.1 Arquitetura geral de DOSNs. Adaptado de Datta et al. (2010).

A camada inferior da arquitetura é composta pela Rede Física de Comunicação, a Internet. Os serviços de comunicação entre os pares acontecem através dos protocolos disponíveis na infraestrutura da Internet (e.g., Protocolo de Controle de Transmissão (do inglês, *Transmission Control Protocol* (TCP)), Protocolo de Datagrama de Usuário (do inglês, *User Datagram Protocol* (UDP)), Protocolo de Transferência de Hipertexto (do inglês, *Hypertext Transfer Protocol* (HTTP)), Protocolo de Transferência de Correio Simples (do inglês, *Simple Mail Transfer Protocol* (SMTP)).

A camada de Gerenciamento de Sobreposição Distribuída oferece funcionalidades para gerenciar recursos na infraestrutura de suporte, como rede de servidores distribuídos ou sobreposição. Segundo Datta et al. (2010) essa camada apresenta maneiras de buscar recursos, rotear mensagens e recuperar informações de forma confiável entre os nós.

Na camada de Sistema de Armazenamento Distribuído são implementadas funcionalidades necessárias para consultar, inserir e atualizar objetos persistentes nos sistemas. Furht (2010) garante que essa é uma camada de grande importância para toda infraestrutura, pois garante que os dados estejam sempre disponíveis de alguma forma. Por fim, na camada de Suporte a Rede Social, são implementadas todas as funcionalidades e recursos básicos fornecidos por OSNs.

Pesquisadores apresentaram e implementaram várias soluções de DOSNs, Diáspora[1] por exemplo, foi uma das primeiras iniciativas e obteve uma popularidade considerável (GUIDI et al., 2018). Em sua arquitetura, a Diáspora permite aos usuários atuar como servidor local, onde o usuário mantém controle total sobre seus dados, ou possibilita que cada usuário configure seu Armazenamento de Dados Pessoais Online (do inglês, *Personal Online Datastore* (POD)) fornecido pela Diáspora (KOLL; LI; FU, 2017). Na Seção 2.4 é possível ter mais detalhes sobre PODs e sua estrutura.

2.2 PROVENIÊNCIA DE DADOS

Moreau (2010) afirma que a proveniência de dados se refere a origem ou linhagem dos dados. Já Ram e Liu (2006) complementam afirmando que a proveniência de dados diz respeito a fatos relativos à criação, processamento e arquivamento dos dados e que além de ter se tornado uma preocupação para pesquisadores em Ciência da Computação, através desses dados é possível identificar características de proveniência tais como: quando, por que e como os itens de dados foram produzidos.

Para Magliacane (2012), os dados de proveniência são fundamentais para várias aplicações e têm sido pesquisados sob diferentes perspectivas, levando em consideração questões de agregação, qualidade e confiança. Logo, a proveniência é uma propriedade intrínseca dos dados e como salienta Moreau (2010), se capturados com precisão geram benefícios potenciais em vários contextos como: *e-science*, curadoria de bancos de dados e Web Semântica.

De acordo com Pérez, Rubio e Sáenz-Adán (2018), conhecer dados de proveniência não se refere apenas a aspectos de origem ou fases de processamento, mas permite também compreender informações adicionais referentes ao contexto e dependências dos dados. Assim, dois tipos de informações de proveniência são consideradas ao analisar itens de dados: (i) Proveniência de fonte: informações de dados sobre sua origem ou fonte, e (ii) Proveniência de transformação: alterações envolvidas na criação de um item de dado, ou seja, identificando tanto sua criação quanto a derivação dos dados.

Para Herschel, Diestelkämper e Lahmar (2017), existem duas tarefas muito relevantes para proveniência dos dados: a captura e o rastreamento. A importância dessas tarefas está ligada à avaliação da qualidade, a garantia da reprodutibilidade e ao reforço da confiança dos dados como produtos finais de um encadeamento de procedência.

Com possibilidade de aplicação em diferentes contextos, Meester et al. (2017) destacam que a proveniência de dados possui diversas funcionalidades, como: avaliar a qualidade dos dados, rastrear caminhos percorridos pelas informações, permitir trilha de replicação, estabelecer autoria e fornecer contexto. No contexto de ambientes dis-

[1] <https://joindiaspora.com>

tribuídos, Reilly e Naughton (2006) afirmam que a captura de dados de proveniência é uma tarefa importante, e sua execução deve ocorrer durante o fluxo de trabalho para que informações não se percam quando o trabalho for concluído.

Segundo Tan (2004), a proveniência pode ser capturada a partir de duas abordagens: (i) preguiçosa e (ii) ansiosa. A abordagem preguiçosa coleta os dados de proveniência somente quando solicitada, dessa forma apenas informações necessárias são capturadas e armazenadas. Na abordagem ansiosa a proveniência é coletada em todo tempo, estando sempre disponível para futuras consultas.

A captura dos dados de proveniência e a análise dos rastros produzidos por elas são explorados para produzir respostas sobre todas as etapas percorridas pela informação, incluindo sua geração (WOODMAN; HIDEN; WATSON, 2017). Para isso, existem padrões já estabelecidos que têm como objetivo representar a coleta e o rastreamento de proveniência. Um desses padrões é o Prov do *World Wide Web Consortium* (W3C) que será apresentado na Seção 2.5.

2.3 LINKED DATA

A *Web* 3.0 é comumente conhecida como Web Semântica, foi idealizada por Tim Berners-Lee, criador da *World Wide Web*. A ideia central da Web Semântica é tornar a *Web* compreensível por máquinas e não somente por humanos (AGHAEI; NEMATBAKHSH; FARSANI, 2012). Além disso, para Berners-Lee et al. (2001), a Web Semântica não parte do princípio da criação de uma nova *Web* e sim da extensão da *Web* atual.

Portanto, a Web Semântica é resultado da necessidade de comunicação de máquinas e pessoas com os dados da *Web*. Porém, segundo Berners-Lee (2006), não basta apenas publicar os dados, eles precisam ser explorados de maneira eficiente, ou seja, se conectando quando relacionados. Essa eficiência se torna possível através do uso de Dados Vinculados (do inglês, *Linked Data*) (AGHAEI; NEMATBAKHSH; FARSANI, 2012).

Linked Data significa publicar dados de maneira estruturada na *Web*, permitindo a vinculação de dados de bases diferentes. Para que a vinculação e publicação dos dados seja possível na *Web*, *Linked Data* faz uso de tecnologias fundamentais como a Estrutura de Descrição de Recursos (do inglês, *Resource Description Framework* (RDF)).

O RDF é um modelo associado a um conjunto de práticas, recomendado pelo W3C e utilizado para representar informações sobre recursos na *Web* (MA; CAPRETZ; YAN, 2016). O modelo RDF descreve os dados da *Web* em conjuntos de triplas no formato: sujeito, predicado e objeto, como representado na Figura 2.2.

De acordo com Berners-Lee et al. (2001), essa estrutura de descrição dos dados possibilita a formação de grafos direcionados, onde as arestas representam conexões entre os recursos, e os recursos são caracterizados pelos nós (BIZER et al., 2008). Para a persistência desses grafos são utilizadas *triplestores* (i.e., bancos de dados de grafos desenvolvidos para armazenamento e recuperação de triplas).

O modelo RDF descreve os metadados de recursos da *Web* de forma clara e flexível, podendo representar dados estruturados e não estruturados, se tornando um padrão para definição e interligação de dados na *Web* de Dados (MA; CAPRETZ; YAN, 2016). Desse modo, a forma de representar os recursos na *Web* acontece por meio de Identificador

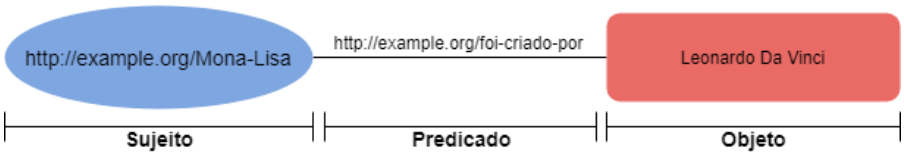


Figura 2.2 Grafo de amostra de tripla. Adaptado de Schreiber e Raimond (2014)

Uniforme de Recursos (do inglês, *Uniform Resource Identifier* (URI)), que são identificadores sintaticamente semelhantes ao Localizador Uniforme de Recursos (inglês, Uniform Resource Locator(URLs)) (ARENAS; GUTIERREZ; PÉREZ, 2010).

Para publicar e conectar esses dados na *Web* foi proposto um conjunto de regras impostas sobre os dados e a maneira como devem ocorrer as publicações. Essas regras foram propostas por Berners-Lee (2006) e são conhecidas como Princípios *Linked Data*:

1. Use URIs como nomes para coisas;

2. Use URIs HTTP para que as pessoas possam procurar esses nomes;

3. Quando alguém procurar um URI, forneça informações úteis, usando os padrões (RDF, SPARQL);

4. Inclua links para outras URIs, para que eles possam descobrir mais coisas.

O exemplo mais efetivo de aplicação e utilização dos princípios *Linked Data* ocorrem no projeto *Linking Open Data* (LOD). O projeto LOD foi fundado em 2007 pelo *Semantic Web Education and Outreach Interest Group* (SWEO)[2], com objetivo de tornar dados de licença aberta em dados padronizados de acordo com os princípios *Linked Data* e publicá-los na *Web* (BIZER et al., 2009).

A Figura 2.3 representa a nuvem LOD, contendo 1.260 conjuntos de dados e 16.187 interligações, possibilitando a observação de publicações e demonstrando o alcance do projeto. Conjuntos de dados como DBpedia[4] e Geonames[5], conforme afirmam Bizer et al. (2008), funcionam como *hubs* de *links* por fornecer URIs e descrições RDF para muitas entidades e conceitos comuns, além de serem constantemente referenciados em outros conjuntos de dados.

Com essa grande quantidade de dados interligados é necessário que existam linguagens e protocolos para recuperação e manipulação de grafos RDF publicados na *Web* ou em diferentes bases de dados. Assim, para efetuar consultas aos dados no formato RDF é necessário utilizar uma linguagem de consulta específica. O Protocolo Simples e Linguagem de Consulta RDF (do inglês, *Simple Protocol and RDF Query Language* (SPARQL))

[2]<https://www.w3.org/blog/SWEO/>
[3]<https://lod-cloud.net/>
[4]<https://wiki.dbpedia.org/>
[5]<http://www.geonames.org/>

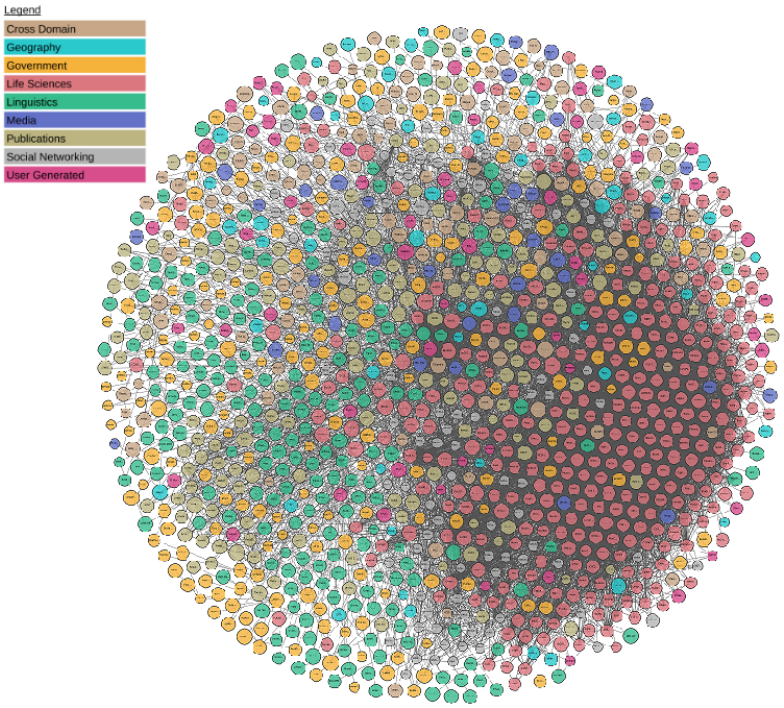


Figura 2.3 Diagrama de nuvem LOD, maio de 2020[3].

é um protocolo de acesso a dados em RDF e uma linguagem de consulta baseada em triplas recomendada pelo W3C, além de ser considerada a linguagem padrão para a consulta desses tipos de dados (ARENAS; GUTIERREZ; PÉREZ, 2010).

A Figura 2.4 apresenta um exemplo de consulta SPARQL, onde na primeira linha é definido o *namespace* utilizado na consulta. A cláusula SELECT define o campo que será selecionado como resultado, ou seja, quais dados se deseja obter, neste caso, o campo idade. Na cláusula WHERE, temos a referência de subgrafo de triplas com a lógica da consulta: sujeito, predicado e objeto. Na linha 6 é atribuída a variável ?idade o valor do atributo foaf:age da pessoa com foaf:name igual a "JoaquinaFarias".

```
1    PREFIX foaf:<http://xmlns.com/foaf/0.1/>
2    SELECT ?idade
3    WHERE{
4        ?pessoa foaf:Person
5        ?pessoa foaf:name "JoaquinaFarias"
6        ?person foaf:age ?idade.
7    }
```

Figura 2.4 Exemplo de Consulta SPARQL

2.4 ECOSSISTEMA SOLID

Nos últimos anos a *Web* vem avançando no sentido contrário ao que foi criada. Os dados pessoais de usuários têm sido centralizados em servidores de dados de grandes empresas. Como exemplos dessa centralização temos OSNs como Facebook[6], Twitter[7], LinkedIn[8] e várias outras redes centralizadas que podem ser consideradas silos de dados pois, segundo Verborgh (2018), cada uma delas controlam os dados dos usuários e determinam seus próprios mecanismos de controle de acesso e autenticação.

Caminhando em sentido oposto a essa centralização de dados da *Web*, surgiu o projeto de Dados Sociais Vinculados (do inglês, *Social Linked Data* (SOLID)[9]), liderado por Tim Berners-Lee. Tal projeto tem a finalidade de propor um conjunto de convenções, protocolos e padrões abertos para criar um ambiente social descentralizado e baseado nos princípios *Linked Data*, em que os dados dos seus usuários são gerenciados independentemente dos aplicativos que criam e consomem esses dados (MANSOUR et al., 2016).

De acordo com Sambra et al. (2016), para a construção desse ambiente social descentralizado o projeto conta com padrões abertos e convenções SOLID fundamentadas nas tecnologias da Web Semântica e tem o objetivo de fornecer mecanismos de gerenciamento e independência de dados de forma simples e eficiente, já que cada usuário armazena seus dados em PODs.

Os PODs são serviços de armazenamento que podem ser acessados através da *Web*, em que cada POD representa um usuário na rede SOLID, onde os usuários podem persistir seus dados. Além disso, os PODs são espaços de armazenamento seguro, onde o proprietário dos dados pode conceder acesso aos dados e pode revogar esse acesso quando necessário (SOLANKI, 2021). Para obter um POD o usuário pode escolher entre as seguintes opções:

1. Hospedar seu próprio servidor;

2. Obter seu POD em fornecedores de armazenamento de nuvem (i.e., Solid Community[10] ou Inrupt[11]).

Dessa forma, cada usuário pode ter um ou mais PODs de diferentes provedores e pode alternar facilmente entre esses provedores (MANSOUR et al., 2016). Os aplicativos SOLID podem trabalhar com qualquer servidor de POD, independente de provedor de serviço e localização. A Figura 2.5 ilustra a arquitetura comum encontrada em servidores POD, com uma visão geral dos recursos implementados em diferentes PODs.

Como mostra a Figura 2.5, PODs podem armazenar recursos RDF e não-RDF, permitem suporte a Linked Data Plataform (LDP) para efetuar operações básicas em containers e recursos e suporta Lista de Controle de Acesso (do inglês, *Access Control List* (ACL))

[6] <https://www.facebook.com/>
[7] <https://twitter.com/>
[8] <https://www.linkedin.com>
[9] <https://solidproject.org/>
[10] <https://solidproject.org/users/get-a-pod>
[11] <https://inrupt.com/>

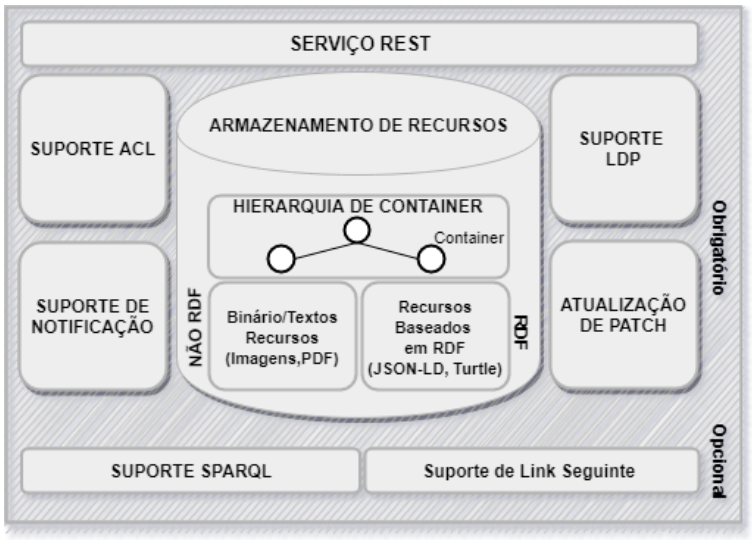


Figura 2.5 Arquitetura de servidores POD. Adaptado de Mansour et al. (2016).

para controlar o acesso ao conteúdo de recursos e containers. Além disso, PODs possuem suporte a notificações e atualizações em caso de modificações de recursos e podem suportar SPARQL para recuperação de dados (MANSOUR et al., 2016).

No ecossistema SOLID as aplicações são executadas como aplicativos da *Web* que rodam do lado do cliente em um navegador. Esses aplicativos usam protocolos de autenticação para descobrir dados de identidade de perfil de usuários, *links* relevantes que apontam para dados de PODs e aplicações de usuários. O SOLID usa *WebID* para fornecer recursos de autenticação descentralizada. Segundo Tramp et al. (2014), *WebID* é uma identificação digital para usuários finais. Um perfil *WebID* contém triplas RDF com Identificador de Recursos Internacionalizado (do inglês, *Internationalized Resource Identifier* (IRI)), identificando o proprietário como sujeito vinculado a um Documento de Perfil.

Os *WebIDs* podem ser utilizados para permitir mais confiança a *Web*, fazendo uso de vocabulários como *Friend of a Friend* (FOAF), para que usuários vinculem seus perfis de forma pública ou protegida, pois são mecanismos de identificação simples, distribuídos e que melhoram a privacidade e o controle dos usuários sobre suas identidades e forma de identificação (SAMBRA; STORY; BERNERS-LEE, 2013).

Segundo Mansour et al. (2016), através do documento de perfil RDF, geralmente armazenado em seu servidor POD, o usuário controla sua identidade e pode carregar uma aplicação SOLID a partir de um provedor de aplicação. A aplicação obtém o POD a partir do perfil de identidade, segue os links do perfil para descobrir seus dados armazenados no POD e no POD de outros usuários, efetuando autenticação quando necessário, como apresentado na Figura 2.6.

No ecossistema Solid, os dados são criados nos containers a partir de operações HTTP POST ao URI do container, são atualizados com HTTP PUT e podem ser deletados com HTTP DELETE. Vale ressaltar que os servidores de PODs são independentes de

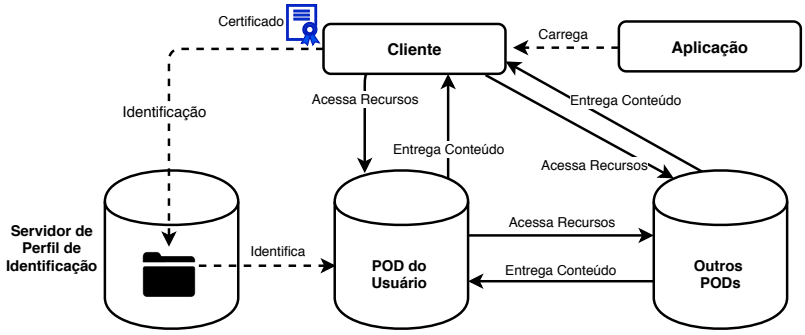


Figura 2.6 Arquitetura Solid. Adaptado de Sambra et al. (2016)

aplicações, ou seja, mudanças em aplicações que utilizam PODs não implicam mudanças nos PODs ou dados armazenados nos PODs dos usuários.

2.5 ONTOLOGIAS

O palavra ontologia pode ter significados diferentes em comunidades distintas. Dentre as diversas definições das diferentes áreas, os significados que mais se contrastam são as diferenças de sentido filosófico e sentido computacional. Os significados de sentido computacional partem de uma definição informal de ontologia como "especificações explícitas de conceituações"(GUARINO; OBERLE; STAAB, 2009). Portanto, ontologia é a especificação explícita de algum domínio de interesse, sendo essa a definição no contexto da Computação, feita por Gruber (1995) a partir da Filosofia.

Ontologias são utilizadas com frequência nas áreas de extração de informação, gerenciamento de conhecimento e Web Semântica (BRANK; GROBELNIK; MLADENIC, 2005). Segundo Berners-Lee et al. (2001), ontologia é o terceiro componente da Web Semântica e pode ser classificada como coleções de informações, fazendo uso de taxonomias e regras de inferência. Para Guarino (1998), as ontologias podem receber uma classificação do ponto vista do conteúdo que apresenta ou de uma tarefa específica:

- Ontologias de alto nível: descrevem conceitos independentes de domínio, conceitos gerais (e.g., espaço, tempo e evento);
- Ontologias de tarefa: descrevem conceitos relacionados a uma tarefa ou atividade (e.g.,vendas);
- Ontologias de domínio: descrevem vocabulários relacionados a um domínio genérico (e.g., medicina);
- Ontologias de aplicação: descrevem conceitos dependentes de um domínio ou tarefa particular, geralmente estendem ou especializam ontologias de tarefa e domínio.

Para Roche (2003), a razão da grande utilização de ontologias se deve em parte ao que ela se propõe a estabelecer: entendimento e comunicação de domínio comum e compar-

tilhado entre pessoas e computadores. Além disso, ontologias buscam descrever conhecimento de domínio consensual. Podem não apresentar a mesma estrutura mas segundo Gruber (1995), possuem características e componentes comuns como: classes, relacionamentos e funções.

Diversos benefícios são identificados na literatura para justificar o uso de ontologias, dentre estes benefícios Noy, McGuinness et al. (2001) destacam: estruturação da informação, compartilhamento, reúso, confiabilidade e interoperabilidade.

2.5.1 Metodologia para Modelagem de Ontologias

De acordo com Noy, McGuinness et al. (2001), não existe um forma correta para modelar ontologias, ou seja, não existe uma metodologia padrão para modelagem de domínios específicos, porém existem opções viáveis de modelagem e a solução ideal depende do que o projetista prioriza e dos objetivos que pretende alcançar. Na literatura existem metodologias importantes para criação de ontologias, nas quais são empregados esforços de rigor e disciplina nas atividades e processos de desenvolvimento ontológico. Por isso, para escolha da metodologia de modelagem é preciso definir objetivos a serem atingidos e conhecimento a ser representado (NOY; MCGUINNESS et al., 2001).

Uma metodologia de grande relevância é a *Ontology Development 101* e conforme destaca Cristani e Cuel (2005) sua proposta é ser um guia descomplicado, que auxilia mesmo os desenvolvedores inexperientes a criar ontologias, por se fundamentar na construção de uma ontologia através de um processo iterativo de suas etapas. De acordo com sua estrutura proposta, são seguidas as 7 etapas de desenvolvimento a seguir (NOY; MCGUINNESS et al., 2001):

1. **Determinação do domínio e escopo da ontologia:** A metodologia sugere que se inicie o processo de desenvolvimento da ontologia definindo seu domínio e escopo, bem como respondendo a uma séria de perguntas básicas como:

   - Qual o domínio coberto pela ontologia?
   - Para o que vamos usar a ontologia?
   - Quem usará a ontologia?
   - A quais tipos de perguntas a ontologia deve fornecer respostas?

2. **Reutilização de ontologias existentes:** Considerar refinar e estender fontes existentes para o domínio ou tarefa da ontologia que estamos desenvolvendo. Muitas ontologias já existem na literatura, em formato eletrônico e podem ser importadas para a ontologia que estamos criando.

3. **Enumeração dos termos importantes da ontologia:** Enumerar todos os termos que gostaríamos de fazer declarações ou explicar a um usuário, quais propriedades esses termos possui e o que dizer sobre estes termos.

4. **Definição de classes e sua hierarquia:** Após listar os termos na etapa anterior, esta etapa sugere especificar as classes e sua hierarquia de classe.

5. **Definição das propriedades das classes:** Apenas as classes não fornecem informações suficientes para responder as questões da etapa 1. Após definir os termos na etapa 3, os termos restantes em sua maioria são propriedades dessas classes, sendo esta etapa caracterizada por definir as propriedades que representam interação entre as classes.

6. **Definição das facetas das propriedades:** Representam as restrições que podem ser impostas às propriedades, podendo ser divididas em três tipos:

   - Tipo de Valor: as propriedades que ligam classes a valores literais devem fazer restrições a tipos de valores aceitos.
   - Cardinalidade: determina a quantidade de valores que um *slot* pode ter. *Slots* são atributos de conceito, seu seja, classes que podem ser chamadas de papéis ou propriedades.
   - Domínio e Imagem: Definir classes que correspondam ao domínio e a imagem para um *slot*.

7. **Criação de instâncias:** Etapa responsável por criar instâncias individuais de classes na hierarquia e compreende: escolher a classe, criar instância individual nesta classe e preencher os *slots*.

2.5.2 Avaliação de Ontologias

Diversas abordagens para avaliação de ontologias foram consideradas na literatura, a depender da sua finalidade e o tipo de ontologia avaliada. De maneira geral, as abordagens de avaliação podem se agrupar nas seguintes categorias (BRANK; GROBELNIK; MLADENIC, 2005):

1. Comparação da ontologia com um padrão-ouro, ou seja, outra ontologia;

2. Uso da ontologia em uma aplicação e avaliação dos resultados obtidos;

3. Comparar a ontologia com uma fonte de dados (e.g., coleção de documentos) referente ao domínio a ser coberto;

4. Avaliação fundamentada em critérios pré-definidos, requisitos e padrões.

A abordagem de Gruber (1995) sobre avaliação de ontologias segue um caminho em direção a usabilidade e qualidade da ontologia. Sendo assim, em sua abordagem são propostos critérios objetivos tanto para criação de uma boa ontologia quanto para sua avaliação. Os critérios adotados por Gruber (1995) para criação e avaliação de ontologias são:

- Clareza: uma ontologia deve comunicar efetivamente o significado pretendido dos termos definidos, suas definições devem ser objetivas.

- Coerência: uma ontologia deve ser coerente, isto é, deve suportar inferências que sejam consistentes com as definições.

- Extensibilidade: uma ontologia deve ser capaz de definir novos termos para usos especiais baseado em definições já existentes, de uma maneira que não exija a revisão das definições anteriormente determinadas.

- Codificação Minimizada: conceitos devem ser especificados no nível de conhecimento sem depender de uma codificação específica no nível do símbolo. O viés de codificação deve ser minimizado, porque os agentes de compartilhamento de conhecimento podem ser implementados em diferentes sistemas e estilos de representação.

- Compromisso Ontológico: uma ontologia deve fazer o mínimo possível de afirmações sobre o mundo que está sendo modelado, permitindo que os utilizadores tenham liberdade, e tanto especializem quanto instanciem a ontologia, conforme necessário.

Outra forma de avaliar ontologias é através de medidas. Gangemi et al. (2006) identificou em seu estudo três diferentes medidas para avaliação de ontologias: medias estruturais, funcionais e relacionadas ao perfil de usabilidade. As medidas funcionais e estruturais dizem respeito a uma avaliação técnica, sintática e semântica formal e uso pretendido da ontologia. Já as medidas de usabilidade dizem respeito ao contexto de comunicação de uma ontologia e são importantes para a avaliação do usuário. Estas medidas de avaliação fazem parte da ferramenta de avaliação de ontologias OOPS![12] (do ingês, OntOlogy Pitfall Scanner!), que será utilizada para validar o modelo PROV-DOSN.

De acordo com Vrandečić (2009), executar uma boa avaliação de ontologia não assegura a inexistência de problemas, mas permite o uso mais seguro de determinada ontologia. Embora existam vários estudos sobre abordagens, métricas e ferramentas sobre avaliação, Degbelo (2017) afirma que ainda não existe consenso quanto a avaliação de ontologias.

2.5.3 PROV-O

A ontologia PROV (do inglês, *Prov Ontology* (PROV-O)), foi desenvolvida com objetivo de permitir a interoperabilidade de informações de proveniência em ambientes heterogêneos. Além disso, PROV-O é genérica e independente de domínio, permitindo que sistemas e aplicações possam estendê-la para seus propósitos de proveniência convenientes (MISSIER et al., 2013).

As formas de representar e compartilhar proveniência de informações na *Web* foram a motivação inicial de um esforço que resultou na criação de um modelo de dados para proveniência na *Web*, criado pelo *W3C Provenance Working Group*. Logo, segundo Moreau et al. (2015) todo esse processo e esforços de criação resultaram no conjunto de documentos que recebeu o nome de Família de Documentos PROV.

A Família de Documentos PROV é formada por doze documentos, incluindo uma visão geral que define sua especificação, como mostrado na Figura 2.7. Dentre os principais documentos estão o PROV-DM, um modelo que especifica a captura de dados, o PROV-CONSTRAINTS que estabelece um conjunto de restrições aplicáveis ao modelo PROV-

[12] <http://oops.linkeddata.es/index.jsp>

DM e regras de inferência e o PROV-O, uma ontologia para mapeamento dos dados (MISSIER, 2016).

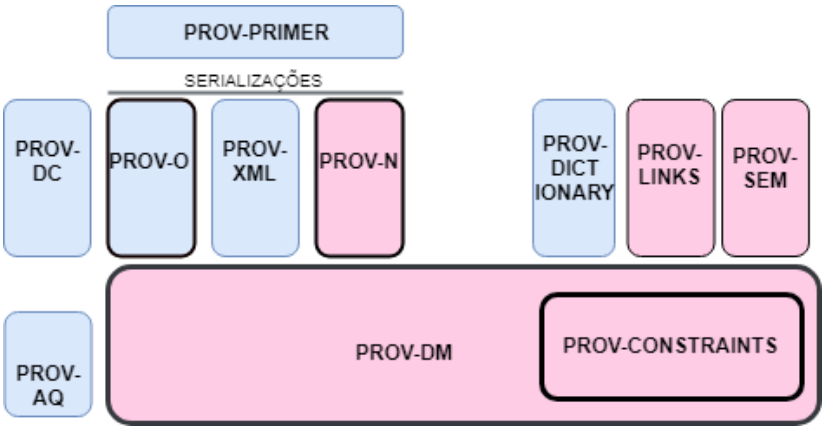


Figura 2.7 Família de Documentos PROV. Adaptado de Groth e Moreau (2013).

A ontologia PROV-O expressa o modelo de dados PROV-DM usando a Linguagem de Ontologia da *Web* (do inglês, *OWL2 Web Ontology Language* (OWL2)). A PROV-O possui classes, propriedades e restrições usadas para representar e trocar informações de proveniência entre diferentes sistemas e contextos. Dessa forma, segundo Closa et al. (2017), ela permite codificar proveniência em formato RDF, possibilitando descrever, capturar e consultar proveniência em ambientes distribuídos.

O ponto de partida para a utilização da PROV-O é um conjunto de classes e propriedades fundamentais usadas para criar descrições simples de proveniência. É possível observar, na Figura 2.8, o grafo de representação do modelo de classes e relacionamentos Prov, contendo as três classes: Entidade (*Entity*), Atividade (*Activity*) e Agente (*Agent*), que fornecem a base para todo PROV-O, em que:

- *prov:Entity* – é um tipo físico, digital ou conceitual (MOREAU et al., 2015). Representam estados imutáveis (CURCIN et al., 2017).
- *prov:Activity* – produtores e consumidores de prov:*Entity* (CURCIN et al., 2017).
- *prov:Agent* – responsável por uma atividade, que aconteceu, está acontecendo ou atividade de outro agente. (MOREAU et al., 2015).

De acordo com o modelo de classes desenvolvido por Behajjame et al. (2013), as Entidades são representadas como elipses amarelas, Atividades como retângulos azuis e Agentes como pentágonos laranja. As atividades acontecem em determinados pontos no tempo e são descritas pelas propriedades *prov:startedAtTime*, determinando início e *prov:endedAtTime* indicando o seu final. Na Figura 2.8 é possível observar os relacionamentos PROV, em que as arestas do grafo representam as relações entre os nós. As três classes básicas ou primárias, *Entity*, *Activity* e *Agent* se relacionam entre si ou com elas mesmas usando as seguintes propriedades:

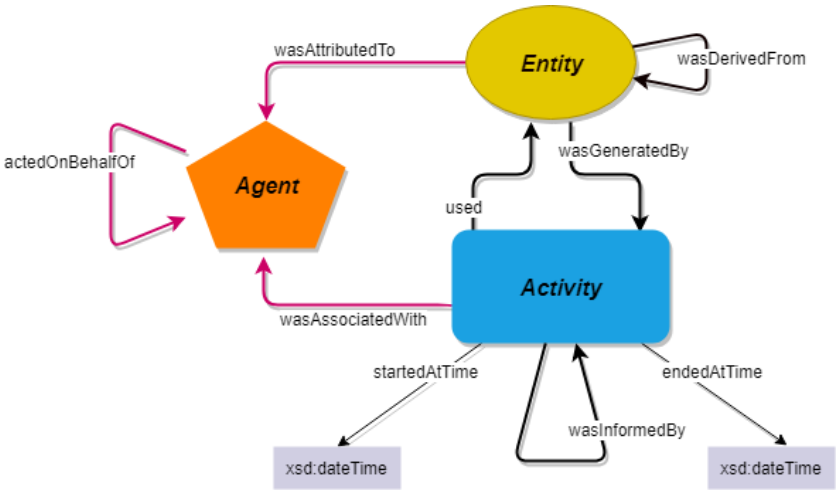


Figura 2.8 Modelo de classes e relacionamentos PROV. Adaptado de Behajjame et al. (2013).

- *wasGeneratedBy* – determina a produção de uma nova entidade por uma atividade. Essa entidade só passou a existir e ficar disponível após sua geração.
- *wasDerivedFrom* – transformação de uma entidade em outra, atualização de uma entidade resultando em uma nova ou a criação de entidade a partir de outra pré-existente.
- *wasAssociatedWith* – associação de responsabilidade de uma atividade a um agente.
- *wasAttributedTo* – atribuição de uma entidade a um agente.
- *wasInformedBy* – uma atividade é dependente de outra por usar entidade em comum.
- *used* – é o inicio da utilização de uma entidade por uma atividade.
- *actedOnBehalfOf* – atribuição ou delegação de autoridade a um agente.

Como uma forma de representar a proveniência e poder compartilhá-la na *Web*, a ontologia PROV-O é leve, podendo ser usada em várias aplicações. Além disso, pode ser usada de forma direta para representar proveniência ou pode ser utilizada para modelar ontologias de domínios específicos. Nesse sentido, Behajjame et al. (2013) afirmam que os usuários são capazes de utilizar apenas parte da ontologia que julgar necessária, dependendo de quanto deseja detalhar e das suas necessidades de proveniência específicas. Os termos da ontologia PROV-O reutilizados pelo modelo DOSN-PROV são apresentados na Figura 2.9.

2.5.4 FOAF

O vocabulário FOAF é resultado da colaboração de desenvolvedores no projeto que leva o mesmo nome, iniciado em 2000 por Dan Brickley e Libby Miller. De acordo com

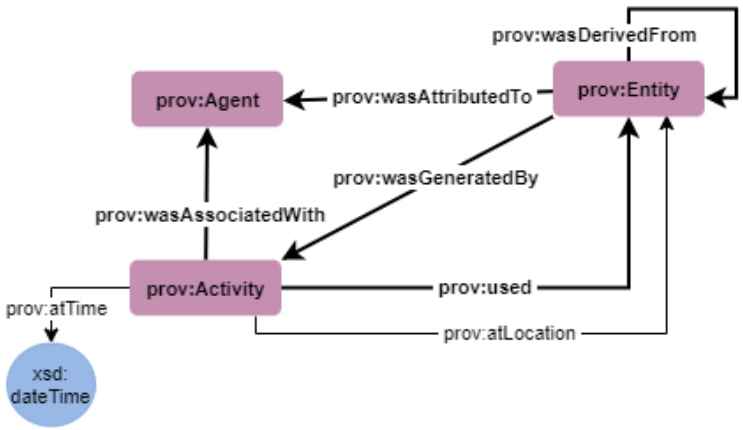


Figura 2.9 Termos da ontologia PROV-O reutilizados no modelo DOSN-PROV.

Golbeck e Rothstein (2008), o projeto FOAF se consolidou como um vocabulário padrão, bastante adotado na representação de redes sociais e usado por sites para produzir perfis de usuários na Web Semântica.

Segundo Ding et al. (2005), o item mais importante de um documento FOAF é o seu vocabulário, identificado pelo URI: `http://xmlns.com/foaf/0.1`. Vocabulários FOAF definem classes (e.g., `foaf:Agent`, `foaf:Person` e `foaf:Document`) e propriedades (e.g., `foaf:name` e `foaf:mbox`) fundamentadas na semântica RDF. Na Tabela 2.1 podem ser observadas as demais classes e propriedades que compõem o vocabulário FOAF (BRICKLEY; MILLER, 2014).

Tabela 2.1 Classes e Propriedades FOAF.

CLASSES	PROPRIEDADES			
Agent	account	focus	firstname	holdsAccount
Document	accountName	theme	surname	lastName
Group	age	name	fundedBy	pastProject
Image	aimChatId	tipjar	geekcode	schoolHomepage
LabelProperty	based_near	title	gender	mbox_sha1sum
OnlineAccount	birthday	topic	givenName	topic_interest
OnlineChatAccount	currentProject	page	knows	msnChatId
OnlineEccomerceAccount	depiction	weblog	homepage	myersBriggs
OnlineGamingAccount	dnaChecksum	sha1	icqChatId	jabberId
Organization	familyName	status	interest	isPrimaryTopicOf
Person	family_name	openId	skypeId	membershipClass
PersonalProfileDocument	publications	img	plan	workInfoHomepage
	member	logo	made	workplaceHomeplace
	phone	mbox	surname	
	primaryTopic	maker	thumbnail	

No FOAF, as descrições de perfis de usuários e representações de conexões não só são baseadas em OWL/RDF, como também incluem métodos capazes de representar informações pessoais e ligações sociais usando tecnologias da Web Semântica. Logo, de acordo com Sohn e Chung (2013), esses métodos tornam mais simples o compartilhamento de dados em redes sociais.

Brickley e Miller (2010) afirmam que o vocabulário FOAF se baseia na intenção de unir redes de descrições descentralizadas, partindo da ideia de pessoas publicarem informações em formato de documentos FOAF e as máquinas poderem fazer uso dessas informações. Os termos FOAF reutilizados no modelo DOSN-PROV são apresentados na Figura 2.10.

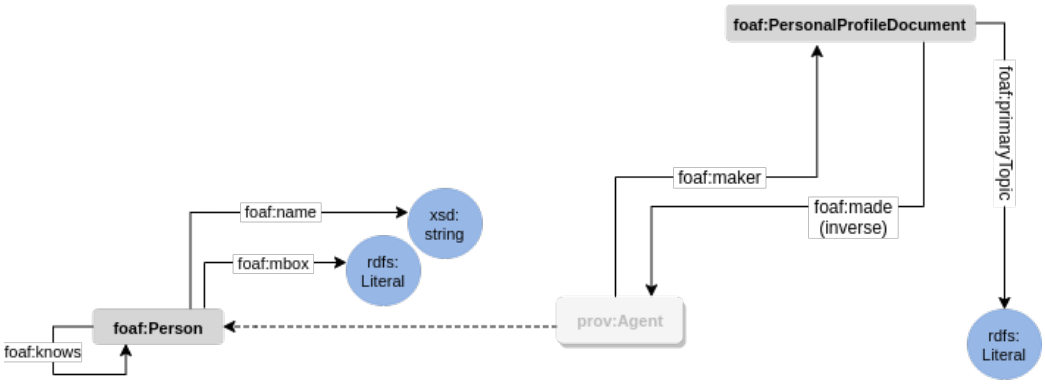


Figura 2.10 Termos FOAF reutilizados no modelo DOSN-PROV.

2.5.5 SIOC

O projeto de Comunidades Online Interligadas Semanticamente (do inglês, *Semantically Interlinked Online Communities* (SIOC)) objetiva permitir a integração de informação de comunidades *online*. Além disso, segundo Bojars et al. (2010), oferece uma ontologia da Web Semântica para representar dados da Web Social em RDF, sendo comumente utilizada em conjunto com o vocabulário FOAF para representar perfis pessoais e informações em redes sociais.

O projeto SIOC[13] tem foco em manter a integração através de interações sociais legíveis por máquinas, fornecendo um modelo de dados para representação de comunidades online e suas atividades de maneira homogênea (PASSANT et al., 2009).

Para proporcionar facilidade de uso e legibilidade, a ontologia é dividida e seus módulos são separados em acesso, serviços e tipos. Segundo Bojars et al. (2010), essa divisão possibilita o uso inicial de um esquema simples, começando geralmente com o *Core Ontology* e utilizando outros módulos conforme a necessidade de uso da aplicação.

Nas Ontologias básicas SIOC são utilizados termos que descrevem áreas da *Web*, como blogs e fóruns. Como exemplos de alguns termos desse vocabulário nos ambientes citados temos: `sioc:Post`, permitindo aos usuários a descrição de postagens, organizadas em

[13] <http://sioc-project.org/>

fóruns usando o `sioc:Forum`. Com a evolução de ambientes sociais online os termos citados se tornaram subclasses de outros termos adicionados ao SIOC (BOJARS et al., 2010). Na Figura 2.11 são apresentadas as principais classes e propriedades da ontologia SIOC.

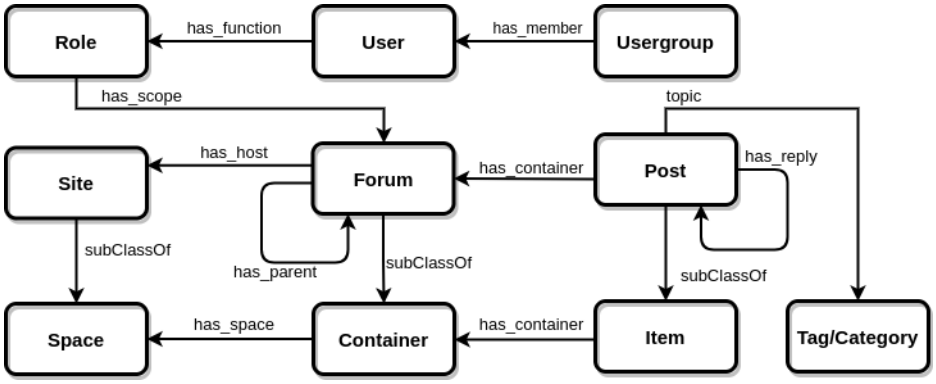


Figura 2.11 Principais classes e propriedades Sioc. Adaptado de Bojars et al. (2010)

Classes e propriedades de outras ontologias podem ser utilizadas em conjunto com o SIOC. Os conceitos das outras ontologias não são incluídos no SIOC, mas são utilizados diretamente como termos SIOC para descrever informações (PASSANT et al., 2009). Como exemplo de reutilização, temos o já citado FOAF, o Dublin Core para definir atributos de conteúdo criado e Sistema Simples de Organização do Conhecimento (do inglês, *Simple Knowledge Organization System* (SKOS)), para modelar tópicos de discussão. Os termos SIOC reutilizados pelo modelo DOSN-PROV são: sioc:Post, sioc:content e sioc:like.

2.5.6 vCard

vCard é uma especificação utilizada para descrever pessoas e organizações, compreendendo adicionalmente informações de localizações e grupos de entidades. A ontologia vCard contou com suas primeiras especificações em 1995 e atualmente continua utilizando o *namespace* `<http://www.w3.org/2006/vcard/ns#>`, com objetivo de manter a compatibilidade com suas versões anteriores (IANNELLA; MCKINNEY, 2014).

A Figura 2.12 apresenta a representação de um exemplo de uso da ontologia vCard na sintaxe *turtle*. A sintaxe turtle, de acordo com a definição de Beckett et al. (2014), permite que grafos RDF sejam escritos em formato de texto natural e compacto, utilizando tipos de dados comuns e abreviações para padrões de uso.

Os termos vCard reutilizados pelo modelo de proveniência PROV-O estão relacionados à identificação de usuário e endereço. Estes termos de reuso são apresentados na Figura 2.13.

```
@prefix vcard: <http://www.w3.org/2006/vcard/ns#> .
@prefix rdfa: <http://www.w3.org/ns/rdfa#> .
<http://example.com/me/corky> a vcard:Individual;
    vcard:hasEmail <mailto:corky@example.com>;
    vcard:fn "Corky Crystal";
    vcard:hasAddress [ a vcard:Home;
      vcard:country-name "Australia";
      vcard:locality "WonderCity";
      vcard:postal-code "5555";
      vcard:street-address "111 Lake Drive" ];
    vcard:hasTelephone [ a vcard:Home,
      vcard:hasValue <tel:+61755555555> ];
    vcard:nickname "Corks" .
```

Figura 2.12 Exemplo de uso da ontologia vCard em formato turtle. Adaptado de Iannella e McKinney (2014).

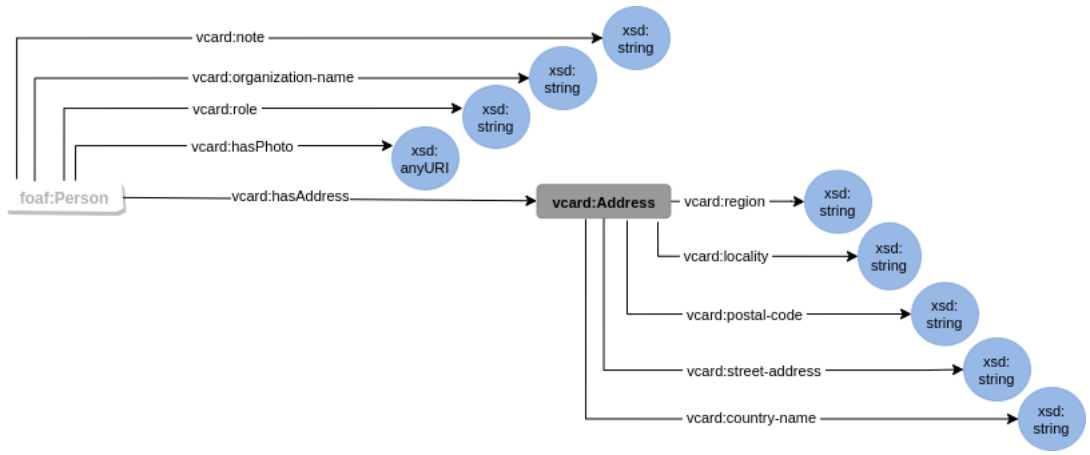


Figura 2.13 Termos vCard reutilizados no modelo DOSN-PROV.

2.6 CONSIDERAÇÕES FINAIS

Este capítulo apresentou os principais conceitos e tecnologias que fundamentam nesta pesquisa. Foram apresentados os conceitos de DOSN, proveniência de dados e *Linked Data*. Foram apresentados detalhes do ecossistema SOLID, ontologias, metodologia de modelagem de ontologia e avaliação de ontologia. Por fim, foram descritas as ontologias PROV-O e os vocabulários FOAF, SIOC e vCard, reutilizadas para criação do modelo DOSN-PROV.

Capítulo 3

DOSN-PROV

Este capítulo apresenta a arquitetura de proveniência, que inclui o modelo DONS-PROV e os serviços de suporte à captura e rastreamento de proveniência em DOSNs. A proposta é apresentada como segue: a Seção 3.1 apresenta a arquitetura de proveniência. Na Seção 3.2 são detalhados os requisitos de proveniência utilizados para elaboração do modelo DOSN-PROV, apresentado na Seção 3.3. A Seção 3.4 apresenta os serviços de suporte a proveniência. A implementação da aplicação utilizada nos nossos testes, que é baseada na arquitetura do Solid, é apresentada na Seção 3.5. Na Seção 3.6 são apresentados trabalhos relacionados à coleta, ao rastreamento e ontologias de proveniência. Por fim, na Seção 3.7 são discutidas as considerações finais do capítulo.

3.1 ARQUITETURA DE PROVENIÊNCIA

A arquitetura de proveniência proposta é apresentada na Figura 3.1 e é dividida em: (i) Rede Social Descentralizada; (ii) Servidor de Serviços de Proveniência, composto pelos serviços de Captura, Rastreamento e pelo Modelo DOSN-PROV; e (iii) Servidor de Proveniência, formado por *triplestores* responsáveis por armazenar a proveniência capturada. Dessa forma, os dados de proveniência seguem o seguinte fluxo na arquitetura:

- **Figura 3.1 - Passo 1**: Postagens e informações dos usuários de redes sociais serão persistidas em PODs de usuários através de aplicações de DOSNs;
- **Figura 3.1 - Passo 2**: No momento da execução de postagens a aplicação executa requisições REST (HTTP) ao serviço de proveniência para captura dos dados de proveniência;
- **Figura 3.1 - Passo 3**: Os dados de proveniência capturados no Passo 2 são persistidos em *triplestores* de proveniência;
- **Figura 3.1 - Passo 4**: A proveniência capturada e armazenada em *triplestores* pode ser recuperada através de consultas ao serviço de rastreamento por meio de consultas SPARQL;

- **Figura 3.1 - Passo 5**: O serviço de rastreamento processa as consultas SPARQL e acessa as *triplestores*, onde estão armazenados os dados de proveniência;
- **Figura 3.1 - Passo 6**: O serviço de rastreamento devolve os resultados da consulta de rastreamento recuperados.

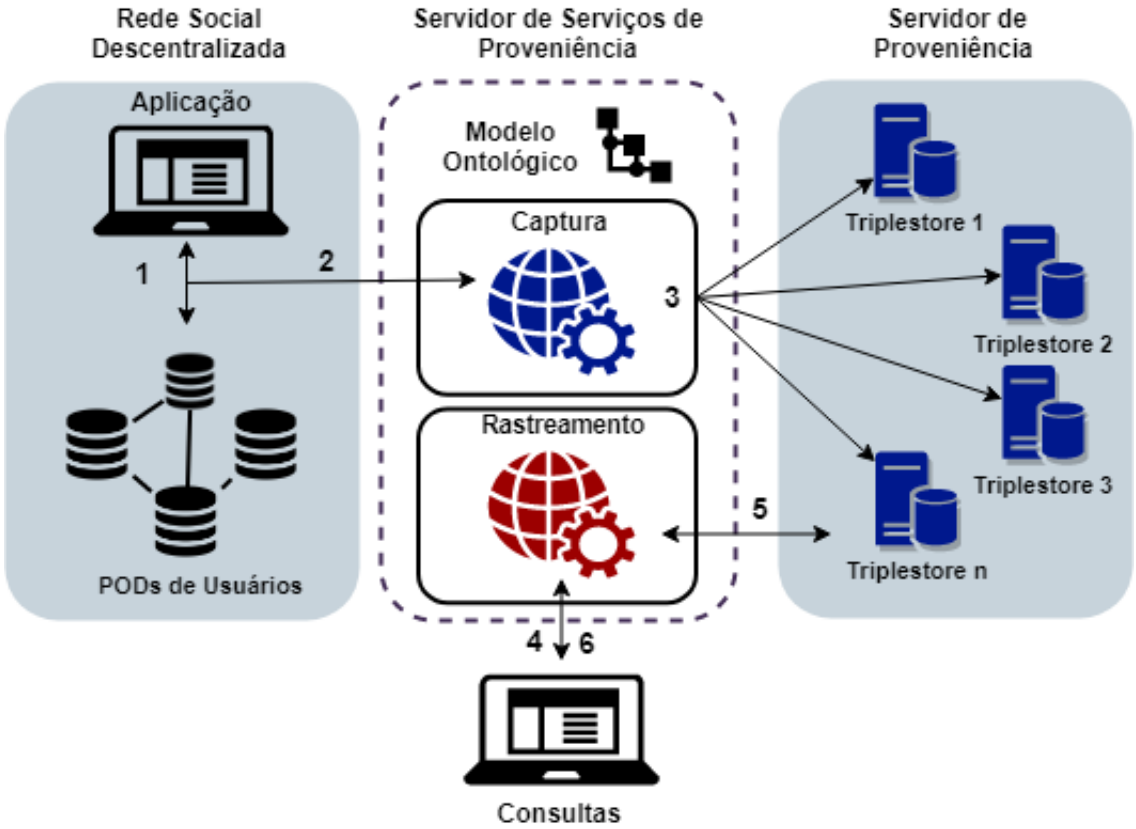


Figura 3.1 Arquitetura de Proveniência.

3.2 REQUISITOS DE PROVENIÊNCIA

Buscando maior gerenciamento da qualidade e da grande variedade de dados e informações da *Web*, o W3C através do *Provenance Incubator Group*[1] definiu um conjunto de padronizações relativo à proveniência em aplicações da Web Semântica. Adicionalmente, o *Provenance Incubator Group* produziu um relatório com as principais dimensões de proveniência e mais de trinta casos de uso que exemplificam essas dimensões por meio de um vasto conjunto de requisitos.

Para Groth et al. (2012), os requisitos de proveniência podem ser sintetizados em várias dimensões, contudo são agrupados em três aspectos-chave: (i) gerenciamento dos

[1] <https://www.w3.org/2005/Incubator/prov/charter>

registros de proveniência, (ii) conteúdo de proveniência e (iii) utilização das informações de proveniência. Gil et al. (2010) definiram requisitos de proveniência motivados por casos de uso coletados pelo *Provenance Incubator Group* e esses requisitos também foram agrupados de acordo com os mesmos aspectos-chave: gerenciamento, conteúdo e utilização.

Tabela 3.1 Requisitos de Proveniência.

Requisito	**Aspecto-Chave**	**Descrição**
R1	Conteúdo	O modelo deve ser capaz de identificar a fonte e a entidade que são referidas por proveniência.
R2	Conteúdo	O modelo deve deixar clara a noção de "criador de dados".
R3	Conteúdo	O modelo deve ser capaz de determinar a influência de um agente em uma determinada atividade.
R4	Conteúdo	O modelo de proveniência deve ser rico o suficiente em termos de descrição de processos geradores de proveniência, sendo capaz de identificar os principais recursos de um processo.
R5	Conteúdo	O modelo deve determinar e registrar quando o conteúdo foi gerado.
R6	Gerenciamento	O modelo deve permitir rastreamento de proveniência.
R7	Gerenciamento	O modelo deve permitir rastrear as postagens republicadas novamente até sua fonte original.
R8	Uso	O modelo deve ser rico o suficiente para cobrir a maioria dos possíveis comportamentos para um determinado domínio.
R9	Uso	O modelo deve fornecer os meios para consultar a proveniência de uma entidade.
R10	Uso	Consultas devem ser capazes de identificar aspectos específicos de proveniência.

Para a elaboração do modelo DOSN-PROV, foram utilizados requisitos de proveniência baseados no documento gerado pelo *Provenance Incubator Group*. Para a obtenção dos requisitos de proveniência adequados ao domínio de DOSNs, foi realizada uma curadoria nos documentos de requisitos do usuário, requisitos técnicos, bem como em todos os casos de uso disponíveis pelo referido grupo. Os requisitos de proveniência aplicáveis a este trabalho de mestrado foram filtrados e adaptados ao domínio de DOSNs e são apresentados na Tabela 3.1, onde estão descritos e agrupados por aspecto-chave. A seguir, são realizadas considerações sobre os requisitos elicitados para construção do modelo DOSN-PROV.

- Os requisitos R1 e R2 referem-se à atribuição de proveniência. Esses requisitos demonstram a responsabilidade sobre entidades, deixando evidente os criadores de

dados, atribuindo prova de autoria para um agente sobre uma determinada entidade. Estes requisitos consideram que o criador ou fonte de postagem é responsável pelo seu conteúdo, não sendo possível fazer atribuição de responsabilidade a entidades externas a DOSN.

- Os requisitos R3 e R4 são referentes aos processos, que são as atividades executadas para criar entidades. Essas atividades sofrem influência de agentes e precisam ter seus recursos identificados.
- O requisito R5 está relacionado ao tempo, onde é definido que deve haver um marco temporal de criação dos registros.
- Requisitos R6 e R7 tratam de permitir a rastreabilidade da proveniência, sendo um requisito vital para um dos serviços propostos neste trabalho, pois abrange possibilidade de criar trilhas de postagens no caso de compartilhamentos.
- O requisito R8 define que o modelo deve abranger as principais funcionalidades do domínio. No contexto deste trabalho, as funcionalidades que o modelo deve cobrir se referem ao domínio de redes sociais.
- Os requisitos R9 e R10 se referem às consultas de proveniência, tornando possível que sejam executadas consultas específicas por entidades e também possam ser feitas consultas específicas por parâmetros.

3.3 MODELO DOSN-PROV

DOSN-PROV é um modelo ontológico de proveniência desenvolvido com o objetivo de permitir manipular proveniência em DOSNs. O modelo DOSN-PROV é leve e foi elaborado utilizando padrões de modelagem de ontologias e requisitos de proveniência recomendados pelo W3C.

3.3.1 Desenvolvimento do Modelo DOSN-PROV

O modelo DOSN-PROV foi desenvolvido utilizando a metodologia de desenvolvimento de ontologia *Ontology Development 101*, descrita no Capítulo 2. A metodologia *Ontology Development 101* foi escolhida por propor dividir a construção da ontologia em etapas e por contar com processo interativo, tornando mais simples o desenvolvimento da ontologia. Seguindo a *Ontology Development 101*, a primeira etapa consiste em determinar o escopo e domínio da ontologia, o que inclui definir seus requisitos. Portanto, DOSN-PROV é uma ontologia de proveniência para redes sociais de arquitetura descentralizada. Os requisitos de proveniência usados para o desenvolvimento do modelo estão listados e descritos na Seção 3.2.

A segunda etapa se refere à reutilização de ontologias existentes. Assim, para a construção do modelo DOSN-PROV priorizamos reutilizar o máximo possível de ontologias já existentes, buscando criar o mínimo possível de termos novos, ou seja, termos ainda não definidos em outras ontologias. Nesse intuito, o modelo DOSN-PROV reutiliza as ontologias PROV-O, FOAF, SIOC e vCard, apresentadas no Capítulo 2.

Enumerar os termos importantes da ontologia faz parte da terceira etapa da metodologia utilizada. Por meio das etapas anteriores, de definição de requisitos e reutilização de ontologias, foi possível enumerar termos importantes para o desenvolvimento desta etapa. Logo, entre os termos importantes encontrados nesta etapa estão: postagem, conteúdo, compartilhamento, entidade, atividade, agente, criador, publicação e comentário. Foram criados apenas seis termos novos nesta etapa de construção do modelo, sendo quatro classes referentes a atividades (*dosn-prov:Publish*, *dosn-prov:Comment*, *dosn-prov:React* e *dosn-prov:Share*) e duas propriedades de dados (*dosn-prov:idEntity* e *dosn-prov:IdActivity*).

As tarefas posteriores são ligadas à implementação do modelo e se referem à definição de classes e sua hierarquia, propriedades e restrições. A Figura 3.2 ilustra o modelo DOSN-PROV, onde é possível ter uma visão geral de toda a ontologia, suas classes, propriedades de objetos e propriedades de dados.

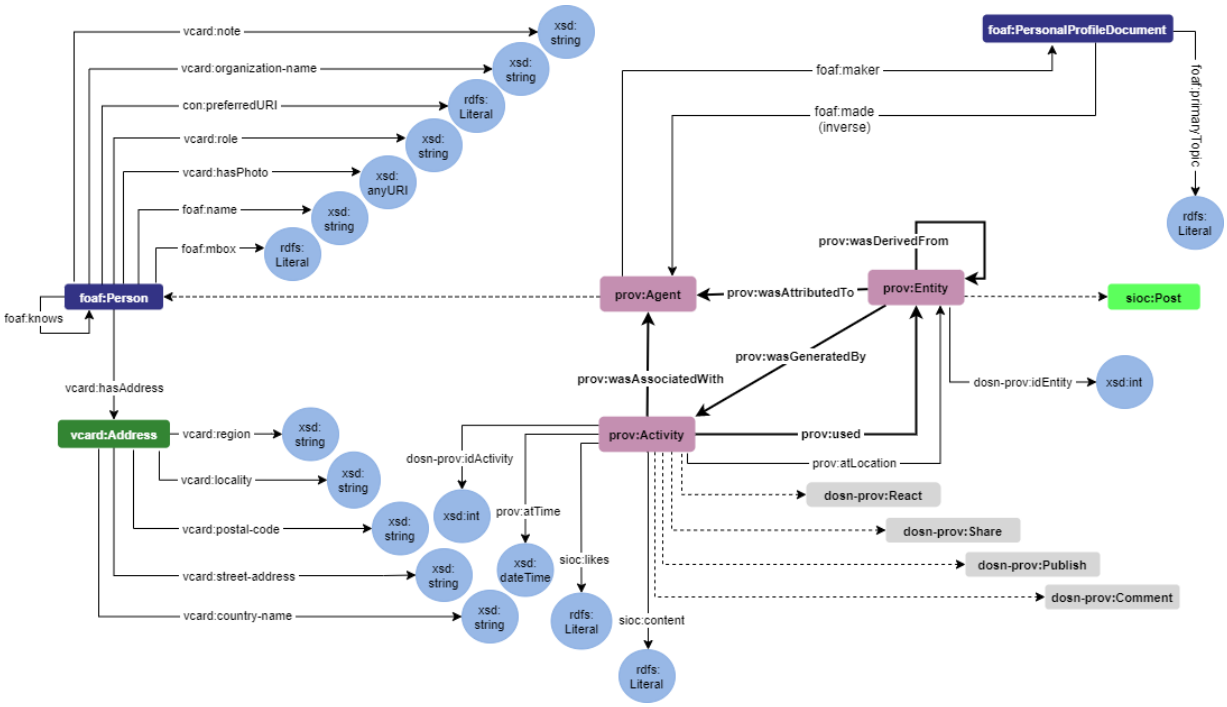


Figura 3.2 Modelo DOSN-PROV.

Para interpretação da Figura 3.2, que contém o modelo DOSN-PROV, é utilizada a seguinte legenda: retângulos representam classes; círculos são equivalentes a tipos de dados, representando dados literais; setas tracejadas interligando retângulos apontam para subclasses; setas sólidas interligando retângulos correspondem a propriedades de objetos; por fim, setas sólidas interligando um retângulo e um círculo, representam propriedades de dados.

Ainda em relação à Figura 3.2, as cores de classes são referentes aos termos criados

ou reutilizados: retângulos azuis representam termos da ontologia FOAF; retângulos rosa são equivalentes a ontologia PROV-O; retângulo verde escuro representa um termo da ontologia VCard; retângulo em verde claro equivale a um termo da ontologia SIOC; por fim, retângulos cinza representam termos DOSN-PROV, criados neste trabalho.

A seguir estão detalhadas as principais classes e propriedades definidas para o modelo DOSN-PROV. Os prefixos utilizados indicam de qual ontologia a classe ou propriedade foi reusada. Além disso, as classes e propriedades criadas neste trabalho estão indicadas com o prefixo *dosn-prov*.

- *prov:Agent* - Classe onde serão atribuídas responsabilidades. Um *prov:Agent* se relaciona com *prov:Entity* através da propriedade *prov:wasAttibutedTo*, o que faz com que uma entidade seja atribuída a um agente, seja ele um *foaf:Person* ou *foaf:Organization*, pois são subclasses de *prov:Agent*.

- *prov:Entity* - Classe que tem como subclasse *sioc:Post* e é utilizada para obter informações sobre as postagens da classe *prov:Agent*. Por meio da propriedade *prov:wasGeneratedBy*, é possível identificar qual entidade foi gerada por uma atividade. Portanto, *sioc:Posts* são entidades geradas por atividades como: *prov:Share*, *dosn-prov:React*, *dosn-prov:Publish* ou *dosn-prov:Comment*.

- *prov:Activity* - Classe que define a ocorrência de uma atividade e a que tempo. Uma *prov:Activity* tem seu tempo definido através da propriedade *prov:atTime*. As atividades desempenhadas nesse domínio podem ser: *dosn-prov:Comment*, *dosn-prov:Publish*, *dosn-prov:React* e *dosn-prov:Share*, representando respectivamente comentários, publicações, reações e compartilhamentos de entidades. Através da propriedade *prov:used*, é possível identificar o *sioc:Post* utilizado para gerar determinada *prov:Activity*. Por fim, uma *prov:Activity* é associada a um agente através da propriedade *prov:wasAssociatedWith*.

- *sioc:Post* - É uma subclasse de *prov:Entity*. Um *sioc:Post* é realizado pelo usuário *prov:Agent* e o local da postagem é adicionado através da propriedade de dados *prov:atLocation*. A propriedade *prov:atLocation* define o local da postagem, se na página do usuário ou na página de terceiros. Já o conteúdo da postagem é adicionado através da propriedade de dados *sioc:content*. É importante ressaltar que *sioc:Post* é uma das classes mais importantes do modelo DOSN-PROV, pois toda sua proveniência é capturada, tanto relacionamentos quanto conteúdo.

- *foaf:Person* - subclasse de *foaf:Agent*, representa pessoas, onde toda pessoa é considerada um agente. Uma *foaf:Person* se relaciona com outra por meio da propriedade de objetos *foaf:knows*, indicando algum tipo de interação recíproca.

- *vcard:Address* - Classe que especifica o endereço de moradia de uma *foaf:Person*. A ligação existente entre um foaf:Person e *vcard:Address* se dá através da propriedade *vcard:hasAddress*. Desse modo, para representar um endereço de forma completa as seguintes propriedades de dados são utilizadas: *vcard:country-name*, *vcard:street-address*, *vcard: postal-code*, *vcard:locality* e *vcard:region*.

- *foaf:PersonalProfileDocument* - Classe que representa o documento em que serão descritas as propriedades de um *prov:Agent* que criou seu documento de perfil pessoal. A atribuição de um *foaf:PersonalProfileDocument* ao agente é determinada pela propriedade *foaf:maker* sendo a sua inversa a propriedade *foaf:made*. A propriedade *foaf:primaryTopyc* relaciona o documento ao seu objeto principal, o criador do documento de perfil. As seguintes informações de um *foaf:Person* serão representadas através dos relacionamentos obtidos pelas propriedades de dados no foaf:PersonalProfileDocument:
  - *foaf:mbox* - relacionamento de um *foaf:Person* com uma caixa de correio.
  - *foaf:name* - Nome atribuído a um *foaf:Person*.
  - *vcard:hasPhoto* - Imagem ou informação de imagem.
  - *con:preferredURI* - Uma URI de uma pessoa, em que o WebId de um *foaf:Agent* é armazenado.
  - *vcard:organization-name* - Nome de uma organização associada ao *foaf:Person*.
  - *vcard:note* - Nota associada a um *foaf:Person*.
- As propriedades de dados *dosn-prov:IdEntity* e *dosn-prov:idActivity* são utilizadas como identificador de *prov:Entity* e *prov:Activity*. Essas propriedades foram criadas neste trabalho com o objetivo individualizar postagens e atividades como publicações, comentários, compartilhamento e reações.

3.4 SERVIÇOS DE PROVENIÊNCIA

Os serviços de proveniência fazem parte da arquitetura de proveniência apresentada na Seção 3.1 e foram desenvolvidos com objetivo de oferecer suporte à captura e ao rastreamento de dados de proveniência em DOSNs. Os serviços foram implementados utilizando Java 8.0, Framework Spring Boot[2], um framework Java para construção de *RESTful Web Services* e o Apache Jena[3], um framework Java gratuito e de código aberto para aplicações da Web Semântica e *Linked Data*.

3.4.1 Serviço de Captura de Proveniência

O serviço de captura de proveniência tem a função de coletar dados de proveniência postados por usuários de DOSNs. Para fazer uso do serviço de captura de proveniência, as DOSNs devem fundamentar suas postagens e atividades de acordo com o modelo DOSN-PROV. As aplicações, no momento da publicação de postagens nas suas redes, ou seja, no momento da persistência dessas informações nos PODs dos usuários, farão chamadas ao serviço via protocolo HTTP, com objetivo de efetuar a captura dos dados de proveniência conforme o modelo DOSN-PROV.

A proveniência das postagens dos usuários serão persistidas em *triplestores*, servidores de proveniência que armazenarão os dados para posteriores recuperações pelo serviço de

[2]<https://spring.io/>
[3]<https://jena.apache.org/index.html>

rastreamento de proveniência, como apresentado na Seção 3.4.2. O processo de coleta de proveniência executado pelo serviço através do método HTTP POST é representado pelo Algoritmo 1, onde o método é utilizado para capturar a proveniência de dados de uma atividade (publicação, comentário, compartilhamento ou reação) e gera uma entidade *sioc:Post*. Qualquer dessas atividades seguem o comportamento de criar uma *sioc:Post*, conforme modelado pela ontologia.

Na linha 3 do Algoritmo 1, o modelo DOSN-PROV é carregado do servidor de serviços de proveniência. Na linha 4 é criada a estrutura do objeto de acordo com o modelo já carregado. Na linha 5 são preenchidos os dados com a proveniência capturada e na linha 6 o *post* é finalmente inserido na *triplestore* de proveniência através do método gravarPost(), encerrando o serviço para essa requisição. Vale ressaltar que as chamadas ao serviço de captura são realizadas através do método POST nos caminhos /publish, /comment, /share e /react.

Algoritmo 1: Captura de Proveniência

```
   Entrada: dados_de_proveniencia // passados no corpo da requisição)
1  Servico_de_Captura(dados_de_proveniencia)
2  início
       // Carrega o modelo DOSN-PROV
3      modelo = carregarModelo(dosnProv)
       // Cria um post de acordo com o modelo
4      post = criarPost(modelo)
       // Preenche o post com os dados capturados
5      post = dados_de_proveniencia
       // Grava o post na Triplestore
6      gravarPost(post)
7  fim
```

Ao receber uma chamada HTTP, o serviço recebe no corpo da requisição os dados referentes à atividade e ao *sioc:Post*, que terá sua proveniência coletada. Pois, *posts* são gerados por alguma atividade como: publicação, comentário, compartilhamento ou reação. Desse modo, todos os casos efetuam a coleta do *sioc:Post* e tem os seguintes dados passados no corpo da requisição: *dosn-prov:idEntity*, *prov:wasAttribuitedTo*, *prov:wasGeneratedBy* e *prov:wasDerivedFrom*. Adicionalmente a esses dados que se referem ao post são passados os dados da atividade que gerou esse *sioc:Post*.

Por exemplo, se uma atividade de captura de comentário for chamada pelo serviço, além dos dados de *post* deverão ser passados no corpo da requisição também as propriedades: *prov:used*, *prov:wasAssociatedWith*, *sioc:content*, *dosn-prov:idActivity*, *prov:atLocation*, e *prov:atTime*. Dessa mesma forma acontece com chamadas a outros serviços, deverão passar dados do *sioc:Post* e dados específicos de cada atividade.

Para fazer uso do serviço de proveniência, é preciso que a aplicação que o utilize faça a modelagem de seus dados de acordo com os prefixos utilizados pelo modelo, pois o serviço captura os dados de proveniência de acordo com o modelo DOSN-PROV.

3.4.2 Serviço de Rastreamento de Proveniência

O Serviço de Rastreamento é responsável por permitir consultar as informações de proveniência persistidas nas *triplestores* de proveniência através do serviço de captura. Para ter acesso à proveniência são executadas consultas SPARQL, por meio de requisições HTTP ao serviço. O serviço recebe a requisição, acessa a *triplestore* de proveniência e retorna o resultado da requisição contendo as informações consultadas.

Algoritmo 2: Algoritmo de rastreamento de proveniência

```
   Entrada: String_de_consulta
   Saída: Resultado
1  Servico_de_Rastreamento(String_de_consulta)
2  início
       // Cria uma consulta SPARQL a partir da string de consulta
3      consulta = criarConsulta(string_de_consulta)
       // Cria um objeto de execução de consulta
4      execucaoConsulta = criarExecucaoConsulta(url_servico_sparql, consulta)
       // Executa a consulta à Triplestore
5      resultado = executarConsulta(execucaoConsulta)
       // Retorna os dados de proveniência encontrados
6      retorna resultado
7  fim
```

O Algoritmo 2 representa o método HTTP GET, que executa consulta SPARQL na *triplestore* de proveniência, contendo os dados capturados pelo método HTTP POST, representado pelo Algoritmo 1. O algoritmo mostra como o serviço efetua o rastreamento de dados. Na linha 3 do Algoritmo 2, o serviço cria uma consulta SPARQL recebendo como parâmetro uma String de consulta. Na linha 4 do Algoritmo 2, é criado um objeto para execução da consulta, recebendo como parâmetros a URL do serviço SPARQL e a consulta. Nas linhas 5 e 6 do Algoritmo 2, é executada a consulta na *triplestore* de proveniência e são retornados os dados encontrados para a consulta executada.

3.5 IMPLEMENTAÇÃO

Com objetivo de auxiliar na etapa de avaliação, apresentada no Capítulo 4, foi desenvolvida uma aplicação de rede social baseada no ecossistema Solid, apresentado no Capítulo 2. Idealizada por um grupo de pesquisa liderado por Tim Bernners-Lee (SAMBRA et al., 2016), o Solid contribui para a construção de um ambiente social descentralizado baseado nos princípios *Linked Data*.

A aplicação foi desenvolvida utilizando as bibliotecas *JavaScript* do *Inrupt*[4]: *solid-*

[4]<https://inrupt.com/>

client-authn[5], *solid-client*[6] e *vocab-common-rdf*[7]. A biblioteca *solid-client-authn* é utilizada para autenticação em Provedores de Identidade Solid, que permitem o acesso aos PODs de usuários. A *solid-client* é uma biblioteca cliente desenvolvida para garantir o acesso aos dados armazenados em PODs Solid, fornecendo uma camada de abstração entre RDF e os princípios Solid. Portanto, essa biblioteca foi utilizada na aplicação desenvolvida com o objetivo de gravar e acessar os dados de postagens e informações de usuários armazenados nos PODs Solid.

A biblioteca *vocab-common-rdf* possui bibliotecas de vocabulários que fornecem constantes estáticas para identificadores usados no SOLID, tornando a codificação mais simples, sem a necessidade de adicionar todo o URI, apenas a classe do identificador oferecido pela biblioteca. Dessa forma, a biblioteca *common-vocab-rdf* foi utilizada neste trabalho para relacionar identificadores de objetos ao seus vocabulários RDF (e.g, FOAF, VCARD e PROV).

A arquitetura de desenvolvimento de aplicações Solid é dividida em duas camadas: (i) camada cliente e (ii) camada de dados. Para o desenvolvimento da aplicação de rede social foram utilizados: Linguagem de Marcação de HiperTexto (do inglês, *HyperText Markup Language* (HTML)), Folhas de Estilo em Cascata (do inglês, *Cascading Style Sheets* (CSS)) e JavaScript na camada cliente e na camada de dados, o POD Solid. A aplicação permite efetuar postagens em sua estrutura, utiliza a estrutura de PODs Solid para persistência dos dados dessas postagens de usuários e consome os dados dos usuários armazenados no POD. Além disso, a aplicação realiza chamadas ao serviço de captura no momento das postagens para coleta da proveniência e permite a recuperação dos dados de proveniência por meio de chamadas ao serviço de rastreamento.

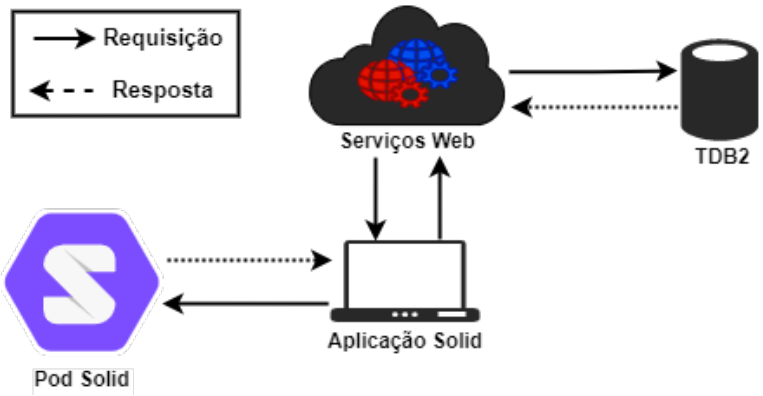


Figura 3.3 Estrutura desenvolvida para avaliação

A estrutura desenvolvida para avaliação do modelo é apresentada na Figura 3.3 e foi baseada na arquitetura de proveniência apresentada no Seção 3.1. Assim, a aplicação grava e lê dados no POD Solid e faz chamadas aos serviços de proveniência hospedados em

[5]<https://github.com/inrupt/solid-client-authn-js>
[6]<https://github.com/inrupt/solid-client-js>
[7]<https://github.com/inrupt/solid-common-vocab-rdf>

nuvem na Plataforma Heroku[8]. O serviço de captura armazena a proveniência coletada na *triplestore* TDB2. Já o serviço de rastreamento acessa a *triplestore* através de requisições HTTP e executa consultas SPARQL aos dados de proveniência capturados.

Para utilizar a aplicação, o usuário precisa estar autenticado em seu POD Solid. O *login* é efetuado por meio de autenticação do usuário em seu POD, que permite a aplicação ter acesso às informações públicas do usuário para apresentação na aplicação. Para a etapa de desenvolvimento da funcionalidade de *login*, foi utilizada a biblioteca *solid-client-authn*. A autenticação do usuário é apresentada na Figura 3.4 e segue o seguinte fluxo na aplicação: **Figura 3.4-Passo 1** - a aplicação inicia o processo enviando o usuário ao Provedor de Identidade Solid; **Figura 3.4-Passo 2** - usuário faz *login* no Provedor de Identidade Solid, utilizando seu nome de usuário e senha; **Figura 3.4-Passo 3** - o Provedor de Identidade Solid envia o usuário novamente para a aplicação e processa as informações para finalizar o processo de login e permitir acesso ao usuário.

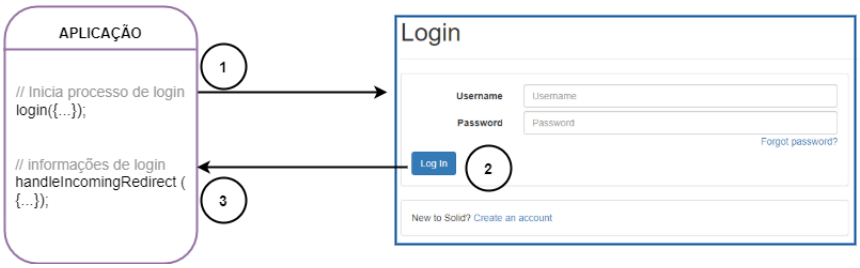


Figura 3.4 Fluxo de login da aplicação no Provedor de Identidade Solid. Adaptado da Documentação 2021 do Inrupt In.[10]

A Figura 3.5 apresenta a aplicação de rede social desenvolvida neste trabalho, em que é exibida a página de um usuário chamado Bob Henry contendo a postagem de uma publicação. No início da página encontramos seu nome e *WebId*. Ao lado dessas informações pessoais se encontra o botão de login que neste caso está desabilitado, indicando que o usuário está autenticado e pode fazer postagens em sua página.

[8] <https://www.heroku.com/>

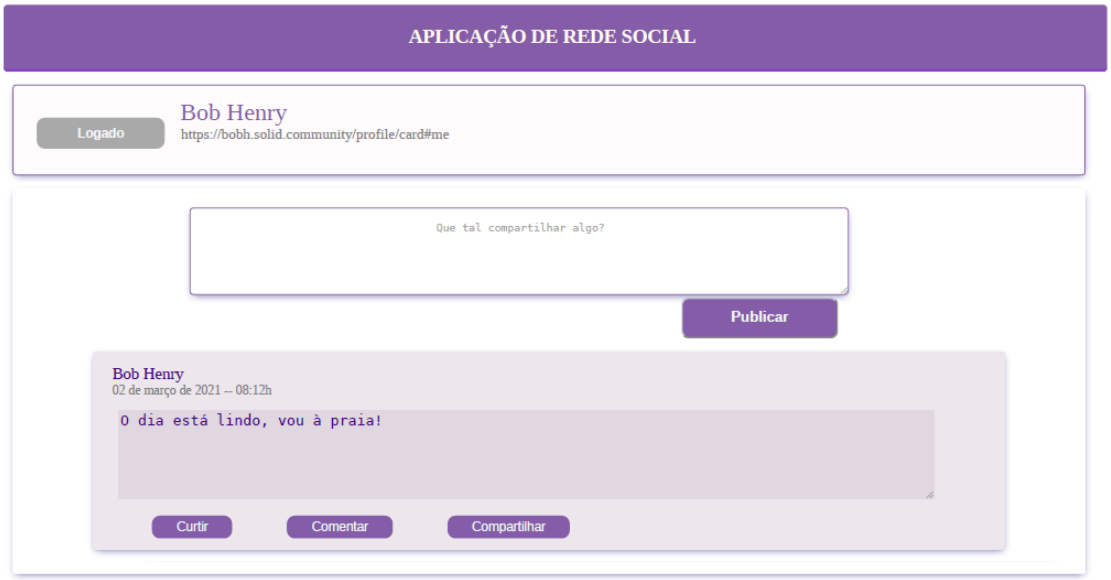


Figura 3.5 Aplicação de Rede Social - Página do usuário Bob Henry.

Bob efetuou uma publicação em sua página e o conteúdo da publicação foi persistido em seu POD Solid. Os dados das postagens são armazenados em um *container*, que são estruturas semelhantes a pastas em sistemas de arquivos, criado no diretório raiz (/) do POD Solid e nesta aplicação específica o diretório recebe o nome de redesocial. Para ilustrar essa estrutura, na Figura 3.6, é possível observar como são armazenadas as informações de postagens no POD de usuários.

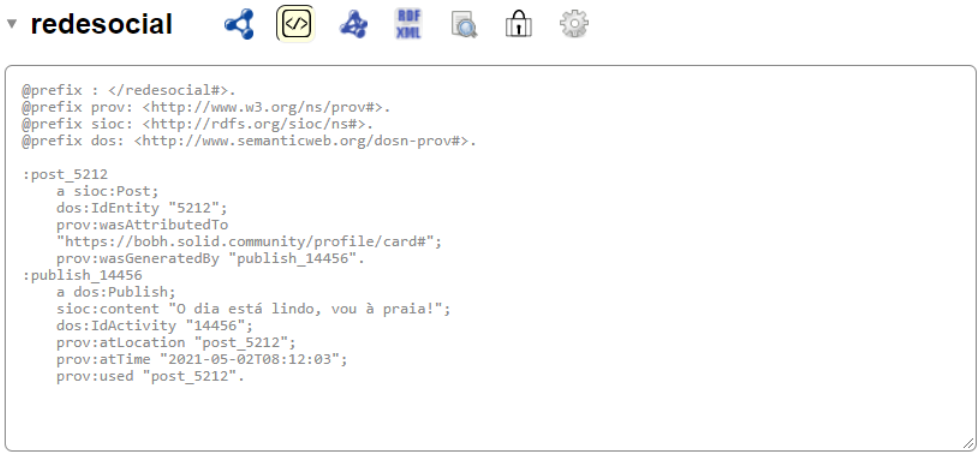


Figura 3.6 POD de um usuário com informações de publicação.

Para criar postagens é necessário utilizar funções disponíveis na biblioteca *solid-client* e *common-vocab-rdf*. A Figura 3.7 apresenta a função *createPublish()*, utilizada para

criação de publicações no POD. Nas linhas 3 e 4 da Figura 3.7 é utilizada a função *createThing()* para criar uma entidade de dados, onde são criados *post* e *publish*, pois cada atividade de publicação gera uma entidade de *post*.

```
1 function createPublish() {
2   for (let i = 0; i < postagens.length; i++) {
3    let post = createThing({ name: 'post_' + i })
4    let publish = createThing({ name: 'publish_' + j })
5    //informações de posts
6    post = addUrl(post, RDF.type, 'http://rdfs.org/sioc/ns#Post')
7    post = addStringNoLocale(post, PROV_O.wasGeneratedBy,
8        'publish_${i}')
9    post = addStringNoLocale(post, PROV_O.wasAttributedTo, webId)
10   post = addStringNoLocale(post,
11       http://www.semanticweb.org/dosn-prov#IdEntity',i)
12   // infomações de publish
13   publish = addUrl(publish,RDF.type,
14       'http://www.semanticweb.org/dosn-prov#Publish')
15   publish = addStringNoLocale(
16       publish,'http://rdfs.org/sioc/ns#content', postagens[i])
17   publish = addStringNoLocale(
18       publish,'http://www.semanticweb.org/dosn-prov#IdActivity',j)
19   publish = addStringNoLocale(publish, PROV_O.used, `post_${i})
20   publish = addStringNoLocale(publish, PROV_O.atLocation, 'post_${i})
21   publish = addStringNoLocale(publish, PROV_O.atTime, dataHora)
22
23   //adiciona post e publish ao Pod
24   mySolidDataset = setThing(mySolidDataset, post)
25   mySolidDataset = setThing(mySolidDataset, publish)
26   }
27 }
```

Figura 3.7 Função para criação de publicação.

Nas linhas 6 a 11 da Figura 3.7 são adicionadas as informações de postagem utilizadas na publicação. É utilizada a função addURL() para adicionar a propriedade *rdf:type* que define o *post* como um *sioc:Post*. Também é utilizada a função addStringNoLocale para adicionar as informações das propriedades *prov:wasGeneratedBy*, *prov:wasAttribuitedTo* e *dosn-prov:Entity*, sendo todas referentes ao *sioc:Post*.

Nas linhas 13 a 21 da Figura 3.7 são adicionadas as informações da publicação. Na linha 13 é definido o tipo da publicação utilizando a função addUrl() para definir *dosn-prov:Publish* como o tipo dessa publicação. As demais atribuições acontecem por meio da função addStringNoLocale, e são adicionadas as propriedades *dosn-prov:IdActivity*, *prov:used*, *prov:atLocation*, *sioc:content* e *proc:AtTime*. Finalmente, nas linhas 24 e 25 a função setThing() adiciona post e publish ao SolidDataset (mySolidDataset) do usuário, que é armazenado em seu POD.

Para efetuar qualquer interação na página de um usuário, é necessário inserir o *WebID* de quem está interagindo na página. Para efetuar um comentário, reação ou compartilha-

mento em postagens, o usuário insere seu *webID*, para ter sua identidade carregada junto com sua interação, possibilitando a sua atribuição de responsabilidade de postagem. A Figura 3.8 ilustra a página de um usuário (Eve Thomas), acessada por terceiros, onde é possível efetuar uma publicação. A publicação somente será criada e salva no POD após a inserção do *WebID* no campo: insira seu *WebID*.

Figura 3.8 Tela de interação de usuário.

3.6 TRABALHOS RELACIONADOS

Dado que a literatura não reporta muitos trabalhos relacionados à proveniência de dados em redes sociais e que não foram encontrados trabalhos relacionados à proveniência de dados em DOSN, os trabalhos discutidos nesta seção empregam a estratégia de utilização de modelos ontológicos para efetuar captura ou rastreamento de dados de proveniência em diferentes domínios.

Para Tas, Baeth e Aktas (2016), ainda existe a necessidade de serviços que capturem e gerenciem proveniência. Para chegar a essa conclusão os autores realizaram experimentos de desempenho e escalabilidade afim de investigar se sistemas de proveniência como PrevServ (GROTH; MILES; MOREAU, 2005), Karma (SIMMHAN; PLALE; GANNON, 2008) e Komadu (SURIARACHCHI; ZHOU; PLALE, 2015) são capazes de lidar com proveniência social de grande porte. Assim, para suprir essa necessidade, descoberta após os experimentos, os autores propõem uma arquitetura de serviços para ambientes de redes sociais. A estrutura da arquitetura proposta conta com uma API XML para requisitos de domínios de redes sociais e um padrão para semântica de representação dos dados. Fazem parte da composição da estrutura uma série de componentes como: (i) Mecanismo de Captura de Proveniência, (ii) Ouvinte de Proveniência e (iii) uma API de Análise de Proveniência.

Ainda segundo Tas, Baeth e Aktas (2016), o serviço de captura é utilizado para registrar todas as modificações nos dados, efetuando captura em tempo real e com possibilidade de verificação posterior em arquivos de log. Assim, os dados podem ser alterados em ambientes distribuídos de diferentes nós, utilizando um mecanismo de publicação-assinatura para registrar modificações. O serviço de proveniência proposto na arquitetura de Tas, Baeth e Aktas (2016) faz uso da ontologia PROV-O para representação dos dados, porém a especificação atual não deixa claro se a solução atende ao domínio específico de redes sociais descentralizadas. O modelo DOSN-PROV proposto neste tra-

balho de mestrado estende o modelo PROV-O para atender aos requisitos de redes sociais descentralizadas.

O trabalho de Arya, Abhishek e Deepak (2019) propõe um modelo capaz de determinar pegada digital organizacional, utilizando um modelo de proveniência e padrões de documentos do W3C Provenance para rastrear informações. Na proposta dos autores, as informações são coletadas e manipuladas como arquivos de fluxo e posteriormente são identificados agentes, entidades, atividades e relacionamentos com a extração dos seus metadados em notação Prov-N, gerando um rastro digital. A abordagem utilizando modelo Prov, para coleta de fluxos de informações organizacionais e criação de rastreamento de dados digitais das organizações para gestão de questões financeiras, se mostrou adequada para identificação de inadimplência, bem como suporte a tomada de decisões. A proposta de Arya, Abhishek e Deepak (2019) é utilizada para formar rastros e prover confiança para o domínio de proveniência em organizações, o que difere do modelo DOSN-PROV que tem como domínio a proveniência de redes sociais descentralizadas.

Os trabalhos de Trinh et al. (2017) e Meester et al. (2017) concentram seus esforços em capturar proveniência de *Linked Data*. A captura de dados realizada nesses trabalhos são executadas de forma automática e seus modelos de dados são baseados na ontologia de proveniência PROV-O. Os autores apresentam formas de integrar *Linked Data* de maneira transparente e reutilizável, por meio da geração automática de rastros de proveniência. Para avaliar a estrutura e o modelo desenvolvidos por Trinh et al. (2017), foi utilizada uma plataforma de *mashup* colaborativa, proporcionando um ambiente aberto de troca e integração de dados. Foi constatado que reutilizar a ontologia PROV-O contribui para satisfazer requisitos de proveniência como: disponibilidade e acessibilidade na *Web*. Desse modo, a criação automatizada de captura de proveniência como proposta por Trinh et al. (2017) e Meester et al. (2017) foram importantes para a definição deste trabalho de mestrado, pois sugerem uma forma de captura e rastreamento de proveniência individual e em etapas de processamento até a etapa final dos dados, agregando confiabilidade ao resultado final desse processo. Entretanto, Trinh et al. (2017) e Meester et al. (2017) também não tratam do domínio de redes sociais descentralizadas.

O trabalho de Riveni et al. (2019) investiga as vantagens e benefícios que a proveniência pode oferecer à computação social. Os autores propuseram um modelo de proveniência estabelecido em competências sociais que utiliza como base especificações da ontologia PROV-O. Riveni et al. (2019) realizaram experimentos para visualização de dados de proveniência e outro para avaliação do tempo de processamento de tamanhos diferentes de dados de proveniência. Os experimentos mostraram que os dados de proveniência podem apresentar visões gerais detalhadas dos sistemas sociais, o que permite, por sua vez, um gerenciamento eficiente, por conter informações extraídas dos dados de proveniência.

Riveni et al. (2019) realizaram o experimento de tempo de processamento usando a ferramenta Komadu desenvolvida no trabalho de Suriarachchi, Zhou e Plale (2015). Os resultados indicaram que processar grandes quantidades de dados com ferramentas de proveniência é uma tarefa eficiente. A partir dos resultados do trabalho de Riveni et al. (2019) é possível constatar a viabilidade de processar grande quantidade de informações de sistemas sociais. Apesar do trabalho não trazer soluções quanto a redes sociais descentralizadas, como é o propósito desta pesquisa de mestrado, o trabalho processa a

proveniência detalhada, contribuindo com a nossa proposta.

Diversas bibliotecas têm publicado seus dados em *Linked Data*, o que gera a necessidade de garantir a confiabilidade e qualidade desses dados. A proposta de McKenna, Debruyne e O'Sullivan (2019) garante a publicação de *interlinks* confiáveis, pois o modelo proposto adiciona proveniência à descrição e justificativa dos *links* criados, além de atender aos padrões de requisitos de metadados de bibliotecas. O trabalho apresenta um modelo de proveniência de processos utilizando *Linked Data* para o domínio de bibliotecas chamado NaiscProv. Esse modelo desenvolvido faz parte de uma abordagem de interligação geral e sua estrutura incorpora três grafos: (i) *Interlink Graph* contendo um conjunto de interligações, (ii) *Provenance Graph* incluindo um *prov:Bundle* com dados de origem de *interlinks* de grafos de interligação e (iii) *Relationship Graph* que representa um *prov:hasProvenance*, uma interligação entre um grafo de interligação e um grafo de proveniência. Os grafos de proveniência permitem obter informações de origem, criar e revisar os *interlinks*. Nessa abordagem de proveniência, McKenna, Debruyne e O'Sullivan (2019) utilizam o modelo PROV-O e as ontologias Dublin Core e FOAF, com objetivo de permitir descrições ricas de sujeito e entidades. Para realizar descrições de conjuntos de dados ou fontes de entidades, os autores utilizaram *VOID Vocabulary*. A abordagem de McKenna, Debruyne e O'Sullivan (2019) contribuiu para esta pesquisa de mestrado por utilizar proveniência em *Linked Data*, de modo a acrescentar confiabilidade aos dados. Entretanto, os autores trataram de proveniência específica para o domínio de biblioteca, ou seja, não levaram em consideração os requisitos de redes sociais descentralizadas.

A Tabela 3.2 apresenta um comparativo entre a abordagem proposta neste trabalho de mestrado e os trabalhos relacionados discutidos anteriormente nesta seção. Entre os critérios de comparação utilizados estão:

- Domínio: verifica o domínio de uso da proveniência;
- Modelo de Proveniência: verifica qual modelo específico de proveniência foi utilizado;
- Captura: verifica se o trabalho propõe captura de dados de proveniência;
- Rastreio: verifica se o trabalho propõe rastreamento de dados de proveniência;
- Implementa Solução: verifica se o trabalho implementa a solução proposta.

A Tabela 3.2 faz um comparativo dos trabalhos relacionados e através de seus critérios fica evidente que existem modelos de proveniência para diferentes domínios e que garantem captura e rastreamento de proveniência de dados. Contudo, diferente das abordagens propostas pelos trabalhos relacionados apresentados, a abordagem de pesquisa desta dissertação concentra esforços na construção de um modelo ontológico específico para DOSNs e propõe serviços para suporte a captura e rastreamento de proveniência de dados em DOSNs.

Tabela 3.2 Comparativo de Trabalhos Relacionados.

Autores	Proposta	Domínio	Modelo de Proveniência	Captura	Rastreio	Implementa Solução
(TAS; BAETH; AKTAS, 2016)	Arquitetura para gerenciamento de proveniência social	Rede social	PROV-O	SIM	SIM	NÃO
(TRINH et al., 2017)	Modelo de captura e integração de Linked Data	Genérico	PROV-O	SIM	SIM	SIM
(MEESTER et al., 2017)	Mecanismo de captura automática de proveniência	Genérico	PROV-O	SIM	NÃO	SIM
(RIVENI et al., 2019)	Modelo de proveniência para definir competências sociais	Computação social	PROV-O	SIM	SIM	SIM
(ARYA; ABHISHEK; DEEPAK, 2019)	Modelo de proveniência para determinar pegada digital organizacional	Organizações e empresas	PROV-N	NÃO	SIM	SIM
(MCKENNA; DEBRUYNE; O'SULLIVAN, 2019)	Modelo de proveniência para biblioteca	Biblioteca	PROV-O	SIM	NÃO	NÃO
Abordagem Proposta	**Modelo e serviços de suporte a proveniência**	**DOSNs**	**PROV-O**	**SIM**	**SIM**	**SIM**

3.7 CONSIDERAÇÕES FINAIS

Neste capítulo foi detalhada a Arquitetura de Proveniência, que conta com o modelo DOSN-PROV, baseado nos Requisitos de Proveniência e os Serviços de Captura e Rastreamento de proveniência. O capítulo também apresentou a estrutura criada para dar suporte à avaliação do modelo DOSN-PROV e a implementação da aplicação Solid desenvolvida. Por fim, foram apresentados trabalhos relacionados a este trabalho de mestrado.

Capítulo 4

AVALIAÇÃO E RESULTADOS

Este capítulo apresenta a avaliação do modelo DOSN-PROV e dos serviços de captura e rastreamento de proveniência. O modelo DOSN-PROV foi avaliado de duas formas: primeiro verificamos a adequação do modelo aos requisitos de proveniência; posteriormente validamos a ontologia quanto a possíveis erros, falhas e inconsistências. Em relação aos serviços, os mesmos foram avaliados por meio de uma avaliação de desempenho aplicada a cada um dos serviços com a intenção de investigar seus comportamentos quando submetidos a cargas diferentes de usuários simultâneos e variação na quantidades de triplas.

Na Seção 4.1 é apresentada a verificação do modelo a partir de casos de aplicação de exemplo e consultas SPARQL ao modelo. Na Seção 4.2 é apresentada a validação do modelo quando submetido à ferramenta de validação de ontologias. Na Seção 4.3 é executada a avaliação de desempenho, com a finalidade de serem avaliados os serviços de proveniência quanto à variável alvo tempo de resposta. No capítulo 4.4 os resultados das avaliações são apresentados e, por fim, as considerações finais do capítulo na Seção 4.5.

4.1 VERIFICAÇÃO DO MODELO

Com objetivo de testar a nossa hipótese, de que um modelo de proveniência específico, baseado em PROV-O pode garantir uma estrutura capaz de coletar e rastrear dados de proveniência em DOSNs de forma eficaz, implementamos o modelo DOSN-PROV e a arquitetura descritos no Capítulo 3.

Nesta etapa de avaliação, verificamos a conformidade do modelo ontológico com os requisitos de proveniência para o domínio de DOSNs, utilizando a estrutura da implementação apresentada no Capítulo 3. Essa estrutura é composta por uma aplicação Solid que, ao efetuar postagens na aplicação, persiste os dados no POD Solid do usuário e faz chamadas ao serviço de coleta de proveniência, para que a proveniência das postagens seja coletada e armazenada na triplestore de proveniência TDB2.

Os modelos de proveniência devem abordar os aspectos-chave de gerenciamento, uso e conteúdo, pois, segundo Groth et al. (2012), estes são adequados para qualificar casos de

uso comuns, sendo a estrutura usada para avaliar modelos e sistemas de softwares relacionados a proveniência. Portanto, avaliamos nossos resultados na etapa de verificação tendo como referência os requisitos de proveniência relacionados ao conteúdo, gerenciamento e uso apresentados na Tabela 3.1.

Para verificação do modelo, foram modelados casos de aplicação, cada caso representando algum tipo de postagem de DOSNs. Essas postagens terão seus dados de proveniência modelados, capturados e rastreados. Os casos de aplicação foram criados artificialmente, simulando interações de usuários reais em DOSNs. Essa escolha foi motivada pela falta de dados de DOSNs disponíveis, abertos e em formato *Linked Data*, tal como é proposto neste trabalho de mestrado. Os casos de aplicação contam com as seguintes postagens: publicação, comentário, compartilhamento e reação.

4.1.1 Casos de Aplicação de DOSNs

Os casos de aplicação desenvolvidos para verificação contam com quatro usuários que interagem em uma aplicação de DOSN. Esses usuários possuem *WebIDs*, como identificadores únicos, e PODs para armazenamento de seus dados. Bob é o autor da publicação inicial desse caso de aplicação, onde seu *post* será comentado, compartilhado e reagido por outros usuários.

A Figura 4.1 modela todos os casos de aplicação desenvolvidos, onde é possível identificar agentes, atividades e entidades envolvidas na verificação. Nessa modelagem da Figura 4.1, os agentes são representados por losangos laranjas, atividades como retângulos azuis e entidades como círculos amarelos. As atividades têm seu marco temporal definido por retângulos cinza. As setas em rosa determinam propriedades de atribuição de responsabilidade. A seguir são apresentados e discutidos, separadamente, cada um dos casos de aplicação que integram esta fase da avaliação.

1. **Caso de Aplicação - Publicação (*Publish*)**

   A Figura 4.2 apresenta a modelagem da publicação *publish_14456*. Essa publicação gera a entidade *post_5212*, cujo relacionamento é estabelecido através da propriedade *prov:wasGeneratedBy*. Bob é o agente dessa publicação e tem suas responsabilidades sobre o *post_5212* e a *publish_14456* atribuídas respectivamente pelas propriedades *prov:wasAttributedTo* e *prov:wasAssociatedWith*. O marco temporal desta publicação é representado pela tripla (*publish_14456*, prov:atTime, "2021-05-02T08:12:03"^^xsd:dataTime).

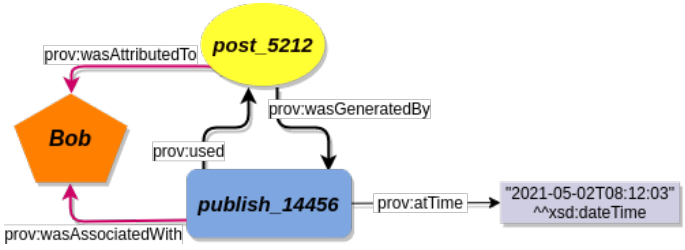


Figura 4.2 Caso de aplicação – publish_14456

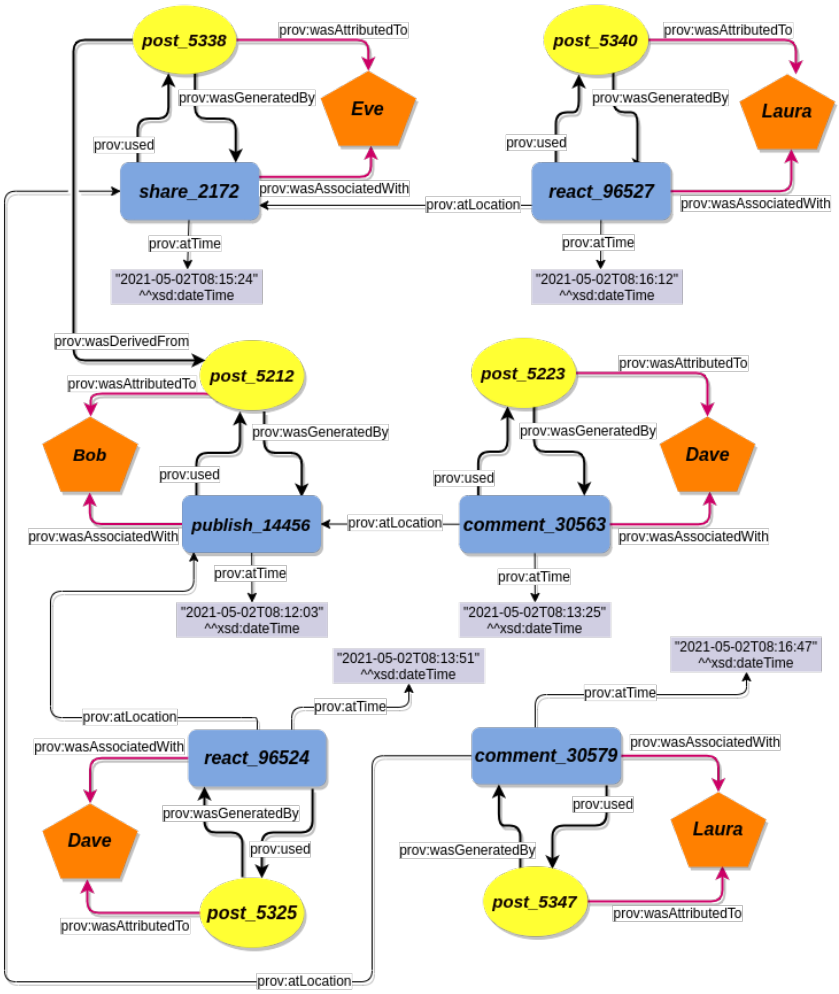


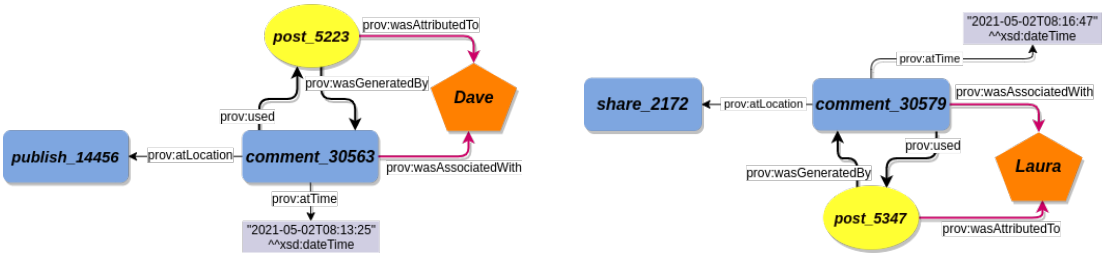
Figura 4.1 Modelagem Completa dos Casos de Aplicação.

2. **Caso de Aplicação - Comentário (*Comment*)**

A Figura 4.3 representa a modelagem de dois casos de aplicação referentes a comentários postados pelos agentes Dave e Laura em uma publicação e um compartilhamento. Na Figura 4.3(a), a tripla (*comment_30563, prov:used, post_5223*) indica que a entidade *post_5223* foi usada pela atividade *comment_30563*. Comentários são realizados em locais específicos como publicações e compartilhamentos e, nesse caso, o comentário de Dave foi efetuado na *publish_14456*, postada por Bob. As propriedades de atribuição apontam Dave como agente criador do *comentário_30563*.

A Figura 4.3(b) se refere à modelagem do comentário de Laura no compartilhamento *share_2172*. A tripla (*comment_30579, prov:atLocation, share_2172*) se refere ao local onde foi postado o comentário. A propriedade *prov:atTime* marca o tempo em que o comentário aconteceu e a propriedade *prov:wasAssociatedWith* define Laura

como agente responsável por esse comentário.



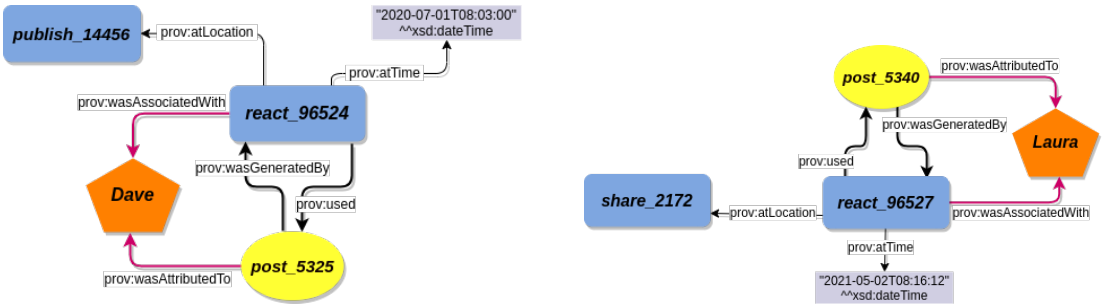
(a) comment_30563. (b) comment_30579.

Figura 4.3 Casos de aplicação – comments.

3. **Caso de Aplicação – Reação (*React*)**

A Figura 4.4 apresenta os casos de aplicação de reação. Na Figura 4.4(a) temos o primeiro caso de reação, em que Dave é o agente autor da postagem e podemos perceber sua responsabilidade através da tripla (*post_5325, prov:wasAttributedTo, Dave*). Dave, em sua postagem, interage com a publicação *publish_14456*, sendo este o local em que Dave posta sua reação. A tripla (*react_96524, prov:atLocation, publish_14456*) demonstra o local de publicação da reação de Dave.

A Figura 4.4(b) representa a reação de Laura ao compartilhamento *share_2172*. A tripla (*react_96527, prov:atLocation, share_2172*) modela a reação de Laura ao compartilhamento. Laura posta sua reação no dia 02/05/2021 às 08:16:12 horas. Podemos ver essa modelagem de data e hora de postagem através da tripla (*react_96527, prov:atTime, "2021-05-02T08:16:12"*^^xsd:dateTime).



(a) react_96523. (b) react_96527.

Figura 4.4 Casos de aplicação – reacts.

4. **Caso de Aplicação – Compartilhamento (Share)**

   A Figura 4.5 representa a modelagem do cenário de compartilhamento de uma publicação. Como agente do compartilhamento temos Eve, conforme descreve a tripla (*share_2172, prov:wasAssociatedWith*, Eve). A propriedade *prov:wasDerivedFrom* determina que a postagem *post_5338* é uma derivação da postagem *post_5212*, sendo esta a propriedade que caracteriza um compartilhamento. A atividade de compartilhamento somente pode existir a partir da criação da sua atividade de derivação, que, nesse caso, é a publicação *publish_14456* publicada por Bob e que está implícita nessa modelagem por ser a atividade contida no *post_5212*.

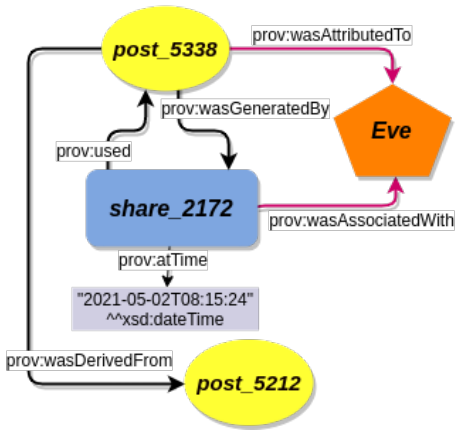


Figura 4.5 Caso de aplicação – share_2172

A seguir é representada em formato *turtle* a modelagem dos casos de aplicação desenvolvidos. São representadas as triplas com as informações de perfil de um usuário (Bob Henry) e as triplas de interação da modelagem completa, conforme a Figura 4.1.

```
1  ..............................CASOS DE APLICAÇÃO..............................
2  @prefix dosn-prov: <http://www.semanticweb.org/dosn-prov#> .
3  @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
4  @prefix con: <http://www.w3.org/2000/10/swap/pim/contact#> .
5  @prefix vcard: <http://www.w3.org/2006/vcard/ns#> .
6  @prefix prov: <http://www.w3.org/ns/prov#> .
7  @prefix sioc: <http://rdfs.org/sioc/ns#> .
8
9  Identificação do usuário Bob Henry
10 <https://bobh.solid.community/profile/card#> rdf:type owl:
       NamedIndividual ,
11          foaf:Person ;
12      vcard:hasAddress dosn-prov:address_4571 ;
13      foaf:made dosn-prov:PersonalProfileDocument_2587 ;
14      con:preferredURI "https://bobh.solid.community/profile/card#me"
           ^^xsd:anyURI ;
15      vcard:hasPhoto "image_bob.jpeg"^^xsd:anyURI ;
```

```
16      vcard:role "Estudante"^^xsd:string ;
17      foaf:mbox "mailto:bobhenry@email.com" ;
18      foaf:name "Bob Henry"^^xsd:string ;
19      rdfs:label "Bob" .
20
21  Postagem - post_5212
22  dosn-prov:post_5212 rdf:type owl:NamedIndividual ,
23          sioc:Post ;
24      prov:wasAttributedTo <https://bobh.solid.community/profile/card
            #> ;
25      prov:wasGeneratedBy dosn-prov:publish_14456 ;
26      dosn-prov:idEntity "5212"^^xsd:int .
27
28  Publicação - gerada pelo post_5212
29  dosn-prov:publish_14456 rdf:type owl:NamedIndividual ,
30          dosn-prov:Publish ;
31      dosn-prov:idActivity "14456"^^xsd:int ;
32      prov:used dosn-prov:post_5212 ;
33      prov:wasAssociatedWith <https://bobh.solid.community/profile/
            card\#me> ;
34      sioc:content "O dia está lindo, vou à praia!"^^rdfs:Literal ;
35      prov:atLocation dosn-prov:post_5212 ;
36      prov:atTime "2021-05-02T08:12:03"^^xsd:dateTime .
37
38  Comentário - gerado pelo post_5223
39  dosn-prov:comment_30563 rdf:type owl:NamedIndividual ,
40          dosn-prov:Comment ;
41      prov:atLocation dosn-prov:publish_14456 ;
42      prov:used dosn-prov:post_5223 ;
43      prov:wasAssociatedWith <https://dave.miller.org/profile/#me> ;
44      sioc:content "Espero que o sol continue lindo, Bob!"^^rdfs:
            Literal ;
45      dosn-prov:idActivity "30563"^^xsd:int ;
46      prov:atTime "2021-05-02T08:13:25"^^xsd:dateTime .
47
48  Reação - gerada pelo post_5325
49  dosn-prov:react_96524 rdf:type owl:NamedIndividual ,
50          dosn-prov:React ;
51      prov:atLocation dosn-prov:publish_14456 ;
52      prov:used dosn-prov:post_5325 ;
53      prov:wasAssociatedWith <https://dave.miller.org/profile/#me> ;
54      sioc:likes "Like"^^rdfs:Literal ;
55      dosn-prov:idActivity "96524"^^xsd:int ;
56      prov:atTime "2021-05-02T08:13:51"^^xsd:dateTime .
57
58  Compartilhamento - gerado pelo post_5338
59  dosn-prov:share_2172 rdf:type owl:NamedIndividual ,
60          dosn-prov:Share ;
61      prov:atLocation dosn-prov:post_5338 ;
62      prov:used dosn-prov:post_5338 ;
63      prov:wasAssociatedWith <https://solid.eve\_thomas.net/profile/
            card#me> ;
64      sioc:content "Sol, praia e calor!"^^rdfs:Literal ;
```

```
65      dosn-prov:idActivity "2172"^^xsd:int ;
66      prov:atTime "2021-05-02T08:15:24"^^xsd:dateTime .
67
68  Reação - gerada pelo post_5340
69  dosn-prov:react_96527 rdf:type owl:NamedIndividual ,
70          dosn-prov:React ;
71      prov:atLocation dosn-prov:share_2172 ;
72      prov:used dosn-prov:post_5340 ;
73      prov:wasAssociatedWith <https://solid.laurat.net/profile/card#me
            > ;
74      sioc:likes "Like"^^rdfs:Literal ;
75      dosn-prov:idActivity "96527"^^xsd:int ;
76      prov:atTime "2021-05-02T08:16:12"^^xsd:dateTime .
77
78  Comentário - gerado pelo post_5347
79  dosn-prov:comment_30579 rdf:type owl:NamedIndividual ,
80          dosn-prov:Comment ;
81      prov:atLocation dosn-prov:share_2172 ;
82      prov:used dosn-prov:post_5347 ;
83      prov:wasAssociatedWith <https://solid.laurat.net/profile/card#me
            > ;
84      sioc:content "Boa ideia, Eve."^^rdfs:Literal ;
85      dosn-prov:idActivity "30579"^^xsd:int ;
86      prov:atTime "2021-05-02T08:16:47"^^xsd:dateTime .
87  ..............................................................................
```

A representação dos casos de aplicação modelam cenários de postagens em uma aplicação de rede social, criando uma estrutura de DOSN, em que agentes podem realizar postagens. As informações dessas postagens são armazenadas em PODs, gerando autonomia sobre as informações, já os dados de proveniência são persistidos em *triplestores*. Nas linhas de 10 a 19 são descritas informações de perfil do usuário que inicia todo caso de aplicação, Bob Henry.

Nas linhas 22 a 26 a postagem *post_5212* é criada. Nas linhas 29 a 36 a publicação *publish_5212* é efetuada, contendo todas as propriedades fundamentais para realizar uma publicação. Vale ressaltar, que na linha 35 é definido o local da publicação, e quando as propriedades *prov:atLocation* e *prov:used* são iguais, significa que a publicação ocorreu na página do próprio agente.

As linhas 39 a 46 mostram a criação de um comentário associado ao usuário com webID <https://dave.miller.org/profile/#me> na publicação *publish_14456*, com conteúdo: Espero que o sol continue lindo, Bob!. As linhas 49 a 56 e 69 a 76 modelam reações de agentes à publicação *publish_14456* e ao compartilhamento *share_2172* respectivamente.

As linhas 59 a 66 modelam um compartilhamento associado ao webID <https://solid.-eve_thomas.net/profile/card#me>. Eve faz o compartilhamento em sua página de perfil, podemos concluir isso por meio das propriedades *prov:atLocation* e *prov:used* com valores iguais. A postagem que deu origem ao compartilhamento é definida pela propriedade *prov:wasDerivedFrom*, que é modelada no *post_5338*.

As linhas 79 a 86 modelam a criação do comentário associado ao usuário de webID <https://solid.laurat.net/profile/card#me>. A tripla (*dosn-prov:comment_30579, prov:atLocation, dosn-prov:share_2172*) indica que o comentário foi realizado no compartilhamento *share_2172*.

4.1.2 Recuperação de Informações

Com o objetivo de recuperar as informações coletadas pelo serviço de captura foram aplicadas consultas *SPARQL* na triplestore através do serviço de rastreamento. O propósito dessas consultas é verificar a conformidade do modelo DOSN-PROV aos requisitos de proveniência. As consultas e respostas a essas consultas são apresentadas a seguir:

1. Dado determinado nome de usuário (Bob Henry), determinar seus dados de perfil (webId, email, endereco, profissao e foto).

```
1  ..........................  Consulta 1 ..............................
2  PREFIX con: <http://www.w3.org/2000/10/swap/pim/contact#>
3  PREFIX vcard: <http://www.w3.org/2006/vcard/ns#>
4  PREFIX foaf: <http://xmlns.com/foaf/0.1/>
5
6  SELECT ?nome ?webId ?email ?endereco ?profissao ?foto
7  WHERE {
8    ?pessoa a foaf:Person;
9            foaf:name ?nome;
10           con:preferredURI ?webId ;
11           foaf:mbox ?email;
12           vcard:hasAddress ?endereco;
13           vcard:role ?profissao;
14           vcard:hasPhoto ?foto.
15 FILTER (CONTAINS(?nome, "Bob Henry"))
16 }
17 ..............................................................
```

Tabela 4.1 Resultado da Consulta 1

Campo	Resultado
nome	"Bob Henry"
webId	"https://bobh.solid.community/profile/card#me
email	"mailto:bobhenry@email.com"
endereco	dosn-prov:address_4571
profissao	"Estudante"
foto	"image_bob.jpeg"xsd:anyURI

2. A partir de determinada publicação (publish_14456), determinar dados referentes à publicação (conteudo, postagem, dataHora, WebIdAutor e nomeAutor).

```
1  ........................... Consulta 2 ............................
2  PREFIX dosn-prov: <http://www.semanticweb.org/dosn-prov#>
3  PREFIX prov: <http://www.w3.org/ns/prov#>
4  PREFIX foaf: <http://xmlns.com/foaf/0.1/>
5
6  SELECT ?conteudo ?postagem ?dataHora ?webIdAutor
7      ?nomeAutor
8  WHERE {
9    dosn-prov:publish_14456 sioc:content ?conteudo ;
10   prov:used ?postagem ;
11   prov:atTime ?dataHora ;
12   prov:wasAssociatedWith ?webidAutor.
13   ?autor foaf:name ?nomeAutor
14 }
15 ................................................................
```

Tabela 4.2 Resultado da Consulta 2

Campo	Resultado
conteudo	"O dia está lindo, vou à praia!"
postagem	dosn-prov:post_5212
dataHora	"2021-05-02T08:12:03"xsd:dateTime
webIdAutor	<https://bobh.solid.community/profile/card#me>
nomeAutor	"Bob Henry"

3. A partir de determinado identificador de usuário (WebID (https://solid.laurat.net /profile/card#me)), identificar as postagens atribuídas ao autor e identificar as atividades executadas (*publish, comment, react* ou *share*).

```
1  ............................ Consulta 3 ............................
2  PREFIX dosn-prov: <http://www.semanticweb.org/dosn-prov#>
3  PREFIX prov: <http://www.w3.org/ns/prov#>
4
5  SELECT ?pessoa ?post ?atividade
6  WHERE {
7      ?post a sioc:Post.
8            ?post prov:wasAttributedTo ?pessoa .
9            ?post prov:wasGeneratedBy ?atividade
10 FILTER (?pessoa = <https://solid.laurat.net/profile/card#me>)
11 }
12 ................................................................
```

4. A partir de determinado compartilhamento (*share_2172*), retornar dados do post de derivação (autoCompartilhamento, autorPostOriginal, atividadeOriginal, conteudo e dataHora).

Tabela 4.3 Resultado da Consulta 3

Campo	Resultado
pessoa	<https://solid.laurat.net/profile/card#me>
post	dosn-prov:post_5340
	dosn-prov:post_5347
atividade	dosn-prov:react_96527
	dosn-prov:comment_30579

```
1  ......................... Consulta 4 ............................
2  PREFIX dosn-prov: <http://www.semanticweb.org/dosn-prov#>
3  PREFIX prov: <http://www.w3.org/ns/prov#>
4  PREFIX sioc: <http://rdfs.org/sioc/ns#>
5
6  SELECT ?autorCompartilhamento ?autorPostOriginal
7      ?atividadeOriginal ?conteudo ?dataHora
8  WHERE {
9    dosn-prov:share_1354 prov:wasAssociatedWith
10     ?autorCompartilhamento.
11   dosn-prov:share_1354 prov:used
12     ?postCompartilhamento.
13   ?postCompartilhamento prov:wasDerivedFrom
14     ?postOriginal.
15   ?postOriginal prov:wasAttributedTo
16     ?autorPostOriginal.
17   ?postOriginal prov:wasGeneratedBy
18     ?atividadeOriginal.
19   ?atividadeOriginal sioc:content ?conteudo.
20   ?atividadeOriginal prov:atTime ?dataHora.
21 }
22 ..........................................................
```

Tabela 4.4 Resultado da Consulta 4

Campo	Resultado
autorCompartilhamento	<https://solid.eve_thomas.net/profile/card#me>
autorPostOriginal	<https://bobh.solid.community/profile/card#me>
atividadeOriginal	dosn-prov:publish_14456
conteudo	O dia está lindo, vou à praia!"
dataHora	"2021-05-02T08:12:03"xsd:dateTime

4.2 VALIDAÇÃO DO MODELO

Para a validação do modelo ontológico DOSN-PROV, foi utilizada a ferramenta OOPS![1] (do inglês, OntOlogy Pitfall Scanner). OOPS! é uma ferramenta Web que identifica

[1] <http://oops.linkeddata.es/index.jsp>

armadilhas durante o desenvolvimento da ontologia. Além disso, a ferramenta é capaz de identificar erros e inconsistências de acordo com as dimensões estrutural, funcional e perfil de usabilidade, além de avaliar os critérios de concisão, integridade e consistência (POVEDA-VILLALÓN; GÓMEZ-PÉREZ; SUÁREZ-FIGUEROA, 2014). A ferramenta OOPS! classifica as armadilhas encontradas de acordo com os seguintes níveis de prioridade:

- crítico – é crucial que essa armadilha seja reparada pois, afeta a consistência e raciocínio da ontologia.
- importante – apesar de não ser uma armadilha crítica, é recomendado o reparo.
- menor – não é um fator de problema, mas sua correção torna a ontologia mais clara e organizada.

Além de sinalizar prioridade e informar a quantidade de ocorrências de cada armadilha, Poveda-Villalón, Suárez-Figueroa e Gómez-Pérez (2012) afirmam que a OOPS! também descreve de forma breve a armadilha ou erro e pode sugerir para alguns erros sua forma de correção. A OOPS! pode ser considerada uma ferramenta de avaliação flexível, pois algumas armadilhas encontradas podem permanecer na ontologia, seja por requisito de modelagem ou por definição do engenheiro de ontologia.

A ferramenta OOPS! desempenha um papel importante na tarefa de validação, suas funcionalidades são capazes de detectar erros e armadilhas muitas vezes não detectáveis pelos desenvolvedores de ontologia. A ferramenta garante que a ontologia seja avaliada quanto a aspectos que garantam consistência, integridade e conformidade com as melhores práticas no desenvolvimento de ontologias, além de oferecer sugestões de correção aos itens identificados.

A ontologia de proveniência DOSN-PROV foi submetida à ferramenta OOPS!, com propósito de validação de sua estrutura funcional, estrutural e perfil de usabilidade. Os resultados encontrados são apresentados na Tabela 4.6 (explicada posteriormente), onde é possível observar que a ontologia não apresentou armadilhas críticas, ou seja, armadilhas ou erros que afetam o raciocínio ou a utilização da ontologia. Também não foram encontradas armadilhas de dimensão funcional, que demonstram quão completa é a ontologia e sua adequação ao contexto de aplicação. Para os critérios de concisão e consistência não foram encontradas armadilhas ou erros. No entanto, OOPS! encontrou armadilhas de dimensão estrutural e perfil de usabilidade.

4.3 AVALIAÇÃO DOS SERVIÇOS DE PROVENIÊNCIA

Para implementação dos experimentos de avaliação dos serviços de proveniência, o modelo DOSN-PROV e os serviços de captura e rastreamento de proveniência foram hospedados em nuvem na plataforma como serviço (Plataform as a Service - PaaS) Heroku[2]. Os dados de proveniência coletados pelo serviço de captura foram armazenados na triplestore

[2]<https://www.heroku.com>

TDB2 do Apache Jena, hospedada na Google Cloud Plataform[3]. Para realização dos experimentos de avaliação dos serviços, utilizamos a ferramenta JMeter[4], originalmente desenvolvida para executar testes em aplicações Web. A máquina utilizada para execução do Apache JMeter foi um computador com 8GB de memória RAM, processador Intel Core i7 de 2.4Ghz e sistema operacional Windows 10 Home Single Language.

Foram realizados experimentos com objetivo de avaliar a capacidade de resposta dos serviços. Desse modo, foram selecionados fatores e níveis considerados relevantes para o domínio. O experimento analisou a interação entre os fatores e níveis, sendo replicados 10 vezes, obtendo a média das replicações e possuindo como métrica alvo a variável tempo de resposta das requisições.

Para o experimento do serviço de captura, o fator quantidade de usuários se limitou a 700 usuários simultâneos, por gerar uma grande quantidade de erros de requisições a partir deste valor. A opção de 100 triplas como a menor quantidade é baseada no caso de representação utilizada na verificação do modelo da Seção 4.1, em que ocorre apenas um caso de publicação, comentário, reação, compartilhamento e identificadores de agentes. Portanto, a configuração do experimento do serviço de coleta é composta por: quantidade de usuários (200, 500 e 700) e quantidade de triplas (100).

Para o experimento do serviço de rastreamento, o fator quantidade de usuários se limitou a 1.000 usuários simultâneos, por gerar uma grande quantidade de erros de requisições a partir deste valor. Para o experimento do serviço de rastreamento a configuração utilizada foi: quantidade de usuários (200, 500, 700, 900 e 1.000) e quantidade de triplas (100 e 10.000).

4.4 RESULTADOS

Os resultados obtidos após a verificação do modelo apresentado na Seção 4.1 demonstram, por meio da recuperação de informações através dos resultados das consultas SPARQL, a conformidade do modelo DOSN-PROV com os requisitos elicitados no Capítulo 3. A Tabela 4.5 apresenta a conformidade do modelo DOSN-PROV aos requisitos de proveniência, em que a coluna Justificativa mostra porque a consulta exposta na coluna Consulta atende ou não ao requisito da coluna Requisito para que ele seja marcado no *checklist* da coluna Verificação.

Os resultados retornados pelas consultas e suas respostas positivas aos requisitos de proveniência demostram a capacidade do modelo DOSN-PROV em lidar com dados de proveniência de DOSNs. Por meio dos resultados da coluna de Verificação da Tabela 4.5, percebemos que o modelo atende a todos os requisitos de proveniência elicitados no Capítulo 3, sendo capaz de efetuar captura e rastreamento de dados de proveniência em uma estrutura de DOSN.

Resultados obtidos após a validação do modelo efetuada por meio da ferramenta OOPS!, apresentada na Seção 4.2, sinalizaram apenas três armadilhas e uma sugestão, como mostra a Tabela 4.6. Foram encontradas 7 ocorrências e uma sugestão de correção. A armadilha P13 é de nível menor, não sendo um fator problema para a ontologia. A

[3]<https://cloud.google.com/>
[4]<https://jmeter.apache.org/>

Requisito	Consulta	Justificativa	Verificação
R1	C4 e C3	A consulta C4 identifica a fonte de proveniência, recupera dados do compartilhamento share_2172 até sua fonte. A consulta C3 retorna as entidades dosn-prov:post_-5340 e dosn-prov_5347 atribuídas ao usuário de WebId <https://solid.laurat.net/profile/card#me>	✓
R2	C2 e C4	O resultado da consulta C2 atribui ao WebId <https://bobh.solid.community/profile/card#me> a responsbilidade pela criação da entidade publish_14456. Assim como, a resposta da consulta C4 atribui ao WebId <https://solid.eve_thomas.net/profile/card#me> a autoria do compartilhamento share_2172.	✓
R3	C2	A resposta da consulta C2 define que a atividade publish_14456 foi associada ao agente de WebId <ttps://bobh.solid.community/profile/card#me>.	✓
R4	C3	O resultado da consulta C3 retorna atividades que foram utilizadas para gerar postagens, neste caso, a atividade dosn-prov:react_96527 gerou o post_23402 e a atividade dosn-prov:comment_30579 gerou o post_5347.	✓
R5	C2 e C4	Nas consultas C2 e C4 são recuperadas informações de data e hora de criação de atividades, através da propriedade prov:atTime.	✓
R6	C4	Na consulta C4 é possível perceber a capacidade de recuperação de várias informações de compartilhamento, podendo rastrear uma atividade	✓
R7	C4	A consulta C4 recupera informações de proveniência do compartilhamento do share_2172, incluindo ator da postagem original, qual atividade original gerou a entidade post, seu conteúdo e a data e hora da atividade original.	✓
R8	C1,C2,C3 e C4	O requisito R8 é atendido por todas as consultas e casos de aplicação, pois o modelo atende as principais entidades e atividades que demandam proveniência em rede social.	✓
R9	C2, C3	A consulta C2 retorna a informação de que a entidade dosn-prov:post_5212 foi gerada pela atividade de publicação publish_15456. Em C3, a partir do WebId <https://solid.laurat.net/profile/card#me> foram recuperadas as entidades e posts gerados por este usuário.	✓
R10	C1 e C3	O resultado da consulta C1 retornou um usuário específico e a consulta C3 filtrou um usuário específico através de um WebId	✓

Tabela 4.5 Verificação de requisitos.

sugestão para esta armadilha é que propriedades como *prov:used* e *prov:WasGeneratedBy* sejam declaradas relacionamentos inversos, o que não se aplica às propriedades destacadas pela ferramenta. Todas as sugestões para esta armadilha não são aplicáveis para a ontologia em questão, tanto por se tratarem de relacionamentos que não fazem parte da ontologia reutilizada, quanto por questões de modelagem. A armadilha P22 é de nível menor e se justifica na ontologia pela presença de reuso de ontologias como FOAF, SIOC e vCARD, possuindo cada uma delas sua convenção de nomenclatura própria.

Armadilha	Casos	Prioridade	Descrição
P13	7 casos	menor	Essa armadilha aparece quando qualquer relacionamento (exceto aqueles que são definidos como propriedades simétricas usando owl: SymmetricProperty) não tem um relacionamento inverso (owl: inverseOf) definido na ontologia.
P22	Se aplica a toda ontologia	menor	Os elementos da ontologia não são nomeados seguindo a mesma convenção (por exemplo CamelCase ou uso de delimitadores como "−"ou "_").
P41	Se aplica a toda ontologia	importante	Os metadados da ontologia omitem informações sobre a licença que se aplica à ontologia.
Sugestão	1 caso	-	Os axiomas de domínio e intervalo são iguais para cada uma das seguintes propriedades de objeto.

Tabela 4.6 Armadilhas encontradas na ontologia DOSN-PROV

A armadilha P41 indica que nenhuma licença foi declarada para essa ontologia, apesar de ser uma armadilha de prioridade importante e se aplicar a ontologia de maneira geral, ela não implica na validação da ontologia. A sugestão feita pela ferramenta consiste em definir transitividade ou simetria a propriedades que possuem valores de domain e range iguais. A propriedade *prov:wasDerivedFrom*, sinalizada pela ferramenta, não pode ser considerada simétrica ou transitiva pois caracteriza uma relação de compartilhamento de um *sioc:Post*, sua inversa ou simétrica torna o relacionamento inadequado, pois existe diferença de postagem compartilhada na página do usuário autor do compartilhamento ou na página de um amigo.

As armadilhas encontradas não necessitam de reparo e não afetam criticamente ou invalidam a ontologia. Portanto, a partir dos resultados obtidos pela ferramenta OOPS! e apresentados na Tabela 4.6, a ontologia se mostrou consistente, por não apresentar qualquer armadilha relacionada a essa categoria, além de não apresentar armadilhas para a categoria funcional. Para as dimensões estrutural e perfil de usabilidade, foram encontradas armadilhas, porém nenhuma delas com prioridade crítica.

Os resultados obtidos após a execução dos experimentos de avaliação dos serviços apresentados na Seção 4.3 são referentes ao tempo de resposta dos serviços em relação à quantidade de requisições simultâneas por usuário e a quantidade de triplas carregadas no servidor de proveniência. A Figura 4.6 demonstra que existe uma relação positiva entre a quantidade de usuários e o tempo de resposta para o serviço de coleta, pois a medida que a quantidade de usuários aumenta, o tempo de resposta do serviço também aumenta. Esse comportamento é observado em todos os níveis do fator quantidade de usuários e foi verificada uma relação linear positiva, em que quantidades maiores de usuários executando requisições ao serviço de coleta tendem a corresponder a um maior tempo gasto pelo serviço para coletar dados de DOSNs e persistir em *triplestores*. Quando ultrapassada a quantidade de 700 usuários simultâneos o serviço passou a apresentar erros,

demostrando assim o limite de usuários suportado pelo serviço para este experimento.

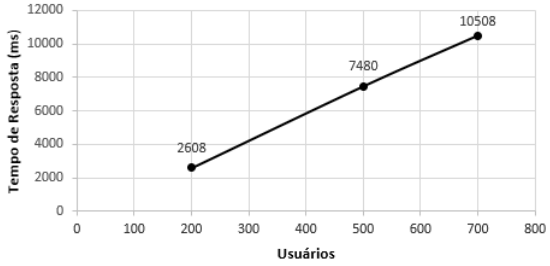


Figura 4.6 Tempo de resposta do serviço de coleta

A Figura 4.7 apresenta o gráfico do serviço de rastreamento, onde é possível observar a média dos tempos de resposta obtidos durante as requisições simultâneas de 200, 400, 600 e 1000 usuários, separados por quantidades de 100 e 10.000 triplas. É possível notar que no início das requisições com 200 usuários, os tempos de resposta relacionados a quantidades de triplas (100 e 10.000) obtiveram resultados próximos, evidenciando que o serviço tem resultados positivos para essa quantidade de usuários, mesmo variando a quantidade de triplas já carregadas no servidor. A diferença dos tempos de resposta para os experimentos de 400, 500, 700 e 900 usuários não são consideradas significativas, dado que a diferença na quantidade de triplas é 100 vezes maior de um caso para outro.

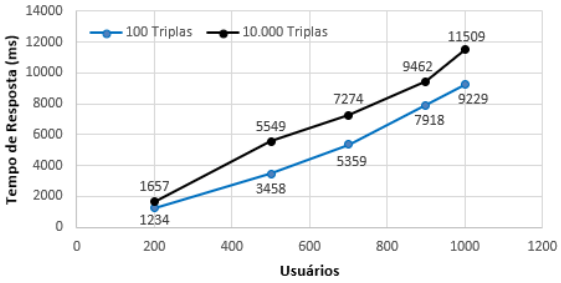


Figura 4.7 Tempo de resposta do serviço de rastreamento por número de triplas.

Os resultados mostram que mesmo submetidos a grandes quantidades de requisições e grande carga de triplas, os serviços de proveniência conseguiram responder de forma positiva aos experimentos, o que demonstra que os serviços podem suportar as demandas de proveniência em DOSNs, visto que nessas redes trafegam quantidades menores de dados que as atuais OSNs.

A limitação da quantidade de usuários (700 ou 1000) nos testes executados é um problema relativo ao uso de Serviços Web, que são serviços com arquitetura monolítica[5].

[5]Na arquitetura monolítica, todo o serviço é executado em uma única máquina.

Uma solução natural para escalar esse número de usuários seria a utilização da arquitetura de microserviços, que permite desenvolver uma aplicação como um conjunto de pequenos serviços distribuídos (assim como as DOSNs) em diversas máquinas. Entretanto, resolver o problema de escalabilidade de usuários não faz parte do escopo deste trabalho de mestrado.

4.5 CONSIDERAÇÕES FINAIS

Este capítulo apresentou em detalhes as avaliações do modelo e dos serviços de proveniência. O modelo passou por duas etapas de avaliação: verificação e validação. Já os serviços passaram por uma avaliação quanto à quantidade de triplas e usuários. Por fim, foram apresentados os resultados obtidos com as avaliações do modelo DOSN-PROV e os serviços de captura e rastreamento de proveniência.

Capítulo 5

CONCLUSÃO

As DOSNs surgiram da necessidade de seus utilizadores de adquirirem o poder sobre seus dados e têm aumentado o número de usuários ativos nos últimos anos. Porém, mesmo sendo uma alternativa às OSNs e seus problemas, as DOSNs também necessitam de soluções a problemas relacionados à proveniência de seus dados, dentre outras questões. A solução apresentada neste trabalho permite que dados das postagens e de usuários sejam capturados e rastreados em DOSNs, garantindo confiança e integridade aos dados que trafegam nessas redes. A solução é composta por um modelo de proveniência DOSN-PROV e serviços para captura e rastreio de dados de postagens e dados de usuários nas redes sociais descentralizadas.

Neste trabalho, apresentamos uma arquitetura de proveniência composta pelo modelo DOSN-PROV e pelos serviços de coleta e rastreamento de proveniência. O modelo DOSN-PROV é responsável por modelar a proveniência, sendo flexível e reutilizável. Os resultados obtidos após a avaliação do modelo evidenciam que o modelo DOSN-PROV atende aos requisitos de proveniência propostos para o domínio de DOSNs e indicam eficiência para os casos gerais de redes sociais. O modelo também se mostra consistente e claro após a sua validação por intermédio da ferramenta OOPS!, uma vez que não foram encontrados erros críticos ou que afetem sua qualidade.

Os resultados obtidos com a avaliação dos serviços de proveniência se mostram favoráveis para o domínio de DOSNs, pois evidenciam que não há diferenças significativas entre os tempos de respostas dos serviços quando variadas as quantidades de triplas e quantidades de usuários. Dessa forma, os serviços apresentaram tempos de resposta aceitáveis para DOSNs.

5.1 CONTRIBUIÇÕES

Esta dissertação apresentou um modelo de proveniência e serviços de suporte capazes de garantir proveniência de dados em DOSNs. As principais contribuições referentes a esta dissertação são resumidas a seguir:

- Um modelo ontológico de proveniência específico para DOSNs, o modelo DOSN-PROV;
- Uma arquitetura de proveniência para DOSNs;
- Seleção e definição de requisitos de proveniência para DOSNs;
- Um serviço de suporte à captura de proveniência de dados de DOSNs;
- Um serviço de suporte ao rastreamento de dados de proveniência em DOSNs.

5.2 LIMITAÇÕES E TRABALHOS FUTUROS

A arquitetura desenvolvida soluciona o problema de rastreabilidade e lacuna de desconfiança encontrada nas DOSNs. Porém essa solução enfraquece a descentralização proposta pelas DOSNs, visto que os dados de proveniência da Arquitetura de Proveniência proposta neste trabalho de mestrado são centralizados em uma *triplestore*. Neste sentido, a descentralização para solucionar os problemas de desconfiança dos dados das DOSNs, estão em oposição a rastreabilidade. Logo, enfraquecer a descentralização foi um ponto limitante para o alcance da estratégia de rastreabilidade e lacuna de desconfiança existente nas DOSNs.

Este trabalho apresentou um modelo ontológico e serviços de apoio à proveniência específicos para DOSNs. Portanto, as OSNs não fazem parte do escopo desta dissertação. Logo, esta lacuna abre caminho a um possível trabalho futuro em direção a estas redes, sugerindo adicionar proveniência a OSNs, propondo captura e rastreamento para dados de OSNs.

O nosso serviço de captura de proveniência não realiza filtragem ou seleção dos dados de proveniência que serão armazenado nas *triplestores*, o que pode gerar uma grande quantidade de dados armazenados sem propósito. Essa limitação gera uma lacuna para um possível trabalho futuro referente ao serviço de captura, com a proposta de desenvolvimento de um mecanismo de identificação de violações, notícias falsas ou possíveis transgressões em *posts* de DOSNs, com propósito de capturar e armazenar somente postagens em que o autor incorra em possíveis transgressões.

A falta de dados de DOSNs abertos e disponíveis em formato Linked Data gerou uma limitação em relação à avaliação do modelo, o que possibilita um trabalho futuro em direção ao aperfeiçoamento da aplicação de DOSN e sua implantação para usuários reais em uma escala controlada, como, por exemplo, uma rede social acadêmica, viabilizando a geração de bases de dados adequadas para avaliações.

REFERÊNCIAS BIBLIOGRÁFICAS

AGHAEI, S.; NEMATBAKHSH, M. A.; FARSANI, H. K. Evolution of the world wide web: From web 1.0 to web 4.0. *International Journal of Web & Semantic Technology*, Academy & Industry Research Collaboration Center (AIRCC), v. 3, n. 1, p. 1, 2012.

ARENAS, M.; GUTIERREZ, C.; PÉREZ, J. On the semantics of sparql. In: *Semantic Web Information Management*. [S.l.]: Springer, 2010. p. 281–307.

ARYA, S.; ABHISHEK, K.; DEEPAK, A. Organizational digital footprint for traceability, provenance approach. In: *Emerging Research in Computing, Information, Communication and Applications*. [S.l.]: Springer, 2019. p. 265–275.

BAHRI, L.; CARMINATI, B.; FERRARI, E. Decentralized privacy preserving services for online social networks. *Online Social Networks and Media*, Elsevier, v. 6, p. 18–25, 2018.

BARBIER, G.; FENG, Z.; GUNDECHA, P.; LIU, H. Provenance data in social media. *Synthesis Lectures on Data Mining and Knowledge Discovery*, Morgan & Claypool Publishers, v. 4, n. 1, p. 1–84, 2013.

BECKETT, D.; BERNERS-LEE, T.; PRUD'HOMMEAUX, E.; CAROTHERS, G. Rdf 1.1 turtle. *World Wide Web Consortium*, p. 18–31, 2014.

BEHAJJAME, K.; CHENEY, J.; CORSAR, D.; GARIJO, D.; SOILAND-REYES, S.; ZEDNIK, S.; ZHAO, J.; LEBO, T.; SAHOO, S.; MCGUINNESS, D. Prov-o: The prov ontology, w3c recommendation rec-prov-o-20130430. *World Wide Web Consortium (Oct. 2013). URL http://www. w3. org/TR/2013/REC-prov-o-20130430*, 2013.

BERNERS-LEE, T. Linked data. *Int. J. on Semantic Web and Information Systems*, v. 4, n. 2, 2006.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. et al. The semantic web. *Scientific american*, New York, NY, USA:, v. 284, n. 5, p. 28–37, 2001.

BIZER, C.; HEATH, T.; IDEHEN, K.; BERNERS-LEE, T. Linked data on the web (ldow2008). In: ACM. *Proceedings of the 17th international conference on World Wide Web*. [S.l.], 2008. p. 1265–1266.

BIZER, C.; LEHMANN, J.; KOBILAROV, G.; AUER, S.; BECKER, C.; CYGANIAK, R.; HELLMANN, S. Dbpedia-a crystallization point for the web of data. *Web Semantics: science, services and agents on the world wide web*, Elsevier, v. 7, n. 3, p. 154–165, 2009.

BOJARS, U.; BRESLIN, J. G.; BERRUETA, D.; BRICKLEY, D.; DECKER, S.; FERNÁNDEZ, S.; GÖRN, C.; HARTH, A.; HEATH, T.; IDEHEN, K. et al. Sioc core ontology specification. *WC Member*, 2010.

BRANK, J.; GROBELNIK, M.; MLADENIC, D. A survey of ontology evaluation techniques. In: CITESEER LJUBLJANA, SLOVENIA. *Proceedings of the conference on data mining and data warehouses (SiKDD 2005)*. [S.l.], 2005. p. 166–170.

BRICKLEY, D.; MILLER, L. *Foaf vocabulary specification 0.98, 2010*. [S.l.]: Accessed, 2010.

BRICKLEY, D.; MILLER, L. Foaf vocabulary specification 0.99, namespace document 14 january 2014-paddington edition. *Recuperado a partir de http://xmlns. com/foaf/spec*, 2014.

BUCHEGGER, S.; SCHIÖBERG, D.; VU, L.-H.; DATTA, A. Peerson: P2p social networking: early experiences and insights. In: ACM. *Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*. [S.l.], 2009. p. 46–52.

CHENEY, J.; CHONG, S.; FOSTER, N.; SELTZER, M.; VANSUMMEREN, S. Provenance: a future history. In: *Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications*. [S.l.: s.n.], 2009. p. 957–964.

CLOSA, G.; MASÓ, J.; PROSS, B.; PONS, X. W3c prov to describe provenance at the dataset, feature and attribute levels in a distributed environment. *Computers, Environment and Urban Systems*, Elsevier, v. 64, p. 103–117, 2017.

CRISTANI, M.; CUEL, R. A survey on ontology creation methodologies. *International Journal on Semantic Web and Information Systems (IJSWIS)*, IGI Global, v. 1, n. 2, p. 49–69, 2005.

CURCIN, V.; FAIRWEATHER, E.; DANGER, R.; CORRIGAN, D. Templates as a method for implementing data provenance in decision support systems. *Journal of biomedical informatics*, Elsevier, v. 65, p. 1–21, 2017.

DATTA, A.; BUCHEGGER, S.; VU, L.-H.; STRUFE, T.; RZADCA, K. Decentralized online social networks. In: *Handbook of Social Network Technologies and Applications*. [S.l.]: Springer, 2010. p. 349–378.

DEGBELO, A. A snapshot of ontology evaluation criteria and strategies. In: *Proceedings of the 13th International Conference on Semantic Systems*. [S.l.: s.n.], 2017. p. 1–8.

DING, L.; ZHOU, L.; FININ, T.; JOSHI, A. How the semantic web is being used: An analysis of foaf documents. In: IEEE. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*. [S.l.], 2005. p. 113c–113c.

DUONG, C. T.; NGUYEN, Q. V. H.; WANG, S.; STANTIC, B. Provenance-based rumor detection. In: SPRINGER. *Australasian Database Conference*. [S.l.], 2017. p. 125–137.

FURHT, B. *Handbook of social network technologies and applications*. [S.l.]: Springer Science & Business Media, 2010.

GANGEMI, A.; CATENACCI, C.; CIARAMITA, M.; LEHMANN, J. Modelling ontology evaluation and validation. In: SPRINGER. *European Semantic Web Conference*. [S.l.], 2006. p. 140–154.

GIL, Y.; CHENEY, J.; GROTH, P.; HARTIG, O.; MILES, S.; MOREAU, L.; SILVA, P. da. *Provenance XG final report, W3C provenance incubator group*. 2010.

GOLBECK, J.; ROTHSTEIN, M. Linking social networks on the web with foaf: A semantic web case study. In: *AAAI*. [S.l.: s.n.], 2008. v. 8, p. 1138–1143.

GROTH, P.; GIL, Y.; CHENEY, J.; MILES, S. Requirements for provenance on the web. *International Journal of Digital Curation*, v. 7, n. 1, p. 39–56, 2012.

GROTH, P.; MILES, S.; MOREAU, L. Preserv: Provenance recording for services. 2005.

GROTH, P.; MOREAU, L. *PROV-Overview: An Overview of the PROV Family of Documents*. 2013. <https://www.w3.org/TR/prov-overview/>. [Online; accessed 02-May-2019].

GRUBER, T. R. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, Elsevier, v. 43, n. 5-6, p. 907–928, 1995.

GUARINO, N. *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*. [S.l.]: IOS press, 1998.

GUARINO, N.; OBERLE, D.; STAAB, S. What is an ontology? In: *Handbook on ontologies*. [S.l.]: Springer, 2009. p. 1–17.

GUIDI, B.; CONTI, M.; PASSARELLA, A.; RICCI, L. Managing social contents in decentralized online social networks: A survey. *Online Social Networks and Media*, Elsevier, v. 7, p. 12–29, 2018.

GUNDECHA, P.; RANGANATH, S.; FENG, Z.; LIU, H. A tool for collecting provenance data in social media. In: ACM. *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.], 2013. p. 1462–1465.

HERSCHEL, M.; DIESTELKÄMPER, R.; LAHMAR, H. B. A survey on provenance: What for? what form? what from? *The VLDB Journal—The International Journal on Very Large Data Bases*, Springer-Verlag New York, Inc., v. 26, n. 6, p. 881–906, 2017.

IANNELLA, R.; MCKINNEY, J. vcard ontology-for describing people and organizations. *W3C Group Note NOTE-vcard-rdf-20140522*, 2014.

KOLL, D.; LI, J.; FU, X. The good left undone: Advances and challenges in decentralizing online social networks. *Computer Communications*, Elsevier, v. 108, p. 36–51, 2017.

LIM, C.; LU, S.; CHEBOTKO, A.; FOTOUHI, F. Prospective and retrospective provenance collection in scientific workflow environments. In: IEEE. *2010 IEEE International Conference on Services Computing*. [S.l.], 2010. p. 449–456.

MA, Z.; CAPRETZ, M. A.; YAN, L. Storing massive resource description framework (rdf) data: a survey. *The Knowledge Engineering Review*, Cambridge University Press, v. 31, n. 4, p. 391–413, 2016.

MAGLIACANE, S. Reconstructing provenance. In: SPRINGER. *International Semantic Web Conference*. [S.l.], 2012. p. 399–406.

MANSOUR, E.; SAMBRA, A. V.; HAWKE, S.; ZEREBA, M.; CAPADISLI, S.; GHANEM, A.; ABOULNAGA, A.; BERNERS-LEE, T. A demonstration of the solid platform for social web applications. In: *Proceedings of the 25th International Conference Companion on World Wide Web*. [S.l.: s.n.], 2016. p. 223–226.

MCKENNA, L.; DEBRUYNE, C.; O'SULLIVAN, D. Modelling the provenance of linked data interlinks for the library domain. In: *Companion Proceedings of The 2019 World Wide Web Conference*. [S.l.: s.n.], 2019. p. 954–958.

MEESTER, B. D.; DIMOU, A.; VERBORGH, R.; MANNENS, E. Detailed provenance capture of data processing. In: *1e Workshop on Enabling Open Semantic Science co-located with 16th International Semantic Web Conference*. [S.l.: s.n.], 2017. p. 1–8.

MISSIER, P. The lifecycle of provenance metadata and its associated challenges and opportunities. In: *Building Trust in Information*. [S.l.]: Springer, 2016. p. 127–137.

MISSIER, P.; DEY, S.; BELHAJJAME, K.; CUEVAS-VICENTTÍN, V.; LUDÄSCHER, B. D-prov: Extending the {PROV} provenance model with workflow structure. In: *5th {USENIX} Workshop on the Theory and Practice of Provenance (TaPP 13)*. [S.l.: s.n.], 2013.

MOREAU, L. The foundations for provenance on the web. *Foundations and Trends in Web Science*, Now Publishers Inc., v. 2, n. 2–3, p. 99–241, 2010.

MOREAU, L.; GROTH, P.; CHENEY, J.; LEBO, T.; MILES, S. The rationale of prov. *Web Semantics: Science, Services and Agents on the World Wide Web*, Elsevier, v. 35, p. 235–257, 2015.

NOY, N. F.; MCGUINNESS, D. L. et al. *Ontology development 101: A guide to creating your first ontology*. [S.l.]: Stanford knowledge systems laboratory technical report KSL-01-05 and . . . , 2001.

PASSANT, A.; BOJĀRS, U.; BRESLIN, J. G.; DECKER, S. The sioc project: semantically-interlinked online communities, from humans to machines. In: SPRINGER. *International Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems*. [S.l.], 2009. p. 179–194.

PÉREZ, B.; RUBIO, J.; SÁENZ-ADÁN, C. A systematic review of provenance systems. *Knowledge and Information Systems*, Springer, v. 57, n. 3, p. 495–543, 2018.

POVEDA-VILLALÓN, M.; GÓMEZ-PÉREZ, A.; SUÁREZ-FIGUEROA, M. C. Oops!(ontology pitfall scanner!): An on-line tool for ontology evaluation. *International Journal on Semantic Web and Information Systems (IJSWIS)*, IGI Global, v. 10, n. 2, p. 7–34, 2014.

POVEDA-VILLALÓN, M.; SUÁREZ-FIGUEROA, M. C.; GÓMEZ-PÉREZ, A. Validating ontologies with oops! In: SPRINGER. *International conference on knowledge engineering and knowledge management*. [S.l.], 2012. p. 267–281.

RAM, S.; LIU, J. Understanding the semantics of data provenance to support active conceptual modeling. In: SPRINGER. *International Workshop on Active Conceptual Modeling of Learning*. [S.l.], 2006. p. 17–29.

RECUERO, R. da C. 12. comunidades virtuais: uma abordagem teórica. *Mídia, imprensa e as novas tecnologias*, Edipucrs, v. 24, p. 221, 2002.

REILLY, C. F.; NAUGHTON, J. F. Exploring provenance in a distributed job execution system. In: SPRINGER. *International Provenance and Annotation Workshop*. [S.l.], 2006. p. 237–245.

RIVENI, M.; NGUYEN, T.-D.; AKTAS, M. S.; DUSTDAR, S. Application of provenance in social computing: A case study. *Concurrency and Computation: Practice and Experience*, Wiley Online Library, v. 31, n. 3, p. e4894, 2019.

ROCHE, C. Ontology: a survey. *IFAC Proceedings Volumes*, Elsevier, v. 36, n. 22, p. 187–192, 2003.

SALVE, A. D.; GUIDI, B.; MORI, P. Predicting the availability of users' devices in decentralized online social networks. *Concurrency and Computation: Practice and Experience*, Wiley Online Library, v. 30, n. 20, p. e4390, 2018.

SALVE, A. D.; MORI, P.; RICCI, L.; AL-AARIDHI, R.; GRAFFI, K. Privacy-preserving data allocation in decentralized online social networks. In: SPRINGER. *Distributed Applications and Interoperable Systems*. [S.l.], 2016. p. 47–60.

SAMBRA, A.; STORY, H.; BERNERS-LEE, T. *Webid 1.0-web identity and discovery. W3C*. 2013.

SAMBRA, A. V.; MANSOUR, E.; HAWKE, S.; ZEREBA, M.; GRECO, N.; GHANEM, A.; ZAGIDULIN, D.; ABOULNAGA, A.; BERNERS-LEE, T. *Solid: a platform for decentralized social applications based on linked data*. [S.l.], 2016.

SCHREIBER, G.; RAIMOND, Y. Rdf 1.1 primer. w3c working group note.(24 june 2014). *W3C. Recuperado a partir de https://www. w3. org/TR/2014/NOTE-rdf11-primer-20140624*, 2014.

SHARMA, R.; DATTA, A. Supernova: Super-peers based architecture for decentralized online social networks. In: IEEE. *2012 Fourth International Conference on Communication Systems and Networks (COMSNETS 2012)*. [S.l.], 2012. p. 1–10.

SIMMHAN, Y. L.; PLALE, B.; GANNON, D. Karma2: Provenance management for data-driven workflows. *International Journal of Web Services Research (IJWSR)*, IGI Global, v. 5, n. 2, p. 1–22, 2008.

SOHN, J.-S.; CHUNG, I.-J. Dynamic foaf management method for social networks in the social web environment. *The journal of supercomputing*, Springer, v. 66, n. 2, p. 633–648, 2013.

SOLANKI, M. R. Solid: A web system to restore the control of users' personal data. In: *ICT Systems and Sustainability*. [S.l.]: Springer, 2021. p. 257–267.

STANTIC, B. Provenance-based rumor detection. In: SPRINGER. *Databases Theory and Applications: 28th Australasian Database Conference, ADC 2017, Brisbane, QLD, Australia, September 25–28, 2017, Proceedings*. [S.l.], 2017. v. 10538, p. 125.

STRUFE, T. Safebook: A privacy-preserving online social network leveraging on real-life trust. *IEEE Communications Magazine*, Citeseer, v. 95, 2009.

SURIARACHCHI, I.; ZHOU, Q.; PLALE, B. Komadu: A capture and visualization system for scientific data provenance. *Journal of Open Research Software*, Ubiquity Press, v. 3, n. 1, 2015.

TAN, W. C. Research problems in data provenance. *IEEE Data Eng. Bull.*, v. 27, n. 4, p. 45–52, 2004.

TAS, Y.; BAETH, M. J.; AKTAS, M. S. An approach to standalone provenance systems for big social provenance data. In: IEEE. *2016 12th International Conference on Semantics, Knowledge and Grids (SKG)*. [S.l.], 2016. p. 9–16.

TAXIDOU, I.; LIEBER, S.; FISCHER, P. M.; NIES, T. D.; VERBORGH, R. Web-scale provenance reconstruction of implicit information diffusion on social media. *Distributed and Parallel Databases*, Springer, v. 36, n. 1, p. 47–79, 2018.

TAXIDOU, I.; NIES, T. D.; VERBORGH, R.; FISCHER, P. M.; MANNENS, E.; WALLE, R. Van de. Modeling information diffusion in social media as provenance with w3c prov. In: *Proceedings of the 24th International Conference on World Wide Web*. [S.l.: s.n.], 2015. p. 819–824.

TRAMP, S.; FRISCHMUTH, P.; ERMILOV, T.; SHEKARPOUR, S.; AUER, S. An architecture of a distributed semantic social network. *Semantic Web*, IOS Press, v. 5, n. 1, p. 77–95, 2014.

TRINH, T.-D.; ARYAN, P. R.; DO, B.-L.; EKAPUTRA, F. J.; KIESLING, E.; RAUBER, A.; WETZ, P.; TJOA, A. M. Linked data processing provenance: towards transparent and reusable linked data integration. In: ACM. *Proceedings of the International Conference on Web Intelligence*. [S.l.], 2017. p. 88–96.

TRIVEDI, H.; BINDU, P.; THILAGAM, P. S. Identifying provenance of information and anomalous paths in attributed social networks. In: IEEE. *2018 Second International Conference on Computing Methodologies and Communication (ICCMC)*. [S.l.], 2018. p. 914–919.

VERBORGH, R. Decentralizing the semantic web through incentivized collaboration. In: *ISWC2018, the 17th International Semantic Web Conference*. [S.l.: s.n.], 2018. p. 1–5.

VRANDEČIĆ, D. Ontology evaluation. In: *Handbook on ontologies*. [S.l.]: Springer, 2009. p. 293–313.

WANG, P.; XU, B.; WU, Y.; ZHOU, X. Link prediction in social networks: the state-of-the-art. *Science China Information Sciences*, Springer, v. 58, n. 1, p. 1–38, 2015.

WOODMAN, S.; HIDEN, H.; WATSON, P. Applications of provenance in performance prediction and data storage optimisation. *Future Generation Computer Systems*, Elsevier, v. 75, p. 299–309, 2017.

XIA, Y.; LIU, Q.; TAN, C.; LENG, J.; XU, S.; ZANG, B.; CHEN, H. Taming distrust in the decentralized internet with pixiu. *arXiv preprint arXiv:1901.06095*, 2019.

YEUNG, C.-m. A.; LICCARDI, I.; LU, K.; SENEVIRATNE, O.; BERNERS-LEE, T. Decentralization: The future of online social networking. In: *W3C Workshop on the Future of Social Networking Position Papers*. [S.l.: s.n.], 2009. v. 2, p. 2–7.