



**UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO**



CARLOS FRAN FERREIRA DANTAS

**ARIDUS: UMA PLATAFORMA ORIENTADA A
SERVIÇOS WEB PARA INTEGRAÇÃO SEMÂNTICA DE
INFORMAÇÕES EM DIFERENTES DOMÍNIOS DE
CONHECIMENTO.**

MOSSORÓ – RN

2013

CARLOS FRAN FERREIRA DANTAS

**ARIDUS: UMA PLATAFORMA ORIENTADA A
SERVIÇOS WEB PARA INTEGRAÇÃO SEMÂNTICA DE
INFORMAÇÕES EM DIFERENTES DOMÍNIOS DE
CONHECIMENTO.**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação – associação ampla entre a Universidade do Estado do Rio Grande do Norte e a Universidade Federal Rural do Semi-Árido, para a obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dra. Cláudia Maria Fernandes Araújo Ribeiro – UERN.

MOSSORÓ – RN

2013

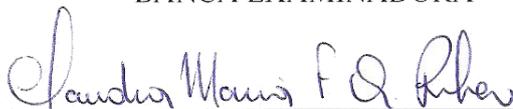
CARLOS FRAN FERREIRA DANTAS

**ARIDUS: UMA PLATAFORMA ORIENTADA A
SERVIÇOS WEB PARA INTEGRAÇÃO SEMÂNTICA DE
INFORMAÇÕES EM DIFERENTES DOMÍNIOS DE
CONHECIMENTO.**

Dissertação apresentada ao Programa de Pós-
Graduação em Ciência da Computação para a
obtenção do título de Mestre em Ciência da
Computação.

APROVADA EM: 27 / 08 / 2013.

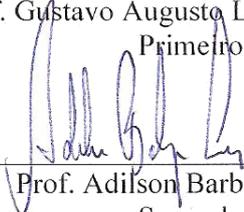
BANCA EXAMINADORA



Prof. Cláudia M^a. Fernandes Araújo Ribeiro – UERN
Presidente



Prof. Gustavo Augusto Lima de Campos – UECE
Primeiro Membro



Prof. Adilson Barboza Lopes – UFRN
Segundo Membro

**Catálogo da Publicação na Fonte.
Universidade do Estado do Rio Grande do Norte.**

Dantas, Carlos Fran Ferreira

Aridus: uma plataforma orientada a serviços web para integração semântica de informações em diferentes domínios de conhecimento. / Carlos Fran Ferreira Dantas. – Mossoró, RN, 2013.

95 f.

Orientador: Prof. Dra. Cláudia Maria Fernandes Araújo Ribeiro

Dissertação (Mestrado) Universidade do Estado do Rio Grande do Norte. Universidade Federal Rural do Semi-Árido. Programa de Pós-Graduação em Ciência da Computação.

1. Plataforma Aridus. 2. Integração de informações. 3. Web Semântica. 4. Arquitetura Orientada a Serviços. I. Ribeiro, Cláudia Maria Fernandes Araújo . II. Universidade do Estado do Rio Grande do Norte. III. Universidade Federal Rural do Semi-Árido. IV. Título.

UERN/BC

CDD 005

Bibliotecária: Jocelania Marinho Maia de Oliveira – CRB 15 319

AGRADECIMENTOS

Agradeço a Deus por tudo, por Ele.

Aos meus pais, Francisco Benedito Belo e Maria José Ferreira Dantas, pelo apoio incondicional, pela formação e, acima de tudo, pelo amor dedicado.

Aos meus amados irmãos, Carliano, Carliésio e Carliolane, por todos os incentivos e auxílios.

À família Nunes Oliveira, em especial, Gladstone, Régia Berlânia e Nathályson, pelo acolhimento em tantos momentos, por todo o carinho e todos os cuidados de sempre.

À minha amiga, companheira, cúmplice e merecedora de todo amor e dedicação, Natalyany <3.

À médica Ana Flávia Sobral de Medeiros e sua equipe, pelo acompanhamento, incentivos e suporte clínico.

À professora orientadora Dra. Cláudia Maria Fernandes Araújo Ribeiro por acreditar na minha capacidade e orientar-me neste trabalho.

Aos meus amigos do Grupo de Sistemas Distribuídos (GSiD) e companheiros do Laboratório LUMEN da UERN, Felipe Costa, Tatiana Sorrentino e Hander Pereira, pelas colaborações diretas e indiretas.

À todos do Programa de Pós-Graduação em Ciência da Computação (PPgCC) UERN/UFERSA pelo suporte acadêmico.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo suporte ao projeto ARIDUS (Edital UNIVERSAL N° 14/2011).

Aos professores da banca pela disponibilidade em avaliar e contribuir para a melhoria deste trabalho.

A todos não citados, mas que, de alguma forma, contribuíram para a conclusão deste trabalho e etapa de vida.

Muito obrigado a todos.

RESUMO

Com o crescente volume de informações digitais, a necessidade de analisá-las de forma integrada torna-se um requisito crítico. Este trabalho apresenta uma plataforma semântica orientada a serviços que visa a integração de informações em diferentes domínios do conhecimento disponíveis na Web. Para alcançar este objetivo, apresenta-se um conjunto de serviços específicos para a anotação e extração semântica, bem como modelos para acessibilidade e personalização de conteúdos. Para fins de validação, dois aplicativos foram desenvolvidos. O primeiro enfatiza a integração da informações distribuídas em diferentes fontes e formatos e o segundo, é um sistema de recomendação com característica de acessibilidade. Outra aplicação desenvolvida mostra a viabilidade dos serviços da ARIDUS no contexto das redes sociais.

Palavras-chave: Integração de informações. Web Semântica. Arquitetura Orientada a Serviços.

ABSTRACT

As the amount of digital information increases, the necessity of analysing them in an integrated way becomes a critical requirement. This work presents a semantic service-oriented platform that aims to integrate information in different domains of knowledge spread out in the web. To accomplish this goal, it presents a set of specific services for semantic annotation and extraction, as well as accessibility and content personalization models. For validation purpose, two applications were developed. The first one emphasizes the integration of information distributed in different sources and formats and the second one is a recommender system with accessibility characteristic. Another application developed shows the feasibility of ARIDUS services in the context of social networks.

Keywords: Information integration. Semantic Web. Service-Oriented Architecture.

LISTA DE FIGURAS

1	Arquitetura da Web Semântica segundo a W3C ¹	18
2	Partes do URI (Uniform Resource Identifier).	19
3	Exemplo de arquivo XML.	19
4	Representação em forma de grafo de uma declaração RDF.	20
5	Exemplo de representação de informação em grafo (RDF <i>Graph</i>).	20
6	Exemplo representação de informação em RDF/XML.	21
7	Consulta SPARQL.	24
8	Representação Visual de Anotação Semântica	26
9	Comparativo entre ferramentas de anotação semântica	27
10	Trecho de código HTML contendo anotações em RDFa.	28
11	Informações exibidas ao usuário pelo navegador web.	28
12	RDF extraído do trecho de código HTML da figura10.	28
13	Cenário de Processo de Negócio sem SOA	33
14	Cenário de Processo de Negócio com SOA	34
15	Paradigma “ <i>Find, Bind and Execute</i> ”	35
16	Exemplo de Orquestração de Serviços	38
17	Exemplo de Coordenação de Serviços	38
18	Exemplo de Coreografia de Serviços	39
19	Padrões em serviços web	40
20	Utilização de tecnologias web em serviços	41
21	Arquitetura do Integrated Agriculture Information Framework (SHOAIB; BASHARAT, 2010)	44

22	Arquitetura do framework de integração de conhecimento em produção agrícola	45
23	Arquitetura do Asio Semantic Query Decomposition (SQD) ¹	47
24	Arquitetura da Plataforma ARIDUS	50
25	Disposição dos Serviços da Plataforma ARIDUS	55
26	Orquestração <i>AlignAndImport</i>	59
27	Exemplo de entrada para o serviço <i>QueryRewriter</i>	61
28	Consulta Exemplo da Saída do Serviço <i>QueryRewriter</i>	61
29	Interface de Usuário da Aplicação “ <i>Reasoning On The Web</i> ”.	63
30	Virtuoso Sponger - Extração e Estruturação de Dados em RDF.	64
31	Orquestração <i>QueryProcess</i> .	67
32	Cenário de Integração de Informação com a ARIDUS.	68
33	Diagrama de Sequência - Aplicação, Serviços e Fontes de Dados.	70
34	Disposição do Modelo PerSoN na Plataforma ARIDUS.	72
35	Disposição do Modelo @dapt na Plataforma ARIDUS.	73
36	IntercropApp – Criação de um consórcio de duas culturas	75
37	Consulta SPARQL utilizada para obter informações da cultura Alface	76
38	Resultado da análise do consórcio entre as cultura selecionadas pelo usuário	77
39	Exibição de informações de um documento da <i>DocBase</i>	79
40	Interface web da DocBase	79
41	SPARQL Endpoint da DocBase	80
42	Portal de Informação Tecnológica em Agricultura da Embrapa (Infoteca-e)	81
43	Página gerada pelo serviço <i>ContentViewer</i>	83
44	Disposição do <i>TwitterRecommender</i> na Plataforma ARIDUS.	84
45	Alinhamento entre vocabulários para estruturação dos perfis de usuários.	87
46	Alinhamento entre vocabulários para estruturação de dados sobre conteúdos	89

LISTA DE SIGLAS

BPEL	Business Process Execution Language
GSiD	Grupo de Sistemas Distribuídos
HTML	HiperText Markup Language
N3	Notation 3
OASIS	Organization for the Advancement of Structured Information Standards
OWL	Ontology Web Language
PerSoN	Personalization Model for Semantic Social Networks
RDF	Resource Description Framework
RDFS	RDF Scheme
REST	Representational State Transfer
SOA	Service-Oriented Architecture
SOMA	Service-Oriented Methodology and Architecture
SPARQL	SPARQL Protocol and RDF Query Language
SWRL	Semantic Web Rule Language
TI	Tecnologia da Informação
UDDI	Universal Description, Discovery and Integration
UERN	Universidade do Estado do Rio Grande do Norte
UFERSA	Universidade Federal Rural do Semi-Árido
URI	Uniform Resource Identifier
W3C	World Wide Web Consortium
WSDL	Web Services Description Language
XML	eXtensible Markup Language

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivos	15
1.1.1	Objetivos específicos	15
1.2	Estrutura do trabalho	15
2	INTEGRAÇÃO SEMÂNTICA DE INFORMAÇÕES	16
2.1	Integração de Dados	16
2.2	Web Semântica	17
2.2.1	URI, XML e RDF	18
2.2.2	Ontologias e OWL	21
2.2.3	SPARQL	23
2.2.4	Anotação, Inferência e Extração	25
2.2.4.1	Anotação Semântica	25
2.2.4.2	Extração Semântica	27
2.2.4.3	Inferência	29
2.3	Integração Semântica de Informações	30
2.3.1	Utilizando Axiomas OWL	30
2.3.2	Utilizando Regras de Usuário	31
2.3.3	Utilizando consultas SPARQL	31
3	SOA E SERVIÇOS WEB	33
3.1	<i>Service-Oriented Architecture</i> (SOA)	33

3.2	Serviços	34
3.2.1	Tipos básicos de serviços	36
3.2.2	Características de serviços	36
3.3	Composição de serviços	37
3.4	Tecnologias em Serviços Web	39
4	TRABALHOS RELACIONADOS	42
4.1	SWIS – <i>Semantic Web-based Integration System</i>	42
4.2	ISENS	43
4.3	IAIF – Integrated Agriculture Information Framework	44
4.4	Pesquisa em tecnologias web semântica para a integração de conhecimento em produção agrícola	45
4.5	Asio Scout	46
4.6	Análise comparativa	48
5	PLATAFORMA ARIDUS	50
5.1	Requisitos e Arquitetura	50
5.2	Serviços	52
5.2.1	<i>DataSetManager</i> (1)	55
5.2.2	<i>Alignment API</i> (2)	56
5.2.3	<i>OWL Syntax Converter</i> (3)	58
5.2.4	<i>AlignmentManager</i> (4)	58
5.2.5	<i>AlignAndImportProcess</i> (5)	59
5.2.6	<i>QueryRewriter</i> (6)	59
5.2.7	<i>Semantic Reasoner</i> (7)	62
5.2.8	<i>(RDF Proxy Service) IExtractor</i> (8) e <i>IAcquisitionProcess</i> (9)	63
5.2.9	<i>Semantic Annotator</i> (10)	65
5.2.10	<i>Query Executor</i> (11)	66

5.2.11	<i>Query Process</i> (12)	67
5.2.12	<i>UserProfileManager</i> (13)	67
5.3	Integração Semântica de Informações com a ARIDUS	68
5.4	Extensão da plataforma	71
5.4.1	PerSoN: Um modelo semântico para personalização e interoperabilidade entre redes sociais	71
5.4.2	@dapt – Modelo Semântico para prover acessibilidade para deficientes visuais na Internet	72
6	ESTUDOS DE CASO E APLICAÇÕES	74
6.1	IntercropApp	74
6.2	Integração DocBase e Infoteca-e	78
6.2.1	As fontes de informações	78
6.2.1.1	DocBase	78
6.2.1.2	Infoteca-e	81
6.2.2	Integração, Personalização e Acessibilidade das Informações	81
6.3	<i>TwitterRecommender</i>	84
7	CONSIDERAÇÕES FINAIS	90
	Referências	92

1 INTRODUÇÃO

A popularização da Internet e o aumento no nível de informatização nas instituições ocasionou em um crescente volume de dados na web e na necessidade emergente de integrar as várias fontes de dados existentes e acessíveis através da web, com o intuito de executar a análise sobre os dados com uma visão global. Além disso, as várias fontes isoladas contêm diferentes informações que podem apresentar entre si relações de significado e se complementar para a resolução de algumas questões (DIMITROV et al., 2009).

A perspectiva do desenvolvimento sustentável é uma amostra da necessidade de se analisar dados de maneira integrada. Como é o caso, por exemplo, dos pesquisadores da área da agricultura que tem sido forçados a analisar de forma integrada dados sobre várias culturas, suas relações com o clima e o solo, os impactos na natureza, os aspectos sociais, econômicos, além de outros, com o intuito de avaliar os projetos nas dimensões da sustentabilidade.

A integração de dados tem por objetivo oferecer uma visão unificada das informações, tornando transparente a complexidade de acesso a cada fonte de dados separadamente, o que facilita a análise integrada. Segundo Macias-Garcia, Sosa-Sosa e Lopez-Arevalo (2009), geralmente o processo de integração é um trabalho difícil, lento e suscetível a erros.

Em geral, as fontes de dados, apesar de correlacionadas, apresentam estruturas diferentes e terminologias com significados próprios em seus contextos que dificultam a integração. Nestes cenários faz-se necessário expandir a interoperabilidade ao nível semântico, permitindo que as relações entre os dados sejam realizadas em função dos seus significados e não pela similaridade de suas sintaxes.

A integração semântica permite que um mesmo dado possa coexistir em diferentes domínios do conhecimento, o que evita redundância e possíveis falhas de integridade. Além disso, diferentes visões ou perspectivas de conhecimento em domínios específicos podem ser associadas aos dados, preservando o vocabulário original da fonte e permitindo que sejam adicionadas informações de outros aspectos aos dados.

Devido à evolução natural dos sistemas, eles atendem a um número cada vez maior de processos de negócios ou necessidades, crescendo em complexidade e volume de dados armazenados. Este cenário, apoiado pela existência comum de um ambiente computacional heterogêneo e a necessidade por reuso de componentes, faz da tecnologia de serviços uma das mais promissoras no atendimento destes desafios, havendo um enorme interesse empresarial em sua adoção (SOMMERVILLE, 2007).

Serviços apresentam características como coesão, autonomia e baixo acoplamento, que proporcionam reuso e a composição. Além disso, por estarem associados às tecnologias Web, serviços têm a capacidade de reuso e interoperabilidade potencializados (ERL, 2009).

Service-Oriented Architecture (SOA) é um modelo arquitetônico que visa aprimorar a eficiência, a agilidade e a produtividade de uma instituição, alinhando os objetivos estratégicos à infraestrutura de tecnologia da informação, permitindo que benefícios como mais opções de fornecedores e maior retorno de investimento sejam alcançados (ERL, 2009).

SOA tem sido largamente empregada no desenvolvimento de soluções de integração de aplicações e sistemas de informação (LU; ZHANG, 2009)(QING-RUI; HAI-TAO, 2010)(GU; ZHANG, 2010)(KEENEY et al., 2011), bem como, tecnologias da Web Semântica (BERNERS-LEE; HENDLER; LASSILA, 2001) são atualmente utilizadas na integração de dados (AMINI; SABOOHI; NEMATBAKHSH, 2008)(ZHOU et al., 2010)(SHI et al., 2010)(CAIPING; HUIJING, 2010).

É importante ressaltar que SOA e Web Semântica também têm sido exploradas em conjunto, como, por exemplo, a aplicação de semântica para descoberta, invocação e composição automática de serviços, e a utilização de serviços web para disponibilização de ferramentas relacionadas às tecnologias semânticas, a exemplo, *OWL Syntax Converter*¹ que traduz uma ontologia em *Ontology Web Language* (OWL) para outras sintaxes.

É neste contexto que se insere a ARIDUS, uma plataforma orientada a serviços web para a integração semântica de informações em diferentes domínios do conhecimento, projeto desenvolvido com suporte do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) – Edital UNIVERSAL N° 14/2011 – e que se relaciona à outros projetos desenvolvidos por membros do Grupo de Sistemas Distribuídos (GSiD) no Laboratório LUMEN (UERN – Campus Natal). Os objetivos deste trabalho estão descritos a seguir.

¹<http://owl.cs.manchester.ac.uk/converter/restful.jsp>

1.1 Objetivos

O principal objetivo deste trabalho é o desenvolvimento de uma plataforma Web orientada a serviços para anotação semântica, extração, apresentação e integração semântica de informações em diferentes domínios do conhecimento.

1.1.1 Objetivos específicos

- Identificar e desenvolver um conjunto de serviços que atendam aos requisitos da integração semântica de informações em diferentes domínios do conhecimento;
- Dar suporte e estender as atividades previstas em projeto de bolsa de mestrado aprovado em edital de Bolsas no País (Edital MCT/CNPq N° 70/2009 – Mestrado/Doutorado – Processo 558489/2010-9) em desenvolvimento pelo Mestrado em Ciência da Computação, associação ampla entre a Universidade do Estado do Rio Grande do Norte (UERN) e Universidade Federal Rural do Semi-Árido (UFERSA), que visa à estruturação e extração de conhecimento através de tecnologias semânticas, e propõe como estudo de caso, a análise integrada de informações sobre o cultivo consorciado de culturas no semiárido, sob as dimensões da sustentabilidade;
- Realizar a extensão da plataforma com modelos desenvolvidos por membros do Grupo de Sistemas Distribuídos – GSiD/UERN para prover acessibilidade e personalização.

1.2 Estrutura do trabalho

Este documento apresenta fundamentos em Integração de Dados, Web Semântica e Integração Semântica no Capítulo 2 e fundamentos em SOA e Serviços Web no Capítulo 3, expondo os principais conceitos e tecnologias abordados nos capítulos seguintes.

No Capítulo 4, são apresentados trabalhos relacionados e discutido o estado da arte em integração semântica.

O Capítulo 5 apresenta os requisitos, arquitetura e serviços da plataforma ARIDUS, cujos estudos de caso e aplicações são apresentadas no capítulo seguinte (Capítulo 6).

Por fim, considerações, contribuições e perspectivas de trabalhos futuros são apresentados.

2 *INTEGRAÇÃO SEMÂNTICA DE INFORMAÇÕES*

Neste capítulo, conceitos fundamentais sobre integração de informações e a aplicação de tecnologias semânticas neste trabalho.

2.1 *Integração de Dados*

Com o crescente volume de dados disponíveis em diferentes fontes na web e a emergente necessidade de analisá-los sob uma perspectiva global, a integração de dados tem sido explorada pelas intuições com diferentes interesses científicos e comerciais no intuito de responder questões para a tomada de decisão e produzir novos conhecimentos.

A integração de dados consiste em combinar dados armazenados em diferentes fontes provendo uma visão unificada desses dados, tornando transparente a complexidade de acesso a cada uma das fontes de dados separadamente e possibilitando a análise integrada (SOARES; TANAKA; BAIÃO, 2010).

Segundo Telang e Chakravarthy (2007) e Wu (2012), as principais abordagens de integração de dados são:

- **Warehousing** – Abordagem baseada na tradicional técnica de *Data Warehouse*. Dados de diferentes fontes distribuídas são obtidos, mapeados para uma estrutura comum e armazenados em único local de forma centralizada;
- **Federação** – Caracterizada por fontes de dados distribuídas e acesso uniforme, e pela obtenção das informações necessárias para responder consultas diretamente das fontes;

- **Mediador** – No contexto de integração de informações, um mediador é um sistema responsável por reformular consultas de usuários em consultas no esquema local das fontes de dados subjacentes. Através de um esquema global, são realizadas uma visão virtual, reconciliada e integrada das fontes de dados, bem como a reformulação das consultas. A reformulação das consultas é baseada nos mapeamentos existentes entre os esquemas locais das fontes de dados e o esquema global.

As abordagens apresentam vantagens e desvantagens em suas respectivas aplicações, sendo que os mediadores são a abordagem mais flexível para a integração de fontes de dados heterogêneas, principal características das fontes de dados acessíveis na Web, e com desempenho de performance mais estável(WU, 2012). Por estes motivos, é a abordagem mais utilizada atualmente, como pode ser verificado nos trabalhos relacionados deste documento.

2.2 Web Semântica

A web baseia-se na linguagem HTML (*HiperText Markup Language*) que descreve como são apresentadas as informações, porém não define o que significam tais informações. Neste sentido, o crescente volume e a heterogeneidade dos dados na web mostram-se como grandes desafios na recuperação de informações. Os sistemas de busca tradicionais baseados em palavras-chave podem ter um significativo melhoramento com a aplicação de tecnologias semânticas.

A Web Semântica é uma extensão da web atual em que a informação é disponibilizada com significado bem definido com o propósito de tornar as informações compreensíveis pelos computadores no intuito de fazê-los trabalharem melhor em cooperação com as pessoas (BERNERS-LEE; HENDLER; LASSILA, 2001).

O significado da informação é explicitado pela representação formal e padronizada do conhecimento, o que possibilita o processamento automático do significado, o relacionamento e integração de dados heterogêneos em nível semântico, e a inferência automática de informações implícitas.

A figura 1 apresenta a arquitetura conceitual da Web Semântica segundo a *World Wide Web Consortium (W3C)*¹ e expõe algumas das tecnologias que permitem a construção das aplicações.

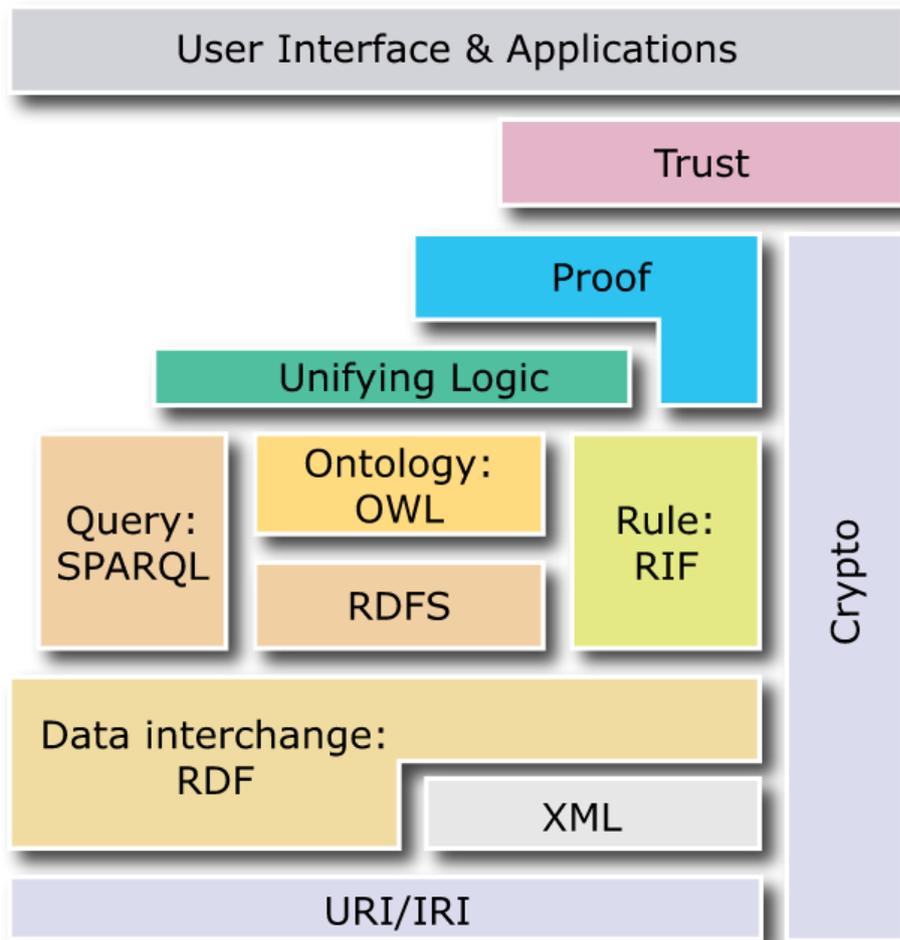


Figura 1: Arquitetura da Web Semântica segundo a W3C¹

É importante frisar que dentre as tecnologias da Web Semântica, há tecnologias experimentais, normatizadas e recém-normatizadas. Destas, as principais tecnologias normatizadas e recém-normatizadas são apresentadas nas subseções seguintes.

2.2.1 URI, XML e RDF

A Web Semântica, também conhecida como Web de Dados (*Web of Data*), necessita de identificação única, representação estruturada e expressão da semântica relacionada ao recurso na Web, para que máquinas, na figura dos agentes de software, possam acessar, decodificar e interpretar o significado das informações. Neste sentido, as tecnologias básicas utilizadas são:

¹<http://www.w3.org/>

- **URI (*Uniform Resource Identifier*)**

O URI trata-se de uma sequência compacta de caracteres possíveis que identifica um recurso físico ou abstrato, sendo um meio simples e extensível de identificação de recursos (RFC 3986)². Utilizando URI é possível identificar unicamente recursos de diferentes tipos com um esquema uniforme de identificação, como ilustra a figura 2, o que facilita o acesso ao recurso.



Figura 2: Partes do URI (Uniform Resource Identifier).

- **XML (*eXtensible Markup Language*)**

A XML é uma linguagem de marcação extensível frequentemente utilizada para a simplificação do armazenamento e intercâmbio de dados.

Semelhante ao HTML, XML possui sua sintaxe baseada em *tags*, porém, não pré-definidas. Assim, um documento escrito em XML tem suas *tags* definidas pelo desenvolvedor da aplicação, obedecendo às regras da sintaxe da linguagem. Desta forma, XML é uma linguagem flexível e de extrema extensibilidade, servindo de base para outras várias tecnologias como por exemplo: RDF (*Resource Description Framework*), OWL (*Ontology Web Language*) e GML (*Geography Markup Language*) (SOARES; TANAKA; BAIÃO, 2010).

A figura 3 apresenta um exemplo simples de arquivo XML.

```

1  <?xml version="1.0" ?>
2  <livro>
3      <titulo>Foundations of Semantic Web Technologies</titulo>
4      <autor>Hitzler P.; Krotzsch M.; Rudolph S.</autor>
5      <editora>CRC Press</editora>
6      <ano>2011</ano>
7  </livro>

```

Figura 3: Exemplo de arquivo XML.

As *tags* <livro>, <titulo>, <autor>, <editora> e <ano>, utilizadas no documento exemplo, são tags definidas pelo desenvolvedor e estruturam os dados do documento.

Embora a linguagem viabilize a codificação de qualquer tipo de estrutura de dados e apoie a troca de dados de forma autônoma de formato de arquivo, tecnologia e plataforma, a linguagem XML não especifica a semântica e a forma de utilização da informação dentro do

²Request For Comments(RFC) 3986 – <http://www.ietf.org/rfc/rfc3986.txt>

contexto do documento(DIAS, 2011). Portanto, os nomes dos elementos e atributos definidos nas *tags* do documento e seus significados precisam ser assegurados pelos recursos envolvidos na troca das informações.

- **RDF (*Resource Description Framework*)**

O RDF trata-se de um modelo para a representação de informações sobre recursos na web. Baseado na identificação de recursos com URI's e na descrição destes recursos em termos de propriedades simples e seus respectivos valores, RDF permite a representação de informações sobre recurso com a declaração de triplas (*RDF Triples*) do tipo *Sujeito-Propriedade-Objeto*, podendo o conjunto de declarações ser representado por um grafo direcionado. A figura 4 ilustra o grafo de uma tripla RDF.

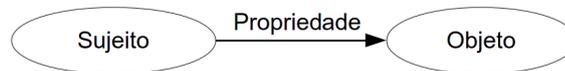


Figura 4: Representação em forma de grafo de uma declaração RDF.

Os nós do grafo representam os recursos e os arcos representam as propriedades. RDF possui algumas sintaxes de representação, como por exemplo, Turtle³, N3⁴, N-Triple⁵ e RDF/XML⁶. De forma ilustrativa, as figuras 5 e 6 apresentam as representações em grafo e RDF/XML de informações sobre um dos documentos disponibilizados pela W3C.

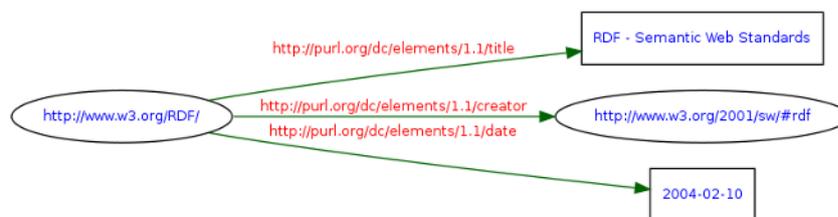


Figura 5: Exemplo de representação de informação em grafo (*RDF Graph*).

³*Terse RDF Triple Language* – <http://www.w3.org/TeamSubmission/turtle/>

⁴*Notation 3* – <http://www.w3.org/TeamSubmission/n3/>

⁵<http://www.w3.org/TR/n-triples/>

⁶<http://www.w3.org/TR/REC-rdf-syntax/>

```

1: <?xml version="1.0"?>
2: <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3:   xmlns:dc="http://purl.org/dc/elements/1.1/">
4:   <rdf:Description rdf:about="http://www.w3.org/RDF/">
5:     <dc:title>RDF - Semantic Web Standards</dc:title>
6:     <dc:creator><rdf:Description rdf:about="http://www.w3.org/2001/sw/#rdf">
7:   </rdf:Description>
8:   </dc:creator>
9:   <dc:date>2004-02-10</dc:date>
10: </rdf:Description>
11: </rdf:RDF>
12:

```

Figura 6: Exemplo representação de informação em RDF/XML.

Com o RDF, tem-se uma semântica processável por computadores e permite a codificação, a interação e o reúso de metadados estruturados .

O RDF possibilita a codificação, a troca e reuso de metadados estruturados, bem como o processamento automático de significado e prover interoperabilidade sintática, semântica e estrutural da informação (DIAS, 2011)

Um exemplo da utilização do RDF para a criação de elementos de metadados é o RDF Schema (RDFS), modelo este, capaz de declarar classes básicas e tipos a serem utilizados nas descrições RDF. O RDFS define vocabulários novos associados à um determinado domínio/aplicação que podem ser referenciados com o uso de um URI. O RDFS pode ser considerado como uma linguagem primitiva para escrever ontologias (SOARES; TANAKA; BAIÃO, 2010).

URI, XML e RDF/RDFS servem de base para várias outras tecnologias da Web Semântica, sendo, portanto, fundamental à compreensão destas tecnologias.

2.2.2 Ontologias e OWL

Ontologia é um termo originado na Filosofia que designa um ramo da metafísica no qual se estuda o ser e seus relacionamentos. Em sistemas de informação, linguística, ciência da informação e em outras área de aplicação, o termo recebe interpretações mais específicas da área como o comumente utilizado pela comunidade de pesquisadores em Web Semântica que define uma ontologia como “uma representação formal de conhecimento por um conjunto de conceitos dentro de um domínio e os relacionamentos entres estes conceitos” (BARRETT et al., 2002).

Podemos destacar como os principais componentes de uma ontologia:

- **Classes** – representam entidades ou conceitos de um domínio modelado;

- **Propriedades** – atributos ou itens de informação que descrevem uma classe;
- **Indivíduos ou Instâncias** – são exemplares das classes e propriedades de uma ontologia;
- **Relacionamentos** – representam as relações entre as instâncias das classes, normalmente definidas pelo uso de propriedades cujo valor é uma instância de uma ontologia;
- **Axiomas** – assertivas aplicadas aos conceitos e às propriedades.

As ontologias permitem o compartilhamento e reúso de conhecimento e são aplicadas na estruturação, troca e recuperação de informações, de maneira que eleva tais tarefas ao nível semântico, pois fornecem vocabulários para a associação de conceitos à documentos ou partes de um documento, inferência de informações implícitas e definição de relações entre termos/conceitos de diferentes ontologias, o que favorece a integração e a recuperação de informações em diferentes domínios de conhecimento.

Para a construção de ontologias é essencial uma linguagem com semântica bem definida e expressiva o suficiente para descrever inter-relacionamentos abstrusos e assertivas/restrições entre conceitos, e neste sentido, a W3C recomenda a OWL⁷ (*Ontology Web Language*) (ELLER, 2008).

A OWL possui 3 sublinguagens que se diferenciam pela expressividade e complexidade formal, características estas que implicam na representatividade do conhecimento e na realização de raciocínio sob as ontologias criadas.

As sublinguagens da OWL são:

- **OWL Lite** – sublinguagem direcionada ao grupo de desenvolvedores que necessitam apenas de uma hierarquia de classes e restrições simples;
- **OWL DL (*Description Language*)** – baseia-se na lógica descritiva e, portanto, passível de raciocínio automático e verificação de inconsistências na ontologia;
- **OWL Full** – é a sublinguagem de maior expressividade, porém não se garante a decidibilidade ou completeza da linguagem, o que torna impossível realizar inferências em ontologias OWL Full.

A OWL é implementada com a utilização das tecnologias XML, RDF e RDFS, sendo considerada uma evolução destas tecnologias para expressão de semântica interpretável por máquinas.

⁷<http://www.w3.org/2004/OWL/>

2.2.3 SPARQL

SPARQL é o acrônimo recursivo para o inglês *SPARQL Protocol and RDF Query Language*, ou seja, SPARQL refere-se a uma linguagem de consulta e protocolo de acesso a dados em RDF, sendo o padrão recomendado pela W3C (DUCHARME, 2011).

A linguagem SPARQL baseia-se na representação *Turtle* do RDF e estabelece o conceito de *Graph Pattern* que é justamente RDF usando algumas variáveis.

A figura 7 exibe uma consulta exemplo.

```

1 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
2 SELECT ?name ?age
3 WHERE {
4   ?x foaf:name ?name .
5   ?x foaf:age ?age
6 }
7 ORDER BY ?name

```

Figura 7: Consulta SPARQL.

As variáveis na linguagem SPARQL são iniciadas com sinal de interrogação e usadas na construção de padrões de triplas e grafos que os mecanismos de processamento de consultas utilizam para selecionar as informações a serem retornadas como resultado da consulta.

Na figura 7, apresenta-se uma consulta SPARQL com os principais componentes, que são:

- **PREFIX** – utilizado para abreviação de URI's. No exemplo, o URI `<http://xmlns.com/foaf/0.1/>` é abreviado por "foaf" e portanto, a propriedade *name* pode ser referenciada utilizando `foaf:name` e não mais `<http://xmlns.com/foaf/0.1/name>`.
- **SELECT** – é uma das 4 cláusulas de retorno possíveis (as demais são apresentadas posteriormente). O *SELECT* indica quais elementos indicados pelas variáveis devem ser retornados como resultado. Assim, o resultado de uma consulta *SELECT* é uma tabela onde as colunas indicam os elementos desejados e as linhas, os valores obtidos para tais elementos;
- **WHERE** – contém o conjunto de *triple pattern* a ser utilizado na seleção das informações;
- **ORDER BY** – é um dos possíveis modificadores da solução utilizados para alterar o resultado das consultas. *ORDER BY* indica como os valores devem ser ordenados. Dentre outros possíveis modificadores estão o *LIMIT* e *OFFSET* que delimitam o conjunto de resultados a ser retornado.

Além do *SELECT*, há outros 3 tipos de cláusulas de retorno de consultas que podem ser usadas. São elas:

- **ASK** – verifica se há pelo menos um resultado que combine com o *graph pattern* da consulta e retorna *true* ou *false*;

- **DESCRIBE** – retorna um grafo RDF que descreve um recurso. A implementação desta cláusula é dependente do mecanismo de processamento de consultas e por isso é utilizada com menos frequência que as demais;
- **CONSTRUCT** – retorna um grafo RDF criado a partir de um *template* especificado na consulta. A cláusula utiliza os valores que combinam com o *graph pattern* para construir o novo grafo segundo o template especificado. CONSTRUCT é usual na transformação de informações de um vocabulário para outro.

O protocolo SPARQL permite consultas através de requisições HTTP (*HiperText Transfer Protocol*), caracterizando um endpoint SPARQL como um simples serviço web que implementa o protocolo SPARQL.

SPARQL permite a extração de dados, exploração e descoberta de relações desconhecidas através de consultas, sendo uma das maneiras mais diretas de extração semântica. Na sua versão 1.1, permite atualização de dados, uso de funções de agregação, subconsultas e consultas federadas, tornando-se uma linguagem ainda mais robusta.

2.2.4 Anotação, Inferência e Extração

Anotação, Inferência e Extração são três tarefas de aplicação das tecnologias básicas da Web Semântica que possibilitam a compreensão do significado e a realização de tarefas automáticas pelas máquinas. Nas seções seguintes, uma rápida visão sobre tais tarefas são apresentadas.

2.2.4.1 Anotação Semântica

Anotação Semântica trata-se de uma associação de conceitos a documentos ou partes de um documento, otimizando a indexação e a recuperação de informações em documentos pelos mecanismos automáticos de busca(ELLER, 2008). As recomendações da W3C definem que os conceitos devem ser especificados em ontologias e as anotações estruturadas com RDF, de maneira que os documentos ou partes deles possam ser descritos.

Na figura 8 está representada visualmente as anotações de um pequeno texto com conceitos armazenados em um repositório semântico. Tais anotações podem ser realizadas de forma intrusiva ou não-intrusiva, através de processos automáticos, semi-automáticos ou híbridos de anotação.

As anotações intrusivas são mantidas junto ao conteúdo anotado. Já as não-intrusivas, são armazenadas em repositório ou arquivo separado do conteúdo. A W3C recomenda a utilização do RDF para a criação das anotações semânticas. Mais especificamente para anotações em páginas web (XHTML e HTML5) ou documentos baseados em XML, a W3C recomenda o *RDFa* para anotações intrusivas, porém, destacam-se o *Microformats*⁸, proposto em 2005, e mais recentemente, 2011, o *HTML5 Microdata*⁹, uma iniciativa da Google, Microsoft e Yahoo, que assemelha-se ao *microformats* e usa o *schema.org* como vocabulário base para anotações.

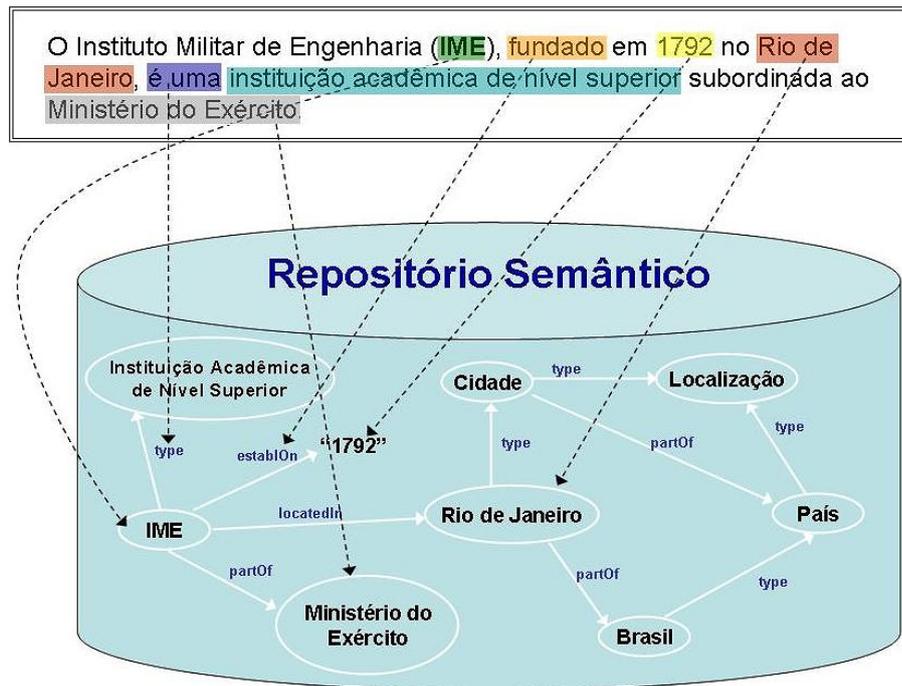


Figura 8: Representação Visual de Anotação Semântica (FONTES; MOURA; CAVALCANTI, 2010)

Os processos de anotação semântica são, ainda, um desafio na Web Semântica. Os processos automáticos esbarram nas limitações da identificação e classificação de entidades, apresentando resultados nem sempre satisfatórios. Já os processos semi-automáticos, que dependem do julgamento humano, são exaustivos pela enorme quantidade de conteúdos disponíveis na Web, lentos e estão sujeitos a falhas humanas (SOARES; TANAKA; BAIÃO, 2010).

Existem ferramentas híbridas, ou seja, que utilizam-se tanto das técnicas de inteligência artificial como do julgamento humano, para criação das anotações. Porém, assim como os demais processos, as ferramentas híbridas são limitadas à determinados tipos de conteúdos e formatos, bem como apresentam restrições de vocabulários a serem utilizados para as anotações.

⁸<http://microformats.org/>

⁹<http://www.w3.org/html/wg/drafts/microdata/master/>

A figura 9 apresenta um comparativo entre ferramentas de anotações quanto ao tipo de processo, linguagem da ontologia utilizada, entrada de ontologia¹⁰, representação da anotação, evolução da ontologia¹¹, formato do documento de entrada e tipo de anotação.

Nome da Ferramenta	Tipo de Ferramenta	Linguagem da Ontologia	Entrada de Ontologias	Representação das Anotações	Evolução de Ontologia	Formato do Documento	Tipo de Anotação
Ont-O-Mat	Semi-automática	DAML + OIL/RDF	Apenas uma por tarefa	Dentro de páginas da web	Sim	HTML	Intrusivo
MnM	Híbrida	DAML + OIL/RDF	Uma ou mais de uma por tarefa	Dentro de páginas da Web e em uma base de conhecimento	Sim	HTML, texto	Intrusivo
KIM	Automática	RDF/OWL	Ontologia do KIM (Proton)	Dentro de páginas da web	Sim	HTML	Intrusivo
Smore	Semi-automática	OWL	Uma ou mais de um por tarefa	Dentro de páginas da web	Sim	HTML, e-mail, imagens, texto	Intrusivo
Annotea	Semi-automática	RDF	Uma ou mais de um por tarefa	Servidores de anotações	Não	HTML	Não-intrusivo

Figura 9: Comparativo entre ferramentas de anotação semântica(ELLER, 2008)

De fato, anotações semânticas são uma das mais importantes tarefas para o sucesso de uma aplicação web semântica, pois é através das relações entre conceitos e conteúdos disponíveis na web que é possível a busca e extração de informações por significado.

2.2.4.2 Extração Semântica

Descrevemos a recuperação de informações em documentos anotados semanticamente, seja com base em um conjunto de regras ou não, como extração semântica.

A extração pode ser realizada com ferramentas que obtêm as informações anotadas de forma intrusa em um documento ou em bases RDF por meio de consultas SPARQL e expõem as informações obtidas, preferencialmente, em RDF por ser um modelo estrutural básico na Web Semântica.

¹⁰Se a entrada da ontologia para anotação é configurável ou pré-definida, se uma ou mais ontologias podem ser utilizadas no processo de anotação de uma vez

¹¹Se a ontologia pré-definida pode ser evoluída

Como exemplo de extração semântica, considere o trecho de código HTML anotado semanticamente com RDFa apresentado na figura 10.

```

1: <div vocab="http://schema.org/" typeof="Person">
2:   <a property="image" href="http://manu.sporny.org/images/manu.png">
3:     <span property="name">Manu Sporny</span></a>,
4:   <span property="jobTitle">Founder/CEO</span>
5:   <div>
6:     Phone: <span property="telephone">(540) 961-4469</span>
7:   </div>
8:   <div>
9:     E-mail: <a property="email"
10:      href="mailto:msporny@digitalbazaar.com">msporny@digitalbazaar.com</a></div>
11:   <div>
12:     Links: <a property="url" href="http://manu.sporny.org/">Manu's homepage</a>
13:   </div>
14: </div>

```

Figura 10: Trecho de código HTML contendo anotações em RDFa.

As informações contidas no trecho da página cujo código é exibido na figura 10 são apresentadas ao usuário no navegador como ilustrado na figura 11.

[Manu Sporny, Founder/CEO](#)
 Phone: (540) 961-4469
 E-mail: msporny@digitalbazaar.com
 Links: [Manu's homepage](http://manu.sporny.org/)

Figura 11: Informações exibidas ao usuário pelo navegador web.

Utilizando-se de uma ferramenta de extração, pode-se obter, em RDF, as informações anotadas em um documento, possibilitando desta forma a execução de consultas SPARQL, bem como, a inferência de possíveis informações implícitas. A figura 12 apresenta as informações em RDF extraídas do código HTML anotado com RDFa (figura 10).

```

@prefix rdfa: <http://www.w3.org/ns/rdfa#> .
@prefix schema: <http://schema.org/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

<http://rdfa.info/play/>
  rdfa:usesVocabulary schema: .
_:1
  rdf:type schema:Person;
  schema:image <http://manu.sporny.org/images/manu.png>;
  schema:name "Manu Sporny";
  schema:jobTitle "Founder/CEO";
  schema:telephone "(540) 961-4469";
  schema:email <mailto:msporny@digitalbazaar.com>;
  schema:url <http://manu.sporny.org/> .

```

Figura 12: RDF extraído do trecho de código HTML da figura10.

2.2.4.3 Inferência

A inferência é o processo de raciocínio no intuito de explicitar fatos ou relações a partir de um conjunto adicional de informações sobre o vocabulário ou conjunto de regras. Por exemplo:

- Se tivermos a seguinte afirmação (instância): “Beethoven é um Cachorro.’;
- E na ontologia tivermos o axioma que diz: “Cada Cachorro é também um Mamífero.’;
- Pode-se inferir que: “Beethoven é um Mamífero.’.

O processo de raciocínio pode ser implementado por mecanismos de inferência, também conhecidos como motores de inferência, ou nos mecanismos de processamento de consultas.

Um mecanismo de inferência deve ser capaz de inferir novas triplas ou axiomas a partir de um conjunto de instâncias recebido (*ABox Inference*) ou uma ontologia (*TBox Inference*), como também, ser capaz de raciocinar sobre os conceitos e instâncias baseando-se em regras definidas pelo usuário (*Rule Inference*). Como exemplo de tecnologias para realização de inferência na Web Semântica, pode-se citar os mecanismos de inferência Pellet, Hermit e JFact e a linguagem SWRL¹²(*Semantic Web Rule Language*).

Alguns mecanismos de processamentos de consultas incorporados nos repositórios RDF implementam opções de inferência usados na extração de informações através de consultas SPARQL. Exemplos destes repositórios são o AllegroGraph¹³, Virtuoso¹⁴, BigOWLIM¹⁵, Sesame¹⁶ e Jena TDB¹⁷.

Dentro da Web Semântica, inferência permite que processos automáticos estabeleçam novos conjuntos de regras e relações, alterando o fluxo de execução e o comportamento do sistema ou aplicação baseando-se na compreensão do significado dos dados disponíveis, sejam eles explícitos ou implícitos.

¹²<http://www.w3.org/Submission/SWRL/>

¹³<http://www.franz.com/agraph/allegrograph/>

¹⁴<http://virtuoso.openlinksw.com/>

¹⁵<http://www.ontotext.com/owlim>

¹⁶<http://www.openrdf.org/>

¹⁷<http://jena.apache.org/documentation/tdb/>

2.3 Integração Semântica de Informações

Tecnologias da Web Semântica têm sido empregadas na estruturação e integração de dados com o objetivo de romper barreiras da heterogeneidade estrutural, sintática e semântica. Um dos pontos-chave da integração de informações é o estabelecimento de relações entre os elementos do esquema e os elementos do modelo semântico formalizado (ZHOU et al., 2010). Neste sentido, a utilização de ontologias e das operações de coordenação¹⁸, como o alinhamento, soluciona o problema da heterogeneidade semântica de dados sem preocupação com a origem e forma de armazenamento destes dados(WACHE et al., 2001)(ZHOU et al., 2010).

De uma perspectiva semântica, as relações entre os elementos podem ser 1) automaticamente inferidas usando os axiomas do RDFS/OWL, 2) identificados através de regras definidas pelo usuário ou 3) derivadas de resultados de consultas SPARQL (KEENEY et al., 2011).

2.3.1 Utilizando Axiomas OWL

Nesta abordagem, uma ontologia é criada para integrar conhecimentos estruturados em outras ontologias ou alteram-se as ontologias existentes adicionando axiomas que relacionem os conceitos das diferentes ontologias.

Se considerarmos a aplicação desta abordagem para a integração de informações armazenadas em fontes de dados FOAF (*Friend Of A Friend*) e DBLP (*Digital Bibliography & Library Project*), podemos então relacionar as propriedades foaf:knows e dblp:acquaintance simplesmente adicionando um dos seguintes axiomas:

```
(dblp:acquaintance
  owl:equivalentProperty foaf:knows)
ou
(dblp:acquaintance
  rdfs:subPropertyOf foaf:knows)
```

A diferença entre a representação com owl:equivalentProperty e rdfs:subPropertyOf é que a segunda impõe um baixo acoplamento semântico. Por exemplo, de cada instância de DBLP (X dblp:acquaintance Y) infere-se uma FOAF (X foaf:knows Y), porém, partindo da instância de FOAF não é possível inferir (X dblp:acquaintance Y).

¹⁸A realização automática de operações de coordenação de ontologias pode ser sintetizada como processos que buscam similaridades ou dissimilaridades e estabelece, ou não, relações entre os conceitos especificados formalmente nas ontologias.

Esta abordagem é uma das mais simples e, em geral, a primeira a ser considerada para a integração. No entanto, a mesma requer conhecimento prévio no desenvolvimento de ontologia e as ontologias disponíveis para realizar as alterações e inferências necessárias.

2.3.2 Utilizando Regras de Usuário

Para realizar a integração com regras, é preciso defini-las com o intuito de declarar as relações que seriam adicionadas pelos axiomas. Para o exemplo dado na seção anterior, teremos a seguinte regra equivalente:

```
dblp:acquaintance(?x, ?y) -> foaf:knows(?x, ?y)
```

A utilização de regras é mais flexível que a abordagem por axiomas, pois não requer alterações nas ontologias. Além disso, as linguagens para definição de regras apresentam maiores recursos para expressar as relações entre conceitos e instâncias dos mais diversos vocabulários formalizados em ontologias.

2.3.3 Utilizando consultas SPARQL

Com SPARQL é possível extrair dados de uma fonte e transformar estes dados de um vocabulário para outro com consultas do tipo CONSTRUCT.

Para exemplificar a utilização de consultas na integração, considere o exemplo adotado para demonstrar as outras abordagens no qual deseja-se obter dados da base DBPL como dados FOAF. Neste sentido, a consulta CONSTRUCT a ser executada na base DBPL será:

```
PREFIX dblp: <http://prefix.cc/dblp>
PREFIX foaf: <http://xmlns.com/foaf/>

CONSTRUCT {
    ?x foaf:knows ?y
}WHERE {
    ?x dblp:acquaintance ?y
}
```

Na consulta, para todas as triplas que satisfazem o *graph pattern* do SPARQL WHERE são usadas na construção do grafo especificado em CONSTRUCT.

A vantagem da utilização de consultas na integração dos dados é que pode-se obter dados diretamente das fontes, porém, por ser uma linguagem de consulta de dados RDF, todas as inferências devem ter sido previamente materializadas nas fontes. A materialização prévia das inferências garante o melhor desempenho para abordagem quando comparada às demais.

Devido a heterogeneidade semântica dos vocabulários, ou seja, um conceito pode apresentar relações de significado com diversos outros, e ao fato de que dados em diferentes vocabulários podem estar armazenados em fontes distribuídas, a utilização de reescrita de consulta e execução distribuída das mesmas tem recebido ênfase em aplicações de integração de informações na Web, cujas fontes apresentam heterogeneidade e granularidade, e caracterizam-se como fontes de dados autônomas.

Reescrita de consulta são baseadas em mapeamento ou alinhamentos de vocabulários, de maneira que é possível traduzir uma consulta de um vocabulário para outros e fundir os resultados da execução das consultas em diferentes fontes de dados. A seção 5.2.6 deste documento demonstra como consultas são reescritas.

3 *SOA E SERVIÇOS WEB*

Este capítulo tem por objetivo apresentar fundamentos em SOA (*Service-Oriented Architecture*), caracterização e composição de serviços, além das principais tecnologias em serviços web.

3.1 *Service-Oriented Architecture (SOA)*

Por trás da substituição de tarefas manuais por sistemas de computadores, tem-se o objetivo de atingir resultados como agilidade nos processos, redução de custos, inovação em produtos, melhor atendimento ao cliente, dentre outros, que visam melhorar o desempenho da instituição, aumentando a competitividade e lucros.

Recursos tecnológicos têm sido empregados pelas instituições em diversas áreas para melhorarem seus desempenhos. Em consequência disto, o aumento constante de sistemas computacionais acarretam a necessidade de integrá-los. Outro problema é que muitos dos sistemas podem apresentar funções ou regras de negócio iguais, e em caso de alterações, pode-se demandar bastante tempo para refletir as alterações em todos os sistemas, em casos mais extremos, sendo inviável a realização das alterações. A figura 13 ilustra o cenário, onde os blocos representam os sistemas e as linhas indicam a comunicação entre eles para realização de um processo de negócio.

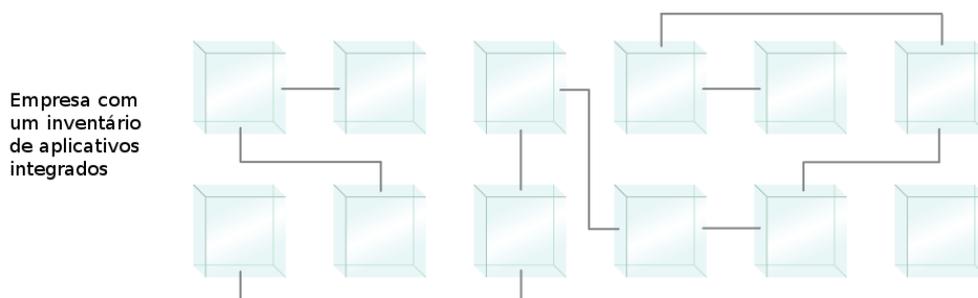


Figura 13: Cenário de Processo de Negócio sem SOA(ERL, 2009)

Arquitetura Orientada a Serviços (SOA, do inglês *Service-Oriented Architecture*) é um modelo arquitetônico, portanto, independente de tecnologia, que visa aprimorar a eficiência, a agilidade e a produtividade de uma instituição, alinhando os objetivos estratégicos à infraestrutura de tecnologia da informação (ERL, 2009).

Considerando o cenário ilustrado pela figura 13, em uma solução SOA, os sistemas aplicativos são construídos em função de partes lógicas independentes que podem ser combinadas em diferentes processos, aumentando a reusabilidade e interoperabilidade da infraestrutura. A figura 14 exemplifica o cenário com SOA.

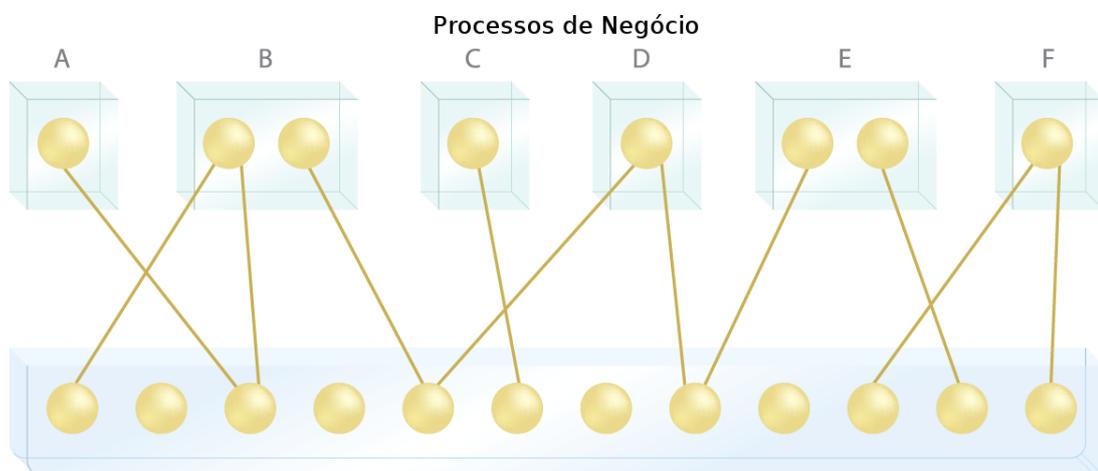


Figura 14: Cenário de Processo de Negócio com SOA(ERL, 2009)

No cenário, os processos de negócio A e B, assim como, D e E, compartilham partes lógicas independentes utilizadas por ambos os processos. Desta forma, as funções providas pelos serviços são compartilhadas nos diferentes processos e facilmente integrados e reutilizados em novos processos que vierem à ser construídos.

SOA segue os princípios da orientação a serviços, paradigma de projeto que tem como unidade fundamental o *serviço*.

3.2 Serviços

Segundo Thomas Erl(2009), serviços são programas fisicamente independentes, com características de projeto distintas que dão suporte à obtenção dos objetivos estratégicos associados à computação orientada a serviços.

De acordo com a OASIS¹ (*Organization for the Advancement of Structured Informa-*

¹<https://www.oasis-open.org>

tion Standards), em seu modelo de referência para SOA (OASIS STANDARD, 2006), um serviço é um mecanismo que permite acesso à uma ou mais capacidades, onde o acesso é provido usando uma interface pré-estabelecida e é exercido de acordo com as restrições e políticas especificadas na descrição do serviços.

Com base nestas definições, pode-se ressaltar que serviços apresentam características de projeto distintas e que possuem uma descrição, também chamada de contrato, utilizada para explicitar informações de acesso e restrições às suas capacidades. Assim, serviços seguem o paradigma “*Find, Bind and Execute*”, no qual, o consumidor do serviço pode encontrar e executar serviços automaticamente utilizando-se de um diretório de serviços e de contratos/descrições publicadas pelos provedores de serviços.

A figura 15 ilustra o fluxo das informações entre provedor, consumidor e o diretório de serviços que armazena os registros dos serviços (contratos).

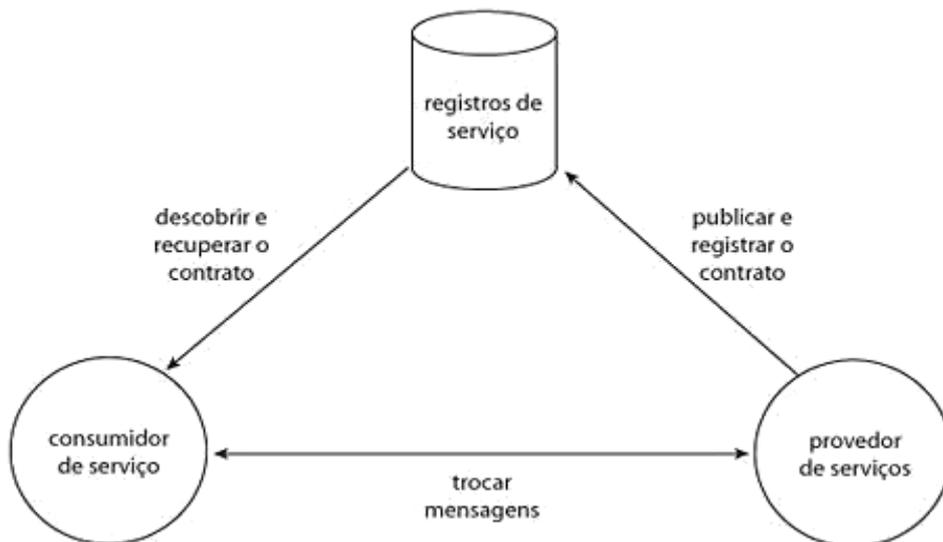


Figura 15: Paradigma “*Find, Bind and Execute*” (ERL, 2009)

O provedor disponibiliza seus serviços e registra-os no diretório enviando os contratos dos serviços disponibilizados. Um consumidor então realiza a descoberta de serviços no diretório e recupera o contrato do serviço que deseja. Utilizando as informações do contrato, o consumidor troca mensagens com o provedor do serviço para a realização de uma operação.

3.2.1 Tipos básicos de serviços

Serviços são classificados na literatura em 3 categorias (JOSUTTIS, 2008) (ERL, 2009). Sendo elas:

- **Entidade** – Assemelham-se às classes em programação orientada a objetos, serviços entidade, também conhecidos como serviços básicos por alguns autores, representam conceitos principais em uma organização ou domínio. É considerado um serviço altamente reutilizável porque é agnóstico à maior parte dos processos de negócio;
- **Tarefa** – Serviços que representam tarefas do processo de negócio, sendo os responsáveis por garantir precondições de outros serviços. São serviços de baixo potencial de reutilização, porém, quando reutilizados, obtém-se grande ganho;
- **Utilitário** – Serviços que não se encaixam diretamente aos processos de negócios e também não estão associados aos principais conceitos do domínio. Exemplos de serviços utilitários são os serviços para conversão de unidades de medida ou validação de uma data. Estes serviços podem assumir caráter mais tecnológico ou de infraestrutura e possuem alta reutilização.

3.2.2 Características de serviços

Seguindo o conjunto de princípios de projeto do paradigma de orientação a serviços, as soluções construídas como serviços apresentam uma série de características que possibilitam o sucesso de SOA. Dentre estas características estão:

- **Padronização** – Contratos de serviços devem seguir os mesmos padrões de projeto estabelecendo um inventário de serviços padronizado;
- **Baixo acoplamento** – Serviços tendem a ser fracamente acoplados, ou seja, em geral, são desacoplados de seu ambiente adjacente, não apresentando dependências com outros serviços e consumidores;
- **Abstração** – Apenas informações realmente necessárias para que o serviço seja funcionalmente útil aos consumidores são publicadas nos contratos de serviços. Serviços abstraem uma lógica e aspectos funcionais de seus clientes;
- **Reúso** – Serviços contêm e expressam lógica e podem ser considerados recursos reutilizáveis de uma organização;

- **Autonomia** – Devido às características anteriores, serviços têm a confiabilidade, o desempenho e previsibilidade comportamental para suportar a execução autônoma, exercendo um alto controle sobre a lógica e recursos em tempo de execução;
- **Independência de Estado** – Serviços são projetados para minimizar o tempo em que eles existem em uma condição de dependência de informações de estado, de modo que, a lógica do serviço leva à realização de uma ação com o estado atual da informação, o que aumenta a escalabilidade do serviço;
- **Visibilidade** – Serviços são desenvolvidos para serem automaticamente descobertos e efetivamente interpretados, de modo que seu propósito e capacidades sejam compreendidas claramente;
- **Composibilidade** – Serviços podem participar de composições, tornando flexível a construção de processos independentemente do tamanho e complexidade.

As características dos serviços permitem o alcance de benefícios como maior interoperabilidade, federação, mais opções de fornecedores, maior alinhamento de negócio e tecnologia, maior agilidade organizacional, menor carga de trabalho do setor de TI e maior retorno de investimento (ERL, 2009).

3.3 Composição de serviços

Uma das principais características de serviços em SOA, e que deve ser tomada como meta de realização, é a capacidade de participarem de uma composição com outros serviços.

A composição de serviços trata da combinação de um conjunto de serviços para que os mesmos alcancem objetivos que superam suas capacidades individuais, realizando um processo de negócio, ou seja, realizem um conjunto de tarefas de um processo na organização.

Na literatura, são descritos 3 modelos de implementação de composição de serviços:

- **Orquestração** – Composição realizada com uma descrição executável do processo, onde as interações com os serviços participantes são definidas previamente de acordo com as tarefas no processo de negócio. O processo descrito é visível ao consumidor como um serviço.

A figura 16 apresenta um exemplo de orquestração.

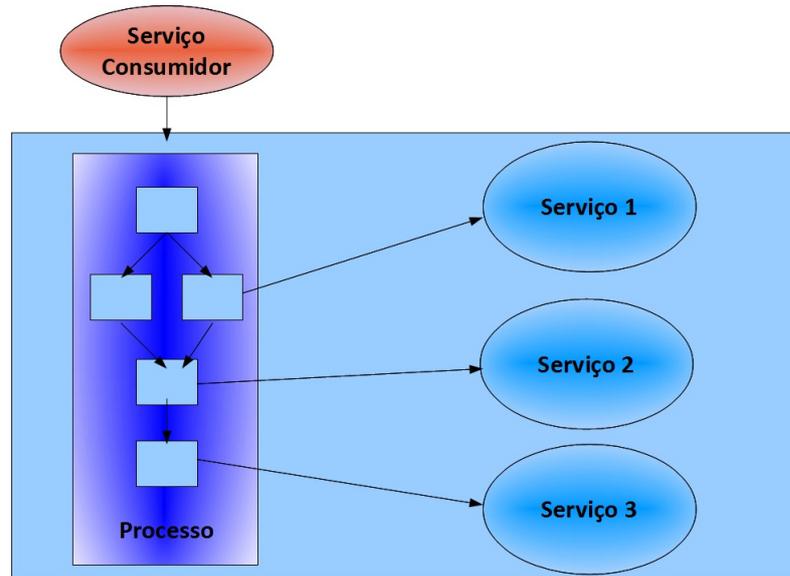


Figura 16: Exemplo de Orquestração de Serviços (ACQUAVIVA, 2011)

- **Coordenação** – Na qual a interação com os serviços participantes é realizada por um serviço específico, denominado *coordenador* e, em geral, não se tem uma sequência definida na troca de mensagens entre o coordenador e os participantes. A figura 17 ilustra a coordenação envolvendo dois serviços participantes.

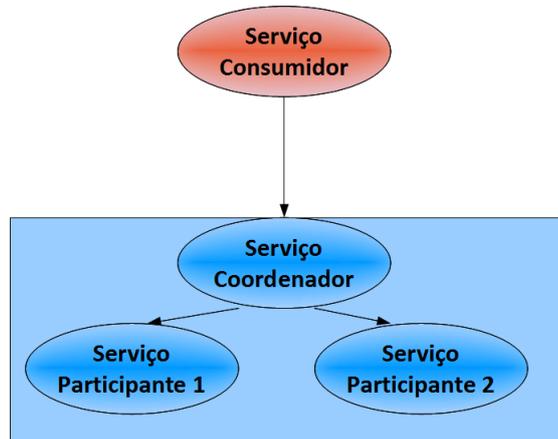


Figura 17: Exemplo de Coordenação de Serviços (ACQUAVIVA, 2011)

A coordenação é bastante útil quando o processo a ser implementado envolve operações complexas para utilização de serviços de terceiros, dificultando a implementação por orquestração de serviços.

- **Coreografia** – trata da composição de serviços sem a utilização de um centralizador, seja um coordenador ou descrição executável do processo, sendo de responsabilidade de cada

participante assumir o seu papel na colaboração, portanto, todos os serviços participantes sabem previamente a ação a ser realizada e quando fazê-la. A figura 18 ilustra uma coreografia de serviços.

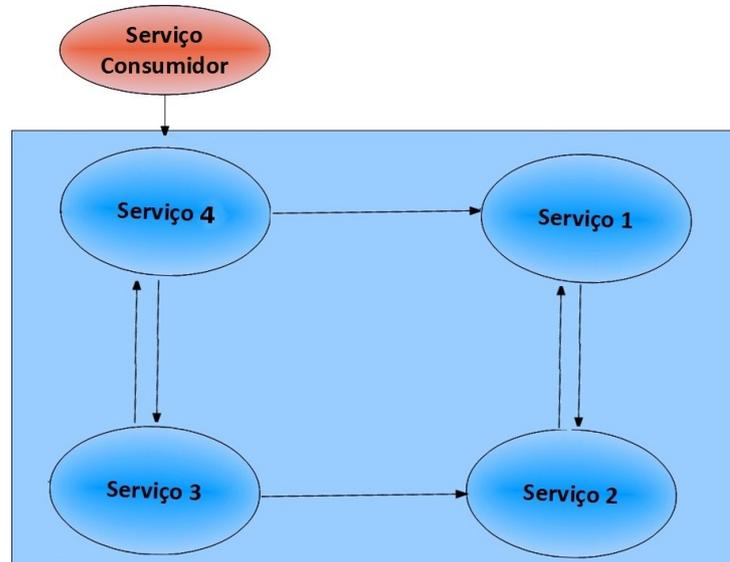


Figura 18: Exemplo de Coreografia de Serviços

A orquestração é uma das primeiras técnicas a serem consideradas, pois a implementação do processo é mais abstrata que na coordenação, ou seja, alterações são feitas a nível de processo sem a preocupação com detalhes da comunicação com serviços participantes.

A orquestração também apresenta algumas vantagens em relação à coreografia, como, por exemplo, gerenciamento dos participantes de forma centralizada, incorporação de novos serviços sem que eles estejam cientes de estarem participantes de um processo de negócio e a possibilidade de implementação de cenários alternativos em caso de falhas (ALBRESHINE; FUHRER; PASQUIER, 2009).

3.4 Tecnologias em Serviços Web

SOA não determina tecnologias para a implementação dos serviços, no entanto, tecnologias web são empregadas com sucesso devido o baixo acoplamento e uso de padrões abertos que aumentam a interoperabilidade das soluções.

A figura 19 apresenta algumas das principais tecnologias em serviços web para desenvolvimento de SOA.

Composição/processos WS-BPEL, WS-CDL	
Coordenação/Transações WS-Coordination, WS-Transaction, WS-BusinessActivity	
Descrição WSDL, XML schema, WS-Policy	Publicação UDDI
Mensagem SOAP, WS-Security, WS-ReliableMessaging, WS-Addressing	
Transporte HTTP, HTTPS, SMTP	Formato XML

Figura 19: Padrões em serviços web

Dentre as tecnologias de descrição, publicação e mensagem, pode-se destacar como as mais básicas:

- **Web Services Description Language (WSDL)²** – é uma linguagem baseada em XML, utilizada para descrever serviços web em termos de mensagens que podem ser trocadas entre o serviço e o consumidor, assim como, descrever as interfaces dos serviços de forma independente de plataforma;
- **Universal Description, Discovery and Integration (UDDI)³** – trata-se de um serviço de diretório onde as descrições WSDL dos serviços podem ser publicadas pelos provedores de serviços, permitindo que consumidores possam descobrir e recuperar contratos/descrições de serviços a serem utilizados;
- **Simple Object Access Protocol (SOAP)⁴** – é um protocolo para trocas de mensagens utilizado na comunicação entre provedores e consumidores de serviços. O SOAP tem suas mensagens baseadas no formato XML e utiliza-se de tecnologias como o HTTP para o transporte das mesmas.

²<http://www.w3.org/TR/wsdl>

³http://uddi.org/pubs/uddi_v3.htm

⁴<http://www.w3.org/TR/soap/>

No cenário simples de interação entre provedor e consumidor de serviços, estas tecnologias estão dispostas como ilustrado na figura 20.

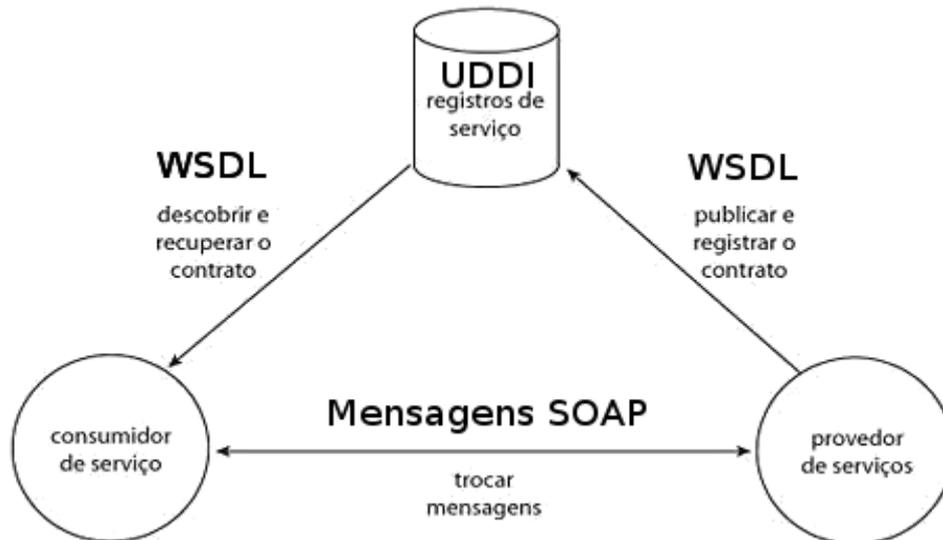


Figura 20: Utilização de tecnologias web em serviços – Adaptado de (ERL, 2009)

Serviços web também podem ser implementados com REST (*Representational State Transfer*), técnica de Engenharia de Software que utiliza-se de URI e requisições HTTP para construção de serviços web.

Em REST, além de não existir monitoramento de estados na comunicação, todos os recursos devem possuir identificadores únicos, serem acessíveis pela URI/URL e terem suas operações padronizadas. As operações são executadas através da URI do recurso associado a um método HTTP (GET, POST, PUT e DELETE) na requisição.

REST permite múltiplas representações, ou seja, saída em diferentes formatos, como JSON⁵ (*JavaScript Object Notation*), XML e RDF, e apresenta vantagens de performance quando comparado ao SOAP.

Os padrões WS-* da OASIS tratam de uma série de especificações relacionadas a serviços web que abordam diversos aspectos, como segurança e composição. Dentre estes padrões da OASIS, destaca-se a WS-BPEL⁶ (*Web Services Business Process Execution Language*), uma linguagem baseada em XML para representação de orquestração de serviços web e execução de processos de negócio. Além de todas as características que serviços podem apresentar, associados às tecnologias web, serviços têm a capacidade de reuso e interoperabilidade potencializados (ERL, 2009).

⁵<http://www.json.org/>

⁶<http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>

4 TRABALHOS RELACIONADOS

Dentre os trabalhos que foram analisados, não foi encontrado nenhum abordando a construção da solução para integração semântica de informações com SOA, somente alguns trabalhos que envolvem serviços web na solução construída. Assim, esta seção apresenta alguns trabalhos semelhantes à proposta de pesquisa, tratando da aplicação de tecnologias web semântica para a integração de informações.

4.1 SWIS – *Semantic Web-based Integration System*

O SWIS (*Semantic Web-based Integration System*), proposto por Cutunda, Fuentes-Lorenzo e Sánchez(2011), realiza integração com a abordagem de ontologia global e faz uso de serviços RESTful para tornar os recursos acessíveis.

O sistema é composto por dois módulos, os quais são:

- **Módulo de Mapeamento** – Cujos dados estruturados de fontes heterogêneas são mapeados com o domínio global definido pela ontologia;
- **Módulo de Acesso** – Pelo qual os usuários possuem acesso aos resultados de consultas.

O SWIS suporta os seguintes tipos de consultas:

- `http://[server]/[ontology]` – Obtém o conjunto de classes de uma certa ontologia [ontology];
- `http://[server]/[ontology]/[class]` – Obtém o conjunto de indivíduos de uma certa classe de uma ontologia.
- `http://[server]/[ontology]/[class]/[value]` – Obtém todas as propriedades e valores das mesmas, para um indivíduo particular identificado por [value];

- [http://\[server\]/\[ontology\]/\[class\]/\[value\]/\[prop\]](http://[server]/[ontology]/[class]/[value]/[prop]) – Obtém o conjunto de indivíduos que se relacionam através da *Object Property* [prop] com um indivíduo específico identificado por [value].
- [http://\[server\]/\[ontology\]/\[class\]/by/\[prop\]/\[value\]](http://[server]/[ontology]/[class]/by/[prop]/[value]) – Obtém o conjunto de indivíduos de uma certa classe que possuem determinado valor [value] para uma determinada propriedade [prop] do tipo *Datatype Property*.

4.2 ISENS

O ISENS (DIMITROV et al., 2009) é um sistema de integração de informação, exploração e consulta de fontes de dados de várias ontologias. O sistema consiste de:

- Integrador de informações baseado em ontologia (*Ontology-Based Information Integrator* – OBII);
- Interface gráfica de usuário *Web-based*;
- Conjunto de ontologias em OWL¹(*Web Ontology Language*);
- Fontes de dados com indivíduos das ontologias (armazenadas utilizando o Sesame²);
- Mapeamentos entre ontologias realizados com a OWLII, uma linguagem de mapeamento inspirada na OWL *Description Logic* e compatível com o algoritmo de seleção da fonte, ambos, linguagem e algoritmo, desenvolvidos pelos autores;
- Metadados sobre relevância das fontes de dados – informações utilizadas pelo algoritmo para a seleção das fontes mais relevantes à consulta.

As consultas são manipuladas pelo integrador (OBII), que se baseia no mapeamento entre as ontologias e as fontes de dados, nos metadados de relevância das fontes e nas ontologias que descrevem os dados para determinar quais são as fontes alvo da consulta, gerar consultas individuais e executá-las utilizando o mecanismo de inferência KAON2³ com os dados das fontes selecionadas.

Apesar de um desenvolvimento bastante elaborado, não há indicação da sua utilização em um cenário real.

¹<http://www.w3.org/TR/owl2-overview/>

²<http://www.openrdf.org/>

³<http://kaon2.semanticweb.org/>

4.3 IAIF – Integrated Agriculture Information Framework

Shoaib e Basharat (2010) apresentam o IAIF, um framework cujo principal objetivo é permitir a extração de conhecimento de vários domínios relacionados à agricultura.

O framework proposto pelos autores foca-se na combinação, fusão e agregação de dados de sistemas de informação ligados à agricultura, através do uso de metadados e ontologias de domínio. A arquitetura é mostrada na Figura 21.

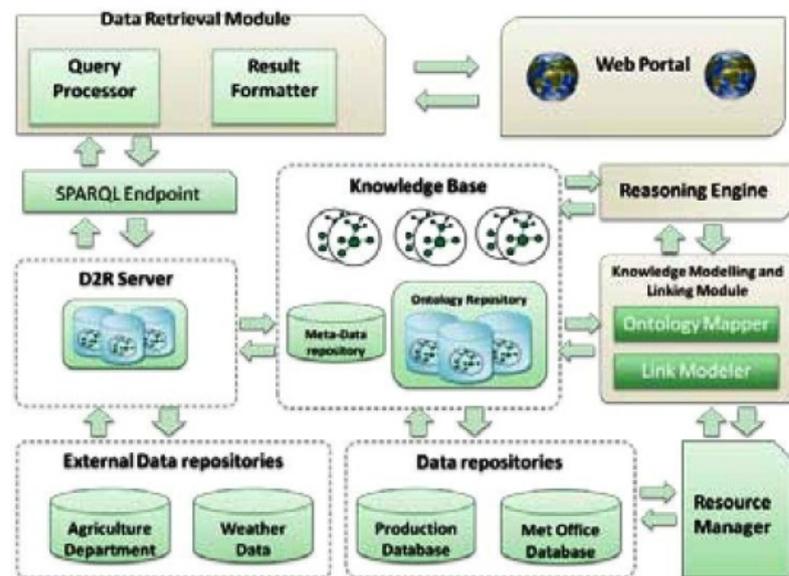


Figura 21: Arquitetura do Integrated Agriculture Information Framework (SHOAIB; BASHARAT, 2010)

O IAIF possui uma arquitetura híbrida dividida nos seguintes módulos:

- Módulo de recuperação de dados – fornece suporte à consulta e extração de conhecimento. O módulo é desenvolvido utilizando como base o Jena⁴;
- D2R Server⁵ – ferramenta que torna possível a publicação de bases de dados relacionais de maneira que permite a execução de consultas em SPARQL⁶.
- Base de conhecimento – componente constituído de uma ontologia central contendo apenas terminologias, sem instâncias, que é mapeada utilizando-se regras com as ontologias que representam as fontes de dados. Além da ontologia central, o componente contém um repositório de ontologias que representam os conhecimentos nas fontes de conhecimentos

⁴<http://jena.sourceforge.net>

⁵<http://www4.wiwiw.fu-berlin.de/bizer/d2r-server/>

⁶<http://www.w3.org/TR/rdf-sparql-query/>

integradas ao framework, e um repositório de metadados sobre as fontes de conhecimentos.

- Gerenciador de Recursos e Motor de Inferência – tais componentes são responsáveis, respectivamente, pela gerência das bases de conhecimento e inferência de novos conhecimentos a partir de um conhecimento existente.

Dentre as tecnologias utilizadas para o desenvolvimento estão o Jena, JESS⁷ e SWRL⁸.

4.4 Pesquisa em tecnologias web semântica para a integração de conhecimento em produção agrícola

Xiang et al. (2009) apresentam um framework semelhante ao proposto por Shoaib e Basharat (2010), que possibilita a integração de conhecimento relacionado à produção agrícola utilizando tecnologias web semântica. A Figura 22 ilustra a arquitetura.

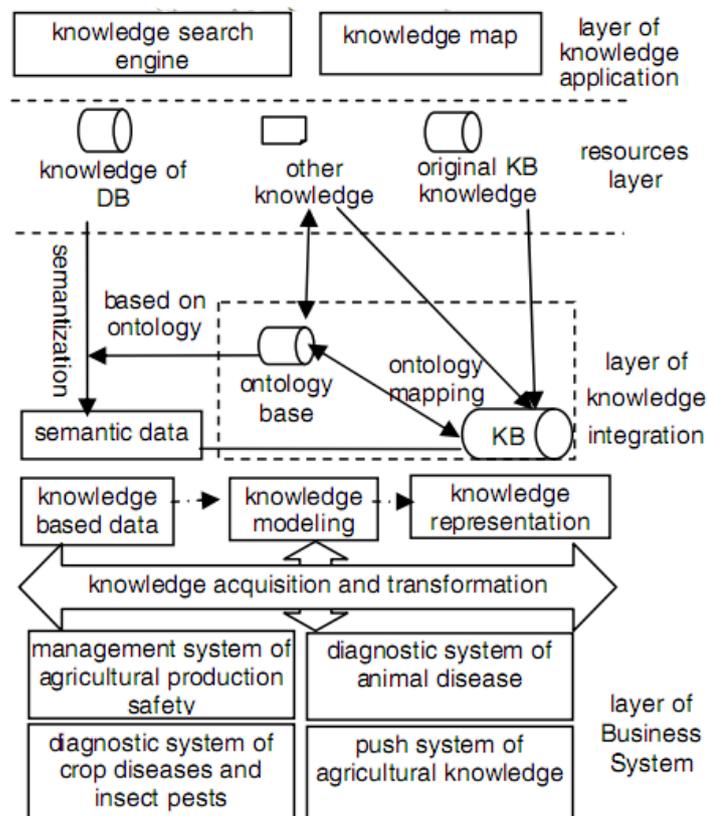


Figura 22: Arquitetura do framework de integração de conhecimento em produção agrícola (XIANG et al., 2009)

⁷<http://www.jessrules.com/>

⁸<http://www.w3.org/Submission/SWRL/>

O framework é dividido em 4 camadas:

- **Camada de Recursos** – constituída de bases de dados com algum conhecimento existente;
- **Camada de Integração de Conhecimento** – composta por um repositório de conhecimento e uma base de ontologias, utilizadas para o mapeamento e semantização dos dados das bases na camada de recursos;
- **Camada de Aplicação de Conhecimento** – fornece aplicações como mecanismo de pesquisa de conhecimento e mapa de conhecimento;
- **Camada de Sistemas de Negócios** – composta por uma coleção de vários sistemas de negócio ligados à produção agrícola, como: sistema de gerência de segurança da produção, sistema de diagnóstico de doenças e pestes na colheita, etc. A aquisição e transformação do conhecimento disponível nestes sistemas é realizada por uma sequência de passos, dentre eles, a conversão e mapeamento entre os dados e a ontologia global da produção agrícola.

Um diferencial entre este *framework* e o trabalho de Shoaib e Basharat (2010) é a transformação de regras de sistemas especialistas para utilização sobre o conhecimento nas ontologias. No entanto, o trabalho é descrito superficialmente e o processo de aquisição e transformação de conhecimentos nos sistemas de informação ligados à produção agrícola não é abordado e nem referenciado.

4.5 Asio Scout

Asio Scout⁹ é um conjunto de ferramentas proprietárias e não-gratuitas, que permitem a construção de pontes entre fontes de dados e tecnologias web semântica. A solução é propriedade da empresa Raytheon BBN's *Semantic Web Applications* e constitui-se das seguintes ferramentas:

- **Asio Semantic Bridge for Relational Databases (SBRD)** – ferramenta que traduz consultas em SPARQL para SQL e o conjunto de dados resultado para RDF¹⁰ (*Resource Description Framework*);

⁹http://www.bbn.com/technology/knowledge/asio_scout

¹⁰<http://www.w3.org/RDF/>

- **Asio Semantic Bridge for Web Services (SBWS)** – ferramenta que traduz consultas SPARQL para entradas em WSDL¹¹ (*Web Services Description Language*) e os resultados da execução dos serviços para RDF;
- **Asio Semantic Query Decomposition (SQD)** – Módulo que divide as consultas SPARQL submetidas à ontologia de domínio, para as fontes de dados individuais apropriadas, ou seja, aquelas cujas ontologias da base foram mapeadas com a ontologia de domínio). Os resultados são reunidos e retornados ao usuário sem expor as fontes de dados.

A interação das ferramentas citadas e o SQD para integração de dados é ilustrada na Figura 23.

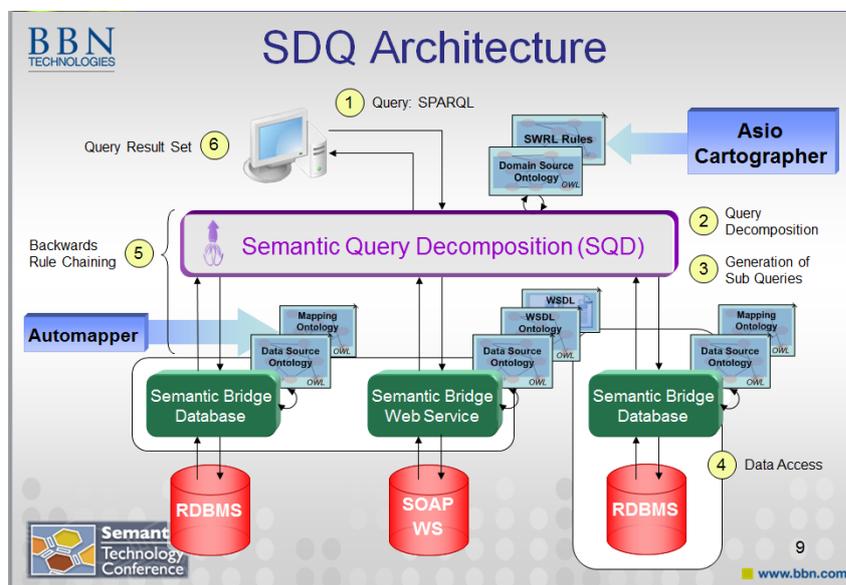


Figura 23: Arquitetura do Asio Semantic Query Decomposition (SQD)¹²

Para utilização do Asio Scout é necessário criar uma ontologia (OWL) da fonte de dados e mapear tal representação com uma ontologia de domínio. Este processo é facilitado com o uso das ferramentas Automapper¹³ (FISHER; DEAN; JOINER, 2008) e Snoggle¹³. A primeira é uma ferramenta baseada na ontologia da D2RQ¹⁴ para gerar automaticamente uma ontologia da fonte de dados. Já a segunda trata-se de uma ferramenta web baseada em SWRL (*Semantic Web Rule Language*) para mapeamento entre ontologias(OWL), cuja interface permite interação do usuário no estilo arrastar-e-soltar (*Drag and Drop*).

¹¹<http://www.w3.org/TR/wsdl>

¹²http://bbn.com/technology/knowledge/asio_sqd

¹³<http://projects.semwebcentral.org/projects/snoggle/>

¹⁴D2RQ é uma linguagem declarativa utilizada para descrever mapeamentos entre esquemas de bases de dados relacionais e ontologias OWL/RDFS. Mais informações no endereço <http://www4.wiwiw.fu-berlin.de/bizer/d2rq/>

4.6 Análise comparativa

No intuito de avaliar os trabalhos apresentados em relação à integração semântica e abordagem de construção da solução, consideremos os seguintes aspectos:

1. **Heterogeneidade das fontes de dados** – Se a solução permite a integração de diferentes tipos de fontes de dados, como por exemplo, banco de dados relacionais, documentos na web, serviços web, bases de conhecimento (bases RDF) e ontologias OWL;
2. **Abordagem de Integração Semântica** – Determina qual das abordagens de integração semântica foi utilizada na solução, se via axiomas, regras ou consultas (Para mais detalhes, veja seção 2.3);
3. **Utilização de Ontologias** – Se a integração envolve mais de uma ontologia (*Multi-ontology*) ou concentra-se em apenas uma ontologia referenciada como global (*Single-Ontology*);
4. **Flexibilidade nas consultas** – Determinar quais soluções predeterminam as consultas possíveis aos dados integrados ou se o usuário pode elaborar suas próprias consultas;
5. **Seleção de fontes de dados na integração** – Se durante a integração, é realizada uma seleção das potenciais fontes de dados a serem integradas para a resolução de uma consulta com o objetivo de otimizar o tempo de resposta ao usuário;
6. **Suporte a extração e conversão de dados** – Se há suporte das soluções à tais tarefas que, como visto na seção 2.2.4, são importantes para aumentar o volume de informações disponíveis e a qualidade das respostas à serem atendidas;
7. **Utilização de Inferência** – Se em algum ponto da solução utiliza-se inferência, pois como relatado anteriormente, a inferência explícita informações nos dados disponíveis, o que auxilia na qualidade das respostas às consultas;
8. **Suporte a anotação** – Se as soluções oferecem algum suporte a anotação semântica de dados;
9. **Estratégia de decomposição da solução** – Como as soluções foram construídas quanto à paradigma e estratégia de decomposição utilizadas, se baseado em componentes, módulos ou serviços;
10. **Interface com usuários** – Como a solução apresenta-se ao usuário que deseja realizar integração semântica de informações;

Comparando os trabalhos apresentados e a plataforma ARIDUS proposta neste documento, temos o resultado exibido na tabela 1.

Tabela 1: Comparativo dos Trabalhos Relacionados

X	Soluções							Asio Scout
	ARIDUS	SWIS	ISENS	IAIF	Xiang			
Aspectos								
1	Sim	Não	Não	Não	Sim		Sim	Sim
2	Consultas	Regras	Regras	Regras	Regras		Regras	Consultas
3	<i>Multi-ontology</i>	<i>Single</i>	<i>Multi-ontology</i>	<i>Multi-ontology</i>	<i>Multi-ontology</i>		<i>Multi-ontology</i>	<i>Multi-ontology</i>
4	Sim	Não	Sim	Sim	Sim		Sim	Sim
5	Sim (em 2 níveis)	Não	Sim	Sim	Não		Não	Sim
6	Sim	Não	Não	Não	Sim (Conversão)		Sim (Conversão)	Não
7	Sim (parcialmente)	Não	Sim	Sim	Sim		Sim	Não
8	Não	Não	Não	Não	Não		Não	Não
9	Serviços	Módulos	Módulos	Módulos	Módulos		Módulos	Módulos
10	<i>Services and GUI Web-based</i>	Serviço (REST)	<i>GUI Web-based</i>	<i>GUI Web-based</i>	<i>GUI Web-based</i>		<i>GUI Web-based</i>	–

Podemos verificar que a ARIDUS apresenta muitas semelhanças entre o trabalho de Xiang et. al (2009) e o Asio Scout. No entanto, a ARIDUS se sobressai nos aspectos relacionados à seleção de fontes de dados, suporte a extração e conversão de dados em RDF, utilização de inferência, interface com usuários e diferencia-se das por ser uma solução construídas segundo os princípios da orientação a serviços, fazendo com que a plataforma apresente características como flexibilidade, interoperabilidade e reuso.

5 PLATAFORMA ARIDUS

Este capítulo descreve a plataforma ARIDUS. Na seção 5.1, é apresentada a visão conceitual da plataforma, abordando os requisitos e a arquitetura em camadas da mesma. Na seção 5.2, são descritos os serviços presentes em cada camada, bem como, o detalhamento daqueles em funcionamento no protótipo. Por fim, na seção 5.4, trata-se da extensão da plataforma com modelos para prover personalização e acessibilidade.

5.1 Requisitos e Arquitetura

A ARIDUS foi concebida sob os princípios que a tornam uma infraestrutura aberta e flexível para a integração semântica de informações em diferentes domínios do conhecimento, armazenadas em diversas fontes na Web. Deste modo, para lidar com as questões de interoperabilidade em ambientes computacionais heterogêneos e permitir o reuso de suas funcionalidades, a plataforma ARIDUS foi definida em camadas e adota tecnologias web e princípios do paradigma de orientação a serviço.

A arquitetura, de forma conceitual, é apresenta na figura 24.

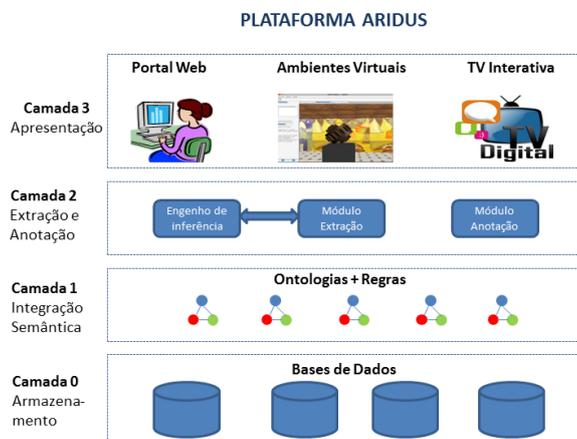


Figura 24: Arquitetura da Plataforma ARIDUS

As camadas definidas são:

0. Camada de Armazenamento

Nesta camada, estão situadas as fontes de dados com diferentes tipos e formatos de documentos e dispostas em diferentes localidades. Desta forma, o principal objetivo dos serviços que constituem esta camada é garantir acesso às informações das mais variadas fontes de maneira transparente.

Tendo em vista o cenário cujas informações de diferentes domínios de conhecimento encontram-se em diferentes fontes, de tipos e formatos diferentes, tradutores (*wrappers*) e conversores são utilizados para fornecer o acesso às informações. Em geral, os tradutores são utilizados em dados estruturados ou semi-estruturados, e os conversores aplicados aos conteúdos binários, como, por exemplo, arquivos de imagens.

1. Camada de Integração Semântica

Esta camada é responsável pela integração semântica das informações, ou seja, pelo mapeamento de conceitos entre os diferentes domínios de conhecimento especificados em ontologias. Para tal, a camada dispõe de serviços para coordenação de ontologias, gerência de alinhamentos, gerência de metadados sobre as fontes de dados, reescrita de consultas de um vocabulário para outros, bem como serviço para conversão do formato de ontologias.

2. Camada de Anotação e Extração

Camada idealizada com o objetivo de aumentar o volume de informações a ser integrada, bem como, fornecer um meio para obter um conjunto de informações específicas através de consultas ou extração de informações anotadas em determinados conteúdos. Desta forma, os seguintes aspectos devem ser considerados:

- (a) Anotação – Serviço que permite realizar anotação automática de conteúdos diversos, por exemplo, textos, imagens, áudios e vídeos, com a possibilidade de definir qual o conjunto de ontologias que serão utilizadas no processo;
- (b) Extração – Serviço que obtém semanticamente informações sobre recursos existentes nas diversas bases registradas na plataforma, assim como, extrai informações anotadas em documentos de diversos tipos e formatos;
- (c) Conversão – Pequenos mecanismos para gerar triplas RDF a partir da estrutura sintática de conteúdos nos principais formatos existentes.

Os processos automáticos de anotação e extração podem ser auxiliados por um mecanismo de inferência com o objetivo de explicitar conhecimento que relaciona conhecimentos já explicitados nas diferentes fontes de informação.

Esta camada auxilia a integração lidando com a granularidade de informações contidas em pequenos conteúdos disponibilizados em várias mídias e formatos.

3. Camada de Apresentação

Nesta camada, é imprescindível que os dados integrados e serviços na plataforma possam ser utilizados em diferentes ambientes, como por exemplo: Web, Ambientes Virtuais e TV Digital Interativa.

Na seção seguinte, são apresentados os serviços que atendem aos requisitos relatados nesta seção.

5.2 Serviços

Para nortear o desenvolvimento da plataforma, adotou-se a metodologia SOMA (*Service-Oriented Methodology and Architecture*). A escolha foi fundamentada no trabalho de (LOPES, 2009) que realiza um comparativo entre as metodologias SOMA, SOUP (*Service-Oriented Unified Process*) e MSOAM (*Mainstream SOA Methodology*) e identifica um conjunto mínimo de artefatos necessários para o projeto e construção de software com arquitetura orientada a serviços. Além disso, faz a seleção, avaliação e o agrupamento das boas práticas presentes nestas metodologias.

A metodologia SOMA apresenta 4 fases, sendo:

- **Identificação de Serviços** – Foram identificadas um conjunto de funcionalidades candidatas a serviços. Para tanto, atividades como a decomposição de domínio em áreas funcionais, processos, subprocessos e tarefas, análise de ativos existentes e modelagem de meta-serviços foram realizadas com o intuito de definir um conjunto de funcionalidades que poderiam ser oferecidas como serviços.

Para a ARIDUS, foram elaborados alguns modelos de processo de negócio (em BPM - *Business Process Management*) e análise de diferentes plataformas, aplicativos e serviços apresentados ou referenciados em diferentes fontes de pesquisa, bem como nos trabalhos relacionados apresentados neste documento, possibilitando a definição de uma lista de serviços candidatos, categorizados logicamente/funcionalmente por camada da plataforma.

- **Especificação de Serviços** – A fase de especificação é uma das fases mais engessadas da metodologia, e, portanto, em adequação ao projeto ARIDUS, a fase resumiu-se na análise dos serviços candidatos sob os critérios ressaltados na metodologia, como, por exemplo, alinhamentos aos negócios, viabilidade de implementação e composibilidade destes serviços, tendo como resultado a seleção dos serviços a serem construídos, a identificação das composições a serem realizadas e a representação das mesmas.
- **Realização de Serviços** – Esta fase consiste, principalmente, no tempo para a conclusão de um conjunto de tarefas que resultam na integração, construção ou terceirização dos serviços. Dentre essas atividades estão a integração de serviços existentes, transformação de funcionalidades presente em softwares e plataformas para serviços, compra, terceirização ou construção de serviços especificados. Para a ARIDUS, durante esta fase, foram tomadas as decisões referentes à construção dos serviços, dentre elas, a decisão das tecnologias adotadas.
- **Implantação de Serviços** – Fase dedicada à disposição dos serviços para execução, na qual serviços desenvolvidos foram implantados em máquinas para desenvolvimento de estudos de casos e aplicações.

A tabela 2 apresenta os serviços identificados e os status dos mesmos em relação à realização.

Tabela 2: Serviços Candidatos

ID	Nome	Descrição	Camada	Status
1	<i>DataSet Manager</i>	Gerencia informações sobre as fontes de dados na plataforma.	1	R
2	<i>Alignment API (Aserve)</i>	Realiza alinhamento entre ontologias.	1	T
3	<i>OWL Syntax Converter</i>	Converte uma ontologia em OWL para outras sintaxes.	1	T
4	<i>Alignment Manager</i>	Gerência alinhamentos na plataforma	1	R
5	<i>Align And Import Process</i>	Realiza orquestração envolvendo o serviço WS101 e WS102 com o objetivo de alinhar e manter o mapeamento consultável com SPARQL.	1	R
6	<i>Query Rewriter</i>	Traduz uma consulta de um vocabulário para outro baseando-se no alinhamento entre eles.	1	R
7	<i>Semantic Reasoner</i>	Obtém um conjunto de triplas (fatos) a partir de outro.	2	NR
8	<i>IExtractor</i>	Extrai RDF de conteúdo enriquecido semanticamente.	2	T
9	<i>IAcquisition Process</i>	Realiza a conversão, enriquecimento e armazenamento de dados em RDF.	2	T
10	<i>Semantic Annotator</i>	Gera um conjunto de triplas RDF associando partes de um documento à conceitos de ontologia(s).	2	NR
11	<i>Query Executor</i>	Executa de consultas SPARQL em fontes de dados federadas (SPARQL Endpoints).	2	R
12	<i>Query Process</i>	Realiza orquestração dos serviços WS106, WS101 e WS202 com o objetivo de realizar a integração de dados.	2	R
13	<i>UserProfileManager</i>	Gerencia informações básicas do usuário na plataforma.	3	R

Status: R – Realizado; NR – Não-realizado; T – Terceirizado.

Camadas: 0 – Armazenamento; 1 – Integração Semântica; 2 – Anotação e Extração;

3 – Apresentação.

Os serviços realizados e terceirizados apresentados na tabela 2, detalhados nas subseções seguintes, são dispostos nas camadas da plataforma como ilustra a figura 25.

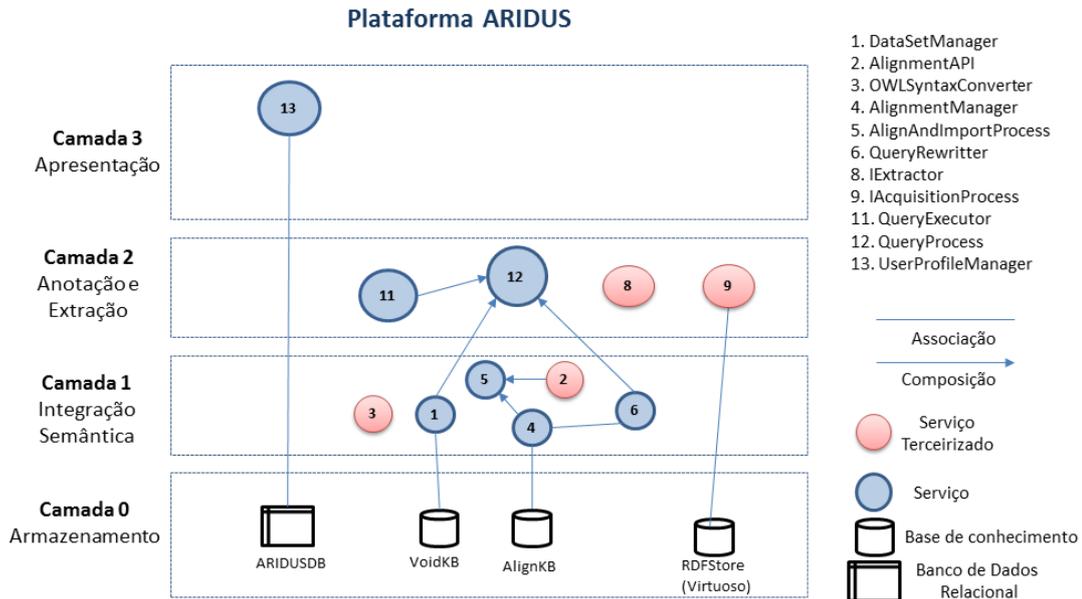


Figura 25: Disposição dos Serviços da Plataforma ARIDUS

Todos os serviços foram implementadas em SOAP e RESTful (*Representational State Transfer*), sendo que, na versão REST, utilizou-se a Jersey¹, a implementação de referência da especificação de serviços web em RESTful.

5.2.1 DataSetManager (1)

O *DataSetManager* gerencia informações sobre as fontes de dados registradas na plataforma. Tais fontes são descritas pelo vocabulário VoID² (*Vocabulary of the Interlinked Data-sets*), do qual, *foaf:homepage*, *dcterms:title*, *dcterms:description*, *void:sparqlEndpoint*, *void:vocabulary* e *void:triples* são os principais termos utilizados³.

O serviço dispõe de um conjunto de operações necessárias à manipulação dos registros das fontes de dados, basicamente, um CRUD⁴ que realiza operações na VoidKB, que é uma base RDF designada para o armazenamento de informações sobre as fontes de dados alvos da integração na plataforma ARIDUS.

¹<http://jersey.java.net/>

²<http://www.w3.org/TR/void/>

³Os prefixos adotadas representam os seguintes namespaces: void=<http://rdfs.org/ns/void#>, foaf=<http://xmlns.com/foaf/0.1/>, dcterms=<http://purl.org/dc/terms/>

⁴Acrônimo de *Create, Read, Update e Delete* em língua Inglesa.

Através deste serviço é possível obter informações sobre as fontes de dados, *endpoints* e vocabulários dos mesmos, tornando possível a seleção das fontes de dados pelos vocabulários que as utilizam. Para demonstrar, estão listadas abaixo, algumas operações possíveis proporcionadas pelos métodos disponíveis no serviço:

- ***create*** e ***delete*** – permite o cadastro e a exclusão de *datasets* na plataforma;
- ***createProp***, ***getProp***, ***updateProp*** e ***deleteProp*** – Métodos para realizar a manipulação de propriedades das fontes de dados. Estes métodos requerem a identificação da fonte de dados e da propriedade a ser manipulada (ou criada);
- ***listEndpointVocabs*** – Lista os vocabulários registrados para um *endpoint*;
- ***listEndpointsByVocab*** – Lista *endpoints* que utilizam um determinado vocabulário recebido como parâmetro;
- ***listEndpointsByVocabs*** – Lista *endpoints* que utilizam pelo menos um dos vocabulários passados como parâmetro;
- ***listEndpointsByUser*** – Lista os *endpoints* registrados por um usuário;
- ***listDatasetVocabs*** – Lista os vocabulários registrados para uma fonte de dados;
- ***listDatasetsByVocab*** – Lista as fontes de dados que utilizam um determinado vocabulário recebido como parâmetro;
- ***listDatasetsByVocabs*** – Lista as fontes de dados que utilizam pelo menos um dos vocabulários passados como parâmetro;
- ***listDatasetsByUser*** – Lista as fontes de dados registrados por um usuário;

Cada fonte de dados (*dataset*) é identificada por um URI, adotando-se o padrão `http://aridus.uern.br/{username}/dataset/{dcterms:title}`, onde {username} corresponde ao nome de usuário, o qual fez o registro, e {dcterms:title} correspondente ao título informado pelo usuário para a fonte de dados.

5.2.2 Alignment API (2)

Um meio de executar a integração semântica de informações é através do alinhamento entre vocabulários utilizados. Deste modo, a ARIDUS utiliza-se do alinhamento entre conceitos de dois ou mais ontologias para reescrever consultas SPARQL e buscar informações nas

diversas fontes de dados registradas na plataforma, que utilizam vocabulários semanticamente equivalentes. Neste sentido, é necessário um serviço que identifique e mapeie os conceitos equivalentes entre duas ontologias.

A *Alignment API* (DAVID et al., 2011) foi adotada como serviço de alinhamento (Aserv) na ARIDUS(WS102) por, dentre outras funcionalidades, permitir o alinhamento entre duas ontologias, representar alinhamentos em formato manipulável por diferentes ferramentas, apresentar alinhamentos em diferentes formatos (RDF, HTML, SWRL, SKOS, JSON, OWL Axioms e outros), dispor de suporte à EDOAL⁵ (*Expressive and Declarative Ontology Alignment Language*) para uso mais elaborado e futuras necessidades de representação, permitir a extensão dos métodos/algoritmos de alinhamento e encontrar-se disponível em <http://aserv.inrialpes.fr/>, tanto em REST quanto em SOAP.

Dentre as operações fornecidas pelo serviço, estão:

- ***match ontologies*** – realiza o alinhamento entre duas ontologias (em formato OWL/XML ou RDF/XML). Dentre os parâmetros estão os URI's acessíveis dos arquivos das ontologias e o método de alinhamento à ser utilizado. A relação dos métodos disponibilizados pode ser vista em <http://aserv.inrialpes.fr/rest/listmethods>;
- ***invert alignmet*** – gera o alinhamento inverso entre as ontologias participantes do alinhamento informado como parâmetro;
- ***store alignment*** – armazena o alinhamento em uma base relacional.
- ***finding alignments*** – busca alinhamentos na base utilizando como parâmetros os *namespaces* das ontologias das quais se deseja buscar alinhamentos;
- ***render alignment*** – apresenta o alinhamento em diferentes formatos, como por exemplo, os citados no segundo parágrafo desta seção;
- ***trim alignment*** – cada alinhamento contém um conjunto de equivalências (*maps*) as quais são compostas de células (*Cell*) formadas por um par de entidades (*entity1* e *entity2*), o relacionamento entre elas (*relation*) e a mensuração da relação (*measure*) que varia de 0 ao máximo 1. A operação *trim* elimina as equivalências cuja célula apresenta o valor de *measure* menor que o passado como parâmetro.

O *Alignment API* utiliza o NeOn Toolkit⁶, *state-of-the-art* em ambiente de engenharia de ontologia, e os métodos/algoritmos de alinhamento disponibilizados por ele. Ambos,

⁵<http://alignapi.gforge.inria.fr/edoal.html>

⁶<http://neon-toolkit.org>

Alignment API e NeOn Toolkit, são licenciados com LGPL (*Lesser General Public License*) e extensíveis com *plug-ins*.

5.2.3 *OWL Syntax Converter* (3)

É um serviço REST adotado na ARIDUS para converter uma ontologia cuja sintaxe não é aceita pelo *Alignment API*.

O *OWL Syntax Converter*⁷ é um serviço disponibilizado pelo grupo⁸ de pesquisa da universidade de *Manchester*, que se dedica ao desenvolvimento de linguagens de ontologias, ferramentas, metodologias e aplicações, com destaque para as contribuições diretas no desenvolvimento da OWL como um padrão W3C.

O serviço realiza a conversão de ontologias entre as principais sintaxes existentes, dentre elas RDF/XML, *OWL Functional Syntax*, *Manchester OWL Syntax*, *OBO Syntax* e *Turtle*.

5.2.4 *AlignmentManager* (4)

Com o intuito de tornar mais flexível a utilização dos alinhamentos na plataforma ARIDUS, importam-se todos os alinhamentos desejados para uma base RDF (*AlignKB*) e tem-se o serviço *AlignmentManager* onde, dentre outras operações disponíveis, encontra-se a *importAlign*.

Desta forma, ao contrário do que acontece no *Alignment API (Asevr)* – que armazena os alinhamentos em um banco de dados relacional e permite apenas um tipo de consulta (consultar alinhamentos que envolve determinada(s) ontologia(s)) – os alinhamentos na ARIDUS são manipuláveis com SPARQL, possibilitando consultas mais elaboradas, como por exemplo, o filtro de células com valor mínimo de *measure*, a consulta de alinhamentos que envolvem um conceito específico, bem como, a possibilidade de inferir novas triplas.

Além da operação *importAlign*, este serviço da ARIDUS dispõe de:

- *getVocabTarget* – recebe o URI de um alinhamento e retorna o *namespace* do vocabulário alvo do alinhamento;
- *getAlignments* – recebe o URI de um vocabulário e retorna uma lista de alinhamentos que o envolve;

⁷<http://owl.cs.manchester.ac.uk/converter/>

⁸<http://owl.cs.manchester.ac.uk/>

- *getMap* – recebe o URI de um alinhamento e retorna o conjunto das células que possuem valor de *measure* igual a 1.

Com a flexibilidade garantida pelo armazenamento em RDF, outras operações e serviços podem ser desenvolvidos com o objetivo de tornar ainda mais flexível a utilização dos alinhamentos presentes na ARIDUS.

5.2.5 *AlignAndImportProcess* (5)

O *AlignAndImportProcess* trata-se de uma orquestração dos serviços *AlignmentAPI* (2) e o *AlignmentManager* (4) automatizando o processo de alinhamento e importação do alinhamento para a base RDF na plataforma ARIDUS, com o objetivo de tornar os alinhamentos consultáveis com SPARQL e mais flexíveis de serem manipulados.

A orquestração foi construída em BPEL e tem como *engine* de execução, a Apache ODE (*Orchestration Director Engine*). Na figura 26 o processo é apresentado, bem como, a comunicação entre o processo e os serviços participantes.

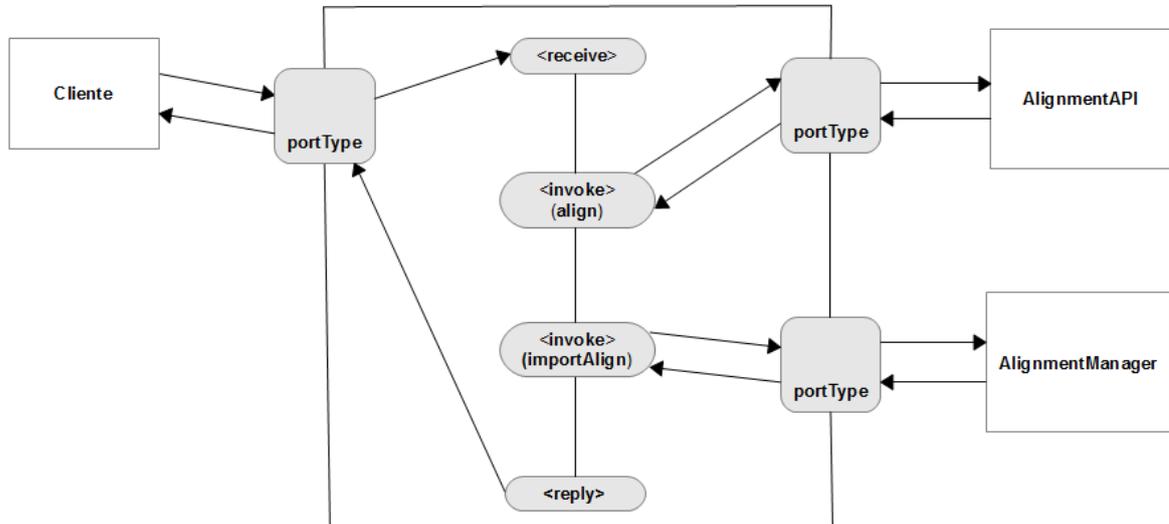


Figura 26: Orquestração *AlignAndImport*

5.2.6 *QueryRewriter* (6)

Dentre as abordagens de integração semântica discutidas, demonstradas e avaliadas por (KEENEY et al., 2011), na plataforma ARIDUS, adota-se a baseada em consultas SPARQL por apresentar-se como a mais rápida e eficiente em relação às outras abordagens – sendo elas:

utilização de axiomas OWL e a utilização de regras definidas por usuário. Deste modo, o serviço *QueryRewriter* (WS106) é responsável pela reescrita das consultas, parte essencial da integração de informações presentes nas diversas fontes de dados.

O *QueryRewriter* realiza a reescrita das consultas SPARQL baseando-se nos alinhamentos importados para a plataforma e armazenados na base *AlignKB*, permitindo desta forma, que uma consulta SPARQL seja executada em cada fonte de dados alvo com o vocabulário específico utilizado na fonte, que é equivalente ao usado na consulta inicial.

O processo de reescrita das consultas pelo *QueryRewriter* é baseado no modelo apresentado por (MAKRIS et al., 2010) composto de um conjunto de funções de reescrita do BGP (*Basic Graph Pattern*) de consultas. É importante ressaltar que tais funções preservam a semântica dos mapeamentos/alinhamentos durante o processo de reescrita das consultas. De forma geral, o processo de reescrita de uma consulta segue os passos descritos no Algoritmo 1, onde a função *Triple Pattern Rewriting* substitui o predicado, objeto ou sujeito por um conceito equivalente preservando o elemento caso o mesmo seja uma variável, literal ou propriedade RDF/RDFS/OWL/IRI.

Algoritmo 1 Reescrita de *Graph Pattern* – Traduzido de (MAKRIS et al., 2010)

Entrada: GP_{in} : *graph pattern* de entrada, M : conjunto de mapeamento.

- 1: Alocar GP_{out} para ser o resultado da reescrita de GP_{in} ;
 - 2: $GP_{out} \leftarrow GP_{in}$ depois de substituir IRI's (classe, propriedade ou indivíduo) dentro das expressões de FILTER utilizando mapeamentos (cardinalidade 1:1) de M ;
 - 3: **Para** cada tripla em GP_{out} **faça**
 - 4: $GP_{out} \leftarrow$ Triple Pattern Rewriting ($GP_{out}, M, predicado$);
 - 5: $GP_{out} \leftarrow$ Triple Pattern Rewriting ($GP_{out}, M, objeto$);
 - 6: $GP_{out} \leftarrow$ Triple Pattern Rewriting ($GP_{out}, M, sujeito$);
 - 7: **fim-para;**
 - 8: $GP_{out} \leftarrow GP_{out}$ após remoção de colchetes desnecessários;
 - 9: **retorne** GP_{out} ;
-

Para exemplificar o processo de reescrita adotado, considere a consulta e o mapeamento obtido através do alinhamento feito entre os vocabulários *person* e *foaf*, como mostra a figura 27.

Desta forma teremos:

- BGP de entrada: { ?s person:name ?name }
- BGP de saída: { ?s foaf:name ?name }

Na ARIDUS, o *QueryRewriter* informa como saída da reescrita, uma consulta que une

```

1 PREFIX person: <http://aridus.uern.br/person/>
2 SELECT DISTINCT ?name
3 WHERE{
4   ?s person:name ?name
5 }
6 LIMIT 10
7
8 # Equivalência obtida do alinhamento entre person e foaf
9 # <http://aridus.uern.br/person/name> = <http://xmlns.com/foaf/0.1/name>

```

Figura 27: Exemplo de consulta e equivalência entre conceitos usados como entrada no serviço *QueryRewriter*.

o BGP inicial e os obtidos através da reescrita com base nos alinhamentos identificados. Desta forma, teremos como saída final do serviço a consulta apresentada na figura 28.

```

1 PREFIX person: <http://aridus.uern.br/person/>
2 PREFIX v0: <http://xmlns.com/foaf/0.1/>
3 SELECT DISTINCT ?name
4 WHERE{
5   {?s person:name ?name }
6   UNION
7   { ?s v0:name ?name }
8 }
9 LIMIT 10

```

Figura 28: Consulta Exemplo da Saída do Serviço *QueryRewriter*

O serviço dispõe dos seguintes métodos:

- ***queryRewriter1*** – Recebe uma consulta SPARQL como parâmetro, lista os prefixos utilizados na consulta, obtém os alinhamentos que envolvem tais vocabulários e reescreve a consulta inicial para cada alinhamento obtido.

Por questões de desempenho, este método não foi implementado como um processo em BPEL (Orquestração envolvendo *AlignmentManager* e *QueryRewriter*). Em futuras versões, tal método será transformado em um processo implementado com BPEL.

- ***queryRewriter2*** – Recebe uma consulta SPARQL e um conjunto de equivalências (conceito A = conceito B) como parâmetros, reescreve a consulta com base no tal conjunto (mapa).
- ***queryRewriter3*** – Recebe uma consulta SPARQL e uma lista de URI de alinhamentos como parâmetros, obtém o conjunto de equivalências entre conceitos e reescreve a consulta para cada um dos alinhamentos da lista recebida.
- ***queryRewriter4*** – Recebe uma consulta SPARQL e um URI de um alinhamento, obtém o conjunto de equivalências entre os conceitos e reescreve a consulta.

- *listVocabs* – Recebe uma consulta SPARQL e lista os prefixos/vocabulários presentes na mesma.

A implementação do *QueryRewriter* foi realizada com o uso do Jena ARQ⁹, utilizando-se das classes `Query`, `AlgebraGenerator`, `Op`, `Prologue`, classes contidas no pacote `jena.sparql.algebra.op` e uma implementação da interface `Transform`.

5.2.7 *Semantic Reasoner* (7)

Como já dito anteriormente, um mecanismo de raciocínio deve ser capaz de inferir novas triplas ou axiomas a partir de um conjunto de instâncias recebido (*ABox Inference*) ou uma ontologia (*TBox Inference*), como também, ser capaz de raciocinar sobre os conceitos e instâncias baseando-se em regras definidas pelo usuário (*Rule Inference*). Neste sentido, um mecanismo de raciocínio pode requerer a manipulação de uma quantidade de dados razoável, o que pode exigir elevada capacidade de armazenamento e poder computacional.

Na plataforma ARIDUS, dada a dificuldade de raciocínio em dados distribuídos em várias fontes de dados, o serviço *Semantic Reasoner* (7), a princípio, não foi realizado. Porém, constatou-se que vários *triple stores* suportam algum tipo de raciocínio, o que permite a descoberta de conhecimentos implícitos nas fontes de dados. Como exemplos destes *triples stores*, pode-se citar o AllegroGraph¹⁰, Virtuoso¹¹, BigOWLIM¹², Sesame¹³ e Jena TDB¹⁴.

A Universidade de Manchester disponibiliza uma aplicação web chamada “*Reasoning On the Web*”¹⁵, a mesma permite a execução de consultas predefinidas em uma ontologia, assim como a definição de qual dos engenhos de inferência disponíveis deve ser utilizado, dentre as opções tem-se o Pellet, HermiT e JFact. A aplicação apresenta limitações de desempenho e bastante erros de execução. Para ilustrar, a figura 29 apresenta a interface da aplicação.

Apesar da não implementação do serviço *Semantic Reasoner* na plataforma ARIDUS, há uma cobertura parcial, porém sem problemas de desempenho na descoberta de conhecimentos implícitos nas fontes de dados.

⁹<http://jena.apache.org/documentation/query/arq-query-eval.html>

¹⁰<http://www.franz.com/agraph/allegrograph/>

¹¹<http://virtuoso.openlinksw.com/>

¹²<http://www.ontotext.com/owlim>

¹³<http://www.openrdf.org/>

¹⁴<http://jena.apache.org/documentation/tdb/>

¹⁵<http://owl.cs.manchester.ac.uk/row/>

Reasoning On the Web

JFact
▼

classhierarchy
▼

cls

depth

Figura 29: Interface de Usuário da Aplicação “Reasoning On The Web”.

5.2.8 (RDF Proxy Service) IExtractor (8) e IAcquisitionProcess (9)

A conversão e extração de dados tratam da obtenção de informações fragmentadas em diversos conteúdos dispersos na Web, aumentando o volume de informações disponíveis para a integração. Desta forma, a aquisição de informações, ou seja, conversão de dados para um formato comum e armazenamento dos mesmos, e a extração de informações implícitas em conteúdos são dois aspectos de significativa importância na integração de informações na Web, cujo volume de dados é crescente e cuja diversidade e granularidade é enorme.

Na ARIDUS, os serviços *IExtractor* e *IAcquisitionProcess* são responsáveis pela extração de informações de conteúdos semanticamente anotados, bem como, a conversão e armazenamento de informações obtidas de conteúdos na Web, permitindo a integração de informações fragmentadas e dispersas nos mais diversos conteúdos.

O *IExtractor* e *IAcquisitionProcess* são realizados pelo Virtuoso Sponger¹⁶, uma plataforma para transformação de dados para RDF. O Sponger realiza, por requisição, a transformação de dados, como páginas HTML (com ou sem microformatos), arquivos binários (texto, planilha, imagem, áudio e vídeo) ou dados advindos de serviços web, para o formato RDF.

O *Sponger* constitui o *middleware Virtuoso* e possui os seguintes módulos:

- **Cartridges** – São combinações de extratores de metadados e mapeador de ontologia, que associam partes ou informações sobre os dados do conteúdo a conceitos especificados em ontologias. As Cartridges atendem à necessidade de ter diferentes conversores dos mais diversos formatos de conteúdo para o formato RDF, simplificando o acesso e manipula-

¹⁶<http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VirtSponger>

ção, pois requer apenas um método para tal. No entanto, caso os dados na fonte original (conteúdo) sejam alterados, é preciso uma nova conversão e tratamento para atualizar os dados obtidos.

- **Meta-Cartridges** – realizam o enriquecimento dos dados gerados pelas *cartridges*.

Além disso, o Sponger utiliza outros componentes do Virtuoso – *Virtuoso Content Crawler* e *RDF Quad Store* – como ilustra a figura 30.

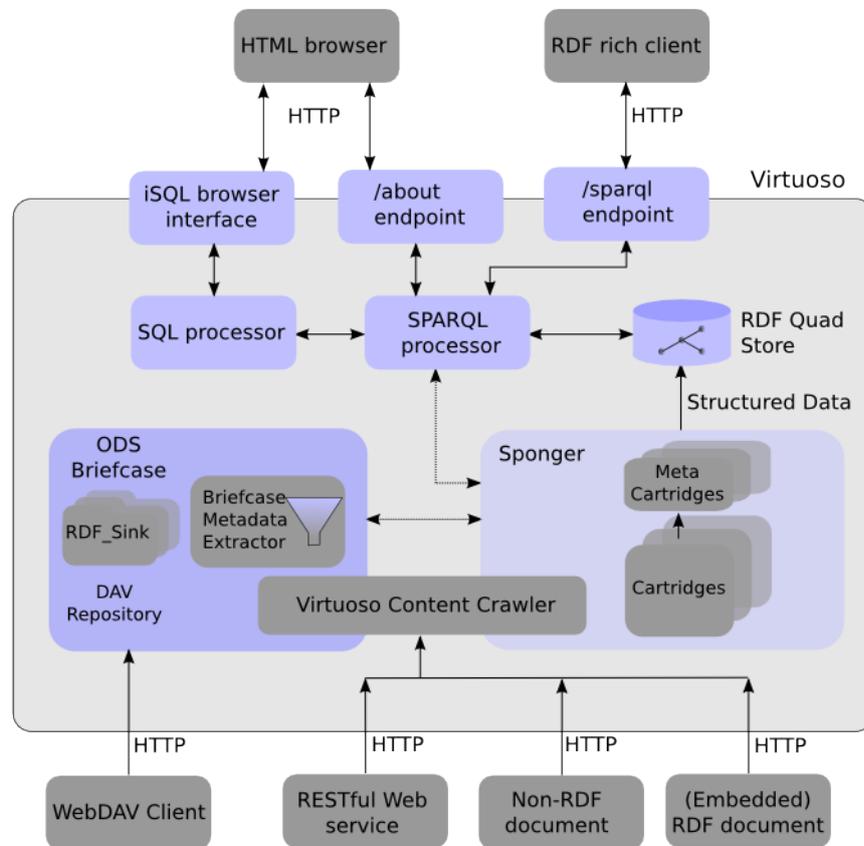


Figura 30: Virtuoso Sponger - Extração e Estruturação de Dados em RDF.

O Virtuoso Sponger dispõe do *RDF Proxy Service* cuja interface permite a extração (*IExtractor*) e/ou aquisição dos dados (*IAcquisitionProcess*). O *RDF Proxy Service* é construído em REST e possui dois métodos: `/about/rdf/` e `/about/html/`. Tais métodos realizam a extração ou conversão de um conteúdo web indicado pela URL na chamada do serviço, como por exemplo: `/about/rdf/http://uern.br/img/uern.png`.

Por padrão, os dados RDF obtidos da extração ou conversão dos conteúdos são armazenados na base RDF e acessíveis através do endpoint SPARQL (utilizado pela plataforma).

Um conjunto de formatos de arquivos e services web suportados pelo Virtuoso Sponger (*Cartridges* disponíveis) são descritos na documentação^{17,18}.

Na ARIDUS, utiliza-se o servidor Sponger acessível em `http://linkeddata.triburner.com/`, cujo *RDF Proxy* e o *endpoint* SPARQL estão disponíveis.

5.2.9 *Semantic Annotator* (10)

Anotação semântica em conteúdos dá-se pela declaração de relações entre trechos dos conteúdos e conceitos especificados em ontologias. Desta forma, um mesmo trecho de um documento pode apresentar relações com conceitos em diferentes domínios de conhecimento, e portanto, pode ser anotado semanticamente com conceitos alvos em diferentes ontologias, nos dando a possibilidade de analisar um mesmo dado sob diferentes perspectivas ou visões de conhecimento. Neste sentido, é notória a contribuição das anotações semânticas para as funcionalidades da plataforma ARIDUS.

A ARIDUS requer um serviço, denominado genericamente de *Semantic Annotator* (para fins de referência), capaz de estabelecer automaticamente relações entre partes de conteúdos e conceitos de ontologias alvos determinados pelo usuário. Além disso, o serviço possui os seguintes requisitos:

1. Anotação automática de conteúdos;
2. Suporte a conteúdo de diversos tipos em diferentes formatos;
3. Parametrização da ontologia a ser utilizada no processo de anotação;
4. Anotação semântica com múltiplas ontologias;
5. Suporte a interoperabilidade de anotações (armazenamento em RDF)
6. Suporte a conteúdos (textos) em diferentes idiomas (pelo menos em Português e Inglês);

A OAT (*Ontology Annotation Tool*)¹⁹ e a Plataforma KIM(POPOV et al., 2003) foram investigados no intuito de elencar ferramentas para a implementação ou terceirização do serviço

¹⁷<http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VirtSpongerCartridgeSupportedDataSourcesNonRDF>

¹⁸<http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VirtSpongerCartridgeSupportedDataSourcesVendorNonRDF>

¹⁹<http://gate.ac.uk/userguide/sec:ontologies:ocat>

de anotação semântica na ARIDUS. No entanto, o OAT (plugin do GATE²⁰ – *General Architecture for Text Engineering*) permite anotação manual de texto com múltiplas ontologias e atende somente aos requisitos 3, 4 e 5 da plataforma. Já a plataforma KIM, realiza anotação automática de texto para uma ontologia em predeterminada na plataforma, na qual apresenta, dentre os principais, os conceitos: *Person*, *Organization*, *Location*, *Money* e *Acquirement*. Deste modo, a plataforma KIM atende somente aos requisitos 1 e 2 identificados para o serviço de anotação da ARIDUS.

É importante ressaltar que o suporte a conteúdos de diversos tipos e anotação com múltiplas ontologias, implica em diferentes abordagens para realizar anotação e em uma representação elaborada das anotações para determinados tipos de mídia. Neste sentido, aponta-se o *Framework Open Semantic Annotation*²¹ que permite a interoperabilidade de anotações semânticas, representação de anotações realizadas com múltiplas ontologias, anotações temporais, espaciais e com restrições de tempos. O framework resolve possíveis problemas de interoperabilidade de anotações e padroniza as anotações semânticas nos diferentes tipos/formatos de dados.

Até o momento, a plataforma ARIDUS não possui um serviço de anotação que atende aos seus requisitos e, portanto, são utilizadas anotações manuais armazenadas em uma base RDF para validação de aspectos da ARIDUS que requerem anotações.

5.2.10 *Query Executor* (11)

O *QueryExecutor* é o serviço da ARIDUS responsável pela execução federada de consultas SPARQL nos diferentes endpoints, assim como, pela fusão dos resultados obtidos. Este serviço utiliza como base o *Framework FedX* (SCHWARTE et al., 2011a) cujo propósito é permitir o processamento eficiente de consultas em fontes distribuídas (*Linked Open Data*).

O FedX realiza otimizações na consulta de entrada e uma seleção prévia das fontes de dados alvos (SCHWARTE et al., 2011b). Tal seleção é feita com a execução de uma consulta SPARQL do tipo ASK para cada tripla do BGP da consulta de entrada, em cada uma das fontes alvos. O resultado das consultas ASK's são mantidas em cache para futuras execuções.

O *QueryExecutor* possui apenas um método que recebe como parâmetros de entrada, a consulta SPARQL e uma lista de endpoints alvos. O resultado do serviço adota como padrão a representação XML/SPARQL *Result*.

²⁰<http://gate.ac.uk/>

²¹<http://www.openannotation.org>

5.2.11 Query Process (12)

O *QueryProcess* é uma orquestração dos serviços *QueryRewriter*, *DataSetManager* e *QueryExecutor* automatizando o processo de reescrita, busca de endpoints alvos e consulta dos dados nos diferentes endpoints.

Assim como o *AlignAndImportProcess* (Seção 5.2.5), o *QueryProcess* foi construído em BPEL e tem como *engine* de execução, a Apache ODE (*Orchestration Director Engine*). Na figura 31, a orquestração é apresentada ressaltando a comunicação entre o processo e os serviços participantes.

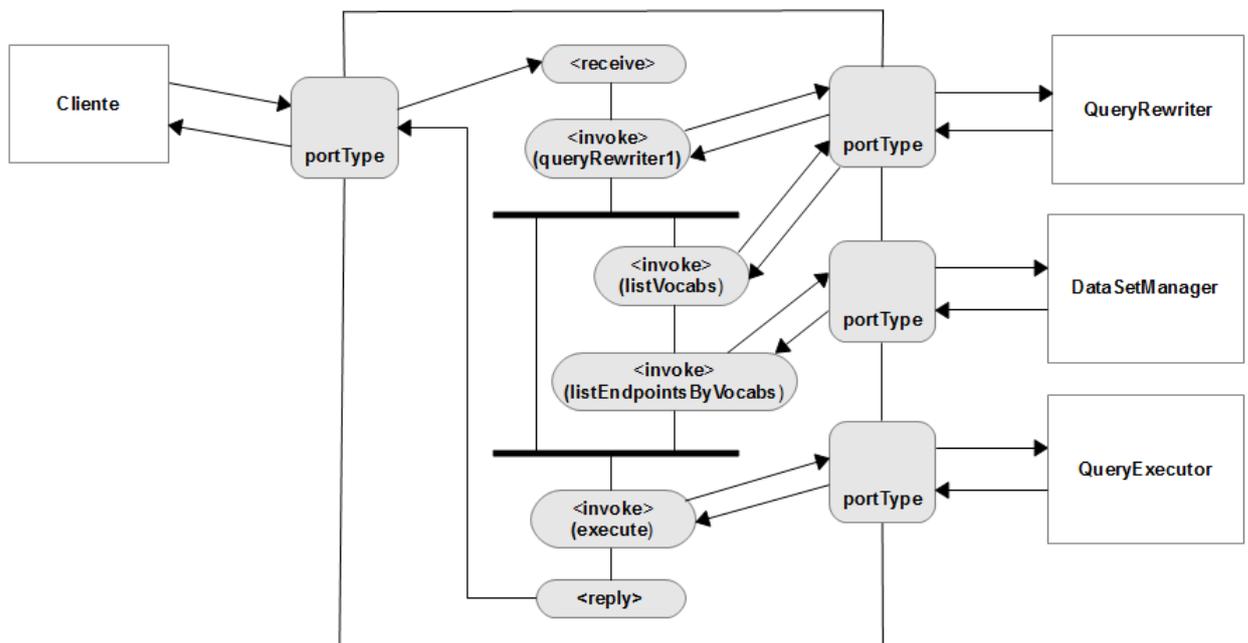


Figura 31: Orquestração *QueryProcess*.

5.2.12 UserProfileManager (13)

O serviço *UserProfileManager* gerencia as informações básicas de identificação do usuário e permite o controle de informações sobre as fontes de dados e documentos registrados pelo mesmo na plataforma, bem como, serve de base para aplicações externas e serviços de extensão da plataforma, descritos na seção seguinte.

O perfil, composto do nome de usuário (*username*), *e-mail*, senha (*password*) e a URI do perfil semântico do usuário (*uriprofile*), como, por exemplo, perfil estruturado com o vocabulário FOAF, é armazenado em uma tabela do banco de dados no MySQL.

O *UserProfileManager* possui operações para a inserção, busca, atualização e exclusão das informações do perfil do usuário na plataforma.

5.3 Integração Semântica de Informações com a ARIDUS

Antes de realizar a integração, são aplicadas tecnologias para fornecer acesso semântico aos dados das fontes estruturadas, semi-estruturas ou não-estruturadas. Para tal, são utilizados conversores e tradutores/wrappers que realizam a transformação dos dados para o modelo RDF, possibilitando o armazenamento dos dados em uma base RDF ou acesso aos dados diretamente da fonte original, respectivamente.

A figura 32 ilustra o cenário de integração com a ARIDUS.

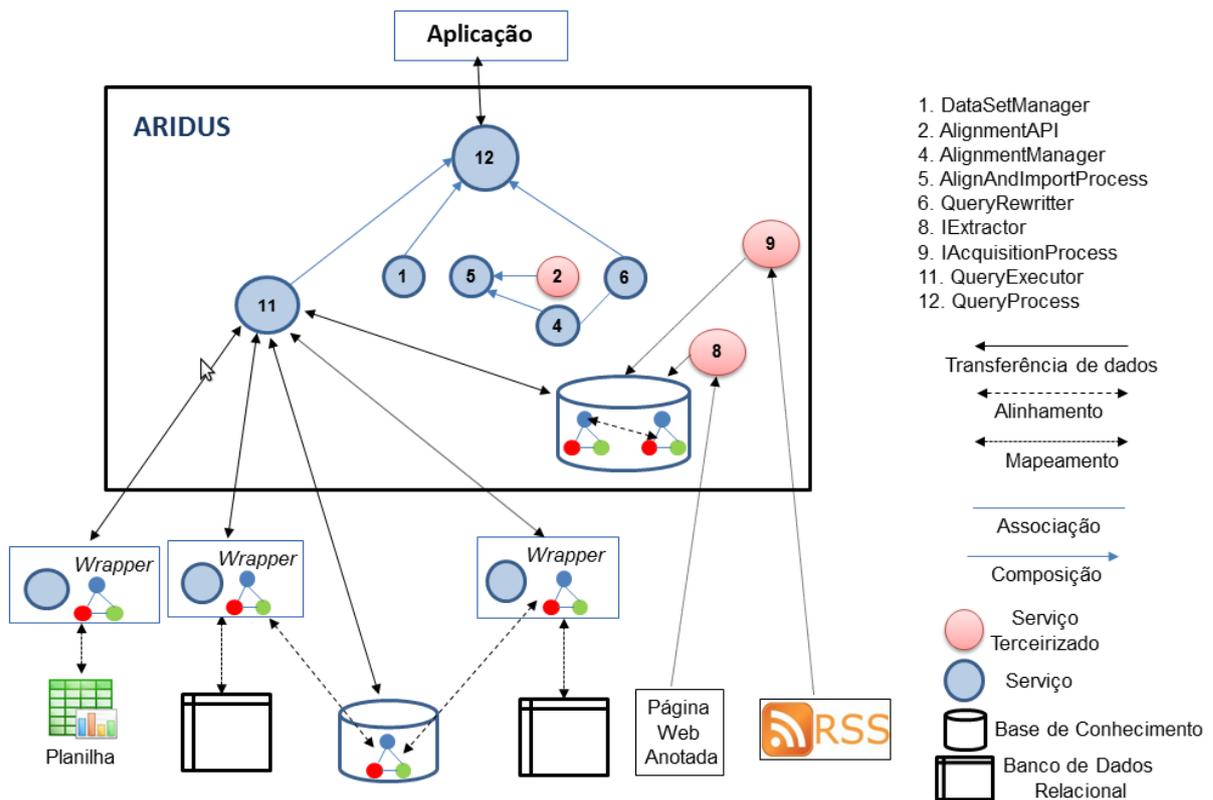


Figura 32: Cenário de Integração de Informação com a ARIDUS.

Com a utilização dos serviços *IExtractor*(8) e *IAcquisitionProcess*(9) da ARIDUS, informações anotadas em documentos e dados de fontes semi-estruturadas ou não-estruturadas são obtidos e armazenados em uma base RDF local, caracterizando uma abordagem de integração *Warehousing*. No caso das fontes de dados estruturadas, como banco de dados relacionais, o acesso por tecnologias semânticas é proporcionado através dos wrappers, como por exemplo,

XLWrap²² e D2R Server²³, que fazem uso do mapeamento do esquema sintático da fonte à vocabulários semânticos para fornecer um acesso *on-the-fly* aos dados, caracterizando-se como uma Federação. Todos os *endpoints* são registrados na plataforma com a utilização do serviço *DataSetManager*(1).

Com os dados das diferentes fontes mapeados aos vocabulários semânticos, seja por conversão ou uso de *wrappers*, realizamos o alinhamento entre os vocabulários utilizados em tais fontes, determinando relações semânticas entre as informações que armazenam, e utilizamos estes alinhamentos para reescrever consultas SPARQL e executá-las nos *endpoints* destas fontes. Na ARIDUS, o alinhamento entre vocabulários, reescrita de consultas SPARQL e execução destas consultas nos *endpoints* são realizadas, respectivamente, pelos serviços *AlignAndImportProcess*(5), *QueryRewriter*(6) e *QueryExecutor*(11).

A aplicação pode submeter consultas SPARQL ao serviço *QueryProcess*(12), uma orquestração dos serviços *QueryRewriter*(6), *DataSetManager*(1) e *QueryExecutor*(11), que realiza a reescrita da consulta, seleção dos endpoints alvos, execução e junção dos resultados, retornando-os à aplicação.

O diagrama de sequência (Figura 33 na página seguinte) apresenta as interações entre a aplicação e os serviços da ARIDUS.

²²<http://xlwrap.sourceforge.net/>

²³<http://d2rq.org/>

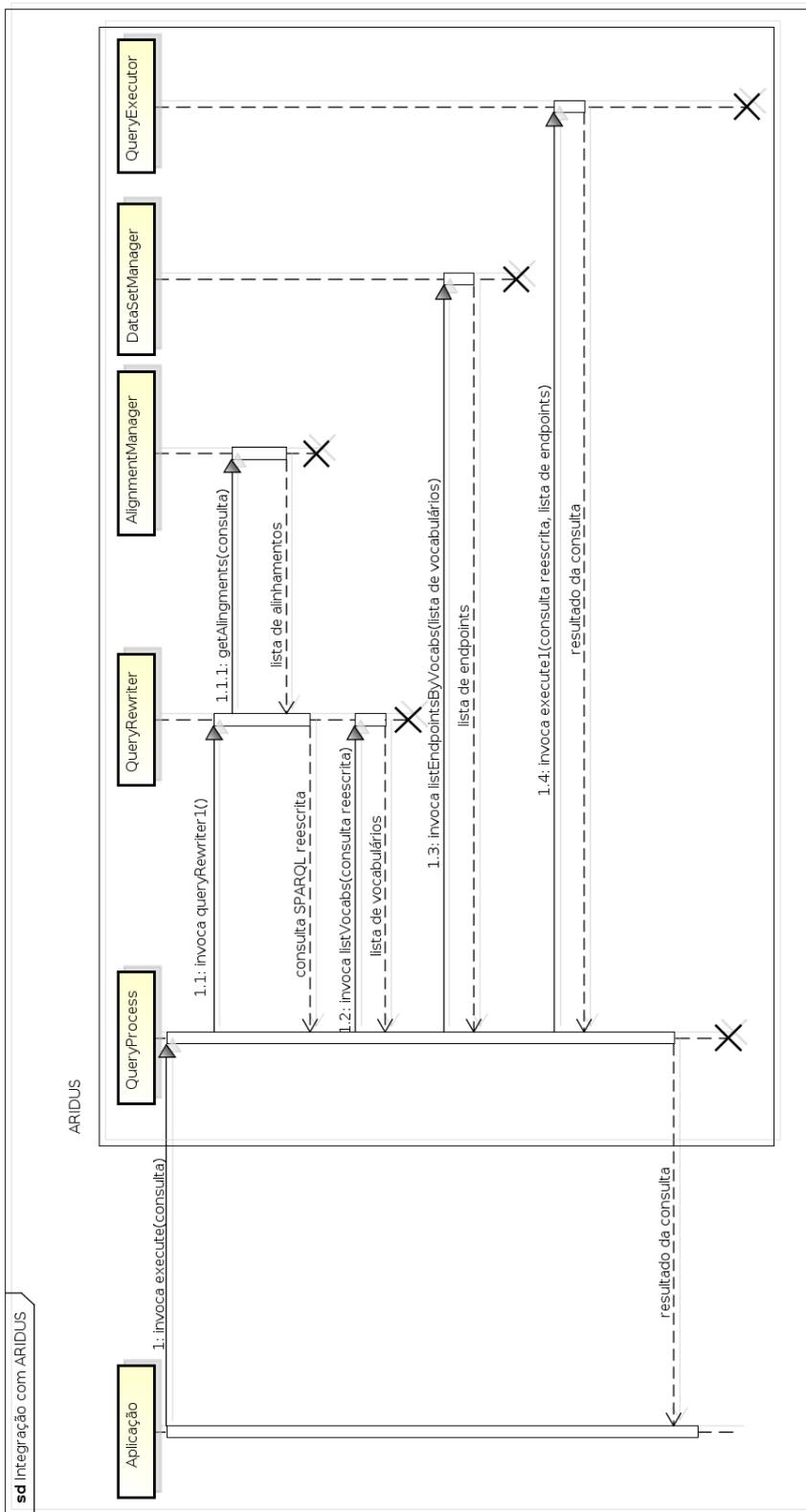


Figura 33: Diagrama de Sequência - Aplicação, Serviços e Fontes de Dados.

O serviço *QueryExecutor*(11) envia consultas aos diferentes *endpoints* e reúne os resultados que serão apresentados à aplicação.

5.4 Extensão da plataforma

Em meio ao grande volume de informações disponibilizadas atualmente na Web, a percepção do que é realmente de interesse do usuário é extremamente difícil. A mudança na forma como a Web é encarada pelos usuários e desenvolvedores propiciou o compartilhamento de conteúdos entre esses usuários e o surgimento das comunidades on-line. No entanto, o surgimento das várias redes sociais tem dificultado a identificação do usuário como um indivíduo único e motivado os sistemas a tratarem usuários como parte de um grupo, recomendando conteúdos por meio de interesses similares dos indivíduos que o compõe, e não pelas particularidades de cada usuário como, por exemplo, uma deficiência.

Neste contexto, busca-se, através da extensão da plataforma com modelos de personalização e acessibilidade, fornecer mecanismos para realizar recomendações de conteúdos e melhorias na acessibilidades dos mesmos.

5.4.1 PerSoN: Um modelo semântico para personalização e interoperabilidade entre redes sociais

Um dos projetos em desenvolvimento pelo GSiD-UERN (Grupo de Sistemas Distribuídos da UERN), visa à construção de um modelo semântico para a personalização de conteúdos que atende à individualidade dos usuários e permite a interoperabilidade de perfis de usuários entre redes sociais.

O modelo, denominado PerSoN (COSTA, 2013) , constitui-se de ontologia de modelagem do perfil do usuário, base com as instâncias dos perfis dos usuários (*PersonKB*), fontes de conteúdos a serem recomendados, serviço (*PersonProfile*) para gerência do perfil semântico do usuário e serviço de recomendação de conteúdos (*ContentRecommender*). Tais serviços estão dispostos na plataforma ARIDUS como é apresentado na Figura 34 (página seguinte).

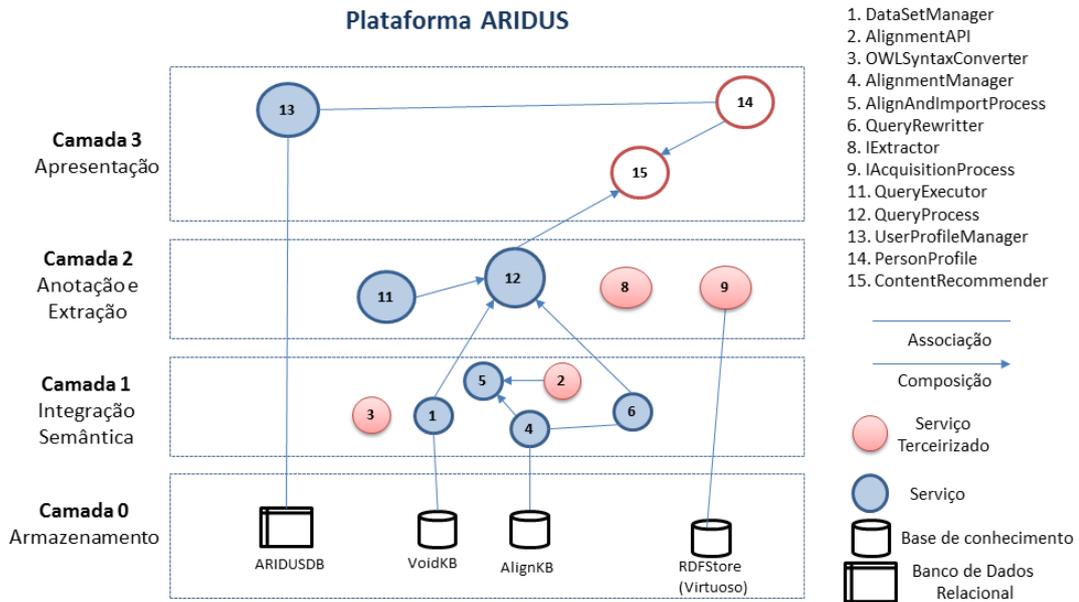


Figura 34: Disposição do Modelo PerSoN na Plataforma ARIDUS.

O PerSoN utiliza a infraestrutura da plataforma ARIDUS para realizar extração semântica de conteúdos na Web (serviço *IExtractor*) e recomendação de conteúdos armazenados nas fontes de dados registradas na plataforma (*QueryProcess*).

5.4.2 @dapt – Modelo Semântico para prover acessibilidade para deficientes visuais na Internet

O @dapt (SORRENTINO, 2013) é um modelo desenvolvido para prover acessibilidade a deficientes visuais na Internet e para tal, baseia-se na utilização de tecnologias semânticas, recomendações de acessibilidade e adaptação dinâmica de páginas web.

O modelo utiliza ontologias para estruturação e representação da base de informações sobre as deficiências (Ontologias Genéricas por Deficiência) e da base de informações (Perfis Semânticos de Conteúdos) para enriquecimento dos conteúdos (imagens e vídeos) no intuito de prover algum nível de acessibilidade aos mesmos.

Disponibilizado como serviço (*ContentAdapter*), o modelo utiliza a plataforma ARIDUS para obter informações de conteúdos a serem adaptados, bem como, conteúdos equivalentes acessíveis a serem utilizados na adaptação de páginas web.

O serviço *ContentAdapter* relaciona-se com o modelo PerSoN, mais especificamente, com o serviço *PersonProfile*. Este serviço gerencia os perfis semânticos dos usuários constituídos de propriedades que descrevem as deficiências do usuário.

A figura 35 ilustra o serviço ContentAdapter na plataforma ARIDUS.

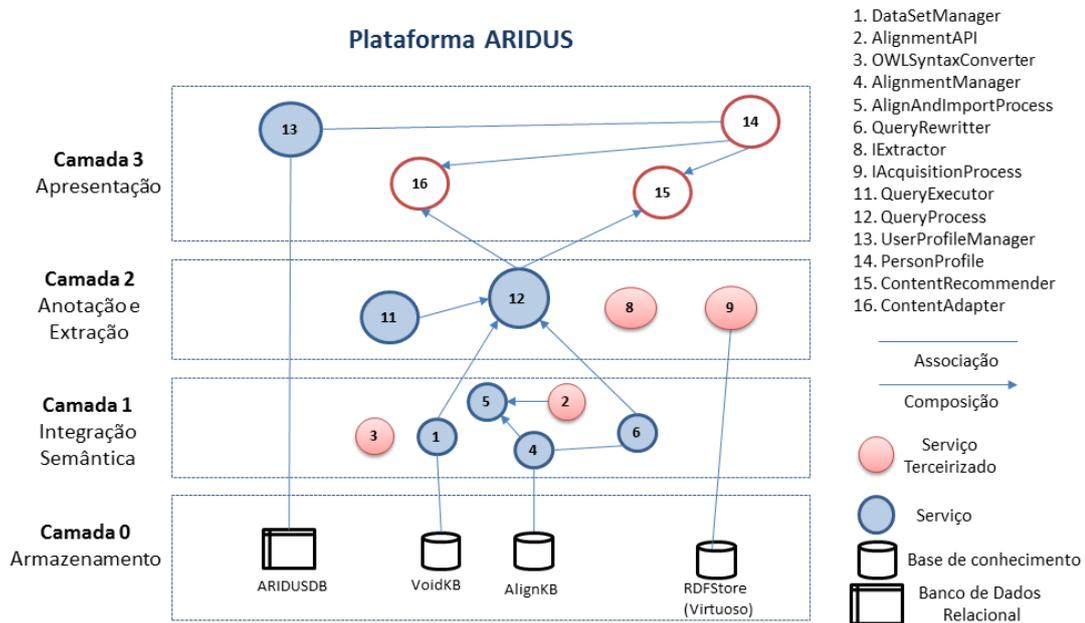


Figura 35: Disposição do Modelo @dapt na Plataforma ARIDUS.

O serviço de anotação (não-realizado até o momento) representa uma ferramenta de significativa importância para alcançar melhorias nas adaptações realizadas pelo modelo, pois permite a associação dos conceitos estabelecidos pelas ontologias aos conteúdos disponíveis na Web.

Diferente dos trabalhos apresentados no capítulo anterior, a plataforma ARIDUS, apresentada neste documento, visa não somente oferecer aos desenvolvedores de aplicações o acesso aos dados integrados mas, também, um conjunto de serviços que dão suporte a extração, personalização de conteúdos e acessibilidade dos mesmos, bem como a flexibilidade, interoperabilidade e reuso oferecidos por uma infraestrutura SOA construída com serviços web, que além dos serviços individuais, possibilita a realização de composições de serviços.

6 ESTUDOS DE CASO E APLICAÇÕES

Este capítulo apresenta estudos de caso e aplicações desenvolvidas que demonstram a utilização dos serviços e validação das funcionalidades da plataforma ARIDUS, bem como, dos modelos que a estendem.

6.1 IntercropApp

A IntercropApp (PEREIRA, 2012), um dos resultados do projeto de mestrado (Edital MCT/CNPq N° 70/2009 – Mestrado/Doutorado – Processo 558489/2010-9), é uma aplicação web com propósito de auxiliar na busca de alternativas sustentáveis para o plantio de hortaliças.

A aplicação foi construída utilizando-se de tecnologias da web semântica e dos serviços da plataforma ARIDUS, e baseia-se em duas técnicas sustentáveis de plantio, sendo elas: Consórcio de culturas e Rotação de culturas.

O Consórcio de culturas se dá pelo plantio de duas ou mais culturas em uma mesma área de plantio. Já a rotação de culturas, trata da alternância de culturas de um consórcio. Desta forma, é necessário analisar as características das culturas para determinar que benefícios existem.

Pereira (2012) apresenta a construção da ontologia na qual estão definidos os conceitos (Classes e propriedades) que permitem a formulação de consórcios entre as culturas pela aplicação. Dentre as propriedades que caracterizam uma cultura, estão:

- **Necessidade de luz** – Representa a quantidade de luz que a cultura precisa;
- **Nome científico** – Representa o nome científico da cultura;
- **Família** – Descreve a família da cultura;

- **Ciclo da cultura** – Determina o ciclo que a cultura leva para a colheita;
- **Época de plantio** – Representa os melhores meses para o plantio da cultura;
- **Raiz profunda** – Representa se a cultura possui raiz profunda ou não;
- **Espaçamento** – Descreve o espaço que a cultura precisa para ser plantada;
- **Necessidade de água** – É o quanto de água a cultura precisa.

Além dessas propriedades, a ontologia possui propriedades que relacionam duas culturas informando se elas são culturas companheiras ou antagônicas. Culturas companheiras são aqueles que quando cultivadas próximas, se beneficiam ou não interferem no desenvolvimento uma da outra. As antagônicas são definidas por apresentarem impactos prejudiciais ocasionados pelo cultivo das culturas em uma mesma área.

Entre as características apresentadas acima, as mais relevantes para a consorciação e rotaciação de culturas são: Ciclo da cultura, Época de plantio, Raiz profunda e é/tem companheira (PEREIRA, 2012).

A IntercropApp permite que o usuário formule um consórcio que posteriormente é analisado de acordo com as dimensões da sustentabilidade. A figura 36 exibe a interface na qual o usuário seleciona duas culturas para um consórcio.

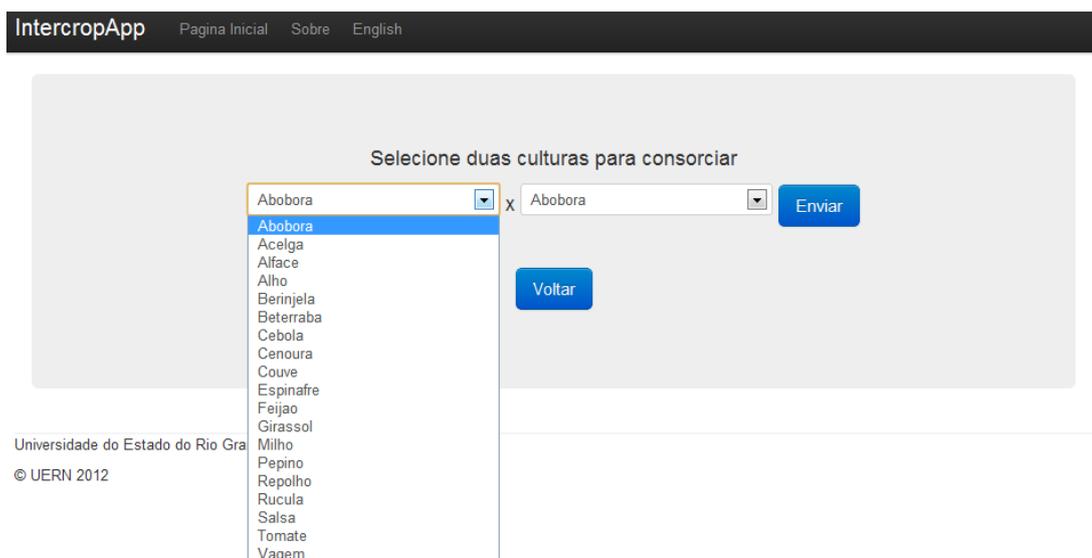


Figura 36: IntercropApp – Criação de um consórcio de duas culturas (PEREIRA, 2012).

Para exemplificar a relação da aplicação com a ARIDUS, consideremos que desejamos analisar consórcios possíveis que envolve a alface e que temos duas bases (icropkb1 e icropkb2)

com informações de culturas, sendo que, dentre as culturas companheiras da alface, apenas tomate e rúcula estão armazenadas na mesma base que ela. Desta forma, para que se obtenhamos informações sobre as outras culturas e ampliemos as opções e análises de consórcios possíveis, é necessário integrar as bases que contêm informações no vocabulário definido por Pereira (2012) ou vocabulários alinhados.

A IntercropApp obtém as informações das culturas através do serviço *QueryProcess* da ARIDUS, que, por sua vez, realiza os seguintes passos:

1. Reescreve a consulta de entrada (como, por exemplo, a figura 37) utilizando o serviço *QueryRewriter*. No entanto, não havendo alinhamentos, o serviço preserva a consulta;

```

1 PREFIX crop: <http://aridus.uern.br/intercrop#>
2
3 SELECT DISTINCT ?RaizProfunda ?temFamilia ?CicloDaCultura ?Espacamento ?EpocaDePlantio
4 WHERE{
5   crop:Alface crop:Raiz_profunda ?RaizProfunda .
6   crop:Alface crop:tem_Familia ?temFamilia .
7   crop:Alface crop:Ciclo_da_cultura ?CicloDaCultura .
8   crop:Alface crop:Espacamento ?Espacamento .
9   crop:Alface crop:Epoca_de_plantio ?EpocaDePlantio
10 }
```

Figura 37: Consulta SPARQL utilizada para obter informações da cultura Alface

2. Obtém o conjunto de endpoints das fontes de dados que utilizam o vocabulário da IntercropApp (icrop). Neste caso, os endpoints das bases icropkb1 icropkb2;
3. Executa a consulta nos endpoints obtidos no passo anterior, a fusão e retorno dos resultados para a aplicação.

Com as informações das culturas, a IntercropApp realiza a análise das informações e exibe um relatório do consórcio como ilustrado na figura 38 (página seguinte).

A ARIDUS beneficia a IntercropApp com a integração de informações e possibilita que a aplicação amplie a formulação de consórcios e realize a análise destes segundo as dimensões da sustentabilidade.

Feijao x Pepino

Ambiental	Econômico	Social	Meses Recomendados
-	+	+	Janeiro, Fevereiro, Maio, Junho, Agosto, Setembro,

Raízes iguais

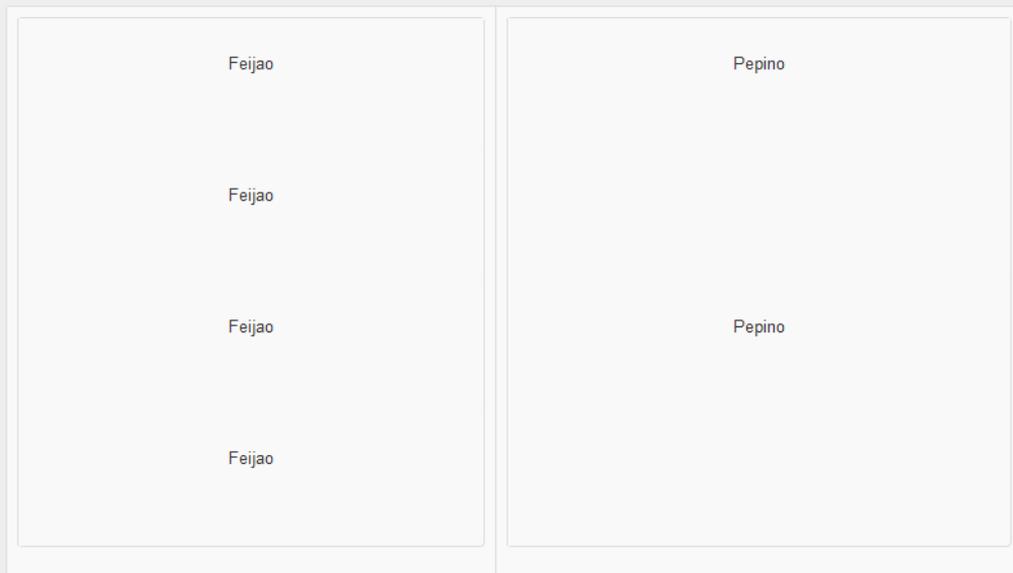
Feijao

Densidade	67000 plantas/ha
Colunas	33
Linhas	335
Produtividade esperada	22110 pes

Pepino

Densidade	13400 plantas/ha
Colunas	33
Linhas	134
Produtividade esperada	8844 pes

Croqui



Próximo cultivo recomendado

Rotação para Feijao	[Berinjela, Beterraba, Cenoura, Girassol]
Rotação para Pepino	[Berinjela, Beterraba, Cenoura, Girassol, Tomate]

Possíveis combinações dos cultivos recomendados

[Cenoura X Tomate]

Voltar

Figura 38: Resultado da análise do consórcio entre as cultura selecionadas pelo usuário

6.2 Integração DocBase e Infoteca-e

Segundo o CENSO 2010¹ do IBGE (Instituto Brasileiro de Geografia e Estatística), cerca de 35 milhões de brasileiros possuem algum tipo de deficiência, sendo o nordeste a região brasileira com maior presença de deficientes e a deficiência visual uma das mais comuns.

No estado do Rio Grande do Norte, constatou-se que em 12% dos municípios, 35% dos investigados têm, pelo menos, uma das limitações consideradas na pesquisa, o que constitui uma parcela da população carente de recursos que atendem às necessidades e especificidades para o acesso com qualidade à Internet.

Nesse contexto do semiárido, realizou-se um estudo de caso na plataforma ARIDUS com o intuito de explorá-la e demonstrar a utilização dos modelos PerSoN e @dapt para fornecer acesso às informações integradas de forma acessível e com recomendações de acordo com os interesses personalizáveis dos usuários. Portanto, visualizemos o cenário onde usuários deficientes visuais acessam conteúdos na web através da ARIDUS, e a mesma se encarrega de fornecer os serviços necessários para que as informações dos conteúdos sejam acessíveis e que recomendações de conteúdos relacionados aos interesses do usuário sejam apresentados.

6.2.1 As fontes de informações

Neste estudo de caso, realiza-se a integração da DocBase e Infoteca-e, duas fontes de informações relacionadas ao semiárido e à agricultura brasileira.

6.2.1.1 DocBase

A DocBase (*DocumentBase*) trata-se de uma base de documentos construída para armazenar projetos de pesquisa e seus resultados. A princípio, a base mantém um pequeno conjunto de informações e os arquivos PDF (*Portable Document Format*) de alguns projetos desenvolvidos na UERN.

As informações dos documentos na DocBase são gerenciadas pelo serviço *DocBase-Manager*, que, dentre outras operações, permite a visualização das informações de um documento em HTML, como ilustra a figura 39 (página seguinte).

¹<http://www.deficientefisico.com/resultados-do-censo-2010-feito-pelo-ibge-sobre-pessoas-com-deficiencia/>

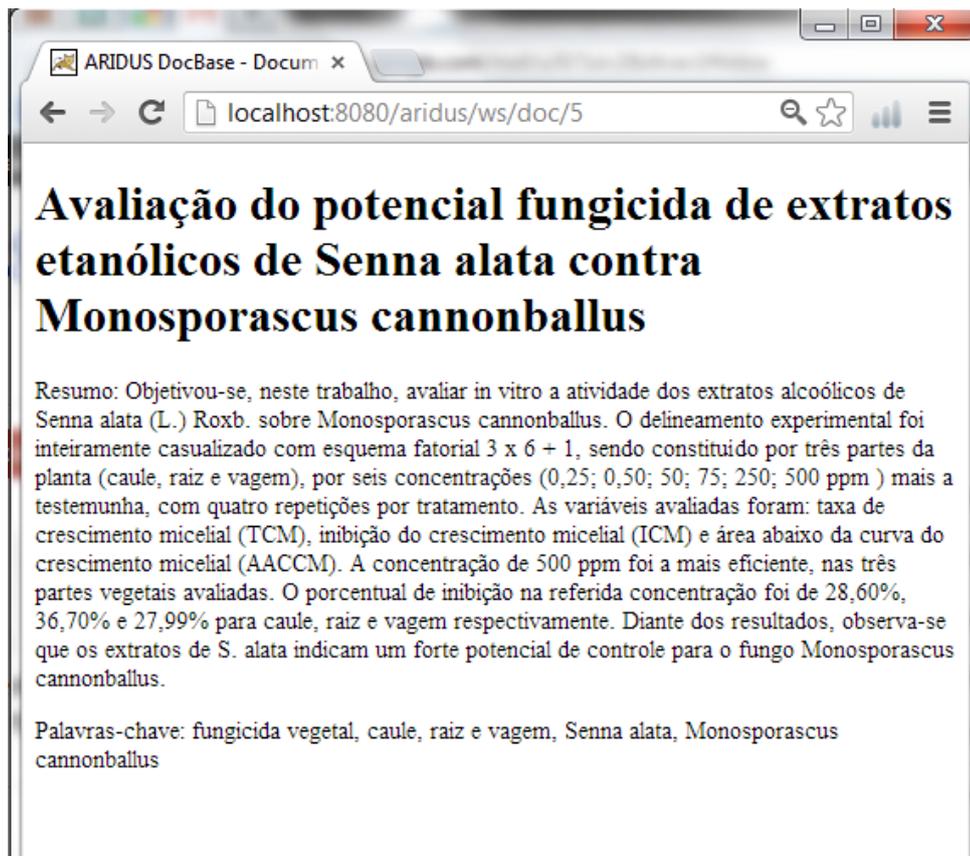


Figura 39: Exibição de informações de um documento da *DocBase*

A base possui também uma interface web que permite a inserção, busca e visualização de documentos. A figura 40 apresenta a listagem de alguns documentos cadastrados na DocBase.



Figura 40: Interface web da DocBase

As páginas da interface web, assim como, as saídas HTML dos serviços, foram cuidadosamente elaboradas para serem simples e atenderem às recomendações de acessibilidade da W3C.

Para fornecer acesso às informações de forma semântica, utilizou-se o D2R Server² para disponibilizar um endpoint SPARQL para as informações armazenadas na *aridusdb* que foram mapeadas ao vocabulário *Dublin Core*. A solução constituída da base relacional (*aridusdb*), D2RServer e o mapeamento realizado entre a *aridusdb* e o *Dublin Core* é referenciada na plataforma como o serviço *DocBaseWrapper*.

A figura 41 exibe a execução direta de uma consulta SPARQL no endpoint da *DocBase*.



Figura 41: SPARQL Endpoint da DocBase

A fonte DocBase (DocumentBase) da ARIDUS engloba uma base relacional (*aridusdb*), serviço para gerência dos documentos (*DocBaseManager*), interface web de usuário e endpoint SPARQL (a solução *DocBaseWrapper*), permitindo que as informações sejam acessadas e manipuladas por diferentes tipos de usuários (agricultores, agrônomos, desenvolvedores de aplicações, pesquisadores em computação, dentre outros).

²<http://d2rq.org/>

6.2.1.2 Infoteca-e

A Infoteca-e³ é o portal de Informação Tecnológica em Agricultura da Embrapa (Empresa Brasileira de Pesquisa Agropecuária). O portal reúne informações sobre tecnologias produzidas pela instituição, bem como, conteúdos editados pela mesma. Estes conteúdos, em forma de cartilhas, livros para transferência de tecnologia, programas de rádio e de televisão, se relacionam às áreas de atuação dos centros de pesquisa da Embrapa e não podem ser consultados utilizando tecnologias semânticas, como SPARQL, por exemplo.

The screenshot shows the Infoteca-e portal interface. At the top, there is a navigation breadcrumb: "Infoteca-e > Embrapa Recursos Genéticos e Biotecnologia (CENARGEN) > Comunicado Técnico (CENARGEN)". Below this, a message asks the user to use a specific identifier for citation: "Por favor, utilize esse identificador para citar ou referenciar esse registro: http://www.infoteca.cnptia.embrapa.br/handle/doc/184737".

The main content is a table with two columns: "Formato Registro" and "Conteúdo". The table contains the following information:

Formato Registro	Conteúdo
Unidade da Embrapa/Coleção:	Embrapa Recursos Genéticos e Biotecnologia - Comunicado Técnico (INFOTECA-E)
Identificador:	23742
Data de Envio:	19-Ago-2003
Tipo do Material:	Comunicado Técnico (INFOTECA-E)
Autoria:	OLIVEIRA, M. R. V. de SILVEIRA, C. C. da LIMA, L. H. C. PAIVA, I. F. de LIRA, G. S. de LAGO, W. N. QUEIROZ, P. R. de FERNANDES, E. R. SANTOS, E. A. de.
Título:	Efeito da temperatura na viabilidade de Bemisia tabaci biótipo B, em plantas de melão.
Edição:	2003
Fonte/Imprensa:	Brasília: Embrapa Recursos Genéticos e Biotecnologia, 2003.
Páginas:	6 p.
Série:	(Embrapa Recursos Genéticos e Biotecnologia. Comunicado Técnico, 79).
Idioma:	pt_BR
Palavras-chaves:	Temperatura Viabilidade Bemisia tabaci Melão Mosca branca Cucumis melo

Figura 42: Portal de Informação Tecnológica em Agricultura da Embrapa (Infoteca-e)

A Infoteca-e exibe as informações sobre os vários conteúdos disponibilizados em páginas HTML, por exemplo, a página apresentada na figura 42, e destas páginas foram extraídas mais de 531 mil triplas RDF posteriormente armazenadas em um base RDF (infotecaekb), tornando tais informações acessíveis através de um endpoint SPARQL.

6.2.2 Integração, Personalização e Acessibilidade das Informações

A plataforma ARIDUS utiliza-se dos modelos @dapt e PerSoN para realizar melhorias na acessibilidade das páginas do portal Infoteca-e e recomendar conteúdos relacionados aos

³<http://www.infoteca.cnptia.embrapa.br>

interesses dos usuários.

O serviço *ContentViewer* é o responsável por receber a requisição do usuário e operar os serviços dos modelos que estendem a ARIDUS, fornecendo acesso às informações segundo as particularidades de cada usuário.

O perfil do usuário é gerenciado pelo serviço *PersonProfile* e contém informações dos seus interesses e a indicação da sua deficiência. Neste sentido, uma requisição ao *ContentViewer* é feita passando como parâmetros a URL da página desejada e o URI do perfil PerSoN.

Ao receber uma requisição, o *ContentViewer*, passando os parâmetros recebidos, faz as seguintes solicitações:

1. Solicita ao serviço *ContentRecommender*, um conjunto de conteúdos que apresentem relações com os interesses do usuário e que sejam diferentes do conteúdo que está sendo acessado. O resultado desta solicitação é recebido como um trecho de código HTML que contém uma lista de links.
2. Solicita ao serviço *ContentAdapter*, a adaptação da página para o usuário. Para realizar a adaptação, o *ContentAdapter*:
 - Identifica URI's de links e imagens no código fonte da página;
 - Busca informações sobre os recursos (links e imagens) na plataforma ARIDUS através do serviço *QueryProcess*. Tais informações, como, por exemplo, descrições ou conteúdos audiodescritos equivalentes, são utilizadas na adaptação dinâmica da página web;
 - Realiza a adaptação da página web inserindo ou alterando informações dos atributos (*title*, *src*, *href*, *alt*) das tags HTML (<A> e) utilizando-se das informações obtidas pela ARIDUS. No caso das imagens () a adaptação para os deficientes visuais inibe todas as imagens da página, removendo a informação do atributo *src*.
3. Ao receber o resultado da solicitação 1, o *ContentViewer* faz uma nova solicitação ao *ContentAdapter* para que possíveis melhorias sejam realizadas no trecho de código HTML recebido.

Com o resultado das solicitações 2 e 3, uma página web (figura 43) é gerada e exibida para o usuário. Nesta página, são apresentadas as informações com adaptações realizadas segundo as particularidades do usuário, e recomendações de conteúdos relacionados aos interesses do mesmo (área inferior da página).

The screenshot shows a web interface with a green header. On the left, there is a sidebar with navigation options like 'Unidades da Embrapa', 'Data de Envio', 'Autor', etc. The main content area displays document details in a table format. Below the table, there are social media links and a 'Recomendações' section with three links.

Formato Registro	Conteúdo
Unidade da Embrapa/Coleção:	Embrapa Recursos Genéticos e Biotecnologia - Comunicado Técnico (INFOTECA-E)
Identificador:	23742
Data de Envio:	19-Ago-2003
Tipo do Material:	Comunicado Técnico (INFOTECA-E)
Autoria:	OLIVEIRA, M. R. V. de SILVEIRA, C. C. da LIMA, L. H. C. PAIVA, I. F. de LIRA, G. S. de LAGO, W. N. QUEIROZ, P. R. de FERNANDES, E. R. SANTOS, E. A. de.
Título:	Efeito da temperatura na viabilidade de Bemisia tabaci biótipo B, em plantas de melão

Recomendações:

- [Extrato etanólico de Senna alata no controle de Fusarium oxysporum, causador da murcha-de-fusarium do meloeiro](#)
- [MANUAL de segurança e qualidade para a cultura do melão.](#)
- [Perfil da exploração do melão nas regiões de Mossoró e Açú- RN e Baixo Jaguaribe- CE.](#)

Figura 43: Página gerada pelo serviço *ContentViewer*

Segundo análise realizada com a ferramenta SortSite⁴, as páginas adaptadas pelo ContentAdapter tiveram 16,7% menos erros.

Os conteúdos recomendados aos usuários, neste estudo de caso, são advindos da integração da DocBase e Infoteca-e (*infotecaekb*). Na ilustração (figura 43), apesar de o usuário estar acessando informações de um documento na Infoteca-e, os conteúdos recomendados são da DocBase. Isto demonstra a disseminação de informações proporcionada pela integração de diferentes fontes e da recomendação realizada considerando os interesses particulares de cada usuário.

⁴<http://www.powermapper.com/products/sortsite/index.htm>

6.3 *TwitterRecommender*

Este estudo de caso apresenta a aplicação da ARIDUS em um domínio diferente dos abordados anteriormente. O *TwitterRecommender* é um serviço de recomendação/notificação de conteúdos compartilhados na rede social Twitter⁵, uma das redes mais populares e, através da qual, segundo Tao et. al., (2012), pode-se descobrir bastante sobre os interesses e necessidades atuais dos usuários.

O *TwitterRecommender* apresenta-se como uma ferramenta que permite ao usuário do Twitter o recebimento de notificações dos conteúdos compartilhados por outros usuários da rede. Para tal, o serviço aplica o modelo PerSoN (COSTA, 2013) e utiliza-se da plataforma ARIDUS para prover recomendações de conteúdos tendo como fonte de informações a base de dados da rede e diversas outras fontes que tenham seus vocabulários alinhados ao vocabulário utilizado no Twitter.

A figura 44 apresenta a disposição do *TwitterRecommender* na ARIDUS.

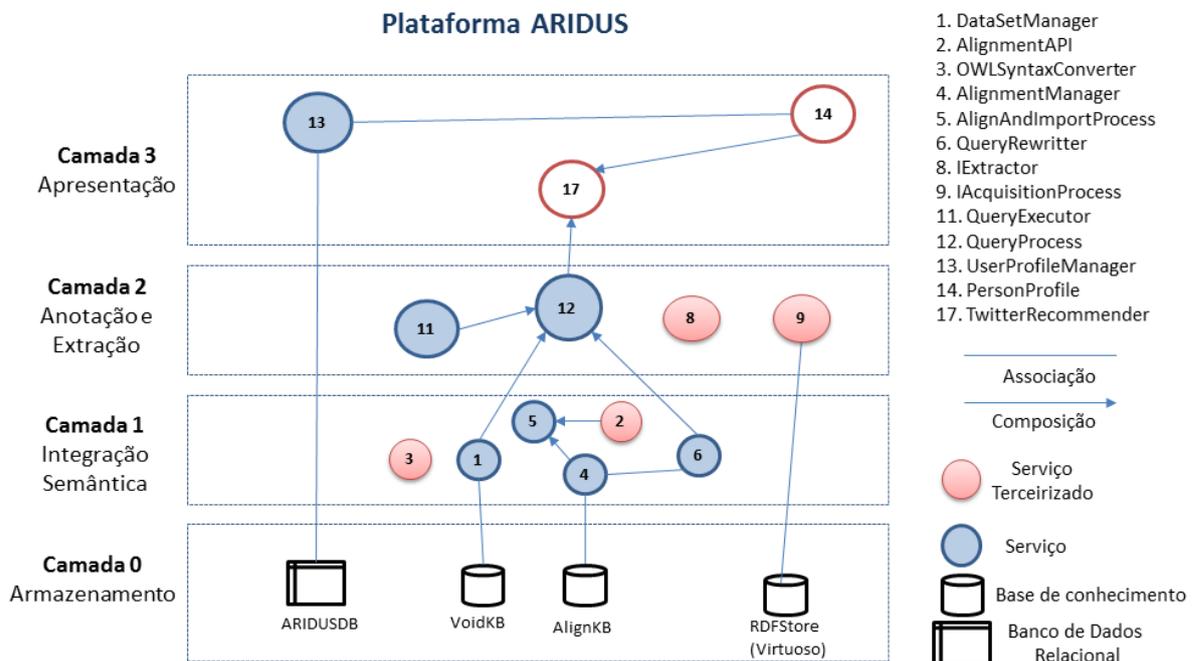


Figura 44: Disposição do *TwitterRecommender* na Plataforma ARIDUS.

Para obter recomendações, é necessário que o usuário cadastre-se na plataforma ARIDUS e crie seu perfil segundo o modelo PerSoN. Para tal, os serviços UserProfileManager e PersonProfile são utilizados para as respectivas ações.

⁵<http://twitter.com>

No *TwitterRecommender*, há dois métodos disponíveis, *tweetRecommender* e *twitterUserRecommender*, nos quais são necessários dois parâmetros, a URI do perfil PerSoN e um *timestamp* (*timing*) usado para filtrar todos os conteúdos publicados antes do momento informado.

Os métodos do *TwitterRecommender* são responsáveis por obter as informações e interesses do usuário, gerar consultas SPARQL com base em modelos pré-estabelecidos (ilustrados abaixo) e executá-las através do *QueryProcess* da ARIDUS, obtendo, desta forma, conteúdos das diversas bases registradas na plataforma que apresentem relações com os interesses do usuário.

Os modelos de consultas contêm os principais elementos fixos da consulta, como o *graph pattern* e as variáveis que indicam os elementos a serem apresentados no resultado da consulta. Para ilustrar, abaixo os modelos de consulta configurados:

Modelo de Consulta SPARQL para Recomendação de Tweets (Conteúdo)

```

PREFIX=tw:<http://www.openlinksw.com/schemas/twitter#>;
foaf:<http://xmlns.com/foaf/0.1/>;
dc:<http://purl.org/dc/elements/1.1/>;
dcterms:<http://purl.org/dc/terms/>;
bibo:<http://purl.org/ontology/bibo/>;
sioc:<http://rdfs.org/sioc/ns#>;
rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>;
rdfs:<http://www.w3.org/2000/01/rdf-schema#>;
xsd:<http://www.w3.org/2001/XMLSchema#>;
wdrs:<http://www.w3.org/2007/05/powder-s#>
SELECT=source,date,name,user,content,tweet
DISTINCT=true
BGP=?tweet rdf:type tw:Tweet .
    ?tweet dc:source ?source .
    ?tweet dcterms:created ?date .
    ?tweet dcterms:creator ?creator .
    ?creator tw:public_profile_url ?user .
    ?creator foaf:name ?name .
    ?tweet bibo:content ?content .
    ?tweet sioc:topic ?topic .
    ?topic rdfs:label ?hashtag
ORDERBY=?date

```

LIMIT=10

Modelo de Consulta SPARQL para Recomendação de Usuário no Twitter

```
PREFIX=tw:<http://www.openlinksw.com/schemas/twitter#>;
foaf:<http://xmlns.com/foaf/0.1/>;
dc:<http://purl.org/dc/elements/1.1/>;
dcterms:<http://purl.org/dc/terms/>;
bibo:<http://purl.org/ontology/bibo/>;
sioc:<http://rdfs.org/sioc/ns#>;
rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>;
rdfs:<http://www.w3.org/2000/01/rdf-schema#>;
xsd:<http://www.w3.org/2001/XMLSchema#>;
wdrs:<http://www.w3.org/2007/05/powder-s#>
SELECT=creator, hashtag, date
DISTINCT=true
BGP=?tweet rdf:type tw:Tweet .
    ?tweet dcterms:created ?date .
    ?tweet sioc:topic ?topic .
    ?topic rdfs:label ?hashtag .
    ?tweet dcterms:creator ?creator
ORDERBY=?date
LIMIT=1000
```

Segundo dados do IBOPE/NetView⁶, as redes sociais e demais páginas categorizadas como comunidades online, atingiram em janeiro de 2013 mais de 46 milhões de usuários brasileiros, cerca de 86% dos internautas ativos no período.

Cada uma das redes sociais utiliza-se de vocabulário próprio para estruturação dos dados dos perfis dos usuários e conteúdos compartilhados pelos mesmos. Deste modo, o alinhamento entre vocabulários de diferentes redes sociais permite a integração e interoperabilidade de informações/conteúdos entre as redes sociais.

As figuras 45 (pág. 86) e 46 (pág. 88) apresentam, respectivamente, os alinhamentos dos vocabulários utilizados para a estruturação dos perfis de usuários e conteúdos compartilhados no Twitter⁷, Facebook⁸ e Goolge Plus⁹.

⁶<http://www.ibope.com.br/pt-br/noticias/paginas/numero-de-usuarios-de-redes-sociais-ultrapassa-46-milhoes-de-brasileiros.aspx>, publicado em 26 de março de 2013.

⁷<http://www.openlinksw.com/schemas/twitter\#>

⁸<http://www.openlinksw.com/schemas/opengraph\#>

⁹<http://www.openlinksw.com/schemas/googleplus\#>

Twitter	Facebook	Google Plus
bibo:content	oplog:message	oplgp:comment_content
dc:source	oplog:link	oplgp:comment_self_link
oplgp:Tweet	oplgp:Post	oplgp:Comment / oplgp:Activity
dcterms:created	dcterms:created	oplgp:published
dcterms:creator	powder-s:describedby	powder-s:describedby
wdrs:describedby	wdrs:describedby	
sioc:topic		foaf:topic
dc:title		oplgp: activity_title
	oplog:from	opl:mentions

Figura 45: Alinhamento entre vocabulários para estruturação de dados dos perfis de usuários em Redes Sociais (COSTA, 2013).

Com base neste alinhamento, o modelo PerSoN pode construir a consulta SPARQL (abaixo) e, através da ARIDUS, obter, de forma transparente, informações compartilhadas nas diferentes redes sociais que se relacionam com os interesses do usuário.

```

PREFIX tw: <http://www.openlinksw.com/schemas/twitter#>
PREFIX plus: <http://www.openlinksw.com/schemas/googleplus#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX bibo: <http://purl.org/ontology/bibo/>
PREFIX sioc: <http://rdfs.org/sioc/ns#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX wdrs: <http://www.w3.org/2007/05/powder-s#>

SELECT DISTINCT ?source ?date ?name ?profile_url ?content
WHERE{
{ ?creator tw:made_tweet ?tweet .
  ?tweet dc:source ?source .
  ?tweet dcterms:created ?date .
  ?creator foaf:name ?name .
  ?creator tw:public_profile_url ?profile_url .
  ?tweet bibo:content ?content
} UNION {

```

```
?a plus:activity_url ?source .
?a plus:published ?date .
?a plus:activity_object ?ao .
?ao plus:html_content ?content .
?a plus:actor ?actor .
?actor plus:actor_displayName ?name .
?actor plus:actor_profile_url ?profile .
?profile plus:profile_url ?profile_url .
} UNION {
?user fb:posted ?post .
?post dcterms:created ?date .
?post fb:link ?source .
?post fb:message ?content .
?user foaf:name ?name .
?user fb:public_profile_url ?profile_url
}
} LIMIT 100
```

Twitter	Facebook	Google Plus
opltw:id	oplog:id	oplgp:id
opltw:screen_name	oplog:username	oplgp:display_name
foaf:img	foaf:img	oplgp:profile_image
foaf:name	foaf:name	oplgp:name
opltw:public_profile_url	oplog:public_profile_url	oplgp:profile_url
opltw:User	oplog:User	oplgp:Person
foaf:nickname	foaf:nickname	
sioc:follows	foaf:knows	
opltw:made_tweet	oplog:posted	
opltw:has_favorite	oplog:has_interest	
dc:title		dc:title
dc:description		dc:description
bibo:abstract		dc:abstract
	foaf:gender	oplgp:gender
	oplog:relationship_status	oplgp:relationshipStatus
	oplog:birthday	oplgp:date_of_birth
	oplog:languages	oplgp:speaksLanguage
	oplog:location	oplgp:currentLocation
	oplog:email	oplgp:email
	oplog:Link	oplgp:shared_url
opltw:followed_by	oplog:last_name	oplgp:associatedWith
opltw:followers_count	foaf:firstName	oplgp:hasApp
opltw:friends_count	oplog:has_album	oplgp:performed_activity
opltw:favorites_count	oplog:likes_book	oplgp:aboutMe
wdrs:describedby	oplog:likes_movie	oplgp>tagline
foaf:title	oplog:likes_music	oplgp:nickname
dcterms:created	oplog:likes_tv_programme	oplgp:name
foaf:homepage		oplgp:placeLived

Figura 46: Alinhamento entre vocabulários para estruturação de conteúdos em Redes Sociais (COSTA, 2013).

7 *CONSIDERAÇÕES FINAIS*

A popularização das tecnologias de informação e comunicação, principalmente do acesso à Web, e o aumento da informatização das organizações ocasionaram um crescente volume de informações. Por causa destes fatores, existe uma necessidade cada vez maior de integração de informações, dispersas em diferentes fontes e em domínios de conhecimentos.

À esta necessidade, soma-se a de interoperabilidade e reúso de softwares para a evolução e alinhamento da infraestrutura de TI às mudanças das organizações em relação aos seus processos cada vez mais dinâmicos. Neste contexto, a plataforma ARIDUS apresentada neste documento é uma solução inovadora, no sentido da aplicação, em conjunto, do paradigma de orientação a serviços e das tecnologias da web semântica para atender tais necessidades.

Na ARIDUS, todas as funcionalidades para integração semântica de informações são providas como serviços Web, potencializando a interoperabilidade e reúso da plataforma e possibilitando a flexibilidade necessária para aplicações em diferentes cenários e integração de informações heterogêneas.

A plataforma ARIDUS, projeto aprovado (Edital UNIVERSAL N° 14/2011) pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), deu suporte ao projeto de mestrado (Edital MCT/CNPq N° 70/2009 – Mestrado/Doutorado – Processo 558489/2010-9) que, dentre outros resultados, apresenta a aplicação IntercropApp (PEREIRA, 2012) e o artigo (PEREIRA; DANTAS; RIBEIRO, 2012) apresentado na 14^a edição da *International Conference on Information Integration and Web-based Applications & Services (iiWAS2012)*¹.

A extensão da plataforma com os modelos para personalização e recomendação, desenvolvidos por membros do Grupo de Sistemas Distribuídos (GSiD) e discentes do Programa de Pós-Graduação em Ciência da Computação (PPgCC-UERN/UFERSA), pode validar a integração de informações pela ARIDUS e o funcionamento dos próprios modelos defendidos por Soares (2013) e Sorrentino (2013).

¹<http://www.iiwas.org/conferences/iiwas2012/>

São claras as contribuições que a infraestrutura construída oferece, sendo, portanto, grande a perspectiva de trabalhos futuros. Dentre alguns trabalhos possíveis estão o desenvolvimento do serviço para anotação semântica que atenda aos requisitos elencados (seção 5.2.9), a decomposição funcional de recursos de extração para criação de novos serviços e aumento das possibilidades de composição que podem ser realizadas.

O código-fonte do protótipo da ARIDUS é mantido no repositório *SourceForge*², especificamente no endereço <https://sf.net/p/aridus>. Além disso, há interesse em disponibilizar a infraestrutura pronta para utilização, ou seja, execução dos serviços, extensão/adição de novos serviços e no desenvolvimento de novas aplicações em domínios diferentes daquelas apresentadas neste documento.

²<https://sourceforge.net>

Referências

- ACQUAVIVA, C. *Composição de Serviços – Parte 1, 2 e 3*. Maio 2011. Disponível em: <<https://blogs.oracle.com/StrawberryFieldsForever>>.
- ALBRESHINE, A.; FUHRER, P.; PASQUIER, J. Web services orchestration and composition. September 2009.
- AMINI, A.; SABOOHI, H.; NEMATBAKHSI, N. A rdf-based data integration framework. In: *National Electrical Engineering Conference (NEEC), 2008*. Iran: [s.n.], 2008.
- BARRETT, T. et al. Rdf representation of metadata for semantic integration of corporate information resources. In: *Workshop on Real world RDF and Semantic Web Applications, 11th International World Wide Web Conference (WWW 2002)*. [S.l.: s.n.], 2002.
- BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. The semantic web. In: *Scientific American*. [S.l.: s.n.], 2001.
- CAIPING, C.; HUIJING, W. A framework for medical information integration based on ontology and web services. In: *Information Engineering and Computer Science (ICIECS), 2010 2nd International Conference on*. [S.l.: s.n.], 2010. p. 1–4. ISSN 2156-7379.
- COSTA, F. S. da. *PerSoN: Um modelo semântico para personalização e interoperabilidade entre redes sociais*. Dissertação (Mestrado) — UERN/UFERSA - Mestrado em Ciência da Computação, jun. 2013.
- CUADRA, A. et al. A semantic web-based integration framework. In: *Next Generation Web Services Practices (NWeSP), 2011 7th International Conference on*. [S.l.: s.n.], 2011. p. 93–98.
- DAVID, J. et al. The alignment api 4.0. *Semantic Web – Interoperability, Usability, Applicability*, v. 2, n. 1, p. 3–10, 2011. Disponível em: <http://www.semantic-web-journal.net/sites/default/files/swj60_1.pdf>.
- DIAS, T. D. *Web Semântica: Fundamentos e Tecnologias*. Monografia (Graduação) — Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2011.
- DIMITROV, D. et al. Isens: A system for information integration, exploration, and querying of multi-ontology data sources. In: *Semantic Computing, 2009. ICSC '09. IEEE International Conference on*. [S.l.: s.n.], 2009. p. 330–335.
- DUCHARME, B. *Learning SPARQL*. O'Reilly Media, 2011. (Oreilly and Associate Series). ISBN 9781449306595. Disponível em: <<http://books.google.com.br/books?id=GVQcFBh4ebgC>>.
- ELLER, M. P. *Anotações Semânticas de Fontes de Dados Heterogêneas: Um estudo de caso com a Ferramenta Smore*. Monografia (Graduação) — Universidade Federal de Santa Catarina – Departamento de Informática e Estatística, Florianópolis, 2008.

ERL, T. *SOA: Princípios de design de serviço / Tradução Edson Furmankiewicz e Carlos Schafranski*. [S.l.]: Prentice Hall PTR, 2009. ISBN 978-85-7605-1889-3.

FISHER, M.; DEAN, M.; JOINER, G. *Use of OWL and SWRL for Semantic Relational Database Translation*. 2008.

FONTES, C. A.; MOURA, A. M. de C.; CAVALCANTI, M. C. Anotação semântica em documentos. In: *IX Workshop de Teses e Dissertações em Banco de Dados (WTDBD 2010)*. [S.l.: s.n.], 2010.

GU, C.; ZHANG, X. An soa based enterprise application integration approach. In: *Electronic Commerce and Security (ISECS), 2010 Third International Symposium on*. [S.l.: s.n.], 2010. p. 324–327.

JOSUTTIS, N. M. *Soa na Prática*. [S.l.]: Alta Books, 2008. ISBN 978-85-7608-1845.

KEENEY, J. et al. Approaches to relating and integrating semantic data from heterogeneous sources. In: *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*. [S.l.: s.n.], 2011. v. 1, p. 170–177.

LOPES, D. V. *Análise Comparativa: Metodologias para desenvolvimento de software com arquitetura orientada a serviços*. 134 f. Monografia (Graduação) — Faculdade Sete de Setembro - FASETE, Paulo Afonso, 2009.

LU, W.; ZHANG, D. Research of enterprise application integration base on service oriented architecture. In: *Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on*. [S.l.: s.n.], 2009. p. 1–9.

MACIAS-GARCIA, M. A.; SOSA-SOSA, V. J.; LOPEZ-AREVALO, I. A generic approach for data integration using rdf, owl and xml. In: *Machine Learning and Data Mining, 2009. WMLDM '09. First Workshop on*. [s.n.], 2009. Disponível em: <http://www.micai.org/2009/proceedings/complementary/cd/wsmldm/215/20091105_MiguelMacias_Micai09_workshop_ML_and_DM.pdf>.

MAKRIS, K. et al. *SPARQL Rewriting for Query Mediation over Mapped Ontologies*. [S.l.], 2010. Disponível em: <<http://www.music.tuc.gr/reports/SPARQLREWRITING.PDF>>.

OASIS STANDARD. *Reference Model for Service Oriented Architecture 1.0*. [S.l.], October 2006. Disponível em: <<http://docs.oasis-open.org/soa-rm/v1.0/>>.

PEREIRA, D. H. G. *Estruturação e Extração de Conhecimento através de Tecnologias Semânticas: Um estudo de caso no domínio da agricultura do Semiárido*. Dissertação (Mestrado) — UERN/UFERSA - Mestrado em Ciência da Computação, 2012.

PEREIRA, D. H. G.; DANTAS, C. F. F.; RIBEIRO, C. M. F. A. A pragmatic approach for sustainable development based on semantic web services. In: *Proceedings of the 14th International Conference on Information Integration and Web-based Applications & Services*. New York, NY, USA: ACM, 2012. (IIWAS '12), p. 82–90. ISBN 978-1-4503-1306-3. Disponível em: <<http://doi.acm.org/10.1145/2428736.2428752>>.

- POPOV, B. et al. Kim - semantic annotation platform. In: FENSEL, D.; SYCARA, K.; MYLOPOULOS, J. (Ed.). *The Semantic Web - ISWC 2003*. Springer Berlin Heidelberg, 2003, (Lecture Notes in Computer Science, v. 2870). p. 834–849. ISBN 978-3-540-20362-9. Disponível em: <http://dx.doi.org/10.1007/978-3-540-39718-2_53>.
- QING-RUI, G.; HAI-TAO, Z. Research of using soa build enterprise application integration strategy. In: *Information Management and Engineering (ICIME), 2010 The 2nd IEEE International Conference on*. [S.l.: s.n.], 2010. p. 435–438.
- SCHWARTE, A. et al. Fedx: A federation layer for distributed query processing on linked open data. In: ANTONIOU, G. et al. (Ed.). *The Semantic Web: Research and Applications*. Springer Berlin Heidelberg, 2011, (Lecture Notes in Computer Science, v. 6644). p. 481–486. ISBN 978-3-642-21063-1. Disponível em: <http://dx.doi.org/10.1007/978-3-642-21064-8_39>.
- SCHWARTE, A. et al. Fedx: optimization techniques for federated query processing on linked data. In: *Proceedings of the 10th international conference on The semantic web - Volume Part I*. Berlin, Heidelberg: Springer-Verlag, 2011. (ISWC'11), p. 601–616. ISBN 978-3-642-25072-9. Disponível em: <<http://dl.acm.org/citation.cfm?id=2063016.2063055>>.
- SHI, Y. et al. Semantic-based data integration model applied to heterogeneous medical information system. In: *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on*. [S.l.: s.n.], 2010. v. 2, p. 624–628.
- SHOAI, M.; BASHARAT, A. Semantic web based integrated agriculture information framework. In: *Computer Research and Development, 2010 Second International Conference on*. [S.l.: s.n.], 2010. p. 285–289.
- SOARES, P.; TANAKA, A.; BAIÃO, F. *Estudo dos Principais Conceitos sobre Integração de Dados Geoespaciais*. [S.l.], Dezembro 2010. Em português, 38 páginas.
- SOMMERVILLE, I. *Software Engineering*. [S.l.]: Person/Addison-Wesley, 2007. (International computer science series). ISBN 9780321313799.
- SORRENTINO, T. A. *@dapt - Um Modelo Semântica para prover Acessibilidade para Deficientes Visuais na Internet*. Dissertação (Mestrado) — UERN/UFERSA - Mestrado em Ciência da Computação, fev. 2013.
- TAO, K. et al. Tums: Twitter-based user modeling service. In: GARCÍA-CASTRO, R.; FENSEL, D.; ANTONIOU, G. (Ed.). *The Semantic Web: ESWC 2011 Workshops*. Springer Berlin Heidelberg, 2012, (Lecture Notes in Computer Science, v. 7117). p. 269–283. ISBN 978-3-642-25952-4. Disponível em: <http://dx.doi.org/10.1007/978-3-642-25953-1_22>.
- TELANG, A.; CHAKRAVARTHY, S. *Information Integration across Heterogeneous Domains: Current Scenario, Challenges and The InfoMosaic Approach*. [S.l.], Dezembro 2007. In english, 31 pages.
- WACHE, H. et al. Ontologybased integration of information – a survey of existing approaches. In: . [S.l.: s.n.], 2001. p. 108–117.
- WU, J. Heterogeneous data integration model based on virtual view. In: *Computer Science Education (ICCSE), 2012 7th International Conference on*. [S.l.: s.n.], 2012. p. 815–817.

XIANG, S. et al. Research on the semantic web-based technology of knowledge integration for agricultural production. In: *Fuzzy Systems and Knowledge Discovery, 2009. FSKD '09. Sixth International Conference on*. [S.l.: s.n.], 2009. v. 2, p. 361 –366.

ZHOU, J. et al. A survey of semantic enterprise information integration. In: *Information Sciences and Interaction Sciences (ICIS), 2010 3rd International Conference on*. [S.l.: s.n.], 2010. p. 234–239.