

# Arcabouço para Anotação de Componentes de Imagem

**Émerson Muraro**

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Émerson Muraro e aprovada pela Banca Examinadora.

Campinas, 31 de Julho de 2012.

Prof. Dr. Ricardo da Silva Torres  
(IC/Unicamp) (Orientador)

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

FICHA CATALOGRÁFICA ELABORADA POR  
ANA REGINA MACHADO - CRB8/5467  
BIBLIOTECA DO INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E  
COMPUTAÇÃO CIENTÍFICA - UNICAMP

M931a Muraro, Émerson, 1986-  
Arcabouço para anotação de componentes de imagem /  
Émerson Muraro. – Campinas, SP : [s.n.], 2012.

Orientador: Ricardo da Silva Torres.  
Dissertação (mestrado) – Universidade Estadual de Campinas,  
Instituto de Computação.

1. Ontologias (Recuperação da informação). 2. Web semântica.  
3. Componente de software. 4. Sistemas de recuperação da  
informação. 5. Metadados. I. Torres, Ricardo da Silva, 1977-. II.  
Universidade Estadual de Campinas. Instituto de Computação. III.  
Título.

Informações para Biblioteca Digital

**Título em inglês:** A framework for semantic annotation of image components

**Palavras-chave em inglês:**

Ontologies (Information retrieval)

Semantic Web

Component software

Information storage and retrieval systems

Metadata

**Área de concentração:** Ciência da Computação

**Titulação:** Mestre em Ciência da Computação

**Banca examinadora:**

Ricardo da Silva Torres [Orientador]

Carla Geovana do Nascimento Macário

Luiz Camolesi Júnior

**Data de defesa:** 31-07-2012

**Programa de Pós-Graduação:** Ciência da Computação

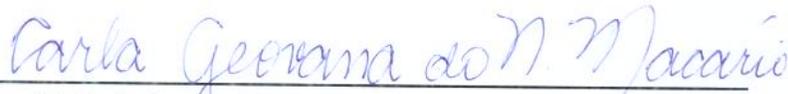
## TERMO DE APROVAÇÃO

Dissertação Defendida e Aprovada em 31 de Julho de 2012, pela  
Banca examinadora composta pelos Professores Doutores:



---

**Prof. Dr. Luiz Camolesi Júnior**  
FT / UNICAMP



---

**Prof.ª. Dr.ª. Carla Geovana do Nascimento Macário**  
CNPTIA/EMBRAPA



---

**Prof. Dr. Ricardo da Silva Torres**  
IC / UNICAMP

# Arcabouço para Anotação de Componentes de Imagem

Émerson Muraro

Julho de 2012

## Banca Examinadora:

- Prof. Dr. Ricardo da Silva Torres  
(IC/Unicamp) (Orientador)
- Prof. Dr. Luiz Camolesi Júnior  
(FT/Unicamp)
- Dr<sup>a</sup>. Carla Geovana do Nascimento Macário  
(Embrapa)
- Prof<sup>a</sup>. Dr<sup>a</sup>. Islene Calciolari Garcia  
(IC/Unicamp)
- Prof. Dr. Luciano Antonio Digiampietri  
(EACH/USP)

# Resumo

Com a disseminação de dispositivos mais baratos para sua aquisição, armazenamento e disponibilização, *imagens* vêm sendo utilizadas em várias aplicações (tais como comerciais, científicas, e pessoais). O uso de imagens nessas aplicações tem motivado a criação de objetos digitais heterogêneos. Imagens não são usadas isoladamente e podem compor outros objetos digitais. Esses novos objetos digitais são conhecidos como *Objetos Complexos*.

Esta dissertação apresenta um arcabouço para anotação semântica automática de componentes de imagem, visando o seu uso na construção de objetos complexos. Esta proposta utiliza diversas formas de busca para encontrar termos para anotação: ontologias, busca por palavras-chaves e por conteúdo visual. Os termos encontrados são ponderados por pesos que definem sua importância, e são combinados por técnicas de fusão de dados em uma única lista de sugestões. As principais contribuições deste trabalho são: especificação do processo de anotação semântica automática de componentes de imagem, que considera o conteúdo visual da imagem, palavras-chaves definidas, ontologias e possíveis combinações envolvendo estas alternativas e especificação e implementação parcial de um arcabouço para anotação de objetos complexos de imagens encapsulados em componentes.

# Abstract

Due to the dissemination of low-cost devices for acquisition, storage, and sharing, images have been used in several applications, (e.g., commercial, scientific, and personal). The use of images in those applications has motivated the creation of heterogeneous digital objects. Images are not longer used in isolation and are used to compose other digital objects, named *Complex Objects*.

In this work, we present a new framework for automatic semantic annotation of image components, aiming at supporting their use in the construction of complex objects. Our proposal uses several approaches for defining appropriate terms to be used in the annotation process: ontologies, textual terms, and image content descriptions. Found terms are weighted according to their importance, and are combined using data fusion techniques. The main contributions of this work are: the specification of an automatic semantic annotation process for image components, that takes into account image visual properties, defined textual terms, ontologies, and their combination, and the specification and partial implementation of an infrastructure for annotating image complex objects encapsulated in components.

# Agradecimentos

Eu gostaria de agradecer em primeiro lugar aos meus pais, Laercio e Margaret, que sempre me apoiaram, por serem pais excelentes que conseguiram me ensinar muitas lições de humildade, honestidade, trabalho, ética, amizade, amor, a ter fé em Deus, o que é ser uma família e um ser humano bom e com grandes princípios. Agradeço à minha família por estar presente nos momentos bons e ruins e me animar no dia-a-dia.

Ao meu orientador por ser um amigo nos momentos bons e difíceis, conseguindo enxergar luz no final do túnel e me incentivando muito em momentos críticos. Agradeço por sua paciência, por me aceitar em tempo parcial e por conseguirmos juntos chegar ao final desse trabalho que marca o começo de uma boa amizade. Sempre apresentou ideias muito boas e levou em consideração as minhas sugestões e interesses. Para mim é um exemplo de orientador.

Agradeço à Jaudete pelas horas de ajuda com o Aondê, pelas sugestões sobre o mestrado e por disponibilizar seu trabalho. Ao Bruno Vilar pela ajuda com o Aondê e suas configurações; estando sempre respondendo meus *e-mails* e chamadas no *talk*.

Ao Prof. Dr. André Santanchè pelas explicações e ajudas com DCC. Por mostrar seu funcionamento e me receber de braços abertos, só querendo ajudar.

À Nádia P. Kozievitch por cooperar com a sua tese de doutorado, me oferecendo conhecimento e grandes ideias.

Não poderia deixar de agradecer a todos do IC e da Unicamp, professores e demais funcionários, que participaram diretamente ou indiretamente durante a realização do meu trabalho.

Agradeço a CAPES, CNPq, FAPESP, AMD e Microsoft pelo apoio relacionado à infraestrutura do laboratório RECOD, no qual esta pesquisa foi desenvolvida.

Por fim, à empresa na qual trabalho, Ci&T, que permitiu uma flexibilidade de horários para que eu conseguisse fazer o mestrado.

# Sumário

Resumo	v
Abstract	vi
Agradecimentos	vii
<b>1 Introdução</b>	<b>1</b>
1.1 Objetivos e Contribuições	3
1.2 Organização da Dissertação	5
<b>2 Revisão Bibliográfica</b>	<b>6</b>
2.1 Objetos Complexos	7
2.1.1 Definição	7
2.1.2 DCC - <i>Digital Content Component</i>	8
2.1.3 Uma Infraestrutura para Objetos Complexos de Imagem	12
2.2 Anotação Semântica	17
2.2.1 Anotação	17
2.2.2 Ontologias	23
2.2.3 Web Semântica	24
2.2.4 Buscas Semânticas	26
2.3 Aondê	27
2.3.1 Arquitetura e Conceitos do Aondê	27
2.3.2 Módulo de Gerência dos Repositórios	28
2.3.3 Módulo de Consultas	29
2.3.4 Módulo de Busca e <i>Ranking</i>	29
2.3.5 Módulo de Visões	31
2.3.6 Módulo de Integração	31
2.3.7 Módulo de Detecção de Diferenças	32
2.4 OntoSAIA: Uma Ontologia Baseada em Busca de Imagem e Uma Ferramenta de Anotação Semi-Automática	33

2.4.1	Arquitetura . . . . .	33
2.4.2	Funcionamento e Aspectos de Implementação . . . . .	34
2.5	Fusão de Dados . . . . .	35
2.6	Considerações do Capítulo . . . . .	36
<b>3</b>	<b>Especificação do Arcabouço de Anotação</b>	<b>38</b>
3.1	Algoritmos de Busca e Recomendação de Termos . . . . .	40
3.1.1	Busca por Texto . . . . .	41
3.1.2	Busca por Conteúdo . . . . .	41
3.1.3	Busca por Conteúdo e Texto . . . . .	42
3.1.4	Busca Semântica . . . . .	43
3.2	Processo de Anotação de Imagem . . . . .	45
3.2.1	Fluxos Básicos de Busca . . . . .	46
3.2.2	Combinações de Fluxos de Busca . . . . .	47
3.2.3	Atribuição de Peso . . . . .	47
3.2.4	Fusão de Dados . . . . .	48
3.2.5	Busca de Ontologia por Termo . . . . .	48
3.2.6	Criação de DCCs . . . . .	49
3.3	Arquitetura do Arcabouço . . . . .	49
3.3.1	Interface . . . . .	50
3.3.2	Camada de Controle . . . . .	50
3.3.3	Camada de Serviços . . . . .	52
3.3.4	Repositórios . . . . .	54
3.4	Considerações do Capítulo . . . . .	55
<b>4</b>	<b>Aspectos de Implementação e Estudos de Caso</b>	<b>56</b>
4.1	Aspectos de Implementação . . . . .	57
4.1.1	Configuração e Adaptação do OntoSAIA . . . . .	58
4.1.2	Configuração e Adaptação do Aondê . . . . .	60
4.1.3	Módulo do Controlador . . . . .	60
4.1.4	Métodos Básicos para Busca de Termos . . . . .	62
4.1.5	Fluxos de Anotação . . . . .	66
4.1.6	Módulo de Atribuição de Peso . . . . .	66
4.1.7	Módulo de Fusão de Dados . . . . .	67
4.1.8	Módulo de DCC . . . . .	67
4.2	Estudos de Caso . . . . .	69
4.2.1	Estudo de Caso Apenas com Imagem . . . . .	71
4.2.2	Estudo de Caso com Imagem e Anotação . . . . .	73

<b>5</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>78</b>
5.1	Contribuições . . . . .	78
5.2	Trabalhos Futuros . . . . .	80
	<b>Bibliografia</b>	<b>82</b>

# Lista de Figuras

2.1	Tipos de objetos digitais. . . . .	8
2.2	Estrutura de um DCC. . . . .	9
2.3	Simulador da taxa de crescimento de café, utilizando DCCs - imagem retirada de [57]. . . . .	10
2.4	Arquitetura da infraestrutura de biblioteca digital baseada em Objetos Complexos - figura retirada de [36]. . . . .	13
2.5	Visão geral da infraestrutura de objetos complexos de imagem para bibliotecas digitais - figura retirada de [36]. . . . .	14
2.6	Exemplo de anotações e metadados da página Web do ontotext [47]. . . . .	19
2.7	Etapas do <i>Framework</i> GeoSpatial - figura retirada de [43]. . . . .	21
2.8	Camadas de padrões e tecnologias da Web Semântica. . . . .	25
2.9	Arquitetura do Serviço Web Aondê (figura retirada de [15]). . . . .	28
2.10	Arquitetura de três camadas do OntoSAIA (figura retirada de [26]). . . . .	34
2.11	Exemplo de fusão de dados com o método Borda Count. . . . .	37
3.1	Macro fluxo do processo de anotação, presente no arcabouço. . . . .	38
3.2	Diagrama de sequência do processo de anotação semântica automática de um DCC de imagem. . . . .	39
3.3	Parte da arquitetura do arcabouço que contém os serviços de busca e o módulo que os controla. . . . .	41
3.4	Exemplo de ontologia com seus respectivos pesos. . . . .	45
3.5	Arquitetura do arcabouço de anotação de componentes de imagem. . . . .	49
4.1	Serviços Web do Aondê e seus métodos disponíveis no ambiente de desenvolvimento ( <i>localhost</i> ). . . . .	61
4.2	Algoritmo do método de combinação por <i>summative union</i> . . . . .	64
4.3	Algoritmo do Borda Count para a fusão das listas de termos sugeridos. . . . .	68
4.4	Esquema de dados das novas tabelas, criadas para atender o arcabouço de anotação. . . . .	69
4.5	Interface do OntoSAIA com o ajuste do arcabouço de anotação. . . . .	70

4.6	Interface OntoSAIA: carregando imagem. . . . .	71
4.7	Interface OntoSAIA: disparando o processo de anotação de imagem e geração de DCC. . . . .	72
4.8	Interface OntoSAIA: final do processo de anotação, mostrando os termos anotados e seu pesos. . . . .	73
4.9	Interface OntoSAIA: inserção dos termos e iniciando o processo de anotação de imagem e geração do DCC. . . . .	74
4.10	Interface OntoSAIA: final do processo de anotação, mostrando os termos anotados e seu pesos. . . . .	75
4.11	Representação simplificado do DCC de imagem anotado semanticamente gerado pelo arcabouço de anotação. . . . .	76

# Capítulo 1

## Introdução

Um grande número de objetos (artefatos) digitais, tais como textos, código e objetos multimídia, vem sendo disponibilizados e compartilhados na Web. Este cenário [59] faz surgir a demanda por novas ferramentas e técnicas para recuperar artefatos digitais apropriados, visando sua (re-)utilização.

Com a disseminação de dispositivos mais baratos para aquisição, armazenamento e disponibilização, *imagens* vêm sendo utilizadas em várias aplicações: comerciais, científicas e pessoais. Do ponto de vista comercial, por exemplo, imagens associadas a objetos vendidos por lojas virtuais vêm sendo empregadas para auxiliar potenciais clientes na escolha de produtos mais adequados a seus interesses. Na área científica destaca-se o uso de imagens em: sistemas de informação de biodiversidade [14, 46]; ou no entendimento de fenômenos ambientais a partir da análise de imagens de sensoriamento remoto [20]. Do ponto de vista pessoal, um grande número de imagens vem sendo compartilhadas em redes sociais, como o Flickr<sup>1</sup>. No caso do *site* Flickr, por exemplo, em média, aproximadamente 560 milhões de imagens foram inseridas no sistema, por mês, em 2011.<sup>2</sup>

O uso de imagens nessas aplicações tem motivado a criação de objetos digitais heterogêneos. Imagens não são usadas em isolamento e passam a compor outros objetos digitais. Esses novos objetos digitais são conhecidos como *Objetos Complexos*. Segundo [41], objetos complexos podem ser definidos como uma única entidade que é composta de múltiplas partes, que podem ser outras entidades compostas ou uma entidade simples definida por si só como um objeto. Exemplos de aplicações que têm usado objetos complexos do tipo imagem incluem: gerenciamento de dados sobre espécies em sistemas de informação de biodiversidade [38, 46]; gerenciamento de grandes coleções de impressões digitais [39] e gerenciamento de histórias de vídeos [37].

Esse grande volume de dados faz aumentar a demanda por sistemas que permitam a

---

<sup>1</sup><http://www.flickr.com/> (último acesso em junho de 2012).

<sup>2</sup><http://www.flickr.com/photos/franckmichel/6855169886/> (último acesso em junho de 2012).

busca de imagens, que compõem ou não objetos complexos. Usualmente, duas técnicas costumam ser empregadas nesse processo: busca textual [33], comumente baseada no casamento de palavras-chaves definidas no momento da busca com anotações associados às imagens; e busca por conteúdo [12,13], na qual propriedades visuais como cor, textura e forma são utilizadas para se encontrar imagens de interesse na coleção.

A anotação de imagens é uma prática cada vez mais comum para viabilizar a realização de buscas. Ela permite associar mais informações a uma imagem, como análises de regiões de interesse em imagens de sensoriamento remoto [18], ou análises de características morfológicas relevantes de seres vivos [38], entre outras. O serviço de anotação tem sido reconhecido como um dos mais importantes serviços em sistemas de bibliotecas digitais, para promover cooperação entre usuários e integração de recursos de informações heterogêneas [1, 29].

O uso de serviço de anotação permite que usuários ou grupos de usuários incluam anotações pessoais em recursos de informações gerenciadas, ou seja, anotar dados armazenados em: banco de dados, bibliotecas digitais e/ou na Web [1]. Anotações podem ser consideradas como metadados, ou seja, é um dado que define ou esclarece as informações de outro dado existente em um sistema de informação [1]. O dado utilizado para anotar serve para esclarecer as propriedades e semântica do conteúdo anotado [1].

Na computação, anotação serve para descrever recursos e o que eles fazem, utilizando conceitos formais e termos de ontologias para descrevê-lo. Os termos das ontologias são referenciados por metadados ou um conjunto de metadados [19]. Uma anotação semântica é representada por um conjunto de metadados que fornece uma referência para cada entidade anotada, por meio de um identificador Web único, como um URI [19]. Dessa forma, anotações semânticas referenciam ontologias, descritas em linguagens específicas, que permitem eliminar ambiguidades e promover um entendimento comum dos conceitos [29].

Recentemente, foi proposta uma infraestrutura para gerenciamento de objetos complexos de imagem [36,40]. Nessa infraestrutura, objetos complexos, que incluem imagens como um dos seus elementos, foram organizados em uma estrutura de componentes. Componentes também foram usados para encapsular *softwares* específicos voltados à busca multimodal de objetos complexos de imagem, baseada em descrições textuais e visuais. Além do encapsulamento dos dados e do *software*, a infraestrutura proposta permite a publicação de objetos complexos.

O modelo de componentes escolhido para o encapsulamento e representação dos processos e dados na infraestrutura proposta em [36, 40] é o *Digital Content Component* (DCC) [3]. DCC é apresentado como uma estrutura para representar qualquer tipo de conteúdo digital, baseada em componentes da Engenharia de Software. O uso de DCCs busca tratar problemas relacionados ao gerenciamento de grande volume de dados em ambientes distribuídos, tais como a heterogeneidade e apoio à recuperação e ao reúso de

objetos digitais. Alguns avanços foram obtidos com relação à busca de DCCs em repositórios na Web. Um deles consiste no uso de marcações semânticas conforme estipulado pelos padrões da Web Semântica, para fornecer um meio mais eficaz de busca e utilização desses componentes [59].

Os esforços com relação ao uso de recursos da Web Semântica foram desenvolvidos em duas direções: sintaxe e semântica comum para troca de dados; e infraestrutura sintática e semântica comum para fornecer interoperabilidade entre processos [59]. Na Web, há um vasto universo de recursos de anotação, que já estão embutidos na organização dos artefatos de dados e processos, dependendo da sua natureza. Esses artefatos e seus recursos podem ser utilizados em ambas as direções citadas acima, mas a dificuldade está em como associar as anotações da Web Semântica com os artefatos considerando sua intrínseca heterogeneidade [59]. Santanchè propôs em seu trabalho [59] o uso de DCCs [3] seguindo os padrões de anotações da Web Semântica para resolver essas dificuldades: RDF [64] e OWL [45]/OWL-S [44] para descrever os dados e o recursos disponíveis na Web; WSDL [9], UDDI [10] e SOAP [32] dos Serviços Web.

A proposta da infraestrutura para gerenciamento de objetos complexos de imagem apresentada em [36, 40] é promissora, mas não contemplou a definição de mecanismos para criação automática de objetos complexos de imagem. Similarmente, embora haja iniciativas, como as descritas acima, para anotação semântica de DCCs, esse processo é manual, dependente de aplicação e não trata especificamente de propriedades visuais comumente empregadas para descrever o conteúdo de imagens. Esta dissertação de mestrado preenche estas lacunas.

## 1.1 **Objetivos e Contribuições**

O objetivo desta dissertação de mestrado é especificar e implementar parcialmente um arcabouço para anotação semântica automática de DCCs de imagens, visando o seu uso na construção de objetos complexos. Dentre as contribuições desta pesquisa, destacam-se:

- especificação do processo de anotação semântica automática de componentes de imagem, considerando-se recomendações de anotações por:
  - conteúdo visual da imagem;
  - palavras-chaves definidas;
  - ontologias;
  - e possíveis combinações envolvendo estas alternativas.
- associação entre anotação, metadado e ontologia;

- especificação de um arcabouço para anotação de objetos complexos de imagens encapsulados em DCCs, conforme o processo de anotação especificado;
- implementação parcial do arcabouço e apresentação de estudos de caso que ilustram o uso de alguns módulos do arcabouço de anotação.

Este trabalho foi motivado pela realização de estudo anterior sobre busca semântica. Foram estudadas algumas técnicas de busca semântica, como expansão de consultas, busca e *ranking* e busca baseada em ontologia. Durante esse estudo e com base nas avaliações obtidas, identificou-se a necessidade de como obter ou converter os dados na Web para as estruturas utilizadas pelas técnicas de busca semântica ou que facilitem a busca. Por exemplo, imagens anotadas ou objetos complexos de imagem. Com isso, foi realizado o estudo de alguns conceitos e ferramentas fundamentais para o desenvolvimento desse trabalho:

- Estudo dos conceitos de anotação, anotação semântica e ontologia, para conseguir definir o processo de anotação semântica automática de componentes de imagem e especificação de formas de se realizar recomendações de anotações;
- Estudo das estruturas e padrões da Web semântica e de busca semânticas, com o intuito de modo a utilizar os conceitos de metadados, ontologias e serviços Web na especificação do processo de anotação e implementação do arcabouço de anotação;
- Estudo sobre objetos complexos e componentes de conteúdo digital (DCC), para encapsulamento dos dados e melhoria dos mecanismos de busca;
- Estudo das ferramentas de busca baseada em ontologia, conteúdo visual de imagem e palavras-chaves, e de ferramentas para armazenamento de imagens, metadados e ontologias;
- Estudo do conceito e métodos de Fusão de Dados para combinar várias listas ordenadas de entidades, obtendo uma única lista com as anotações sugeridas.

Com base nesses estudos, a solução proposta para o processo de anotação semântica automática de componentes de imagem é baseada na reutilização de dois sistemas, recentemente propostos no Instituto de Computação da Unicamp.

- O primeiro, chamado Aondê, é um serviço Web proposto em [15] que teve como objetivo disponibilizar funcionalidades para manipulação e armazenamento de diversas ontologias. A motivação para o desenvolvimento do Aondê consistiu em prover mais semântica associada à informação armazenada ou manipulada, permitindo lidar melhor com problemas como a heterogeneidade de dados. Este serviço

apoia um amplo espectro de funcionalidades, que permitem realizar operações de armazenamento, gerenciamento, busca, *ranking*, análise, comparação e integração de ontologias. Esses recursos são utilizados para implementar as funcionalidades de busca e armazenamento de ontologias do arcabouço de anotação proposto.

- O outro sistema usado como base é o OntoSAIA [26]. O OntoSAIA é uma ferramenta que permite o armazenamento, indexação e recuperação de imagens. O foco principal do OntoSAIA é permitir anotação manual de imagens, a partir de recomendações (anotações sugeridos pelo sistema). Dessa ferramenta são reaproveitados os recursos de busca por conteúdo e busca textual nas anotações armazenadas no repositório do OntoSAIA.

Dessa forma o processo de anotação semântica automático de componentes de imagem pode ser dividido em dois fluxos principais:

- *Fluxos de busca de anotações*, que incluem os fluxos básicos de busca e os fluxos compostos;
- *Fluxo de processamento das anotações sugeridas e criação de objeto complexo da imagem anotada semanticamente*, envolvendo técnicas de classificação dos termos, fusão dos dados, busca e *ranking* de ontologias de domínio visando a anotação e criação de um DCC de imagem;

Com o processo de anotação definido, é proposta a arquitetura do arcabouço de anotação, dividida em quatro camadas: interface, controle, serviços e repositórios. Essa arquitetura é especificada utilizando recursos de outras ferramentas, ou seja, o arcabouço possui módulos de integração com outras ferramentas. Também faz parte deste trabalho a implementação de alguns módulos do arcabouço de anotação proposto e a sua demonstração a partir da definição de estudos de caso com imagens de peixes.

## 1.2 Organização da Dissertação

Esta dissertação é composta por cinco capítulos: o Capítulo 1 caracteriza o problema tratado nesta pesquisa, assim como descreve os principais objetivos e contribuições do trabalho desenvolvido; o Capítulo 2 é dedicado à apresentação dos principais conceitos envolvidos, assim como dos principais trabalhos relacionados a esta dissertação; o Capítulo 3, intitulado Especificação do Arcabouço, apresenta a principal contribuição desta dissertação: o arcabouço para anotação de componentes de imagem; o Capítulo 4 descreve as tecnologias utilizadas na implementação do arcabouço proposto e apresenta estudos de casos que ilustram o seu uso; e, finalmente, o Capítulo 5 apresenta conclusões e possíveis trabalhos futuros.

# Capítulo 2

## Revisão Bibliográfica

Este capítulo apresenta conceitos e alguns dos principais trabalhos correlatos. Como o foco do trabalho são estruturas para uso de mecanismos de busca semântica, este capítulo irá apresentar uma visão dos conceitos utilizados para a geração de objetos digitais complexos por meio do processo automatizado de anotação de imagens. Os conceitos apresentados estão agrupados em *Objetos Complexos* e *Anotações Semânticas*. Neste capítulo, também são apresentados alguns serviços e recursos de outros trabalhos, como as ferramentas Aondê e OntoSAIA, que foram utilizados durante o desenvolvimento do trabalho proposto nessa dissertação de mestrado.

Este capítulo é composto por cinco seções, responsáveis por apresentar os conceitos, trabalhos e ferramentas utilizados no desenvolvimento dessa dissertação de mestrado. A Seção 2.1 contempla os seguintes assuntos: Apresentação da definição do conceito de objeto complexo e a sua relação com os demais tipos de objetos digitais; Apresentação de uma infraestrutura de componentes digitais - DCC, que se enquadra como um tipo de objeto complexo; Apresentação de uma infraestrutura de objetos complexos de imagem [36]. Os conceitos apresentados nessa seção são utilizados na definição da representação adotada para imagens e suas anotações.

A Seção 2.2 apresenta a definição de alguns conceitos importantes relacionados a anotação, Web Semântica e busca semântica. Esta seção está dividida da seguinte forma: Definição do conceito de anotação e anotação semântica, e alguns trabalhos relacionados; Apresentação do conceito de ontologia e da sua estrutura; Apresentação da estrutura, padrões e conceitos da Web Semântica; Apresentação de alguns tipos de busca semântica e alguns trabalhos correlatos. Os conceitos de anotação e ontologia são os mais importantes para o desenvolvimento deste trabalho, pois auxiliam e facilitam a estruturação de objetos digitais voltados para busca.

A Seção 2.3 apresenta o serviço Web Aondê, que disponibiliza funcionalidades para manipulação e busca de ontologias. Este serviço é utilizado neste trabalho para mani-

pulação e armazenamento de ontologias.

A Seção 2.4 apresenta a ferramenta OntoSAIA, que oferece recursos para gerenciamento de imagens e suas anotações. O principal objetivo dessa ferramenta é oferecer um processo manual de anotação de imagem baseado em sugestões do sistema.

A última seção deste capítulo, Seção 2.5, apresenta o conceito de Fusão de Dados, utilizado para combinar diferentes listas de termos em uma única lista ordenada. O estudo deste conceito foi necessário para identificar uma forma de consolidar os termos sugeridos pelas diversas formas de busca de termos, do processo de anotação proposto neste trabalho.

## 2.1 **Objetos Complexos**

Esta seção apresenta os principais conceitos relacionados a objetos complexos e alguns trabalhos que utilizam esses conceitos para definição de infraestruturas ou componentes de dados.

### 2.1.1 **Definição**

Objetos Complexos podem ser definidos como uma única entidade que é composta de múltiplas partes, que podem ser outras entidades compostas ou uma entidade simples definida por si só como um objeto [41].

Os objetos podem ser estruturados de formas diferentes: atômicos, compostos e complexos [36]. A Figura 2.1 mostra a relação entre os tipos de objetos digitais e abrangência de cada um deles. Objetos atômicos consistem basicamente no conteúdo do próprio objeto digital, sendo apenas de um único formato de dado, como a entidade simples mencionada anteriormente. Um exemplo seria um artigo contendo texto, imagens e tabelas representados em um único formato textual. A estrutura composta é formada a partir de arquivos de conteúdo com mesmo formato ou com formatos diferentes. Um exemplo seria um artigo dividido em vários arquivos, sendo um deles o texto e os demais são as imagens em seus formatos específicos, que quando combinados obtemos um único objeto composto. Dessa forma, o objeto composto é uma combinação de objetos atômicos. O objeto complexo, por sua vez, pode ser visto como uma rede de objetos digitais dentro do repositório, ou seja, é uma coleção de objetos digitais: atômicos, composto ou complexo, que podem ser agrupados e manipulados como um único objeto [36]. Um exemplo seria um artigo que pode ser um objeto complexo contendo vários objetos digitais de imagem, um ou mais objetos digitais com o texto e um ou mais objetos com as referências em formatos de link, compondo assim o artigo. O objeto complexo de artigo ainda pode conter um objeto complexo de imagem que é uma referência a uma imagem de outro artigo, e esse objeto

pode conter referências e informações da sua origem. Toda essa estrutura de objetos com texto, imagem, referências e imagem com referência pode ser vista como um único objeto complexo que compõem um artigo científico.

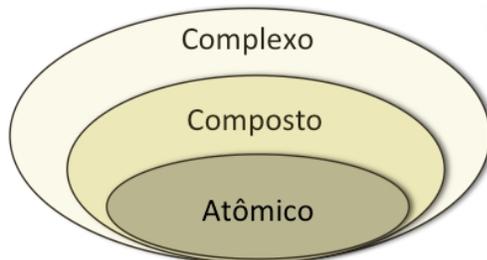


Figura 2.1: Tipos de objetos digitais.

Para trabalhar com um objeto complexo, é preciso definir os limites dos conjuntos e recursos que compõem a entidade de cada objeto complexo. Essa entidade seria como a classe pai que define a entidade e estipula a forma de manipulação desse objeto complexo [41]. Como um objeto complexo pode ser a agregação de qualquer outro objeto, isso abre margens para o gerenciamento de objetos complexos sob diferentes perspectivas:

- Objetos encapsulados por ontologias, que exploram a granularidade do seu conteúdo;
- Explorando as possibilidades e vantagens devido ao estabelecimento de padrões entre os objetos, focados em troca e interoperabilidade de conteúdo (OAI-ORE<sup>1</sup>), entre outros [36, 41].

Dentre esses, destaca-se a proposta de uma estrutura que utiliza objetos complexos para reúso de software e conteúdo [21, 57, 59, 60], além de permitir uma outra forma de classificação e catalogação de artefatos digitais, podendo incluir semântica na sua estrutura e formas de acesso. Na seção 2.1.2, são apresentados mais detalhes sobre essa estrutura.

### 2.1.2 DCC - *Digital Content Component*

DCC é uma unidade de processo e/ou conteúdo de reúso, que foi inspirado no paradigma de componente de software da engenharia de software. Um DCC pode ser visto como um conteúdo (de software ou de dado) encapsulado dentro de uma estrutura de descrição semântica [61]. Um DCC é composto por quatro elementos (Figura 2.2): 1) o próprio conteúdo no seu formato original; 2) uma especificação em XML da estrutura interna usada na organização do componente, permitindo uma hierarquização dos componentes

<sup>1</sup><http://www.openarchives.org/ore/> (último acesso em junho de 2012).

e subcomponentes, especificando assim como estão relacionados os componentes dentro de um DCC; 3) uma especificação RDF [64] da interface do componente, que pode ser representada também em WSDL [9] ou em OWL-S (permite uma melhor representação semântica) [44]; 4) metadados representados em RDF ou em OWL - uma representação semântica mais apropriada no contexto da Web Semântica e de Serviços Web - para descrever funcionalidades, aplicabilidade, restrições de uso, etc [3, 59, 61]. Os componentes de conteúdo digital podem ser recursivamente construídos da composição de outros componentes que possuam as quatro partes da estrutura apresentada na Figura 2.2.

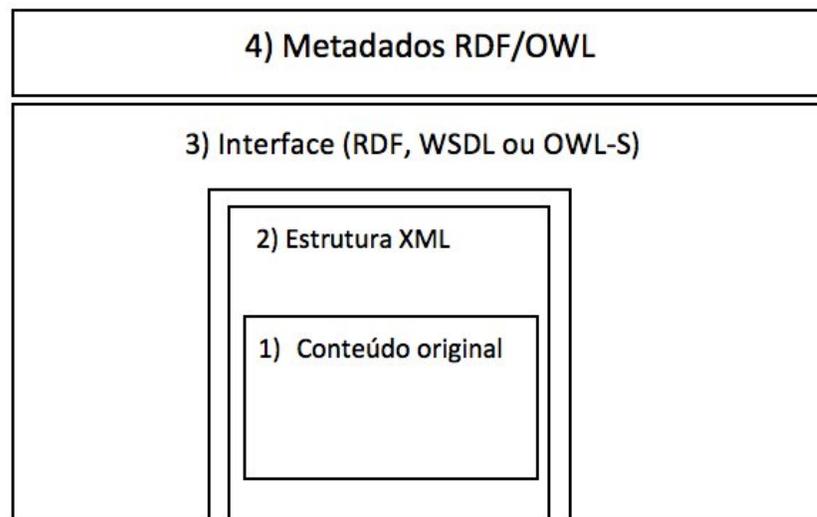


Figura 2.2: Estrutura de um DCC.

Os DCCs podem ser separados em dois grupos: DCCs de processo e DCCs do tipo passivo. Um componente de processo encapsula alguns tipos de descrição de processos (códigos ou sequências de instruções) que podem ser executados por um computador. Um DCC de processo possui uma interface, que a a depende dos valores de entrada definidos, permitem obter resultados diferentes. O DCC passivo concentra-se no conteúdo (texto ou vídeo) e sua interface define como seu conteúdo pode ser acessado [59]. Um DCC passivo pode ser fornecido como entrada de um DCC de processo, que acessa os dados do DCC passivo. A maioria dos componentes de processo não podem ser executados diretamente. Normalmente, é necessário um interpretador ou uma *engine* (módulo de execução), não embutido no DCC, para executar o processo [3].

Por exemplo, o conceito de DCC pode ser utilizado para construção de uma aplicação de simulação da taxa de crescimento de uma planta de café, na região de Campinas [57]. A Figura 2.3 mostra a composição do simulador de café, construído a partir de DCCs.

Este exemplo mostra como DCCs, passivo ou de processos, podem ser utilizados para compor uma aplicação, utilizando serviços Web e tecnologias para orquestração de

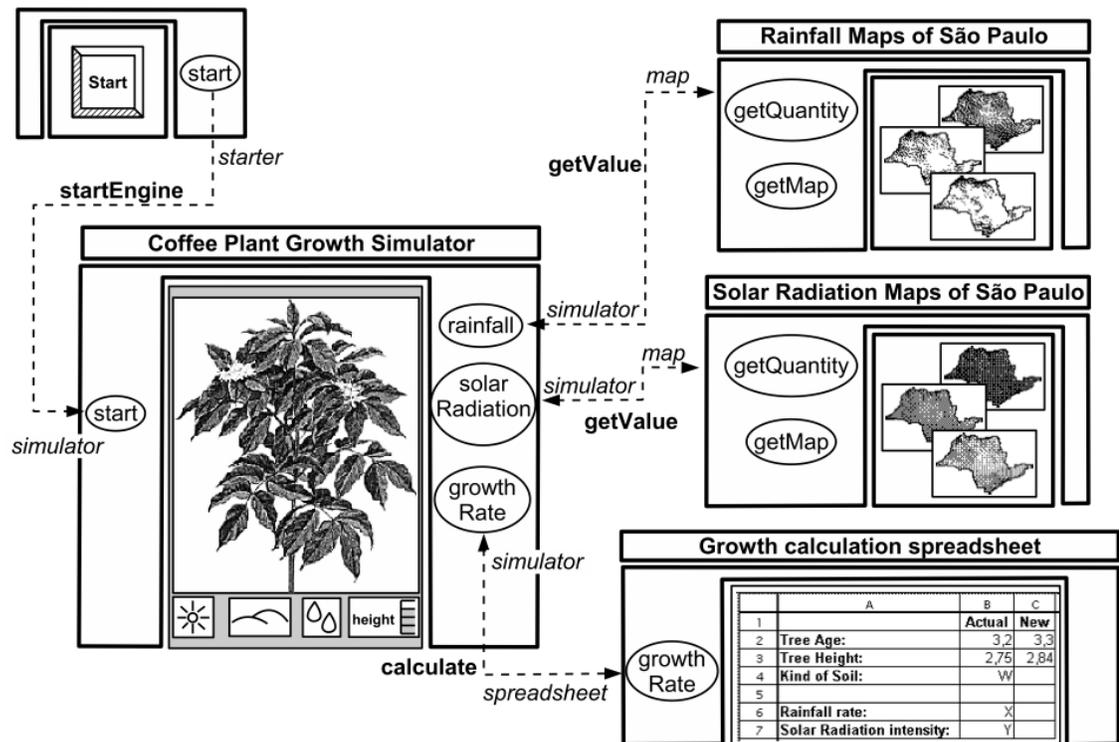


Figura 2.3: Simulador da taxa de crescimento de café, utilizando DCCs - imagem retirada de [57].

serviços. A Figura 2.3 mostra cinco DCCs:

- **Strat**: é um DCC de processo, que inicializa o processo de simulação da taxa de crescimento de uma planta de café;
- **Rainfall Maps of São Paulo**: é um DCC passivo, que contém informações referentes às chuvas na região, para cada período de tempo;
- **Solar Radiation Maps of São Paulo**: é um DCC passivo, que contém informações referentes à radiação solar na região, para cada período de tempo;
- **Growth Calculation Spreadsheet**: é um DCC de processo, que realiza o cálculo da taxa de crescimento da planta de café, baseado nas informações de chuva, radiação solar e outros parâmetros específicos definidos por um especialista da área;
- **Coffee Plant Growth Simulator**: é um DCC de processo, que contém o software responsável por obter as informações necessárias para realizar a simulação e mostrar o resultado, referente ao crescimento da planta de café para os ciclos previamente

configurado. Um ciclo é um intervalo de tempo, e o simulador pode realizar o cálculo para vários ciclos. Este DCC é um exemplo de composição de DCCs, pois ele é composto por outros três DCCs: *Rainfall Maps of São Paulo*, *Solar Radiation Maps of São Paulo* e *Coffee Plant Growth Simulator*.

Os DCCs foram definidos de maneira que fossem armazenados em repositórios na Web. Dessa forma, eles poderiam se beneficiar de recursos da Web, principalmente da Web Semântica, como o uso de metadados e ontologias. Esses recursos ajudariam na recuperação dos DCCs em repositórios na Web [59]. Conforme mencionado anteriormente, um DCC pode ser associado a outro componente (passivo ou de processo). A associação é feita pelo gerenciador de aplicação, que é influenciado pelo contexto da aplicação. O gerenciador de aplicação pode ser especificado de forma que possam ser feitas associações (*binding*) dinâmicas entre DCCs, as quais são transparentes para o usuário final [3].

Um DCC pode ser usado de várias formas, mas três delas são mais comumente utilizadas: i) inserção de componentes de dados e de processos; ii) inserção de conteúdo; iii) construção de aplicação [3]. A inserção de componentes permite inserir DCCs dentro de documentos ou arquivos multimídia, tornando-se parte deles. Essa forma de uso pode ser comparada com a inserção de objetos embutidos em páginas Web (como Java *applets*) [3].

A inserção de conteúdo é semelhante à forma descrita acima. Contudo, neste caso, é inserido apenas o conteúdo no documento ou do arquivo de multimídia [3].

Por fim, a última forma permite que o DCC seja usado como um bloco básico para construção de aplicações, da mesma forma que a composição de componentes de software para construção de uma aplicação em uma rede de componentes interconectados, de acordo com a arquitetura definida [3].

O DCC possui uma interface que possibilita a padronização da forma de acesso a dados, tornando a utilização dos dados mais flexível e aumentando a reutilização de dados e dos componentes de dados.

Para outros exemplos de aplicações que utilizam DCC, recomendam-se os trabalhos [3, 21, 34, 58–61]. Para mais detalhes sobre o uso e implementação de DCCs, recomenda-se [3].

Santacnè especificou uma estrutura que permite encapsular os dados ou software em componentes. Contudo, ele não mostrou como armazenar esses componentes em sistemas de gerenciamento de informação, como banco de dados ou bibliotecas digitais. Isso gerou novas frentes de pesquisa. Dentre elas, destacam-se pesquisas em como associar um componente digital, DCC, com os conceitos de objetos complexos e padrões de bibliotecas digitais. A infraestrutura apresentada na Seção 2.1.3 propõe uma forma de realizar essa tarefa.

### 2.1.3 Uma Infraestrutura para Objetos Complexos de Imagem

Nesta seção, é apresentada a infraestrutura para Objetos Complexos de imagem, baseada em DCC, proposta em [36]. Nessa infraestrutura, objetos complexos que incluem imagens como um dos seus elementos, foram organizados em uma estrutura de componentes. Componentes (DCC) também foram usados para encapsular *softwares* específicos voltados à busca multimodal (baseada em descrições textuais e visuais) de objetos complexos de imagem. Além do encapsulamento dos dados e *software*, a infraestrutura proposta permite a publicação de objetos complexos.

A infraestrutura proposta em [36] tem como objetivo tentar solucionar problemas de especificação e implementação de sistemas de informação que utilizem objetos complexos, com foco em imagens. Apesar do foco da infraestrutura ser o gerenciamento de objetos complexos de imagem, ela foi especificada de forma que possa atender objetos complexos em geral. A proposta do trabalho [36] possui como base dois conceitos: i) a noção de agregação; ii) a interação de conceitos e recursos heterogêneos de objetos complexos com os recursos de bibliotecas digitais, permitindo o gerenciamento e manutenção de grandes volumes de dados heterogêneos. Para definir essa infraestrutura e analisar suas vantagens, o trabalho [36] analisa a formalização, tecnologias de objetos complexos e aplicações relacionadas aos conceitos e uso da infraestrutura para gerenciamento de objetos complexos de imagem. A seguir é detalhada cada uma dessas três análises:

- **Estudos de casos preliminares de aplicações:** o intuito dessa análise é identificar e descrever três aplicações que atendem e trabalham com algumas características importantes para a realização dos estudos de casos da infraestrutura para objetos complexos de imagem. Essas características são: a integração de vários documentos; associação e relacionamento de imagens, metadados, marcas e anotações a um domínio; e a integração de bibliotecas digitais por meio de imagens e metadados.
- **Formalização de objetos complexos de imagem:** nesta etapa, o trabalho [36] apresenta um estudo e análise de objetos complexos e busca de imagens por conteúdo, utilizando o arcabouço 5S como base para descrever objetos complexos de imagem no domínio de bibliotecas digitais. O arcabouço 5S foi estendido para alguns conceitos base de objetos complexos, como agregações e composições.
- **Análise das tecnologias:** são analisadas três tecnologias (DCC, Buckets e OAI-ORE) para o tratamento de objetos complexos, segundo a formalização do arcabouço 5S [36]. Nessa análise, foram considerados apenas quatro aspectos relacionados a objetos complexos: identidade, estrutura, componentes e delimitação.

A infraestrutura proposta em [36] baseia-se no uso de DCCs, para agregar objetos complexos de imagem e encapsular os processos relacionados à busca de imagem por

conteúdo e publicação dos metadados por meio do *Open Archives Initiative* - OAI<sup>2</sup>. É proposta ainda uma arquitetura para bibliotecas digitais baseadas em objetos complexos. Essa arquitetura está dividida em quatro camadas, Figura 2.4:

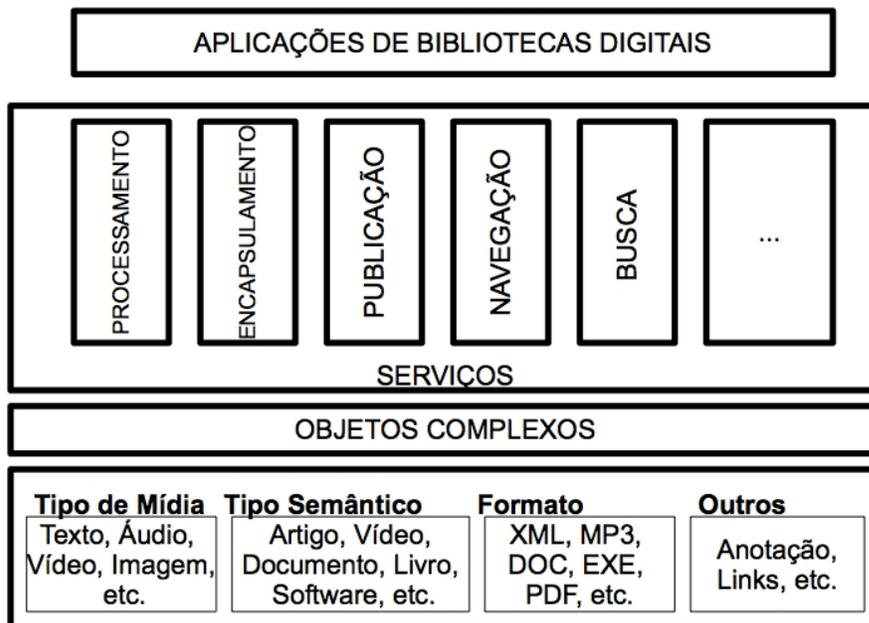


Figura 2.4: Arquitetura da infraestrutura de biblioteca digital baseada em Objetos Complexos - figura retirada de [36].

- Camada de dados: responsável por armazenar os diferentes tipos de dados, tipos de mídia, tipos semânticos e formatos;
- Objetos Complexos: as fontes de dados são agregadas em objetos complexos, disponíveis para a camada de serviços;
- Serviços: são os serviços que acessam e manipulam os dados, que para a infraestrutura do trabalho [36] consistem em serviço de processamento, encapsulamento, publicação, navegação e busca;
- Aplicações de bibliotecas digitais: são as bibliotecas digitais, que podem acessar os serviços, componentes individuais ou objetos complexos como uma única entidade.

As fontes de dados utilizadas pela infraestrutura, discutida nessa seção, são arquivos texto, imagens, softwares e seus respectivos metadados. A infraestrutura foi planejada tendo como base dois tipos de serviços: busca de imagens por conteúdo e publicação de objetos complexos.

<sup>2</sup><http://www.openarchives.org/> (último acesso em junho de 2012).

### Busca de Imagens por Conteúdo

Essa infraestrutura possui três tipos de DCCs (ImageDCC, ImageCODCC, e DescriptorDCC) que são responsáveis por encapsular os dados e oferecer o serviço de busca de imagens. Nesta subseção é apresentado cada um desses DCCs e como realizar a busca de imagens por conteúdo.

A Figura 2.5 mostra uma visão geral da infraestrutura proposta em [36] e como é utilizado e gerencia um objeto complexo de imagem em uma biblioteca digital. Nesta figura, pode-se ver que a forma de acesso da aplicação ao conteúdo e aos recursos se dá a partir do uso de DCCs.

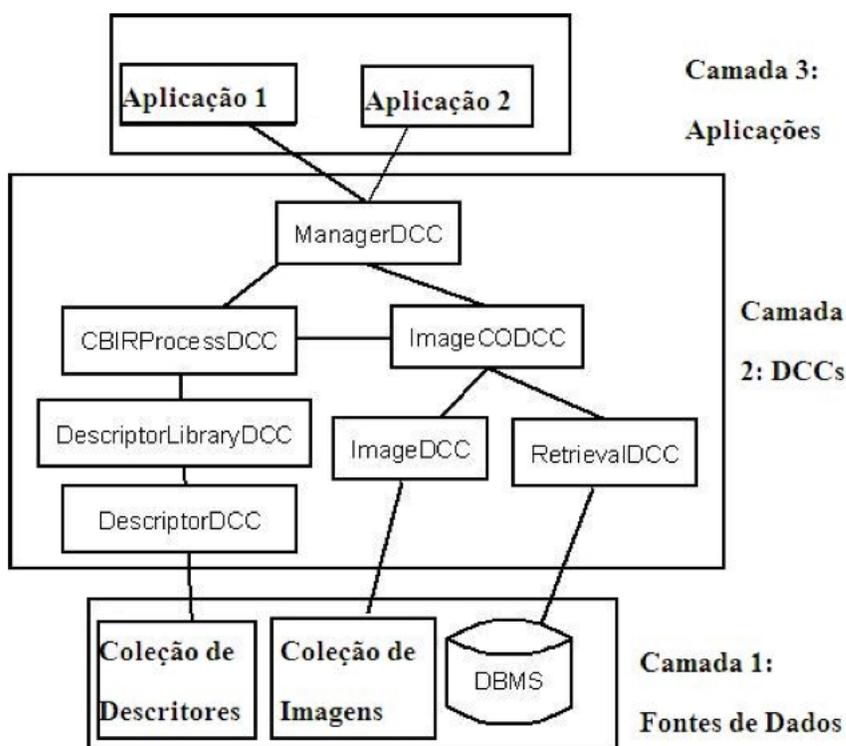


Figura 2.5: Visão geral da infraestrutura de objetos complexos de imagem para bibliotecas digitais - figura retirada de [36].

O ImageDCC é um DCC passivo responsável por encapsular uma imagem, junto com as funções e metadados relacionados. Este DCC possui as seguintes subdivisões:

- **Conteúdo:** a imagem, por exemplo: uma impressão digital de um indivíduo;
- **Estrutura:** um XML, que delimita a imagem;

- **Interface:** a interface do DCC com as suas operações disponíveis, neste caso apenas a operação que retorna a imagem;
- **Metadados:** os metadados que descrevem a imagem.

O ImageCODCC é um DCC passivo que encapsula o DCC de imagem anterior (ImageDCC) e vários elementos que compõem o objeto complexo de imagem. Este DCC é diferente do ImageDCC, pois provê o encapsulamento de vários componentes resultantes do processo de recuperação de imagem por conteúdo (CBIR). Esse DCC possui as seguintes subdivisões:

- **Conteúdo:** o ImageDCC com a imagem de impressão digital, por exemplo, e o vetor de características e as medidas de distâncias dessa imagem em relação a outras de uma coleção;
- **Estrutura:** um XML, com a estrutura das informações da parte do conteúdo desse DCC;
- **Interface:** oferece operações para acessar o vetor de características, as medidas de distância e o ImageDCC;
- **Metadados:** os metadados que descrevem o processo de busca de imagens por conteúdo.

O DescriptorDCC é um DCC de processo responsável por extrair e armazenar características associadas ao conteúdo da imagem, utilizando um descritor. Além disso, disponibiliza-se o método de busca por conteúdo da imagem, baseada nesse descritor. Este DCC possui as seguintes subdivisões:

- **Conteúdo:** é composto por duas funções: a extração, que origina-se no DCC de processo ExtractionDCC, e realiza a extração do vetor de característica da imagem de acordo com o descritor escolhido; e a função de distância que se origina no DCC de processo DistanceDCC, responsável por aplicar a função de medida da distância do descritor escolhido;
- **Estrutura:** um XML, que delimita e conecta o ExtractionDCC e DistanceDCC;
- **Interface:** neste caso, a interface representa as operações dos dois DCCs que compõem este DCC;
- **Metadados:** os metadados representados aqui são a união dos metadados relacionados aos DCCs ExtractionDCC e DistanceDCC.

Existem ainda outros DCCs que completam essa infraestrutura, oferecendo recursos e gerenciando os acessos e manipulações dos DCCs, conforme o exemplo mostrado na Figura 2.5. Os DCCs responsáveis pelo gerenciamento dos processos de alto nível são: *CBIRProcessDCC*, *ImageCODCC* e *ManagerDCC*. Cada DCC que manipula as fontes de dados pode explorar diferentes métodos em sua interface. O DCC de processo *CBIRProcessDCC* é o responsável por oferecer operações de busca de imagens por conteúdo. Este DCC realiza a extração de todos os vetores de características em uma coleção de imagens e calcula a medida de distância entre cada par de imagens. Dessa forma, o *CBIRProcessDCC* permite realizar busca por conteúdo disparada pela aplicação. O *ManagerDCC* aciona o método de busca do *CBIRProcessDCC*, que se comunica com o DCC de imagem (*ImageCODCC*) e com a biblioteca de descritores (*DescriptorLibraryDCC*) e seus descritores (*DescriptorDCC*).

### **Publicação de Objetos Complexos**

Para publicar os objetos complexos o trabalho [36] propõe utilizar o protocolo OAI-PMH<sup>3</sup>, pois ele tem sido amplamente adotado pelos sistemas de informações para realizar intercâmbio de recursos. Além disso, a estrutura do objeto complexo poderia ser mapeada em um arquivo XML da mesma forma que já é feita pela infraestrutura [36], facilitando a sua publicação. O intuito da publicação dos objetos complexos é permitir que outros aplicativos possam encontrá-los e reutilizá-los, possibilitando interoperabilidade, compartilhamento e centralização das informações entre as aplicações. Segundo [36], os objetos disponíveis para publicação são: a imagem, conceitos individuais; o objeto complexo de imagem, agregações dos conceitos individuais; o descritor; e a biblioteca de descritores. Duas requisições específicas foram implementadas em [36] para permitir diferenciar e agilizar a publicação dos metadados relativos a recuperação por conteúdo: *ListFeatureVectors*, que recupera a lista de vetores de características para todas as imagens disponíveis na coleção; e *GetFeatureVector*, que recupera os vetores de características do objeto complexo passado como parâmetro.

### **Conclusão**

A proposta do trabalho [36] é promissora, pois apresenta, tendo como base formalismos bem definidos, uma infraestrutura de bibliotecas digitais com objetos complexos, encapsulados em uma estrutura genérica chamada DCC. O trabalho sugere uma infraestrutura padrão, que permite: reutilização de informações e conteúdos heterogêneos; melhorar a interoperabilidade entre os sistemas, em especial, os de bibliotecas digitais; incluir e associar

---

<sup>3</sup><http://www.openarchives.org/OAI/openarchivesprotocol.html> (último acesso em junho de 2012).

recursos específicos para cada tipo de dados, como busca por conteúdo; e principalmente, definir melhor os dados e informações e o ambiente no qual ele está relacionado. Contudo, o trabalho [36] não contemplou a definição de mecanismos para criação automática de objetos complexos de imagem. Esta dissertação de mestrado preenche esta lacuna.

## 2.2 Anotação Semântica

Esta seção apresenta os conceitos e trabalhos que envolvem anotações, anotações semânticas, ontologia e busca semântica. Esta seção está estruturada da seguinte forma: seção 2.2.1 apresenta a definição dos conceitos de anotação e anotação semântica; seção 2.2.2 apresenta o conceito de ontologia e a sua estrutura; seção 2.2.3 apresenta o conceito de Web Semântica, os seus recursos disponíveis, os principais padrões propostos e exemplifica a utilização de ontologias; seção 2.2.4 apresenta algumas formas de busca semântica.

### 2.2.1 Anotação

Anotação tem sido reconhecida como um dos mais importantes serviços em sistemas de bibliotecas digitais, para promover cooperação entre usuários e integração de recursos de informações heterogêneas [1]. O serviço de anotação permite que os usuários ou grupos de usuários incluam anotações pessoais em recursos de informações gerenciadas, ou seja, em dados armazenados em: banco de dados, bibliotecas digitais e/ou na Web [1]. Além disso, anotações permitem aos usuários inserir, combinar, reaproveitar e ligar informações pessoais com recursos de informações disponíveis em bibliotecas digitais e na Web semântica. Com isso, a anotação contribui com a definição do domínio ou contexto de um conteúdo de interesse [29].

A definição de padrões e estruturas para representação e armazenamento de anotações ajudam a lidar com problemas de ambiguidade, recuperação de informação ou recurso, reaproveitamento e interoperabilidade entre sistemas de bibliotecas digitais [1, 29]. A seguir é apresentada a definição de uma das principais abordagens utilizadas para lidar com anotações: anotação como metadado [1]. Em seguida é apresentada a definição de anotação semântica. Por fim, são apresentados alguns trabalhos que utilizaram o conceito de anotação de imagens e de dados.

#### **Anotação como Metadado**

Anotações podem ser definidas como metadados, ou seja, é um dado que define ou esclarece as informações de outro dado, ou conteúdo, existente no sistema de gerenciamento

de informação. O dado utilizado para anotar serve para esclarecer as propriedades e a semântica do conteúdo anotado [1]. Com isso, a definição de uma anotação segue algumas especificações de estrutura, semântica, sintaxe e também os valores que uma anotação pode assumir. Essas especificações devem considerar que os destinatários dessas anotações podem ser tanto pessoas quanto máquinas, ou seja, a anotação deve estar estruturada e escrita com valores que se encaixem em uma sintaxe de forma que pessoas e máquinas consigam interpretá-las [1, 29].

Um exemplo desse tipo de anotação são os metadados que compõem uma página Web no contexto da Web semântica, seguindo os padrões definidos pelo W3C<sup>4</sup>, como o metadado `<title>` da página Web do ontotext<sup>5</sup> com valor “**Learn More | Ontotext**”, conforme mostrado na Figura 2.6 . Esses metadados fazem parte do conteúdo da página Web, que é legível para uma pessoa, e também permitem que uma máquina consiga entender a semântica dos conteúdos que compõem a página Web, por exemplo, o computador saberia que o texto “**Learn More | Ontotext**” é o título daquela página Web.

### Anotação Semântica

Anotação ou *tagging* são formas de anexar atributos, comentários, descrições entre outras informações a um documento ou parte dele, como um texto [47]. Esses conceitos são utilizados para adicionar informações adicionais ao documento ou texto anotado. *Tagging* é utilizada normalmente para melhorar o desempenho de buscas e encontrar termos mais relevantes e precisos. A anotação semântica vai mais além; ela busca enriquecer dados que estão representados de forma estruturada, complementando e definindo melhor o seu contexto, adicionando diversidade e aprimorando o processo de busca [47]. Essa associação permite encontrar dados que não estão explicitamente associados ao texto ou documento [47].

Na computação, anotação serve para descrever recursos e o que eles fazem, utilizando conceitos formais, como entidades em uma ontologia referenciadas por metadados ou um conjunto de metadados [19]. Uma anotação semântica pode ser representada por um metadado ou um conjunto de metadados, fornecendo uma referência entre o texto e uma entidade da ontologia de interesse. Essa referência é feita por meio de um identificador Web único, como um URI [19]. Dessa forma, anotações semânticas são compostas por ontologias, descritas em linguagens específicas (OWL [45] ou RDF [64]). Estas linguagens permitem eliminar a ambiguidade das linguagens formais e promover um entendimento comum dos conceitos a partir da [29]: reutilização das entidades e ontologias; e da centralização da definição dos conceitos. Além disso, o uso das entidades de uma ontologia

---

<sup>4</sup><http://www.w3.org> (último acesso em junho de 2012).

<sup>5</sup><http://www.ontotext.com/kim/getting-started/learn-more> (último acesso em junho de 2012).

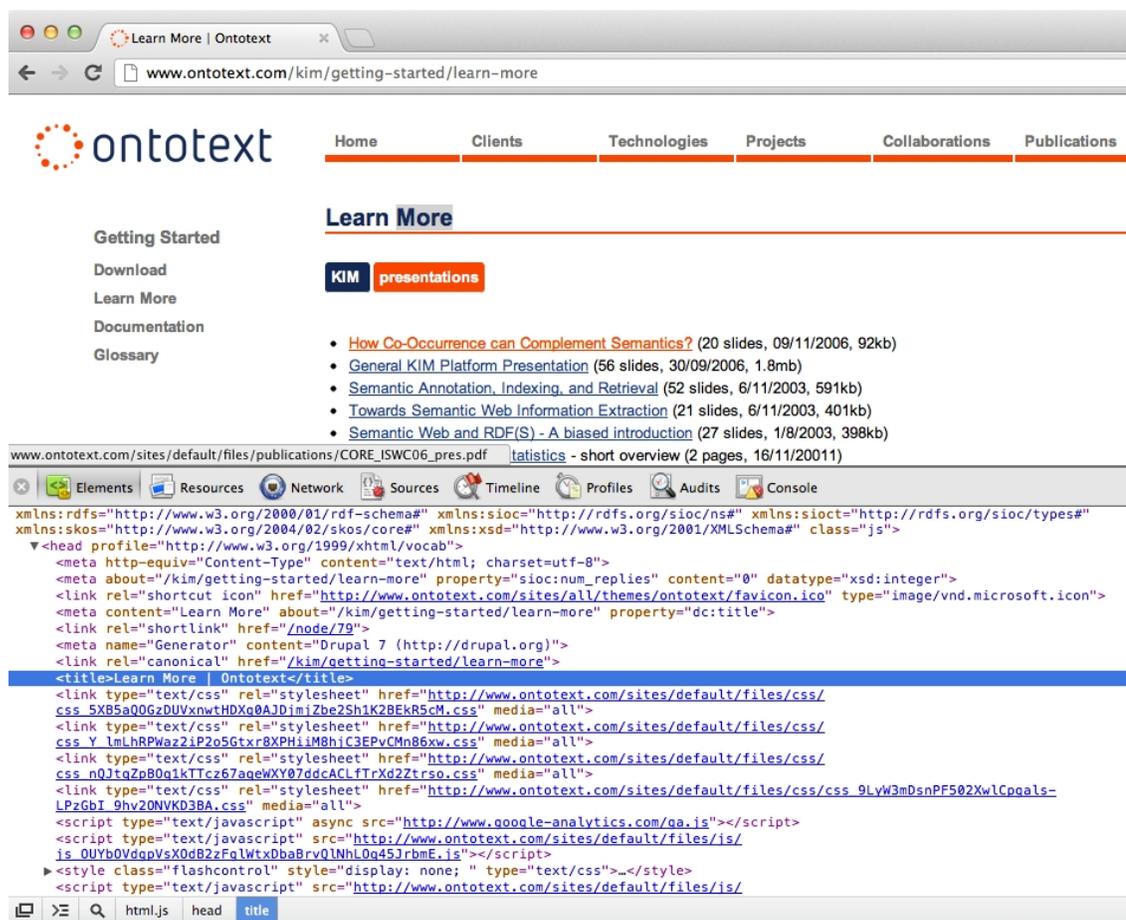


Figura 2.6: Exemplo de anotações e metadados da página Web do ontotext [47].

de domínio como conceitos para anotação é um meio de promover a interoperabilidade entre sistemas.

Com base na Figura 2.6, um exemplo de anotação semântica poderia ser construído trocando o valor do metadado “`http-equiv`” de “`Content-Type`”, que é apenas texto, por uma referência a uma entidade “`Content-Type`”, de uma ontologia: `http://www.w3.org/2007/uwa/context/common.owl#ContentType`. Esta ontologia possui um contexto no qual o texto “`Content-Type`” está inserido, mostrando conceitos associados a ele e definindo a semântica desse texto.

A anotação por meio de entidades das ontologias dificulta a legibilidade por parte do usuário. Com relação a isso, a anotação textual continua sendo mais legível para uma pessoa qualquer do que uma estrutura de anotação baseada em ontologia. Por causa disso, é sugerido em [19] utilizar ambas as anotações: textual (anotação por conteúdo [1]) e semântica. Propõe-se também uma estrutura para representar estas anotações. Em [19],

anotações semânticas combinam conceitos de metadados e ontologias, no qual os campos de metadados são preenchidos com termos de ontologias, que são utilizados para descrever esses campos. A anotação semântica pode ser definida da seguinte forma [19]:

- **Unidade de Anotação:** é uma tripla  $\langle s, m, v \rangle$ , em que  $s$  é o sujeito que está sendo descrito,  $m$  é o nome do metadado e  $v$  o seu valor ou descrição.
- **Anotação:** é um conjunto de uma ou mais unidades de anotação.
- **Unidade de Anotação Semântica:** é uma estrutura semelhante a unidade de anotação mudando apenas o último componente da tripla:  $\langle s, m, o \rangle$ , em que  $o$  é o termo de uma ontologia de domínio.
- **Anotação Semântica:** é o conjunto de uma ou mais unidades de anotação semântica.

Com a estrutura apresentada acima, uma anotação ou anotação semântica é composta de um *schema* e de um *conteúdo*. O *schema* é a sua estrutura definida a partir dos campos dos metadados e o conteúdo e o valor desses campos [19].

### Trabalhos Relacionados

Nesta seção são apresentados dois trabalhos baseados em anotações. O primeiro deles [43] propõe uma estrutura e um *framework* (GeoSpatial) para anotação semântica e manipulação de dados heterogêneos, na área de agricultura utilizando dados geoespaciais. O *framework* proposto em [43] assume duas premissas básicas: os dados geoespaciais podem ser utilizados para acelerar o processo de anotação, aliviando o trabalho do especialista na análise dos dados relacionados as área agrícola; e é possível definir os principais procedimentos de anotação com a ajuda de um especialista em dados geoespaciais, de forma que atendam as demandas na área de agricultura.

O *framework* [43] de anotação semi-automática consiste em um processo de anotação dividido em três etapas:

- **Seleção do *workflow* de anotação:** o *workflow* é o processo de anotação de dados geográficos definido pelo especialista da área. Os *workflows* são criados pelos especialistas conforme a demanda e possibilidade de mais um processo de anotação, que ainda não exista. Essa etapa encarrega-se de selecionar os *workflows* de anotações que serão utilizados na geração das anotações.
- **Execução do *workflow*:** consiste na execução do processo de anotação descrito em cada *workflow*, gerando as unidades de anotações que seguem a estrutura proposta em [19], descrita anteriormente. Cada *workflow* orquestra a execução do seu processo de anotação.

- **Ligação com ontologia:** nesta etapa, cada unidade de anotação é transformada em uma unidade de anotação semântica. A transformação consiste em substituir o texto da unidade de anotação por uma referência a um termo contido em uma ontologia.

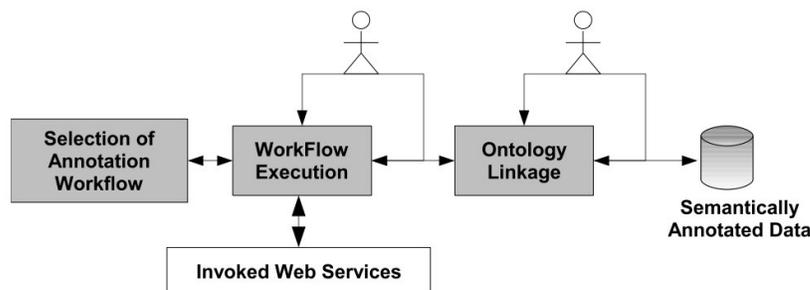


Figura 2.7: Etapas do *Framework* GeoSpatial - figura retirada de [43].

O usuário pode intervir durante o processo de anotação para validar as anotações que foram geradas [43]. A Figura 2.7 mostra as etapas descritas acima.

O outro trabalho [29] propõe um serviço Web para gerenciamento de dados geográficos em sistemas de informação de biodiversidade, chamado GeoNote. Este serviço foi especificado para trabalhar com anotações no formato texto, para dados ou objetos geográficos. O serviço GeoNote busca ajudar com alguns desafios encontrados pelos sistemas de informação em biodiversidade: manipulação de um grande volume de dados, integração de informação de diferentes fontes e em diferentes formatos, manipulação de dados e imagens, e manipulação de referências em informações geoespaciais [29].

A apresentação da especificação e implementação do serviço Web GeoNote pode ser dividida de forma que mostre as suas duas principais contribuições: o modelo de dados baseado em informações sobrepostas e uma API genérica para manipulação de anotação.

O modelo de dados proposto em [29] permite modelar dados de anotações sobrepostas (apenas texto) para objetos geográficos vetoriais. A seguir são apresentadas as entidades que compõem esse modelo, segundo [29]:

- **GeographicObejct:** representa os objetos geográficos com suas coordenadas geográficas, seguindo o modelo da OGC<sup>6</sup>.
- **Excerpt:** é o conteúdo da região marcada, como pontos e linhas, que compõem um objeto geográfico (GeographicObejct) e é referenciada por uma marca (Mark).
- **Mark:** uma marca é criada toda vez que uma anotação sobreposta é criada, relacionando o contexto, informações sobrepostas e metadados.

<sup>6</sup><http://www.opengeospatial.org/> (último acesso em junho de 2012).

- **Context:** é basicamente a informação do conteúdo armazenado, relacionada à região marcada.
- **Element:** são os elementos que compõem um contexto, por exemplo o nível de zoom do mapa.
- **Metadata:** metadados da marca, utilizando apenas cinco elementos do padrão *Dublin Core*<sup>7</sup>.
- **SuperimposedInformation:** é a entidade que representa uma informação sobreposta, que consiste em novas informações relacionadas ou associadas a uma informação base já existente. Este conceito é semelhante ao de metadados, mas permite uma liberdade maior para associar várias informações a uma informação base. A informação sobreposta armazenada pode estar no formato texto, vídeo ou imagem.

A API do serviço Web oferece recursos que permitem realizar as seguintes operações com anotações [28]: inserção, consultas, remoção e atualização. Para cada uma dessas operações existem diversos métodos que aceitam parâmetros diferentes e retornam estruturas diferentes.

## Conclusão

Os trabalhos citados anteriormente possuem focos diferentes para a utilização de anotações. Um deles [29] propõe uma estrutura para associar melhor as informações de uma determinada área de interesse, melhorando a usabilidade e busca dessas informações. Contudo, o processo de anotação ainda é manual e sujeito às falhas humanas.

O outro trabalho [43], por sua vez, propõe uma estrutura padrão para anotação e uma forma semi-automática de gerar anotações. A estrutura da anotação baseia-se no conceito de metadados e triplas RDF [64], para associar o conteúdo ao dado ou conceito anotado, formando um objeto composto. O processo semi-automático de anotação serve para evitar falhas humanas durante a geração da anotação, e não para geração automática de anotações visando um grande volume e facilitar o trabalho de busca de informações.

O processo de anotação proposta nessa dissertação de mestrado utiliza objetos complexos para representar os dados e vincular as anotações e suas semânticas. O foco desse trabalho está no processo automatizado de anotação de imagens e na busca desses tipos de informações.

---

<sup>7</sup><http://dublincore.org/> (último acesso em junho de 2012).

### 2.2.2 Ontologias

Ontologia é uma forma de representação do conhecimento dentro de um domínio. Uma ontologia pode tomar uma variedade de formas, mas necessariamente ela incluirá um vocabulário de termos, e algumas especificações de seus significados. Isso inclui definições e uma indicação de como os conceitos são inter-relacionados, impondo uma estrutura coletiva sobre o domínio e restringindo as possíveis interpretações dos termos. Assim, uma ontologia define um vocabulário específico usado para descrever determinada realidade, mais um conjunto de decisões explícitas, fixando de forma rigorosa o significado pretendido para o vocabulário [31]. Trabalhos como os [15–17, 30] utilizam o conceito de ontologia para representação do conhecimento de forma a facilitar o gerenciamento de dados.

Do ponto de vista computacional, uma ontologia pode ser considerada como um modelo de dados que representa um conjunto de conceitos e conhecimentos, com seus relacionamentos, restritos a um domínio. O conhecimento de domínio pode ser representado em uma ontologia a partir de quatro tipos de componentes: classes, instâncias, propriedades e restrições.

Classes correspondem a conjuntos ou tipos de objetos, que possuem os conceitos ou as categorias de conceitos de domínio [15]. As instâncias são materializações, ocorrências, dos conceitos definidos na classe a qual pertence [15]. Por exemplo: Homem é uma classe e João é uma materialização ou ocorrência da classe Homem.

Para detalhar mais os conceitos dentro de uma classe, são especificadas as suas propriedades. As propriedades são características ou parâmetros das classes ou instâncias, que podem ser usadas para representar relacionamento entre classes [15]. No exemplo da classe Homem podemos ter a propriedade altura, peso, nome, idade, entre outros.

Restrições são algumas condições impostas sobre alguns atributos de modo a definir o domínio dos conceitos representados [15]. Como por exemplo: só é Homem se tiver idade maior que 18.

Ontologias são utilizadas em projetos de domínios específicos como gestão do conhecimento, comércio eletrônico, processamento de linguagens naturais, recuperação da informação na Web, de cunho educacional, e outros.

Berners-Lee propôs a Web-Semântica como uma evolução natural da Web vigente, a fim de viabilizar a manipulação do conteúdo por parte das aplicações com capacidade de interpretar a semântica das informações [7]. O conteúdo da Web, que não passa de informação sem significado para os computadores, pode então ser interpretado por máquinas a partir das ontologias, tornando a recuperação da informação na Web menos ambígua e fornecendo respostas mais precisas às consultas dos usuários.

Para a construção de ontologias é essencial dispor-se de uma linguagem apropriada.

O *World Wide Web Consortium* (W3C<sup>8</sup>) orienta o desenvolvimento, a organização e a padronização de linguagens para promover a interoperabilidade entre aplicações na Web. A linguagem padrão para a codificação de ontologia na Web Semântica é a OWL, segundo o W3C. Como a base para a construção de ontologias é o relacionamento de conceitos e a linguagem mais utilizada para isso é a Resource Description Framework (RDF [64]), assim, a linguagem OWL [45] foi especificada sobre a estrutura da linguagem RDF [30]. Segundo [30] a principal ferramenta para o desenvolvimento de ontologias é o Protégé<sup>9</sup> [27].

Ontologias serão utilizadas neste trabalho para os seguintes propósitos: descrever a interface de um DCC, ou seja, a forma de acesso e de utilização do DCC, realizar busca de termos semelhantes e ser associada ao termo escolhido para anotar a imagem, como uma ontologia de definição do termo.

### 2.2.3 Web Semântica

A Web Semântica é uma extensão da Web atual, no qual os dados possuem definições associadas a eles, permitindo que sejam usados com seus significados bem definidos [7]. Isso aumenta a interpretação dos dados por parte do computador [4]. O objetivo da Web Semântica é aumentar a autonomia dos agentes (módulos, aplicações ou serviços) na Internet de modo que eles possam estabelecer uma comunicação ou negociação entre parceiros [7], no qual o computador consiga processar e identificar os conceitos associados aos dados ou informações trafegados na Web Semântica. Para que os agentes possuam essa autonomia são necessários: protocolo padrão para interoperabilidade, permitindo que agentes desenvolvidos em tecnologias diferentes e por empresas diferentes possam se comunicar; e mecanismos que possibilitem a obtenção de informações, permitindo que o agente possa decidir se iniciará ou não o processo de negociação com algum parceiro [30]. As definições e padrões sobre Web Semântica são coordenadas pelo W3C (*World Wide Web Consortium*)<sup>10</sup>.

Um exemplo de informação que pode ser representada na Web Semântica é:

- Beatles é uma banda popular de Liverpool.
- John Lennon foi um membro dos Beatles.
- “Hey Jude” foi gravada pelos Beatles.

Essas informações podem ser representadas na Web Semântica de forma que o computador consiga entender que Beatles é o nome de uma banda de música que teve origem

---

<sup>8</sup><http://www.w3.org> (último acesso em junho de 2012).

<sup>9</sup><http://protege.stanford.edu> (último acesso em junho de 2012).

<sup>10</sup><http://www.w3.org> (último acesso em junho de 2012).

em Liverpool, ou seja, essas informações não são apenas texto, elas possuem informações semânticas (conceitos e definições) e sintáticas associadas ao texto. Mais detalhes sobre os conceitos da Web Semântica podem ser encontrados no *w3school.com*<sup>11</sup>. Com a evolução da Web para a Web Semântica, tem sido possível propiciar um ambiente bastante adequado à publicação de dados associados a descrições semânticas a partir de descrições/metadados [48].

O ponto chave da Web Semântica é a interoperabilidade, segundo a qual os dados, metadados e agentes devem seguir um padrão de formatos, conceitos e tecnologias, permitindo o intercâmbio e processamento de dados e informações. A Figura 2.8 ilustra as camadas de padrões e tecnologias da Web Semântica, que serão descritas a seguir com base nas referências [4, 25, 30, 48].

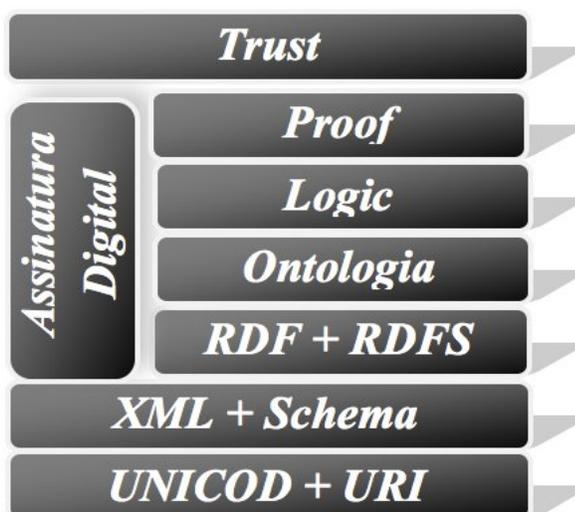


Figura 2.8: Camadas de padrões e tecnologias da Web Semântica.

A primeira camada inferior UNICOD + URI (*Uniform Resource Identifier*) contém a parte de descrição dos dados, codificados em UNICOD, e são usadas para compatibilidade no processamento de textos. A outra parte envolve o conceito de URI que é utilizado para identificar univocamente um recurso abstrato ou físico. A camada de XML consiste em um modelo de dados base para a representação sintática dos dados em XML [8]. Este modelo é estruturado em uma hierarquia de árvore, baseada em marcas (*tag*). Esta camada inclui conceitos de *Namspaces* e definições de esquemas (XMLSchema [23]), para representação de tipos de dados de forma a completar o modelo.

A camada de RDF (*Resource Description Framework*) aborda requisitos semânticos permitindo declarar associações de recursos com suas propriedades, formando a base para

<sup>11</sup>[http://www.w3schools.com/web/web\\_semantic.asp](http://www.w3schools.com/web/web_semantic.asp) (último acesso em junho de 2012).

processar metadados e expressar relacionamentos. O RDFS (*RDF Schema*) possibilita a definição de vocabulários que podem ser usados para associar tipos de recursos a propriedades, podendo ser referenciados a partir de URIs.

A camada de Ontologia enriquece o vocabulário e possibilita sua evolução a partir de extensões no repositório de conceitos e de relacionamentos entre eles. Essa camada faz uso de uma linguagem não ambígua para descrever de modo formal o significado e a terminologia usados em documentos na Web Semântica, sendo OWL (*Web Ontology Language*) a linguagem padrão.

A camada de Assinatura Digital assegura a validade da informação a partir do uso de certificados, que garantem sua origem. As camadas *Logic*, *Proof* e *Trust* ainda estão em desenvolvimento. A primeira expressa o conhecimento por regras, enquanto a segunda usa essas regras para inferir conhecimento. A camada *Trust* provê mecanismos para determinação do grau de confiança do conhecimento inferido.

## 2.2.4 Buscas Semânticas

Nesta seção, são apresentados alguns trabalhos que envolvem conceitos de busca de ontologias.

*Ranking* é um importante conceito para impedir conjuntos de resultados vazios, sobrecarregados ou desordenados [62]. Normalmente, são empregados valores de relevância para fazer o *ranking*. Estes valores restringem sua utilidade quando trabalhamos com diversos fatores que influenciam o resultado da consulta. É proposta uma forma flexível de expressar as relevâncias, de forma que atendam aos diversos fatores, a partir da noção de preferências. O usuário pode adicionar nas consultas para geração do *ranking* os tipos de respostas que ele prefere, deixando a consulta mais flexível. Desta forma, os autores do trabalho [62] propuseram e implementaram uma extensão do SPARQL (linguagem de consulta de ontologia no formato OWL) que suporta a expressão de preferências, no qual é incluída uma sintaxe formal e a semântica de expressões de preferência para SPARQL.

André Santanchè propõe em seu trabalho [59] um mecanismo para descrever e descobrir DCCs. Os mecanismos são validados na implementação de DCCs para um planejamento agro-ambiental, simulação de uma plantação de café. Santanchè especificou os metadados e interfaces dos DCCs em OWL/OWL-S, que podem ser usados na indexação e busca dos componentes (DCC). A ideia é utilizar um modelo de busca e *ranking* de ontologia ou metadados para localizar os DCCs. Além desse trabalho, Santanchè desenvolveu outro trabalho [61], que também utiliza DCC para trabalhar com arquivos de multimídia. Neste trabalho ele apresenta um mecanismo para realizar busca de DCCs em um repositório na Web usando-se o padrão UDDI (*Universal Description, Discovery and Integration*) [10].

Jaudete Daltio apresenta na sua dissertação de mestrado [15] um novo tipo de Serviço

Web, chamado Aondê, que suporta um amplo espectro de operações para armazenamento, gerenciamento, busca, *ranking*, análise e integração de ontologias. Esse serviço e suas operações foram testados com ontologias volumosas e estudos de casos reais na área de biodiversidade. Apresenta-se um estudo sobre projeto, gerenciamento e compartilhamento de ontologias, no qual envolvem pesquisas de algumas técnicas e ferramentas para ontologias, proposta na literatura. Para mais detalhes, ver seção 2.3.

Outros exemplos de busca e/ou *ranking* de ontologias encontrados na literatura são [2, 35, 53, 65], deixando nítido a enorme necessidade de buscas mais eficazes, e como o conceito de ontologia tem sido utilizado para auxiliar esta tarefa.

## 2.3 Aondê

Aondê é um serviço Web proposto em [15] que tem como objetivo disponibilizar funcionalidades para manipulação e armazenamento de diversas ontologias. A motivação para o desenvolvimento do Aondê consistiu em prover mais semântica associada à informação armazenada ou manipulada, permitindo lidar melhor com problemas como o de heterogeneidade de dados. Este serviço apoia um amplo espectro de funcionalidades, que permitem realizar operações de armazenamento, gerenciamento, busca, *ranking*, análise, comparação e integração de ontologias.

Embora seja de caráter geral, o uso do Aondê foi demonstrado no contexto de gerenciamento de dados de biodiversidade. A sua arquitetura, seus repositórios e boa parte de suas funcionalidades foram planejadas e desenvolvidas de forma genérica, podendo ser utilizadas, ou se for o caso adaptadas para o uso em outros domínios. O Aondê apresenta contribuições direcionadas à solução de problemas de heterogeneidade semântica na Web, apresentando um Serviço Web de Ontologias caracterizado por: (i) utilizar repositórios distribuídos para armazenar e gerenciar ontologias e seus metadados; (ii) acessar esses repositórios por meio de Serviços Web; e (iii) operações que permitem a manipulação integrada de conjuntos de ontologias. [15]. Portanto, este serviço disponibiliza meios para que aplicações possam enriquecer sua semântica e interoperabilidade a partir do reúso, consumo e extensão deste serviço e das ontologias publicadas na Web.

### 2.3.1 Arquitetura e Conceitos do Aondê

A arquitetura do serviço Aondê é composta por duas camadas principais: Repositórios e Operações, conforme ilustrado na Figura 2.9. A camada de Repositórios consiste em uma estrutura distribuída de repositórios de ontologias, cuja forma de acesso é via Serviços Web. Os repositórios que compõem essa camada são divididos em dois tipos:

- Repositórios Semânticos: estes são construídos e gerenciados pelos serviços de ontologias. Nesses repositórios são armazenadas as ontologias e metadados utilizados no Aondê.
- Repositórios Externos de Ontologias: são os repositórios que não estão relacionados diretamente aos serviços e seus dados são manipulados via Serviços Web.

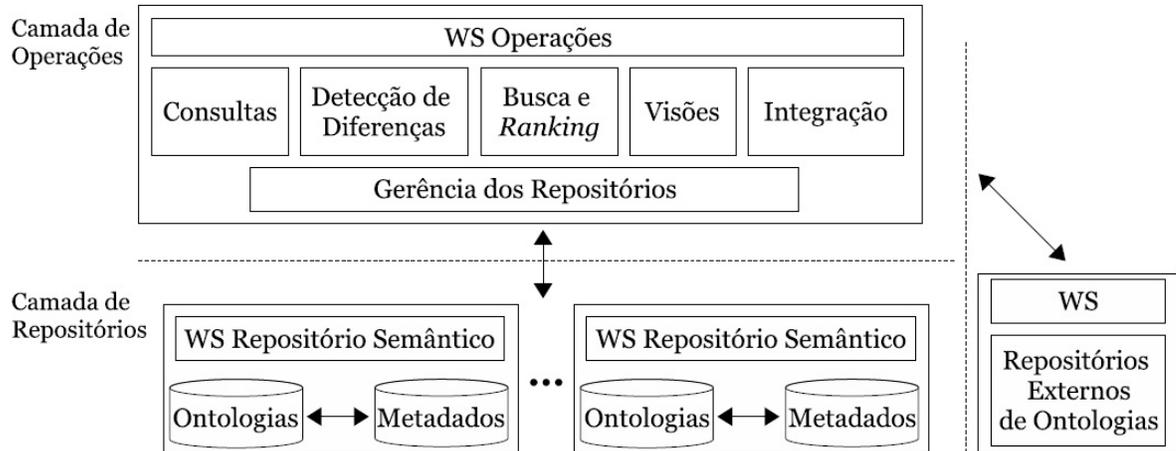


Figura 2.9: Arquitetura do Serviço Web Aondê (figura retirada de [15]).

A camada de repositório é acessada e gerenciada a partir do módulo Gerência dos Repositórios da Camada de Operações. Esta camada é responsável por prover, a partir de serviços Web, as operações avançadas do Aondê. As operações estão divididas em seis módulos: Gerência dos Repositório, Consultas, Detecção de Diferenças, Busca e *Ranking*, Visões e Integração. Em seguida serão detalhados cada um dos módulos.

### 2.3.2 Modulo de Gerência dos Repositórios

Este módulo é responsável pelo gerenciamento, em memória, das ontologias requisitadas pelo serviço, ou seja, por intermediar as operações entre os repositórios semânticos e os outros módulos do serviço. As operações intermediadas são: inserção e eliminação de ontologias e metadados, atualização de ontologias e substituição de metadados. Os repositórios semânticos (com as ontologias e metadados) podem estar distribuídos na Web, permitindo duplicações de ontologias e metadados. Isso dificulta o gerenciamento dos repositório sendo necessário um identificador único para uma ontologia e o seu metadado. Por isso, a identificação única de uma ontologia requer o par:  $\langle idOnto, URLRep \rangle$ , em que  $idOnto$  representa o identificador da ontologia no repositório endereçado por  $URLRep$ . A estrutura de metadados é sempre armazenada no mesmo repositório, sendo

recuperada a partir do identificador da ontologia correspondente. Este módulo gerencia também um catálogo de Repositórios Semânticos, sendo atualizado toda vez que um Repositório Semântico é acessado pela primeira vez (conceito semelhante ao de catálogo de um banco de dados). Com isso, este catálogo facilita operações de busca por repositórios distribuídos.

### 2.3.3 Módulo de Consultas

Este módulo permite a extração de informações armazenadas em uma ontologia. A extração de informação em uma ontologia é realizada a partir de uma consulta, semelhante a uma consulta de banco de dados. Para representar essa consulta, existem diversas linguagens, dentre elas a SPARQL [54]. Este módulo realiza consultas conceituais em ontologias conforme a especificação da linguagem escolhida. A sintaxe para invocação desse módulo é:

*Consulta(idOnto, linguagem, strConsulta, inferencia, formato)*

- ***idOnto***: representa o identificador da ontologia no qual será realizada a consulta;
- ***linguagem***: corresponde à linguagem que deseja utilizar para realizar a consulta (informada em *strConsulta*);
- ***strConsulta***: contém a consulta que deseja realizar, escrita na linguagem informada (*linguagem*);
- ***inferencia***: é uma *flag* que informa se deseja aplicar o método de inferência na ontologia antes de realizar a consulta (o seu valor é *true* ou *false*). Esta *flag* foi inserida para tornar opcional a realização da inferência, pois esta operação é muito custosa;
- ***formato***: especifica o formato de saída do resultado, podendo ser textual (triplas RDF ou partes dessa tripla) ou estruturado em um arquivo XML [6]. O formato textual pode representar apenas resultados cujos elementos possam ser designados por URI's.

### 2.3.4 Módulo de Busca e *Ranking*

Este módulo realiza a busca em um conjunto de repositórios por ontologias que possuam o termo informado. A busca consiste em encontrar exatamente ou parcialmente a palavra-chave informada entre os nomes de classes e instâncias das ontologias. Para realizar a comparação do termo na ontologia, são aplicadas as seguintes métricas de comparação descrita no trabalho [2]:

- **Medida de Combinação de Classes:** essa métrica consiste em comparar o termo com os nomes das classes das ontologias. No caso de passar um conjunto de termos, quanto mais termos forem encontrados como classes de uma ontologia, maior será a pontuação dessa ontologia, indicando maior relevância para o resultado.
- **Medida de Densidade:** o objetivo dessa métrica consiste em verificar se a ontologia possui muitos conceitos associados diretamente ao termo desejado, ou seja, uma ontologia que possui um foco maior para o termo desejado. As medidas são calculadas com base em quatro estruturas de uma ontologia: superclasses, subclasses, relações e irmãos. Para cada uma dessas estruturas são atribuídos pesos de relevância, que podem ser ajustados entre uma consulta e outra.
- **Medida de Similaridade Semântica:** essa medida consiste em determinar quanto dois termos estão relacionados. Para isso, o algoritmo leva em consideração que a representação da ontologia é em um grafo, podendo utilizar como medida o caminho mais curto entre dois nós de um grafo (neste caso dois termos de uma ontologia) [15].
- **Medida de Intermediação:** essa medida é o cálculo do número de caminhos mais curtos entre o termo e cada nó do grafo (cada elemento da ontologias). Esse número permite inferir se o termo é um conceito central da ontologia, ou o quão próximo ele está de ser um conceito central.

Para cada uma das métricas apresentadas acima é atribuído um peso, que representa a sua relevância. Esses pesos são importantes para obter uma melhor combinação dos resultados. Os pesos podem ser alterados de uma busca para outra e a soma deles devem resultar em 1. O Aondê disponibiliza duas opções de busca de ontologias: Busca exata e Busca com *Ranking*.

A Busca exata é utilizada quando se deseja encontrar apenas uma ontologia, para que possa obter mais informações relacionadas ao termo informado. Nessa busca pode-se informar quais o tipos de informações que deseja encontrar. Esses tipos são na verdade os nomes de classes e relacionamentos da ontologia. O serviço espera como parâmetros um conjunto com os tipos, ou diretivas como é chamado no trabalho [15], das informações que deseja encontrar. A sintaxe para a chamada desse serviço de busca é:

$$Busca(\textit{termo}, \{\textit{diretiva}\}, \{\textit{URLRep}\}, \textit{URLRepDestino})$$

A chamada do serviço de busca é composta: pelo *termo* que deseja buscar; por um conjunto de *diretivas* que irão compor o resultado da busca; por um conjunto URL's ( $\{\textit{URLRep}\}$ ) dos repositórios semânticos no qual serão realizadas as buscas; e finalmente,

por uma URL (*URLRepDestino*) de um repositório semântico que irá armazenar a ontologia resultante da busca.

Com relação a Busca e *Ranking* a sintaxe do serviço para esse módulo é:

$$BuscaRank(\{termo\}, \{peso\}, \{URLRep\}, URLRepDestino)$$

O serviço de busca com *ranking* recebe como parâmetro um conjunto de termos (*{termo}*), que para realizar a busca no conjunto de repositórios semânticos (*{URLRep}*) informado. Nos critérios de busca e *ranking* são utilizadas as mesmas medidas da ferramenta AkTiveRank [2], no qual cada uma está associada a um peso de relevância, conforme citado anteriormente. Os pesos são passados como parâmetro dentro do conjunto (*{peso}*). Por fim o resultado da busca com *ranking* da ontologia é armazenada no repositório informado em *URLRepDestino*. Em ambas as formas de busca, o repositório *URLRepDestino* também é encarregado de armazenar a consulta realizada associada a ontologia resultante da busca. Isso servirá para auxiliar buscas futuras.

### 2.3.5 Módulo de Visões

O módulo de extração de uma visão possui um funcionamento muito semelhante ao módulo de Busca, que faz uma busca exata a partir de um termo específico. Contudo, o módulo de visão aceita qualquer termo, e o resultado consiste em uma subontologia da ontologia original, ou seja, uma nova ontologia contendo apenas as classes e os elementos relacionados ao conceito informado por meio de diretivas especificadas. As diretivas são o que define qual é o tipo de associação entre duas classes.

A sintaxe para a chamada desse módulo do serviço é:

$$Visao(idOnto, conceito, \{diretivas\}, URLRepDestino)$$

O parâmetro *idOnto* representa o identificador da ontologia, na qual será procurado pelo *conceito*. O conceito é o nome da classe que representa o conceito central da visão. O conjunto *{diretivas}* é utilizado para especificar quais elementos relacionados ao conceito central devem ser adicionados na visão. Assim que construída, a visão é armazenada no repositório semântico *URLRepDestino* informado.

### 2.3.6 Módulo de Integração

Este módulo consiste em combinar e integrar ontologias, buscando melhorar a definição dos termos e ampliar o domínio da ontologia. Para isso, o Aondê começa com o alinhamento das ontologias, com o qual é construída uma nova ontologia que preserva as ontologias originais e materializa os mapeamentos como relacionamentos de equivalência

entre os conceitos [15]. Normalmente, as técnicas de integração de ontologias focam em suas classes e instâncias, esquecendo que um mesmo conceito pode ser representado de forma diferente entre duas ontologias: classe em uma e instância em outra. Para amenizar esse problema o alinhamento utiliza as seguintes técnicas de combinação de estruturas e conceitos: análise taxonômica, técnicas baseadas em *string*, restrições de tipo e técnicas de grafo (esta é utilizada para comparação em nível de estrutura). Para mais detalhes ver o trabalho [15].

A sintaxe para a chamada do módulo de integração do serviço é:

$$Integracao(idOntoA, idOntoB, \alpha, \beta, confiabilidadeMin, URLRepDestino)$$

Os parâmetros *idOntoA* e *idOntoB* são identificadores das ontologias que serão alinhadas, e a ontologia resultante será armazenada no repositório informado em *URLRepDestino*. O campo *confiabilidadeMin* corresponde à confiabilidade mínima que um mapeamento identificado durante o alinhamento deve ter para ser materializado na ontologia alinhada. A confiabilidade é calculada da seguinte forma:

$$Confiabilidade = \alpha \times \max\{similarElemento\} + \beta \times (similarEstrutura)$$

A expressão  $\{similarElementos\}$  representa o conjunto de valores obtidos utilizando-se técnicas de similaridade de elementos: similaridade de *string* [11] e dicionário de sinônimos. Da mesma forma que o campo *similarEstrutura* representa o valor da similaridade entre as estruturas desses elementos, comparando as suas: propriedades, axiomas, superclasses e subclasses. O cálculo da confiabilidade envolve os outros dois parâmetros:  $\alpha$  e  $\beta$ , que são pesos atribuídos aos resultados das comparações de similaridade de elementos e de estrutura, respectivamente. Para mais detalhes sobre as técnicas de similaridades empregadas, recomenda-se o trabalho [15].

### 2.3.7 Módulo de Detecção de Diferenças

O módulo de detecção de diferenças realiza comparação entre duas ontologias, buscando identificar diferenças em suas estruturas (classes e propriedades) e conteúdos (instâncias). O resultado da comparação é retornado em um arquivo XML, com as diferenças categorizadas em: modificações, inserções e exclusões. A sintaxe utilizada para a chamada desse módulo do serviço é:

$$Diferenca(idOntoA, idOntoB, \alpha, \beta, confiabilidadeMin)$$

O *idOntoA* e o *idOntoB* são os identificadores das ontologias utilizadas na detecção de diferenças. O campo *confiabilidadeMin* corresponde ao mínimo de confiabilidade que um mapeamento precisa ter para ser considerado uma modificação entre as ontologias comparadas. O funcionamento da parte de comparação deste método é semelhante ao do método de integração, incluindo a definição dos pesos  $\alpha$  e  $\beta$ .

## 2.4 OntoSAIA: Uma Ontologia Baseada em Busca de Imagem e Uma Ferramenta de Anotação Semi-Automática

O OntoSAIA, desenvolvido no Instituto de Computação (IC) da Unicamp [26], consiste em uma ferramenta que auxilia no processo de armazenamento, indexação e recuperação de imagens. O foco principal do OntoSAIA é permitir anotação de imagens, a partir de recomendações (termos sugeridos pelo sistema).

### 2.4.1 Arquitetura

A arquitetura do OntoSAIA está dividida em três camadas: Interface, Núcleo e Repositório de Dados, como mostra a Figura 2.10.

**Interface:** É a camada responsável por receber os parâmetros (imagens e/ou anotações) e encaminhar ao *Gerenciador OntoSAIA* no Núcleo. Esta camada também é responsável por exibir os resultados das consultas.

**Núcleo OntoSAIA:** Esta camada é composta de quatro módulos:

- O módulo Gerenciador OntoSAIA é responsável por repassar os parâmetros para os módulos de *Ontologia* e de busca de imagem por conteúdo (CBIR), assim como requisitar as consultas ao módulo de *Banco de Dados*. Dessa forma, ele funciona como um gerenciador entre os outro módulo, recebendo os dados da interface.
- O módulo de Ontologia é responsável por gerenciar as ontologias, realizar consultas por termos e calcular a similaridade entre termos [26]. Contudo, o cálculo de similaridades entre ontologias é muito sensível ao domínio, podendo apresentar relevâncias diferente para um relacionamento. Por causa disso, foi criado um submódulo *Ontology\_Algorithm* para calcular o grau de similaridade entre dois termos para um mesmo relacionamento de uma ontologia.
- O módulo CBISC consiste basicamente do *Content-Based Image Search Component* (CBISC), que realiza busca de imagens por conteúdo, baseada em descritores de imagens [14].

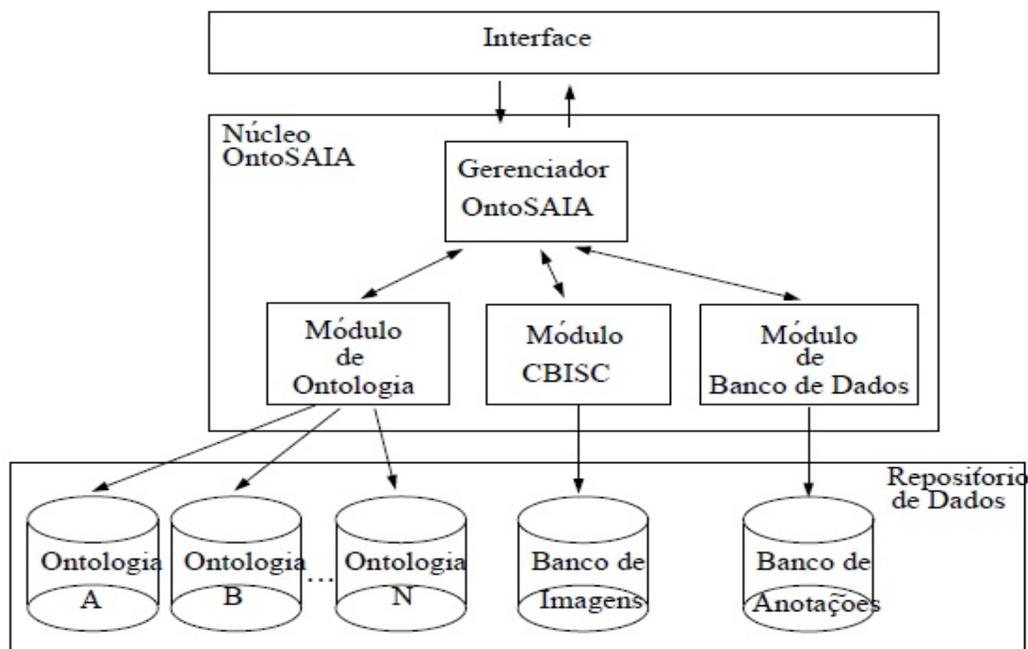


Figura 2.10: Arquitetura de três camadas do OntoSAIA (figura retirada de [26]).

- O módulo de Banco de Dados é responsável pelas operações de inserção e busca de ontologias e imagens nos repositórios de dados.

**Repositório de Dados:** É composto pelas ontologias, imagens e seus vetores de características e pelo banco de anotações. As anotações são gerenciadas pelo módulo Gerenciador OntoSAIA a partir do módulo de Banco de Dados.

## 2.4.2 Funcionamento e Aspectos de Implementação

Com o OntoSAIA é possível incluir um conjunto de palavras-chaves e/ou de conceitos de ontologias como anotação de uma imagem. Esta imagem é armazenada com esses termos, além de vetores com as características extraídas da imagem. As formas de recuperação de imagem da ferramenta existem apenas para auxiliar o usuário na escolha de uma anotação para a sua imagem.

Para utilizar a recomendação de palavras e conceitos, deve-se escolher um dos quatro métodos a seguir:

- Método 0: Este método não utiliza a opção de recomendação.
- Método 1: Este método usa a recomendação baseada em similaridade de termo, que utiliza ontologias para identificar termos similares. Um exemplo é o uso da ontologia

WordNet (banco de dados léxico, semelhante a uma ontologia de dicionário da língua inglesa) para retornar uma lista de termos ordenada pelo peso da similaridade.

- Método 2: Este método é semelhante ao anterior, alterando apenas a medida do peso, que acrescenta ao peso da similaridade a frequência com que termos aparecem na base de dados. Com isso a ordenação é afetada pela frequência.
- Método 3: Este método faz recomendações de acordo com o algoritmo de similaridade de imagens, baseado nos vetores de características (por exemplo, a partir de histograma de cores em RGB). Uma vez identificadas as imagens semelhantes as palavras recomendadas são as que foram usadas para anotar estas imagens.

Para o cálculo da similaridade entre termos, foi proposto no OntoSAIA o algoritmo *Ontology\_Algorithm\_Ternary*. O algoritmo leva em conta três relacionamentos: sinônimos, hiperônimos e hipônimos. Atribui-se o valor 1,0 para sinônimos e desconta-se 0,10 para cada grau de indireção com um relacionamento de hiperônimo ou hipônimo. Estas consultas são realizadas separadamente em uma ontologia, obtendo um arquivo com o resultado de cada uma das três consultas: sinônimo, hiperônimo e hipônimo. A consolidação desses resultados e, quando for o caso, da frequência das anotações é realizada a partir de um algoritmo de combinação. O algoritmo está implementado na classe *Merge\_SummativeUnion* que gera o peso final de um termo ou imagem somando o peso de cada instância [26].

Outra forma de se obter sugestões para as anotações consiste no uso da busca por imagens similares anotadas previamente. Para encontrar essas imagens, o OntoSAIA oferece as seguintes opções de filtros: por conteúdo da imagem, por termos informados pelo usuário ou clicando nos termos sugeridos pela busca em ontologias.

## 2.5 Fusão de Dados

O processo de anotação automática de imagens proposta nessa dissertação de mestrado baseia-se em algumas formas de busca para obter sugestões de anotações. Cada um dos mecanismos de busca retorna uma lista com os termos sugeridos, sendo necessária uma forma de combinar e eleger os termos mais relevantes para anotar as imagens. Essas combinações são realizadas por meio de técnicas de Fusão de Dados.

Fusão de dados (*rank aggregation*) consiste em combinar várias listas ordenadas de entidades, de forma a obter uma única lista ordenada, contendo todos os elementos das listas [42, 56]. Esta lista não possui entidades repetidas e a ordem de relevância das entidades deve levar em consideração as diferentes ordens dos elementos em cada uma das listas envolvidas. Algumas técnicas de fusão de dados consideram um domínio maior do que apenas o conjunto de listas ordenadas utilizadas para aquela fusão de dados, ou

seja, são técnicas que dependem de um treinamento da função com base em exemplos bem sucedidos de fusão de dados.

Existem diversos métodos de fusão de dados na literatura [5, 22, 24, 42, 55, 56]. Eles podem ser classificados em três categorias: fusão baseada em pesos, fusão baseada na ordem dos elementos em uma lista e fusão com treinamento por dados ou não [56]. A técnica de treinamento por dados também é conhecida como aprendizado supervisionado ou não supervisionado. Para mais detalhes sobre as classificações dos métodos de fusão de dados e alguns exemplos ver [22, 42, 56]. A técnica de aprendizado para fusão tem sido bastante mencionado na literatura, como o trabalho [24]. Para entender um pouco mais sobre fusão de dados é apresentado a seguir um dos métodos mais comuns e referenciados na literatura, o Borda Count [22].

**Borda Count:** O método Borda considera a posição do elemento em uma lista ordenada para obter o peso daquele elemento, naquela lista. O peso é obtido contando o número de elementos que estão abaixo ou depois dele. As listas podem ser classificadas em dois tipos: listas completas e listas parciais. As listas completas são um conjunto de listas ordenadas, em que cada uma das listas possuem os mesmos elementos, podendo estar ordenados diferentemente. Portanto, se existem  $k$  listas ordenadas, existirão  $k$  pesos para cada um dos elementos da lista. Por outro lado, as listas parciais podem ser de tamanhos variados e conterem elementos diferentes. Um exemplo de método que considera listas parciais pode ser visto em [55].

Considerando listas completas, o peso calculado pelo método Borda (chamado de peso Borda) para um elemento é a soma dos pesos daquele elemento em cada uma das listas ordenadas. Após obter os pesos Borda para cada um dos elementos da lista, é possível obter uma ordenação final dos elementos da lista por ordem decrescente do peso Borda desses elementos. A Figura 2.11 mostra um exemplo de como funciona o algoritmo Borda Count [22]. O método Borda permite que outras funções possam ser utilizadas no lugar da soma dos pesos das listas para se obter o peso Borda final, como por exemplo, a média aritmética ou geométrica dos pesos [22].

Na Figura 2.11, há três listas ordenadas: **L**, **J** e **K**; na qual é aplicado o método Borda Count obtendo o peso Borda de cada um dos elementos das três listas. Isso é mostrado nas listas **Pesos** e **Resultado** na Figura 2.11. A lista **Resultado** contém o resultado da fusão das três listas.

## 2.6 Considerações do Capítulo

Neste capítulo, foram apresentados conceitos, estruturas e trabalhos que são utilizados na definição e desenvolvimento da proposta dessa dissertação de mestrado: arcabouço para

	Índice	1	2	3	4	5
Listas ordenadas dos elementos	{	L = {	a,	b,	c,	d, e}
		J = {	e,	b,	a,	c, d}
		k = {	e,	a,	b,	d, c}
Resultado = {		a,	b,	e,	c,	d}
Pesos = {		9,	8,	8,	3,	2}

Figura 2.11: Exemplo de fusão de dados com o método Borda Count.

anotação anotação semântica automática de DCCs de imagens. Para realizar este trabalho foi preciso estudar o conceito de objetos complexos e alguns trabalhos relacionados, permitindo escolher ou identificar uma estrutura semântica para armazenar os dados e seus contextos. Essa estrutura também deve facilitar os mecanismos de busca.

Como o intuito é trabalhar com processos de anotações de imagem, uma das formas de atribuir contexto a imagem e melhora mecanismos de busca de imagem consiste no uso de anotações: por metadados ou por estruturas semânticas. Dessa forma, foi realizado estudos dos conceitos e trabalhos relacionados a: Anotação, Ontologia, Web Semântica e Busca Semântica.

Ontologia é bastante utilizada neste trabalho. O objetivo é melhorar cada parte do processo de anotação automática de imagem e o processo como um todo. Para entender melhor o uso de ontologia e como manipulá-la, foi realizado um estudo do serviço Web Aondê, com foco em como reaproveitar as suas funcionalidades e infraestrutura.

Com relação ao armazenamento e manipulação de imagens foi preciso estudar técnicas e ferramentas existentes para definição do processo e do arcabouço de anotação de imagem. Com base nesse estudo foi definida a ferramenta OntoSAIA como a base da arquitetura do arcabouço de anotação.

Por fim, a última seção apresenta um dos conceitos fundamentais no processo de anotação automática de imagens: fusão de dados. Técnicas de fusão são usadas para combinar os resultados de diferentes modalidades, melhorando a relevância das anotações.

# Capítulo 3

## Especificação do Arcabouço de Anotação

O objetivo desta dissertação de mestrado é especificar e implementar um arcabouço para anotação semântica automática de DCCs de imagens, visando o seu uso na construção de objetos complexos. Para isso foi especificado um processo de anotação semântica automática de DCC de imagem, que é a principal contribuição deste trabalho. Com base no processo de anotação foi proposto o arcabouço para anotação de componentes de imagem, que implementa esse processo conforme mostra a Figura 3.1.

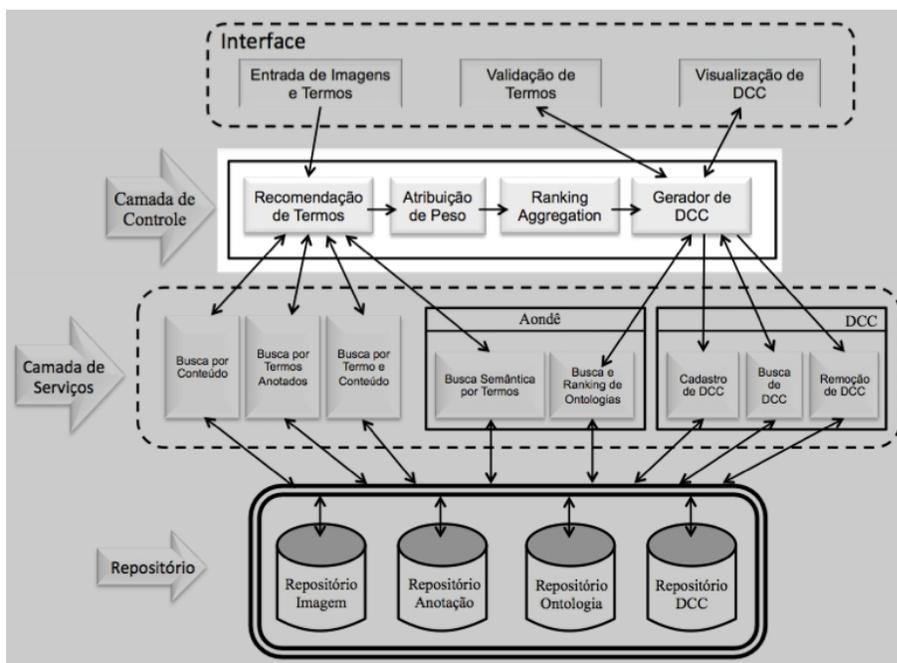


Figura 3.1: Macro fluxo do processo de anotação, presente no arcabouço.

O diagrama de sequência da Figura 3.2 ilustra o funcionamento do processo de anotação semântica automática de DCC de imagem. Esse processo começa com o usuário inserindo uma imagem e/ou termos, como sugestão de anotações, iniciando o fluxo de trabalho de anotação por meio da chamada ao módulo de Recomendação de Termos. Este módulo realiza as buscas multimodais para encontrar termos relacionados aos dados fornecidos pelo usuário. Cada termo encontrado possui um peso, formando um par  $(t, w)$  que compõem as listas de pares resultantes do módulo de Recomendação de Termos. Essas listas e mais os termos do usuário são enviados para o módulo de Atribuição de Peso, que classifica os termos sugeridos pelo usuário e encaminha as listas para o módulo de *Ranking Aggregation*.

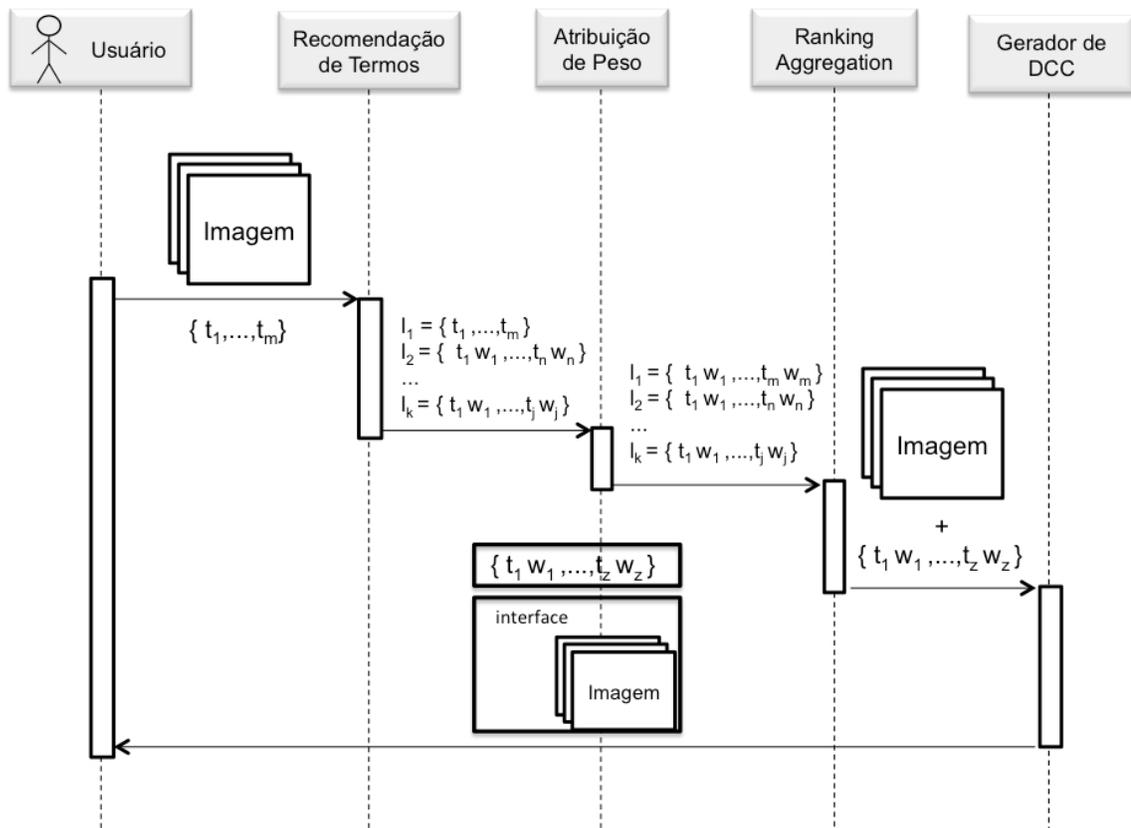


Figura 3.2: Diagrama de sequência do processo de anotação semântica automática de um DCC de imagem.

O módulo de *Ranking Aggregation* combina as listas obtendo uma única lista de termos. Essa lista é enviada para o módulo Gerador de DCC, que filtra os termos com pesos relevantes e busca a ontologia de domínio de cada termo resultante. Com os termos e mais as ontologias encontradas a partir de cada termo são formadas as anotações semânticas

da imagem. O módulo Gerador de DCC constrói o DCC de imagem com essas anotações semânticas. Por fim, o DCC gerado é retornado para o usuário.

Durante a especificação do processo e do arcabouço de anotação foi levado em consideração uma característica importante: como realizar o processo de anotação automática e geração do DCC de imagem para um grande volume de imagens com ou sem anotações. Como o processo é automático e a sua estrutura é um Fluxo de Trabalho orientado a serviço, pode-se facilmente criar um serviço que busca as imagens de um repositório, com ou sem anotações, e para cada imagem realiza o processo de anotação automática de DCC de imagem. Essa adaptação é possível e fácil de ser realizada devida a flexibilidade da Arquitetura Orientada a Serviço (SOA - *Service Oriented Architecture*<sup>1</sup>) utilizada na implementação do processo proposto.

Este capítulo apresenta o processo de anotação semântica automática de DCC de imagem e o arcabouço de anotação que possibilita a realização desse processo. Primeiramente, são apresentados na Seção 3.1 os algoritmos básicos de busca e recomendação de termos: Busca por Texto, Busca por Conteúdo, Busca por Conteúdo e Texto e Busca Semântica. Estes algoritmos são a base para a construção dos Fluxos de Busca. A Seção 3.2 apresenta o Processo de Anotação de Imagem, que foi dividida de acordo com os seus fluxos de trabalho: Fluxos de Busca e Fluxo de Processamento da Anotação.

A Seção 3.3 apresenta a Arquitetura do Arcabouço de Anotação, que utiliza todos os fluxos do processo de anotação automática de DCC de imagem. A arquitetura está dividida em quatro camadas: Interface, Camada de Controle, Camada de Serviços e Repositórios, que são descritas na seção de Arquitetura do Arcabouço de Anotação. Por fim, a Seção 3.4 apresenta as considerações finais do capítulo.

## 3.1 Algoritmos de Busca e Recomendação de Termos

O processo de busca e recomendação de termos para anotação de imagem utilizado possui como base quatro tipos de busca: busca por conteúdo, busca por anotação, busca por conteúdo e por anotação e busca semântica do termo. Cada uma das quatro formas de busca estão disponíveis como serviços, conforme especificado na arquitetura do arcabouço de anotação. A Figura 3.3 mostra a parte da arquitetura correspondente a esses serviços e ao módulo de Recomendação de Termos, que controla as chamadas aos serviços.

A seguir, são definidos os quatro principais algoritmos de busca, que são combinados de diversas formas para atenderem as necessidades dos fluxos de anotação de imagem descritos na Seção 3.2.

---

<sup>1</sup><http://www-01.ibm.com/software/solutions/soa/what-is-soa.html> (último acesso em junho de 2012).

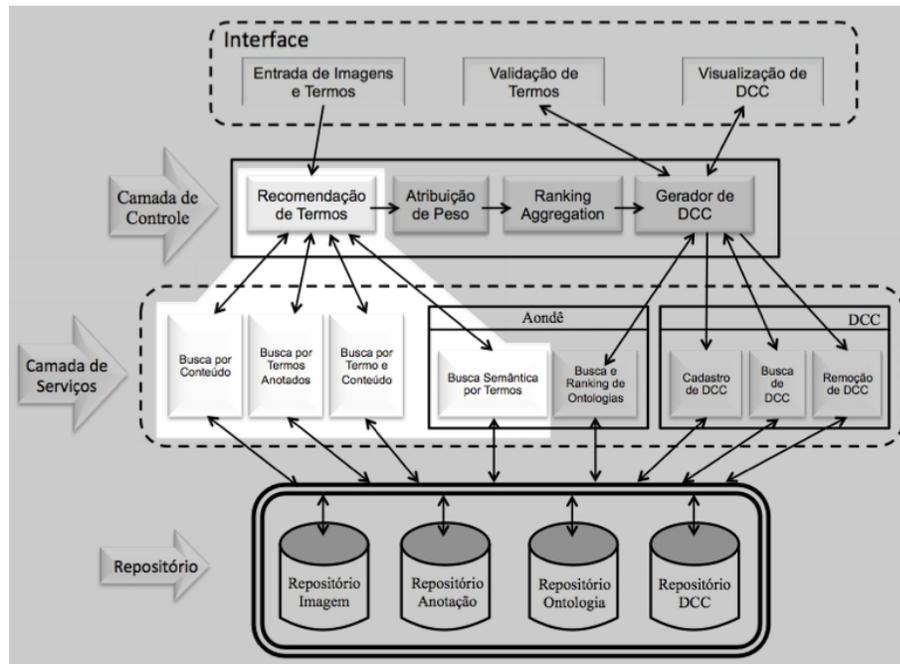


Figura 3.3: Parte da arquitetura do arcabouço que contém os serviços de busca e o módulo que os controla.

### 3.1.1 Busca por Texto

A busca por texto consiste em realizar uma consulta na base de dados de anotações procurando as imagens que possuem exatamente o mesmo texto que o termo pesquisado (*term*), desconsiderando letras maiúsculas e minúsculas. A partir dessas imagens são encontrados os demais termos anotados a elas, para serem retornados na busca. Para cada termo recuperado ( $t_i$ ) é atribuído um peso ( $w_i$ ) que possui o mesmo valor para todos os termos encontrados pelo algoritmo. Com isso, temos um par  $b_i = (t_i, w_i)$  que representa um termo ponderado da lista de recomendações retornada pelo algoritmo de Busca de Texto. No final, o algoritmo retorna uma lista ordenada ( $T$ ) dos  $n$  pares  $(b_0 \dots b_{n-1})$  recomendados pela busca. A seguir, é apresentada a interface do serviço de busca por texto.

$$\text{List}\langle \text{Term}, \text{Weight} \rangle \text{ SearchByText}(\text{Text } term)$$

### 3.1.2 Busca por Conteúdo

O serviço de busca por conteúdo consiste em buscar imagens similares, anotadas anteriormente, para que possa recuperar os termos associados a elas, classificá-los e retorná-los

como uma forma de recomendação de anotação para a imagem desejada. A seguir é apresentada a interface desse serviço.

**List**<**Term**, **Weight**> *SearchByImage*(**ByteArray** *image*, **Number** *similarity*)

Este serviço recebe uma imagem (*image*) na qual é aplicada uma técnica de extração de informação por meio de descritores de imagem, por exemplo, descritor de histograma de cores [63]. Utilizando um ou mais descritores, o algoritmo extrai todos os vetores com as informações (conteúdo) de cada imagem do repositório do sistema, e os compara com os vetores de conteúdo extraídos da imagem passada como parâmetro. Essa comparação entre os vetores é realizada utilizando uma técnica baseada nos descritores, de modo que seja possível determinar quais imagens (*img*) do repositório são mais similares à imagem pesquisada.

Dessa forma, o algoritmo de comparação irá retornar as  $n$  imagens que tenham no mínimo o grau de similaridade definido pelo parâmetro *similarity*, obtendo uma lista ordenada dos pares  $m_i = (img_i, grau_i)$  (para  $i = 0, \dots, n - 1$ ), em que  $img_i$  é uma das imagens encontradas na busca por conteúdo e  $grau_i$  é o grau de similaridade dessa imagem em relação à imagem pesquisada. Na sequência, o algoritmo normaliza o grau de cada imagem recomendada ( $img_i$ ), para que possa ser replicado entre os termos anotados ( $t_j$ ) naquela imagem (para  $j = 0, \dots, k_i - 1$ , em que  $k_i$  é o número de termos anotados para  $img_i$ ), obtendo os conjuntos de pares  $c_j = (t_j, w_i)$  para cada  $t_j$  recuperado de uma  $img_i$ . Esse resultado possui a mesma estrutura do serviço de busca por texto (*SearchByText*), ou seja, uma lista ordenada ( $I$ ) dos  $m$  conjuntos  $C = \{c_0, \dots, c_{m-1}\}$  retornados pelo serviço *SearchByImage* (em que  $m = \sum_{i=0}^n k_i$ , que representa o número total de termos encontrados pelo algoritmo).

### 3.1.3 Busca por Conteúdo e Texto

O algoritmo utilizado por este serviço de busca reutiliza partes dos algoritmos de Busca por texto e Busca por conteúdo apresentados anteriormente. O objetivo deste serviço é realizar uma busca mais refinada por imagens e suas anotações, retornando termos mais relevantes para sugestões como anotações de imagem. A seguir, é apresentada a interface do serviço de busca por conteúdo e por texto.

**List**<**Term**, **Weight**> *SearchByImageAndText*(**ByteArray** *image*,  
**Number** *similarity*, **List**<**Term**> *terms*)

Este serviço primeiro realiza a busca por conteúdo. Esta busca também utiliza descritores de imagens para comparar as imagens e atribuir pesos de relevância, da mesma

forma que o algoritmo de Busca por Conteúdo. Em seguida, para cada imagem encontrada, é realizada uma comparação para descobrir quais termos da lista (*terms*) estão anotados naquela imagem. Se nenhum termo da lista for encontrado, a imagem é descartada. Quanto mais termos iguais ao da lista uma imagem tiver, maior será o seu peso, indicando maior relevância das suas anotações.

Para cada imagem, é feito um controle que permite saber quantos termos da lista estão anotados. Para cada termo anotado, é somado um valor  $\lambda$  ao peso do grau de similaridade do conteúdo da imagem encontrada ( $grau_i$ , para  $i = 0, \dots, n - 1$ , em que  $n$  é o número de imagens encontradas pela busca por conteúdo  $img_i$ ), obtendo o peso ( $w_i$ ) dos termos daquela imagem. Todos os termos anotados em uma imagem são retornados juntamente com os seus respectivos pesos ( $w_i$ ) obtidos nesse processo de busca. Cada termo anotado é representado por  $t_{ij}$ , para  $j = 0, \dots, k_i - 1$  em que  $k$  é o número de termos anotados em cada  $img_i$ . Desta forma, o retorno desse serviço é uma lista ordenada  $P$  de pares  $c_i = (t_i, w_i)$ .

### 3.1.4 Busca Semântica

A Busca Semântica consiste em realizar a busca e *ranking* de ontologias do Serviço Web Aondê, passando como parâmetro um termo ou texto. Dentre as ontologias retornadas, é escolhida aquela que possui a melhor classificação no *ranking* para procurar por outros termos com alguma relação conceitual com o termo (*term*) fornecido ao método de busca. A relação entre os termos depende de como foi definida a representação conceitual dos termos nas ontologias.

Outro fator importante para a busca é a divisão dos repositórios de ontologias por áreas de interesse, por exemplo um repositório de ontologias de cores que contém as definições de cores e todos os possíveis espectros de cores representa um área de interesse. Esses conceitos e seus valores podem ser utilizados para identificar termos que dizem respeito às cores presentes em uma imagem.

Por causa disso, o método aceita até duas URLs de repositórios de ontologias (parâmetros *urlRepSearch1* e *urlRepSearch2*) para que possa pesquisar em mais ontologias. Caso nenhuma URI de repositório seja informada, a busca será no repositório de ontologias do sistema. As ontologias de repositórios externos utilizadas são importadas para o repositório de ontologia do Aondê, que é o mesmo utilizado pelo sistema. A seguir, é apresentada a interface do serviço de busca semântica.

```
List<Object> SemanticSearch(Text term, Text urlRepSearch1,
                          Text urlRepSearch2, Number criterion1,
                          Number criterion2, Number criterion3,
                          Number criterion4)
```

A busca de termo é feita procurando por classes ou instâncias nas ontologias que tenham o mesmo texto que o termo procurado. Quando encontrado o termo, procura-se pelos demais termos ligados direta ou indiretamente a ele. Para os termos associados indiretamente, são considerados apenas os que possuem no máximo um elemento (ou classe) interligando-os, pois, além disso, as associações começam a perder relevância em relação à classe de origem.

Existem algumas relações que são importantes e são amplamente utilizadas: “é um”, “é subclasse de” ou “é subtipo de”. Além dessas, podem ser criadas relações específicas para cada relacionamento dependendo da necessidade. Um exemplo é a ontologia da língua inglesa Wordnet<sup>2</sup>, que é utilizada para encontrar sinônimos, hiperônimos e hipônimos do termo pesquisado.

Com isso, temos as seguintes métricas de comparação semântica:

- o termo buscado é uma classe.
- o termo buscado é uma instância.
- o termo buscado está associado diretamente, ou seja, sem precisar passar por um outro termo até chegar ao termo buscado.
- o termo buscado está associado indiretamente, ou seja, precisa passar por um outro termo até chegar ao termo buscado.
- existe uma relação específica (de acordo com o domínio da ontologia) entre o termo sugerido e o termo buscado.

Para cada uma dessas métricas, o algoritmo de busca atribui um peso proporcional à associação com o termo buscado, e que representa a relevância desse termo encontrado em relação ao termo pesquisado. Um termo recomendado pelo algoritmo pode se encaixar em mais de um desses critérios. Dessa forma, o seu peso será a soma dos pesos atribuídos aos critérios no qual o termo se encaixa.

A Figura 3.4 mostra um exemplo de ontologia com pesos para as métricas mencionadas acima. Realizando uma Busca Semântica pelo termo “*Camaro*”, temos o seguinte resultado:  $S = \{(GM, 3), (Amarelo, 3), (GMFlex, 3), (Gasolina, 2)\}$ . Como o termo buscado é uma instância, ele soma o peso 1 para cada um dos pesos dos outros critérios que forem encontrados. O termo “*Petróleo*” não é retornado, pois possui uma associação indireta por meio de outros dois elementos, o que não se encaixa em nenhum dos critérios de associação. Os Critérios “*é uma classe*” ou “*é uma ocorrência*” não servem para escolher um termo para anotação. Eles servem apenas para aprimorar os critérios de ponderação dos resultados obtidos pelas associações.

---

<sup>2</sup><http://wordnet.princeton.edu/> (último acesso em junho de 2012).

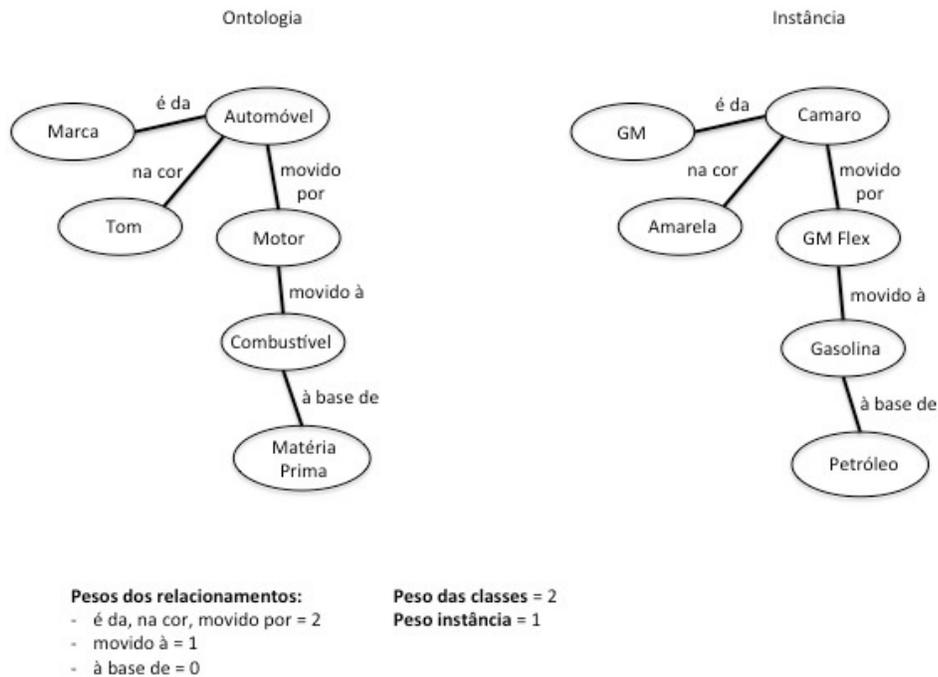


Figura 3.4: Exemplo de ontologia com seus respectivos pesos.

Encontrados os termos ( $t_i$ ) e obtidos os seus respectivos pesos ( $w_i$ ), o serviço *Semantic-Search* retorna uma lista ordenada ( $R$ ) dos  $n$  objetos (**Object**) com os termos sugeridos pelo resultado da busca. A estrutura **Object** contém os seguintes elementos: **term**, **weight** e **ontology**. O elemento **ontology** é a URL da ontologia no qual o termo foi encontrado. A ontologia também é retornada, pois o sistema precisará associar cada termo encontrado a uma ontologia para montar o DCC dessa imagem. O Capítulo 4 apresenta exemplos de uso deste serviço.

## 3.2 Processo de Anotação de Imagem

O processo de anotação de imagem é composto dos Fluxos de Busca e Fluxo de Processamento das Anotações. Os Fluxos de Busca são na verdade fluxos de trabalhos construídos para chamar os serviços de busca, procurando obter as melhores recomendações possíveis

de termos. Os serviços são combinados de forma a obter nove fluxos de recomendações de termos:

- busca por texto;
- busca por conteúdo;
- busca por conteúdo e por texto;
- busca semântica;
- busca por conteúdo e busca semântica;
- busca por texto e busca semântica;
- busca por conteúdo e por texto e busca semântica;
- busca semântica e busca por texto;
- busca semântica e busca por conteúdo e por texto.

O Fluxo de Processamento das Anotações é composto pela chamada dos seguintes serviços, respectivamente: Atribuição de Peso, Fusão de Dados e Criação de DCC. Esse fluxo é processado logo após a execução de todos os Fluxos de Busca e recebe as listas ordenadas resultantes dos Fluxos de Busca como entrada para o serviço de Fusão de Dados. A seguir, são descritos cada um desses fluxos, que juntos compõem o processo de anotação automática de DCC de imagem.

### **3.2.1 Fluxos Básicos de Busca**

Os fluxos básicos são compostos por apenas um serviço, logo, cada um dos serviços de busca apresentados na seção de Algoritmos de busca e recomendação de termos (Seção 3.1) é a definição do fluxo ao qual pertence. Dessa forma, temos os seguintes fluxos básicos: fluxo de busca por texto, fluxo de busca por conteúdo, busca por conteúdo e texto e fluxo de busca semântica, respectivamente. As listas ordenadas resultantes dos fluxos ou serviços, neste caso, são enviadas para o serviço de Fusão de Dados, conforme mencionado anteriormente.

### 3.2.2 Combinações de Fluxos de Busca

Os fluxos compostos existem para tentar encontrar mais termos, melhorando as chances de obter termos relevantes para a anotação. A forma de combinar os fluxos é simples; basta utilizar o resultado do primeiro fluxo como entrada para o segundo fluxo. Para os fluxos que dependem do conteúdo da imagem, é utilizada a mesma imagem que o usuário forneceu. Cada um dos fluxos compostos retorna uma lista (*ranking*) igual ao do último fluxo básico que compõe o fluxo composto. Com isso, temos os seguintes fluxos compostos:

- **Fluxos de Busca por conteúdo e Busca semântica:** Este fluxo começa com o processamento do fluxo básico de busca por conteúdo. A partir da lista resultante desse fluxo, é obtida uma lista de novos termos, que é enviada para o fluxo básico de busca semântica. Finalmente, é obtida a lista resultante do último fluxo básico, o de busca semântica, que é retornado como resultado final.
- **Fluxos de Busca por texto e Busca semântica:** O funcionamento desse fluxo é semelhante ao anterior mudando apenas o primeiro fluxo básico para o fluxo de busca por texto.
- **Fluxos de Busca por conteúdo e por texto e Busca semântica:** O funcionamento desse fluxo é semelhante ao anterior mudando apenas o primeiro fluxo básico para o fluxo de busca por conteúdo e por texto.
- **Fluxos de Busca semântica e Busca por texto:** Este fluxo começa com o processamento do fluxo básico de busca semântica. A partir da lista resultante desse fluxo, é obtida uma lista de novos termos, que é enviada para o fluxo básico de busca por texto. Finalmente, é obtida a lista resultante do último fluxo básico, o de busca por texto, que é o resultado final retornado.
- **Fluxos de Busca semântica e Busca por conteúdo e por texto:** Este fluxo começa com o processamento do fluxo básico de busca semântica. A partir da lista ordenada resultante desse fluxo, é obtida uma lista de novos termos, os quais são considerados como anotações da imagem fornecida pelo usuário. Com isso, o fluxo básico de busca por conteúdo e por texto pode ser executado tendo como entrada essa imagem e suas novas anotações. Finalmente, é obtida a lista resultante do último fluxo básico, o de busca por conteúdo e por texto, que é o resultado final retornado.

### 3.2.3 Atribuição de Peso

O serviço de atribuição de peso pode ser chamado paralelamente aos serviços de busca de termos para anotação, pois ele apenas irá atribuir um peso igual para cada um dos termos

sugeridos pelo usuário no começo no processo de anotação automatizada de imagem. O serviço recebe uma lista de termos (*terms*) e retorna uma lista semelhante as listas retornadas pelos serviços de buscas: uma lista  $W$  contendo os pares  $c_i = (t_i, w)$ , para  $i = 1..n$ , em que  $n$  é o número de termos fornecidos pelo usuário). A seguir é apresentada a interface desse serviço.

$$\text{List}\langle\text{Term, Weight}\rangle \text{ Weight}(\text{List}\langle\text{Text}\rangle \text{ terms})$$

### 3.2.4 Fusão de Dados

Esta parte do fluxo consiste no envio das listas ordenadas dos termos obtidos para recomendação de anotações para o serviço de fusão de dados. Este serviço pode ser qualquer serviço que utiliza alguma técnica de fusão de dados da literatura, para combinar as listas.

A entrada desse serviço deve ser uma lista (**lists**) de listas ( $L_1, L_2, \dots, L_m$ ), em que  $m$  é o número de listas enviadas para o serviço de fusão) de elementos do mesmo tipo que os retornados pelos serviços de busca de termos. Por exemplo, a lista  $L_1$  contém os pares  $c_i = (t_i, w_i)$ , para  $i = 1, \dots, n$ , em que  $n$  é o número de termos daquela lista.

O serviço de fusão deve retornar uma lista ordenada  $Lr$  com as sugestões finais para anotações, ou seja, com os pares  $c_i = (t_i, w_i)$  obtidos ao realizar a fusão de dados. Esta lista será encaminhada para o serviço de busca de ontologia por termo.

Apesar de ser possível mudar o método de fusão utilizado, trocando um serviço de fusão pelo outro, os serviços devem possuir as mesmas entradas e saídas. Portanto, a interface desse serviço deve possuir sempre a mesma estrutura genérica, mostrada abaixo.

$$\text{List}\langle\text{Term, Weight}\rangle \text{ RankingAggregation}(\text{List}\langle\text{List}\langle\text{Term, Weight}\rangle\rangle \text{ lists})$$

### 3.2.5 Busca de Ontologia por Termo

Esta etapa do fluxo recebe a lista ordenada resultante do serviço de fusão de dados. Com isso, o serviço irá procurar por ontologias que definem cada um dos termos da lista. Para isso pode-se utilizar qualquer serviço que recebe uma lista de termos e pesos (*list*) e retorne uma lista ordenada de objetos (**Object**) com os termos que foram encontrados nas ontologias por meio do serviço de busca.

A estrutura **Object** contém os seguintes elementos: **term**, **weight** e **ontology**. O elemento **ontology** é a URL da ontologia no qual o termo foi encontrado. Os termos que não foram associados a alguma ontologia são descartados nesta etapa do fluxo, pois não faz sentido sugerir um termo sem ontologia para uma anotação semântica da imagem. A seguir é apresentada a interface do serviço de busca de ontologia por termo.

$$\text{List}\langle\text{Object}\rangle \text{ SearchOntology}(\text{List}\langle\text{Term, Weight}\rangle \text{ list})$$

### 3.2.6 Criação de DCCs

No fluxo, a parte de criação de DCC pode ser dividida em duas etapas: validação dos termos pelo usuário e criação e armazenamento do DCC. A parte de validação pelo usuário é opcional. A entrada desse serviço é uma lista de termos com os seus respectivos pesos e ontologias e também uma imagem para ser anotada. Em seguida o serviço passa pela etapa de validação, se for requisitado. Caso contrário, os parâmetros de entrada são encaminhados diretamente para a etapa de criação do DCC. Esta última etapa constrói o DCC e armazena sua estrutura no repositório do sistema.

## 3.3 Arquitetura do Arcabouço

A arquitetura do arcabouço de anotação proposto está dividida em quatro camadas: interface, camada de controle, camada de serviços e repositórios, conforme exibido na Figura 3.5.

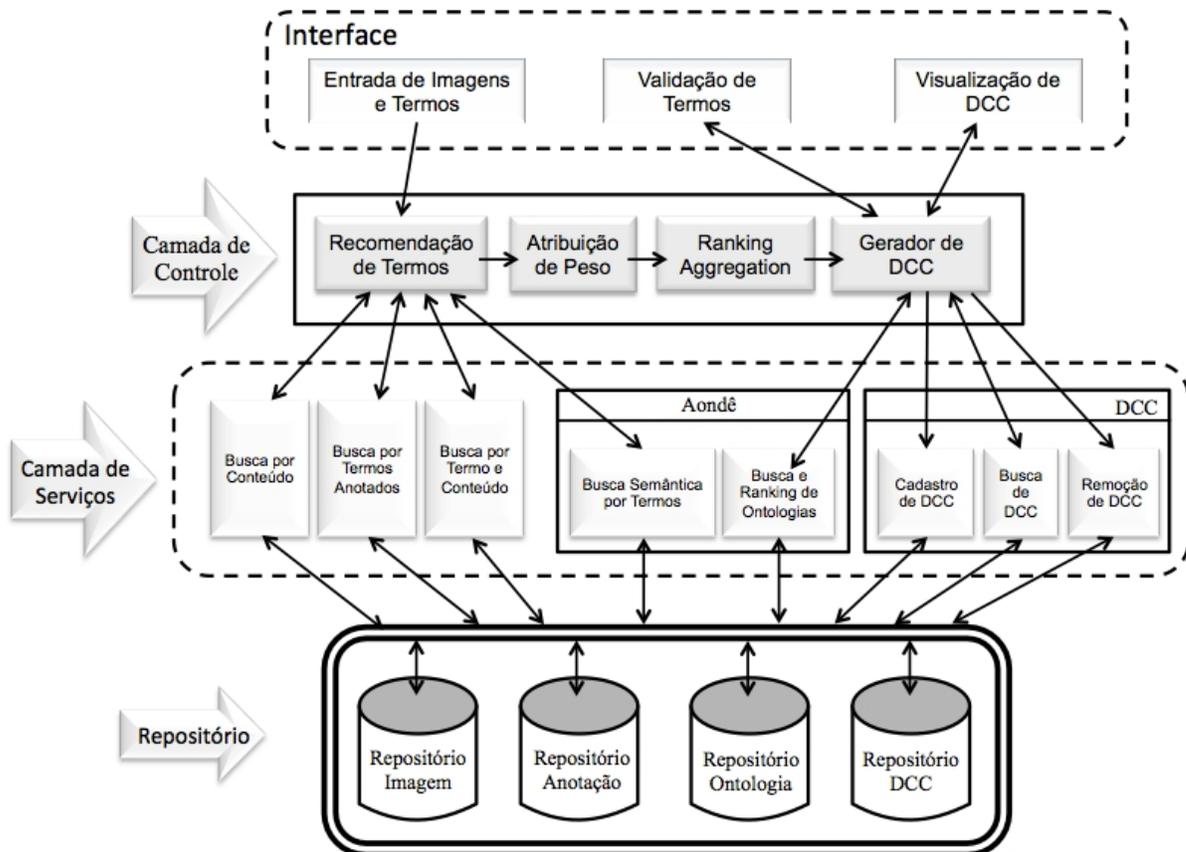


Figura 3.5: Arquitetura do arcabouço de anotação de componentes de imagem.

### 3.3.1 Interface

A interface é a camada de interação com o usuário, possibilitando acesso às funcionalidades do serviço: anotações de imagens, avaliação das anotações, busca e visualização de componentes de dados. Dessa forma, foram projetados módulos para realizar essas funcionalidades: módulo de entrada de imagem e termos, módulo de validação de termos e módulo de visualização de componentes (DCC).

#### Módulo de Entrada de Imagens e Termos

Este módulo consiste em uma interface pela qual o usuário insere a imagem a ser anotada e, se desejar, pode informar alguns termos que gostaria de associar à imagem. Após o usuário informar esses dados, é iniciado o processo de anotação automatizada da imagem fornecida. Este fluxo é gerenciado pela camada de controle, apresentada mais adiante.

#### Módulo de Validação de Termos

Este módulo é uma interface que exibe para o usuário uma lista com os termos sugeridos para a anotação. Estes termos são exibidos juntamente com a sua pontuação, indicando a relevância e associação de cada termo com a imagem informada. Neste momento o usuário pode editar os termos, definindo a combinação que melhor lhe agrada. Assim, o usuário pode concluir esta etapa de avaliação e ajustes dos termos e permitir a criação do DCC.

#### Módulo de Visualização de DCC

O DCC é uma estrutura de objetos digitais complexos que permite representar a imagem informada e a sua relação com os termos e conceitos associados a ela. Portanto, o módulo de visualização de DCC possibilita que o usuário visualize o DCC gerado, o que inclui a imagem, termos, metadados, ontologias e a interface de acesso do DCC. Em resumo, é possível visualizar a estrutura completa do DCC e as suas partes.

### 3.3.2 Camada de Controle

Esta camada é responsável por controlar o fluxo de execução do processo automatizado de anotação de imagens. As responsabilidades desta camada estão divididas entre os seus quatro módulos de processamento: Recomendação de termos, Atribuição de Peso, Fusão de Dados (*Ranking Aggregation*) e Gerador de DCC. Estes módulos compõem o fluxo de trabalho mostrado na Figura 3.1. Cada módulo é responsável por orquestrar as

chamadas dos serviços que necessita para cumprir sua tarefa. A seguir são apresentados mais detalhes sobre cada um dos módulos.

### Módulo de Recomendação de Termos

O módulo de Recomendação de Termos recebe da interface uma imagem com ou sem termos sugeridas pelo usuário. Esses termos influenciam no comportamento deste módulo, pois com os termos são executados mais fluxos de busca, obtendo mais sugestões para a seleção das anotações. Desta forma, o objetivo deste módulo é obter novos termos para recomendação a partir das buscas por termos ou por imagens, resultando em várias listas ordenadas de termos. Cada uma dessas listas é composta por pares, definido por um termo e um peso, ou seja,  $C = \{(t_1, w_1), (t_2, w_2), \dots, (t_n, w_n)\}$  (termo e peso, respectivamente), em que  $n$  é o número de termos encontrados.

As buscas por termos para a anotação utilizam os serviços do Aondê e algumas funcionalidades do OntoSAIA (adaptadas para serem acessadas como serviços). Com isso, temos disponíveis os seguintes serviços: busca por conteúdo, busca por termos anotados, busca por termos e conteúdo, busca semântica por termo, busca e *ranking* de ontologia, cadastro de DCC, busca de DCC e remoção de DCC.

### Módulo de Atribuição de Peso

Cada um dos serviços de busca por termos é responsável por atribuir pesos aos termos encontrados. Contudo, os termos fornecidos pelo usuário também precisam ter pesos que representem a sua relevância no processo de seleção dos termos para anotação (Fusão de Dados). Para isso, o módulo de atribuição de peso irá processar todos os termos sugeridos pelo usuário, atribuindo um peso fixo para cada um deles. Em seguida, monta-se uma lista ordenada de pares formados por termo e peso, igual aos gerados no módulo de recomendação de termos.

### Módulo de Ranking Aggregation

Este módulo recebe as recomendações resultantes dos serviços de busca e de atribuição de peso, ou seja, várias listas dos pares termo e peso ( $l_1 = \{(t_1, w_1) \dots (t_n, w_n)\}, l_2 = \{(t_1, w_1) \dots (t_m, w_m)\}, \dots$ ). A partir destas listas é aplicado o algoritmo de combinação para selecionar os melhores termos de acordo com os seus respectivos pesos. O resultado deste módulo é uma lista de pares (termo, peso), que pode ser submetida para a avaliação do usuário ou utilizada para compor as anotações da imagem, atividade . O gerenciamento da execução dessas duas atividades é responsabilidade do módulo Gerador de DCC.

### Módulo Gerador de DCC

Este módulo é responsável por gerenciar a validação dos termos com o usuário, caso necessário, obtendo assim o conjunto definitivo dos termos a serem anotados na imagem.

De posse da imagem e dos termos, este módulo encarrega-se de buscar as ontologias que definem cada um dos termos recomendados. Para isso é utilizado o serviço de busca e *ranking* de ontologias do Aondê, realizando a busca para cada um dos termos. Cada termo pesquisado é associado a ontologia com a melhor classificação no *ranking* retornado na busca.

Nesta etapa, não é utilizado nenhum padrão de metadado, como o *Dublin Core*<sup>3</sup>, para representação da anotação. A anotação da imagem consiste basicamente no termo recomendado associado a ontologia encontrada a partir desse termo. A ontologia teria a definição desse termo e do que ele representa como anotação daquela imagem.

Para utilizar padrões de metadados, precisa-se ter o cuidado de verificar se aquele metadado faz sentido como anotação daquela imagem com aquele termo recomendado. Por exemplo, dada uma imagem o que representa o metadado *Title* e como buscar termos que possam ser utilizados como título daquela imagem. Para isso, os mecanismos de busca precisariam ser mais refinados e saber o conceito do que está buscando. Pensando em facilitar a definição do processo e do arcabouço de anotação como um todo, foi definido que a anotação seria apenas o termo associado à ontologia.

O próximo passo consiste na geração das estruturas do DCC: XML de representação da imagem, interface do DCC disponibilizando de forma semântica o acesso à imagem e as suas anotações e, por fim, os metadados dos termos anotados. Com as estruturas montadas, o módulo invoca o serviço para cadastrar o DCC gerado.

O cadastro consiste em: armazenar a imagem anotada no repositório de imagens; armazenar as anotações semânticas da imagem, ou seja, os termos e as referências para as suas ontologias no repositório de anotações; e armazenar no repositório de DCCs a estrutura do DCC, os arquivos XML e as relações entre imagem e suas anotações semânticas.

As ontologias que compõem as anotações semânticas já foram armazenadas no repositório de ontologias, pois o Aondê guarda todas as ontologias retornadas pelos seus serviços de busca. Por fim, o módulo retorna para a interface de visualização o DCC construído.

### 3.3.3 Camada de Serviços

Esta camada contém os serviços web que serão utilizados pelo sistema de recomendação de termos e geração de DCCs. Os serviços estão agrupados da seguinte forma: serviços de busca de termos em imagens anotadas (Busca em imagens anotadas), serviços de busca

---

<sup>3</sup><http://dublincore.org/documents/dces/> (último acesso em junho de 2012).

semântica (Aondê) e serviços de DCCs. A seguir é apresentado cada um desses serviços e suas funcionalidades.

### **Busca em Imagens Anotadas**

Os serviços desse grupo são baseados nas funcionalidades do OntoSAIA. Foram criados os seguintes serviços: busca por conteúdo, busca por texto e busca por conteúdo e texto.

O serviço de busca por conteúdo consiste em realizar busca por conteúdo de imagens anotadas na sua base de dados. A busca por conteúdo aplica técnicas de extração de características das imagens que descrevem propriedades visuais, como cor, textura e forma de objetos para realizar a busca na base de dados. O algoritmo que descreve melhor o comportamento dessa busca é o de Busca por Conteúdo, apresentado anteriormente.

A busca por texto é mais simples. Basicamente, realiza-se uma busca por anotações que utilizaram aquelas palavras-chaves definidas, encontrando as imagens cuja anotação contem os termos utilizados e retorna as demais anotações associadas a essas imagens. O algoritmo que descreve melhor o comportamento dessa busca é o de Busca por Texto, apresentado anteriormente.

Por fim, a busca por conteúdo e por texto, que procura por imagens similares que tenham como anotações os textos passados como parâmetro e que sejam similares quanto ao conteúdo visual. O algoritmo que descreve melhor o comportamento dessa busca é o de Busca por Conteúdo e Texto, apresentado anteriormente. Com o uso desses mecanismos de busca de termos é possível encontrar mais termos para recomendação de termos para anotação.

### **Aondê**

Conforme mencionado na Revisão Bibliográfica, o Aondê provê várias funcionalidades de gerenciamento e de busca de ontologias. Para aprimorar a busca por termos na recomendação, será utilizado o serviço de Busca Semântica de Termo. Este serviço consiste em localizar outros termos (conceitos na ontologia) que estão relacionados ao termo informado.

O outro serviço utilizado é o de Busca e Ranking de Ontologias. Este serviço serve para localizar a ontologia de domínio que define ou esteja relacionada com o termo informado. Dessa forma, esse serviço é chamado para cada um dos termos definidos. Caso um dos termos não esteja associado a uma ontologia, o sistema irá reduzir a relevância desse termo na recomendação para anotação. Na construção do DCC, a ontologia será vinculada ao termo buscado.

## DCC

Este módulo contém os principais serviços para gerenciar os DCCs construídos pelo Módulo Gerador de DCC dentre eles estão:

- ***Cadastrar DCC***: serviço responsável por armazenar as informações e estruturas do DCC nos seus respectivos repositórios;
- ***Buscar DCC***: serviço responsável pela busca de DCCs a partir de palavras-chave.
- ***Remover DCC***: serviço responsável por remover as informações e estruturas do DCC dos seus respectivos repositórios.

### 3.3.4 Repositórios

A camada de repositórios serve para armazenar todas as informações relevantes durante o processo de anotação de imagens e geração de DCCs. Esta camada compartilha os repositórios utilizados pelos outros dois sistemas que auxiliam o nosso serviço: Aondê e OntoSAIA.

Os repositórios também são utilizados para guardar informações, como ontologias, que possam aprimorar o processo de busca e recomendação de termos. Com isso, os repositórios podem ser definidos da seguinte forma:

- ***Repositório de imagens***: Este é o repositório no qual serão armazenadas todas as imagens anotadas pelo OntSAIA ou utilizadas na geração do DCC. Neste banco, também são armazenados metadados da anotação da imagem para ajudar na recuperação da imagem;
- ***Repositório de anotação***: Este é o repositório no qual serão armazenados os metadados e termos que compõem a anotação;
- ***Repositório de ontologia***: Este repositório armazena as ontologias de domínio utilizadas pelo DCC para definição dos termos anotados e também as ontologias buscadas no Aondê para auxiliar nas buscas por recomendações de termos;
- ***Repositório de DCC***: Este repositório grava as informações necessárias para obter o DCC de imagem novamente, ou seja, os documentos e ontologias que relacionam as imagens e as anotações.

## 3.4 Considerações do Capítulo

Este capítulo mostrou inicialmente uma visão geral do processo e do arcabouço de anotação de imagem definidos nessa dissertação de mestrado, com o objetivo de prover um meio para anotação semântica automática de DCCs de imagens. A partir dessa visão geral, descrevem-se as principais contribuições dessa pesquisa: Algoritmos de Busca e Recomendação de Termos, Processo de Anotação de Imagem e Arquitetura do Arcabouço.

A base para os fluxos de busca do processo de anotação são os quatro algoritmos especificados na seção Algoritmos de Busca e Recomendação de Termos: Busca por Texto; Busca por Conteúdo; Busca por Conteúdo e Texto; e Busca Semântica, que formam os Fluxos Básicos de Busca.

Com relação ao processo, foi especificado o fluxo de trabalho principal: Recomendação de Termos, Atribuição de Peso, *Ranking Aggregation* e Gerador de DCC. Esse fluxo de trabalho está dividido em Fluxos de Busca e Fluxo de Processamento das Anotações. O módulo de Recomendação de Termos é responsável por gerenciar a execução dos Fluxos de Busca para recomendação de termos. Estes fluxos estão divididos em: Fluxos Básicos de Busca e Fluxos Compostos de Busca.

Os demais módulos do fluxo de trabalho: Atribuição de Peso, *Ranking Aggregation* e Gerador de DCC, compõem o Fluxo de Processamento das Anotações, que é responsável por classificar e combinar os termos sugeridos, obtendo uma lista com os termos mais relevantes. Estes termos são associados às ontologias e juntos formam as anotações semânticas da imagem informada. Por fim, é construído e armazenado o DCC da imagem com as suas anotações.

Com o processo de anotação semântica automática de DCC de imagem definido, foi proposta a arquitetura do arcabouço de anotação que implementa esse processo. A arquitetura proposta está dividida em quatro camadas: Interface, Camada de Controle, Camada de Serviços e Repositório. A Camada de Controle contém o fluxo de trabalho principal do processo de anotação. Esta camada se comunica com a Interface para receber os parâmetros de entrada, o comando de início do processamento e o resultado do processo de anotação. Ela também se comunica com a Camada de Serviço para obter os recursos necessários para o seu processamento.

Dessa forma, temos a definição das principais contribuições dessa dissertação de mestrado: o processo e o arcabouço de anotação. O capítulo seguinte é responsável por apresentar detalhes do desenvolvimento desse arcabouço e dois estudos de caso que exemplificam o seu funcionamento.

# Capítulo 4

## Aspectos de Implementação e Estudos de Caso

Este capítulo apresenta aspectos de implementação do arcabouço para anotação de componentes de imagem, conforme a sua definição no capítulo de Especificação do Arcabouço de Anotação. A implementação do arcabouço possibilita validar as ideias e conceitos do processo de anotação semântica automática de DCCs de imagens, e demonstrar o funcionamento desse processo por meio de estudos de casos.

Os dois principais motivos da implementação do arcabouço são: prover um meio automático para geração de DCCs com estruturas semânticas; e disponibilizar uma ferramenta com arquitetura flexível para validação de diferentes serviços e algoritmos que compõem o processo de anotação semântica automática de DCCs de imagem.

Dessa forma, a Seção 4.1 apresenta aspectos de implementação do arcabouço para anotação de componentes de imagem, definindo tecnologias, ambientes, adaptações e evoluções de outras ferramentas. Esta seção está dividida em oito subseções:

- Configuração e Adaptação do OntoSAIA,
- Configuração e Adaptação do Aondê,
- Módulo do Controlador,
- Métodos Básicos para Busca de Termos,
- Fluxos de Anotação, Módulo de Atribuição de Peso e
- Módulo de Fusão de Dados e Módulo de DCC.

A Seção 4.2 apresenta os estudos de casos realizados para demonstrar o processo de anotação por meio do arcabouço desenvolvido. São dois os estudos de casos apresentados:

o Estudo de Caso Apenas com Imagem, que mostra o funcionamento do processo de anotação fornecendo apenas uma imagem, e como utilizar o arcabouço de anotação para realizar esse processo; e o Estudo de Caso com Imagem e Anotação, que é semelhante ao anterior mas neste caso são fornecidos como entrada uma imagem e alguns termos como sugestão de anotação.

## 4.1 Aspectos de Implementação

A arquitetura do arcabouço de anotação automatizada em DCC, apresentada na Seção 3.3, mostra que o sistema utiliza recursos do serviço Web Aondê e do OntoSAIA. O OntoSAIA foi escolhido para ser a base do desenvolvimento do arcabouço. Os principais motivos que levaram a essa decisão foram: os recursos utilizados do OntoSAIA possuem um alto nível de acoplamento entre si e com o sistema em geral; a grande complexidade para migrar as funcionalidades de busca por conteúdo e armazenamento das imagens e anotações do OntoSAIA para uma nova arquitetura orientada a serviços; a interface do OntoSAIA precisaria de uma pequena adaptação para atender a demanda do arcabouço de anotação proposto nesse trabalho; e a arquitetura e estrutura do OntoSAIA possibilitava a inclusão de um módulo para o arcabouço de anotação.

Esta seção descreve atividades realizadas relacionadas à:

- configuração e adaptação do OntoSAIA, incluindo a adaptação da interface;
- configuração e adaptação do Aondê;
- descrição da implementação de cada um dos quatro métodos básicos de busca de termos para anotação;
- implementação de cada um dos fluxos de anotação de imagem;
- implementação do módulo de atribuição de peso;
- implementação do módulo de fusão de dados; e
- implementação do módulo de criação do DCC.

O arcabouço de anotação foi implementado em Java 1.6 com parte em aplicação *desktop* (OntoSAIA e alguns módulos do arcabouço de anotação) e a outra parte como serviço Web (Aondê). Tanto o OntoSAIA quanto o Aondê foram adaptados para utilizar o PostgreSQL 9.1.2 como sistema de gerenciamento de bancos de dados.

### 4.1.1 Configuração e Adaptação do OntoSAIA

O OntoSAIA está estruturado inteiramente de forma modularizada e seguindo a definição do padrão *Singleton*<sup>1</sup> para controlar a instância dos objetos e o seu tráfego de um pacote para outro. A versão atual do OntoSAIA utiliza duas ontologias: uma é a WordNet<sup>2</sup> disponível na Web, e a outra é uma ontologia sobre taxonomia de espécies aquáticas armazenada na base de dados relacional. Atualmente o OntoSAIA contém sete pacotes:

- ***auxiliar\_pckgd***: contém classes auxiliares para exibir imagens, comparar pesos e uma classe com as propriedades de um peso;
- ***cbisc\_pckgs***: contém uma classe com as propriedades e funcionalidades para se trabalhar com histograma de cor, e outra classe com que gerencia o acesso aos recursos do CBISC;
- ***database\_pckg***: contém as classes responsáveis por conectar e gerenciar o acesso ao banco de dados, e também uma entidade para manipulação de imagem no banco de dados. Essas classes seguem os padrões DAO e Factory, bastante utilizados comercialmente;
- ***merge\_pckg***: contém as classes para realizar a fusão dos termos sugeridos pelos métodos de busca por anotações em imagens e de busca em ontologia. A fusão é realizada utilizando a combinação por *summative union* [14], que consiste na média aritmética dos pesos fornecidos para o método;
- ***ontology\_pckg***: neste pacote, estão as classes que acessam e manipulam as ontologias e, também alguns recursos específicos vinculado ao Wordnet;
- ***ontosaia\_kernel***: contém duas classes: uma com o objeto de imagem e suas anotações e, outra com os métodos que representam todas as funcionalidades acessadas pela interface do sistema. Esses métodos acessam e gerenciam recursos de todos os outros pacotes mencionados anteriormente.

O arcabouço de anotação foi incorporado ao OntoSAIA, de forma que a sua implementação ficasse em pacotes separados, sendo apenas necessária uma adaptação na classe da interface do sistema, que será descrita mais adiante. A arquitetura do arcabouço (mostrada na Figura 3.5 apresentada na Seção 3.3) contém os principais módulos de implementação apresentados na camada de controle: Recomendação de termos, Atribuição de

---

<sup>1</sup><http://java.sun.com/developer/technicalArticles/Programming/singleton/> (último acesso em junho de 2012).

<sup>2</sup><http://wordnet.princeton.edu/> (último acesso em junho de 2012).

Peso, Fusão de Dados e Gerador de DCC. Estes módulos foram criados como classes dentro do pacote *saagen4img-modules*: *SearchModule.java*, *AssignmentWeightModule.java*, *RankAggregationModule.java* e *DCCModule.java*, respectivamente. Também foram criados mais quatro pacotes:

- ***localhost.axis.services.WSOperationsService***: gerado automaticamente pelo *framework* Axis<sup>3</sup>, por causa do cliente do serviço Web do Aondê, que é apresentado na Seção 4.1.4;
- ***saagen4img-constants***: contém a interface *ApplicationConstants.java* com as constantes utilizadas pelo arcabouço, por exemplo o valor do peso atribuído aos termos encontrados pela busca por texto anotados nas imagens;
- ***saagen4img-model***: contém a entidade *Term.java* com os atributos: *String annotation*, *Float weight* e *String ontology* (irá guardar a URI da ontologia);
- ***saagen4img-util***: contém as classes com os recursos utilitários, como os métodos de conversão de tipos e entidades do OntoSAIA para a lista de termos do arcabouço de anotação. Esses recursos podem ser utilizados por qualquer classe do sistema.

As funcionalidades utilizadas do OntoSAIA para implementação do arcabouço de anotação serão tratadas nas próximas seções, conforme forem sendo necessárias.

A alteração na interface do OntoSAIA consistiu em incluir um botão para iniciar o fluxo de anotação automatizada de imagem utilizando o arcabouço de anotação proposto neste trabalho. A interface do OntoSAIA possui quatro classes que estendem *javax.swing.JPanel*, contendo quase toda a implementação de *layout* e tratamento de eventos com o usuário. Essas quatro classes dividem o *layout* da seguinte forma:

- ***enterAnnotationDisplayPanel.java***: entrada das anotações do usuário, escolha do meio de busca de sugestões e identificação do usuário;
- ***openFileDisplayPanel***: carrega a imagem e exibe na tela;
- ***viewAnnotationDisplayPanel***: mostra as anotações escolhidas para aquela imagem, permitindo remover ou finalizar o processo de anotação daquela imagem;
- ***viewSuggestionDisplayPanel***: mostra os termos sugeridos pela busca em ontologia e permite buscar imagens a partir desses termos ou escolher termos para anotação.

Todas essas classes são agrupadas no *JFrame interfaceComSugestao.java*. Um novo botão foi inserido na classe *enterAnnotationDisplayPanel.java*, com o rótulo *Automatic Annotation*.

---

<sup>3</sup><http://axis.apache.org/axis/> (último acesso em junho de 2012).

### 4.1.2 Configuração e Adaptação do Aondê

O Aondê foi construído como uma aplicação Web em Java, para disponibilizar serviços Web que trabalhem com ontologias: busca, armazenamento, entre outros descritos na Seção 2.3. A versão atual do Aondê foi portada para o Java 1.6 e PostgreSQL 9.1.2. A migração da versão do banco obrigou a realização de algumas configurações para habilitar os recursos utilizados da API Jena<sup>4</sup>. Essas adaptações foram em configurações do PostgreSQL com relação ao suporte para o Nuxeo<sup>5</sup>. O guia na internet que ajuda a realizar essa configura chama-se *Configuring PostgreSQL* disponível em <http://doc.nuxeo.com/display/ADMINDOC/Configuring+PostgreSQL> (último acesso em junho de 2012).

O serviço Web Aondê foi desenvolvido utilizando o *framework* Axis versão 1.4 e o servidor Tomcat<sup>6</sup>, versão 5.5. O *framework* Axis auxilia a criação de serviços Web, com todos os padrões definidos pelo W3C: SOAP e WSDL, e no acesso (clientes) e manuseio dos serviços Web. A Figura 4.1 mostra os serviços Web disponíveis do Aondê e os seus métodos.

O serviço *SearchRank* que faz a busca e *ranking* de ontologias depende da ontologia do WordNet para funcionar corretamente. Para isso, foi preciso criar uma base de dados com o nome **wordnet30** e executar os scripts de importação e configuração do banco de dados relacional para o WordNet versão 3.0. Atualmente o Aondê está configurado para trabalhar com as ontologias armazenadas em seu repositório (a partir de buscas ou inserção manual), com a ontologia WordNet e com o repositório de ontologia *Swoogle*<sup>7</sup> na Web. Por causa do OntoSAIA, foi inserida manualmente a ontologia de taxonomia de espécies aquáticas na base do Aondê, com o intuito de centralizar todas as buscas e manipulações de ontologias no Aondê.

### 4.1.3 Módulo do Controlador

O controle de toda a automatização e execução do processo anotação de imagem e geração do DCC é implementada na classe *ControllerModule.java*. A sua implementação consiste basicamente em um método *Controller*, que é chamado pelo evento do clique do botão *Automatic Annotation* na interface do OntoSAIA. Esse método possui a assinatura mostrada a seguir.

```
public List<Term> controller(final String[] userTerms,
                           String imageName)
```

<sup>4</sup><http://jena.apache.org/> (último acesso em junho de 2012).

<sup>5</sup><http://www.nuxeo.com/en/services/connect> (último acesso em junho de 2012).

<sup>6</sup><http://tomcat.apache.org/> (último acesso em junho de 2012).

<sup>7</sup><http://swoogle.umbc.edu/> (último acesso em junho de 2012).



Figura 4.1: Serviços Web do Aondê e seus métodos disponíveis no ambiente de desenvolvimento (*localhost*).

O método *controller* é responsável por chamar cada um dos fluxos de anotação, chamar o método de atribuição de peso, obter a lista de listas de termos e chamar o método de fusão de dados, e por fim chamar o método de criação do DCC.

#### 4.1.4 Métodos Básicos para Busca de Termos

Na Seção 3.1, são apresentados quatro métodos básicos para busca de termos: busca por texto, busca por conteúdo, busca semântica e busca por conteúdo e texto. Apenas os três primeiros métodos foram de fato implementados na classe *SearchModule.java*. A seguir, são apresentados detalhes da implementação de cada um deles.

##### Busca por Texto

Esse método possui a seguinte assinatura:

```
public List<List<Term>> searchByText(final String term)
```

Em que *Term* é a entidade que representa um termo sugerido, conforme apresentada anteriormente. Este método simplesmente chama o método *getImagesByTerm* da classe *OntosaiaManager.java*, que faz a busca por texto nas anotações das imagens. A assinatura desse método é:

```
public OntosaiaImage[] getImagesByTerm(String term)
```

A entidade *OntosaiaImage* representa uma imagem que foi anotada e inserida no banco. Os seus atributos são: *id* (identificado da base de dados) e *annotation* (um *Array* de *String*, com as anotações da imagem). Dessa forma, o resultado da execução desse método precisa ser convertido para o formato de retorno do método de busca por texto (*searchByText*) do arcabouço de anotação. Para fazer isso, foi criado um método de conversão na classe *ConverterModels.java* no pacote *saagen4img\_util*. A assinatura desse método de conversão ficou definida da seguinte forma:

```
public static List<List<Term>> converterOntosaiaImageArrayToTermsList(  
    final OntosaiaImage[] ontosaiaImages, final Float weight)
```

Esse método além de converter de um formato para outro também inclui o peso (*weight*) para o termo sugerido, obtendo o resultado no formato esperado pelo arcabouço de anotação.

## Busca por Conteúdo

Assim como o método de busca por texto, esse método também empacota o método *getObjectSuggestionByImageSimilarity* (busca por conteúdo) da classe *OntoSAIA\_Manager.java* do *OntoSAIA*, incluindo apenas uma lógica para converter os resultados para o formato esperado pelo arcabouço de anotação. O retorno do método do *OntoSAIA* é um vetor do objeto *Weighted\_Object*, que contém os atributos *weight* (peso) e *annotation* (anotação).

O método de busca por conteúdo possui a seguinte assinatura:

```
public List<Term> searchByContent(String fileName)
```

O parâmetro *fileName* é o nome do arquivo de imagem carregada pelo usuário e que fica salvo em memória durante o processo de anotação e armazenamento da imagem. Esse nome é repassado para o método de busca do *OntoSAIA*, que se encarrega de fazer a busca por conteúdo. Atualmente, o *OntoSAIA* está utilizando apenas o descritor histograma de cores. No entanto, esta ferramenta foi projetada para aceitar diferentes tipos de descritores, precisando apenas incluir os métodos ou serviços que gerem esses descritores. A busca por conteúdo do *OntoSAIA* já realiza uma fusão de dados com os termos anotados nas imagens encontradas.

Essa fusão é feita utilizando a técnica *summative union*, que realiza uma média aritmética com base nos pesos dos termos. A Figura 4.2 mostra o algoritmo desse método, que recebe dois parâmetros de entrada: um *vetor* de *Weighted\_Object* com os termos e seus pesos; e um inteiro *normalize*, que pode receber o valor um ou zero. Quando o parâmetro *normalize* recebe o valor um, isso significa que o algoritmo deve normalizar os pesos dos termos, caso contrário não. *Weighted\_Object* é um objeto que contém o texto e o peso de um termo.

O algoritmo percorre todos os elementos do *vetor*, passado como parâmetro para o método, realizando a soma de todos os pesos das diversas ocorrências de um termo com o mesmo texto. Para isso é criada a variável *hashTable*, na qual é adicionada apenas um elemento para cada termo distinto com a soma dos seus pesos. Durante esse processo é identificado o maior peso consolidado dentre os termos distintos e armazenado na variável *pesoMaximo*.

O próximo passo do algoritmo verifica se *normalize* ou *pesoMaximo* é igual a zero e caso essa comparação seja verdadeira é atribuído o valor um para *pesoMaximo*. Essa parte do algoritmo garante que durante a normalização não ocorra divisão por zero.

Em seguida, é convertida a *hashTable* para o formato de retorno do algoritmo, obtendo um *vetorResultante*. Para cada elemento do *vetorResultante* é recuperado o seu peso e dividido pelo *pesoMaximo*, obtendo a média aritmética para cada um dos termos, cujo peso é atualizado para novo valor. Por fim, o algoritmo retorna o *vetorResultante* com a fusão de dados dos termos do *vetor* passado como parâmetro.

**Entrada:** Um vetor de *Weighted\_Object*, um inteiro *normalize*

**Saída:** Um vetor de *Weighted\_Object*

**Início**

*pesoMaximo*  $\leftarrow$  0;

Criar novo *hashTable* vazio;

**Enquanto** não for o final do vetor

**Início**

*term*  $\leftarrow$  *Weighted\_Object*.obterPrimeiroElemento();

**Se** *hashTable*.contem(*term*.obterTexto()) **Então**

*valorDeEntrada*  $\leftarrow$  *term*.obterPeso();

*valorCorrente*  $\leftarrow$  *hashTable*.obterValor(*term*.obterTexto());

*novoValor*  $\leftarrow$  *valorDeEntrada* + *valorCorrente*;

**Se** *novoValor* > *pesoMaximo* **Então**

*pesoMaximo*  $\leftarrow$  *novoValor*;

*hashTable*.remover(*term*.obterTexto());

*hashTable*.colocar(*term*.obterTexto(), *novoValor*);

**Senão**

*hashTable*.colocar(*term*.obterTexto(), *term*.obterPeso());

**Se** *term*.obterPeso() > *pesoMaximo* **Então**

*pesoMaximo*  $\leftarrow$  *term*.obterPeso();

*vetor*.removerPrimeiroElemento();

**Fim**

**Se** *normalize* = 0 ou *pesoMaximo* = 0 **Então**

*pesoMaximo*  $\leftarrow$  1;

Cria *vetorResultante* com os dados de *hashTable*;

**Para** cada elemento do *vetorResultante* **Faça**

*elementoVR*  $\leftarrow$  *vetorResultante*.ObterElementoAtual();

*peso*  $\leftarrow$  *elementoVR*.obterPeso() / *pesoMaximo*;

*elementoVR*.atualizarPeso(*peso*);

**retornar** *vetorResultante*;

**Fim**

Figura 4.2: Algoritmo do método de combinação por *summative union*.

Para converter o resultado das busca do OntoSAIA para o tipo de retorno do método do arcabouço foi criado um método de conversão na classe utilitária *ConverterModels.java*, que possui a seguinte assinatura:

```
public static List<Term> converterWeightedObjectListToTermsList (  
    final Vector sortedTermVector)
```

## Busca Semântica

Este método utiliza o método *SearchRank* do serviço Web *WSOperationsService* do Aondê. Este método é responsável por buscar o termo nas ontologias do repositório especificado como parâmetro. Atualmente o arcabouço está aceitando apenas dois repositórios: o repositório do Aondê e o Swoogle. Na Seção 2.3.4, foram apresentados os objetivos do método e os conceitos envolvidos. Para chamar o serviço Web do Aondê, foi preciso portar a classe com o cliente do serviço, e adaptá-la para a demanda do arcabouço. A classe *WSOperationClient* foi inserida dentro do pacote *ontology\_pckg* e possui um método que chama cada um dos métodos do serviço *WSOperationsService* do Aondê. A assinatura do método que utilizamos é apresentada a seguir.

```
public String[][] SearchRankClient(String[] terms, double[] weights,  
    String[] URLReps, String targetRep) throws Exception
```

O retorno desse método é o identificador da ontologia, a URL do repositório onde ela se encontra e o valor da ontologia utilizada para fazer o *ranking*. A partir desse resultado, é utilizado o método *GetOntology* do serviço Web *WSSemanticRepository*, cuja classe *WSSemanticRepositoryClient* com os respectivos métodos que chamam cada um dos recursos do serviço também foi portada para dentro do pacote *ontology\_pckg*. A assinatura do método cliente que chama o *GetOntology* do serviço Web é mostrada a seguir.

```
public DataHandler GetOntologyClient(String ontName,  
    String[] pathFile) throws Exception
```

O parâmetro *ontName* é o identificador da ontologia no repositório, obtido no momento em que ela foi armazenada no banco. A partir do método *GetOntologyClient* são obtidas cada uma das ontologias, cujo os identificadores foram retornados na chamada do serviço *SearchRank* realizada anteriormente. Para cada uma das ontologias encontradas, é feita a busca dos termos que possuem apenas uma relação direta com o termo pesquisado, e o peso desse termo sugerido é o mesmo peso retornado no serviço *SearchRank* para sua ontologia.

O método do arcabouço de anotação que chama os clientes dos serviços busca os termos para sugestão nas ontologias e monta o resultado final no formato esperado é o *searchByOntology*. A assinatura desse método é apresentada a seguir.

```
public List<List<Term>> searchByOntology(String[] terms,
double[] weights, String[] URLReps, String targetRep)
```

### 4.1.5 Fluxos de Anotação

Conforme mencionado no Capítulo 3, existem dois tipos de fluxos de busca no processo de anotação semântica automática de DCCs de imagem: os fluxos básicos e os fluxos obtidos a partir da combinação dos fluxos básicos. A implementação dos fluxos básicos são os próprios métodos básicos de busca: por texto, por conteúdo, semântica, e por texto e conteúdo. Já para os fluxos compostos, é necessário um método para cada um deles, que ficará responsável por receber a saída de cada um dos fluxos básicos e converter os dados de forma que seja possível passar como parâmetro para o outro método de busca do fluxo. Por exemplo, o fluxo composto de busca por texto e busca semântica possui um método exclusivo para sua execução, que tem a assinatura mostrada a seguir:

```
public List<List<Term>> compositeStreamSearchTextAndSemantic(
List<List<Term>> textTerm, double[] weights,
String[] URLReps, String targetRep)
```

A assinatura desse método é semelhante à de busca semântica, trocando apenas o tipo do parâmetro que contém os termos fornecidos pelo resultado da busca por texto. O método responsável por redirecionar o resultado da busca por texto e fazer a chamada desse método de fluxo composto é o método *controller* descrito na Seção 4.1.3. Os demais fluxos são implementados da mesma forma.

### 4.1.6 Módulo de Atribuição de Peso

A implementação desse módulo é simples, pois todos os serviços de busca já atribuem pesos aos termos sugeridos, assim como os fluxos compostos de anotação. Portanto, apenas os termos sugeridos pelo usuário ainda precisam receber pesos. O peso atribuído a esses termos são definidos em uma propriedade do sistema (arquivo de configuração) e o método *assignmentWeightUserSugestion* da classe *AssignmentWeightModule.java* é o responsável por atribuir o peso e formatar os termos para o retorno. A assinatura desse método é mostrada a seguir.

```
public List<Term> assignmentWeightUserSugestion(final String[] userTerms)
```

### 4.1.7 Módulo de Fusão de Dados

O arcabouço foi projetado de forma que seja possível trocar a técnica utilizada para realizar a fusão de dados, incluindo novos métodos ou chamadas para serviços. Dentre os métodos citados na Seção 2.5, foi escolhido o Borda Count [22], por ser uma técnica básica e bastante referenciada. Em trabalhos futuros pode-se trocar essa técnica e utilizar o arcabouço para comparar sua eficiência e eficácia.

A Figura 4.3 mostra o algoritmo implementado para a fusão das listas com os termos sugeridos, seguindo o método *Borda Count* [5]. Esse algoritmo recebe como parâmetro de entrada uma *lista* de listas de termos.

O primeiro passo do algoritmo é percorrer cada elemento da *lista*, que é uma outra lista com os termos para combinação dos dados. Para realizar a fusão de dados, o algoritmo percorre todos os elementos dessa outra lista e calcula o *valorBorda* para cada um desses elementos. O *valorBorda* consiste no peso Borda calculado pelo método Borda Count [22].

Esse cálculo consiste em subtrair do número de elementos da lista a posição do elemento atual, e multiplicar o resultado pelo peso do termo desse elemento. Existe um *mapa* de chave e valor utilizado para armazenar o termo e o seu peso, auxiliando o processo de soma do *valorBorda* de todas as ocorrências de um termo com o mesmo texto.

Por fim, o *mapa* com o resultado da fusão de dados é convertido para uma única lista de termos e pesos, ordenados decrescentemente por peso, e retornado como resultado do algoritmo.

O método *bordaCount* da classe *RankAggregationModule.java* implementa esse algoritmo. A assinatura desse método é mostrada a seguir.

```
public List<Term> bordaCount(List<List<Term>> sugestinos)
```

### 4.1.8 Módulo de DCC

Conforme apresentado na Seção 2.1.2 e mostrado na Figura 2.2, a estrutura do DCC é composta por quatro partes: metadados, interface, esquema com a estrutura e tipo do dado e o dado em si. Os DCCs criados pelo arcabouço possuem em comum a mesma interface em OWL-S e o mesmo corpo do XMLSchema das imagens, mudando apenas alguns valores como dimensão e formato da imagem. A interface apresenta as informações e recursos que podem ser obtidos daquele DCC. Os métodos disponíveis na interface são:

- *getAnnotation*: não possui parâmetro de entrada e retorna as anotações da imagem encapsulada pelo DCC;
- *getOntologyByAnnotation*: recebe como parâmetro a anotação e retorna a ontologia associada a ela;

**Entrada:** Uma *lista* das listas de termos

**Saída:** Uma lista de termos ordenados

**Início**

**Se** a *lista* não estiver vazia **Então**

Cria um *mapa* vazio com *chave* texto e *valor* decimal;

**Para cada** elementos da *lista* **faça**

*subLista*  $\leftarrow$  Recupera o elemento da *lista*;

*tamanhoSubLista*  $\leftarrow$  Tamanho(*subLista*);

**Para** *i*  $\leftarrow$  0 até *tamanhoSubLista* **faça**

*termo*  $\leftarrow$  *subLista*.obterElemento(*i*);

*valorBorda*  $\leftarrow$  (*tamanhoSubLista* - *i*) \* *termo*.obterPeso();

*anotacao*  $\leftarrow$  converterParaMaiuscula(*termo*.obterAnotacao());

*valor*  $\leftarrow$  *mapa*.obterElementoPorChave(*anotacao*);

**Se** *valor* não existir **Então**

*mapa*.inserir(*anotacao*, *valorBorda*);

**Senão**

*valor*  $\leftarrow$  *valor* + *valorBorda*;

*mapa*.atualizar(*anotacao*, *valor*);

*resultado*  $\leftarrow$  *mapa*.converterParaListaDeTermo();

*resultado*.ordenarLista(decrescente);

**retornar** *resultado*;

**Fim**

Figura 4.3: Algoritmo do Borda Count para a fusão das listas de termos sugeridos.

- *listOntologies*: não possui parâmetro e lista todas as ontologias associadas às anotações;
- *getDescriptor*: não possui parâmetro e retorna todos os descritores da imagem encapsulada pelo DCC;
- *getResolution*: não possui parâmetro e retorna a resolução da imagem encapsulada pelo DCC;
- *getImageFormat*: não possui parâmetro e retorna o formato da imagem encapsulada pelo DCC;
- *getImage*: não possui parâmetro e retorna a imagem encapsulada pelo DCC.

O OWL-S descreve esses métodos de modo semelhante ao WSDL dos serviços Web, mas utilizando ontologia para disponibilizar esses recursos como um serviço Web

semântico. Essa parte dos serviço ainda não foi desenvolvida e é deixada como trabalho futuro.

Os metadados do DCC são as anotações e suas ontologias. As anotações já são armazenadas pelo OntoSAIA, mas não se armazenam referências a ontologias. Por isso, o método responsável pela criação do DCC armazena a anotação e a URI da sua ontologia em uma nova tabela chamada *annotation*. Esta tabela contém os seguintes campos: *ID*, identificador da anotação e chave primária da tabela; *ANNOTATION*, termo da anotação; e *ONTOLOGY*, URL da ontologia associada. Essa tabela possui um relacionamento  $n \times n$  com a tabela *ontosaia* que contém a imagem armazenada. Dessa forma, foi preciso criar mais uma tabela *annotation\_ontosaia*, que faz esse relacionamento. A imagem continua sendo armazenada da mesma forma pelo OntoSAIA. A Figura 4.4 mostra o esquema de dados usando o modelo entidade relacionamento dessas tabelas.

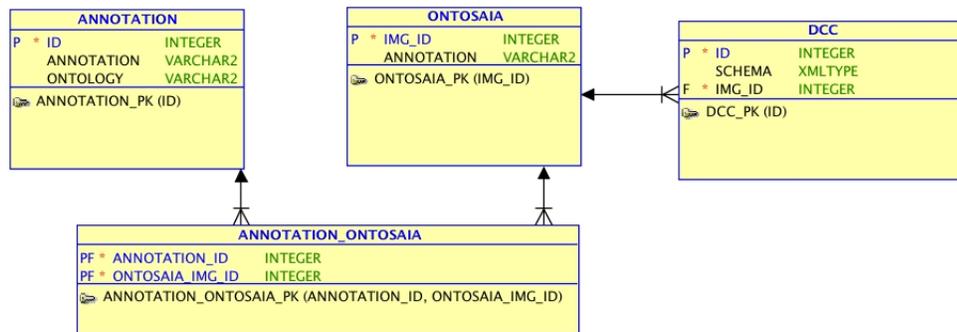


Figura 4.4: Esquema de dados das novas tabelas, criadas para atender o arcabouço de anotação.

A parte que indica que um DCC existe e o relaciona à imagem que ele encapsula e suas anotações está armazenada em uma nova tabela chamada *dcc*, criada dentro da mesma base de dados do OntoSAIA. Essa tabela contém as seguintes colunas: *ID*, que é a chave primária da tabela de DCC; *IMG\_ID*, que é uma chave estrangeira para o identificador da imagem anotada armazenado na tabela *ontosaia*; e uma coluna *SCHEMA*, que guarda o XMLSchema da imagem. Essa tabela também pode ser vista no esquema de dados mostrado na Figura 4.4. Assim, o DCC é gerado e armazenado em uma base de dados relacional.

## 4.2 Estudos de Caso

Nesta seção, são apresentados dois estudos de casos para realizar a comparação das sugestões feitas pelo arcabouço de anotação e visualizar o seu funcionamento. Os cenários de uso apresentados são diferenciados pelos seus dados de entrada, ou seja, se foi fornecida

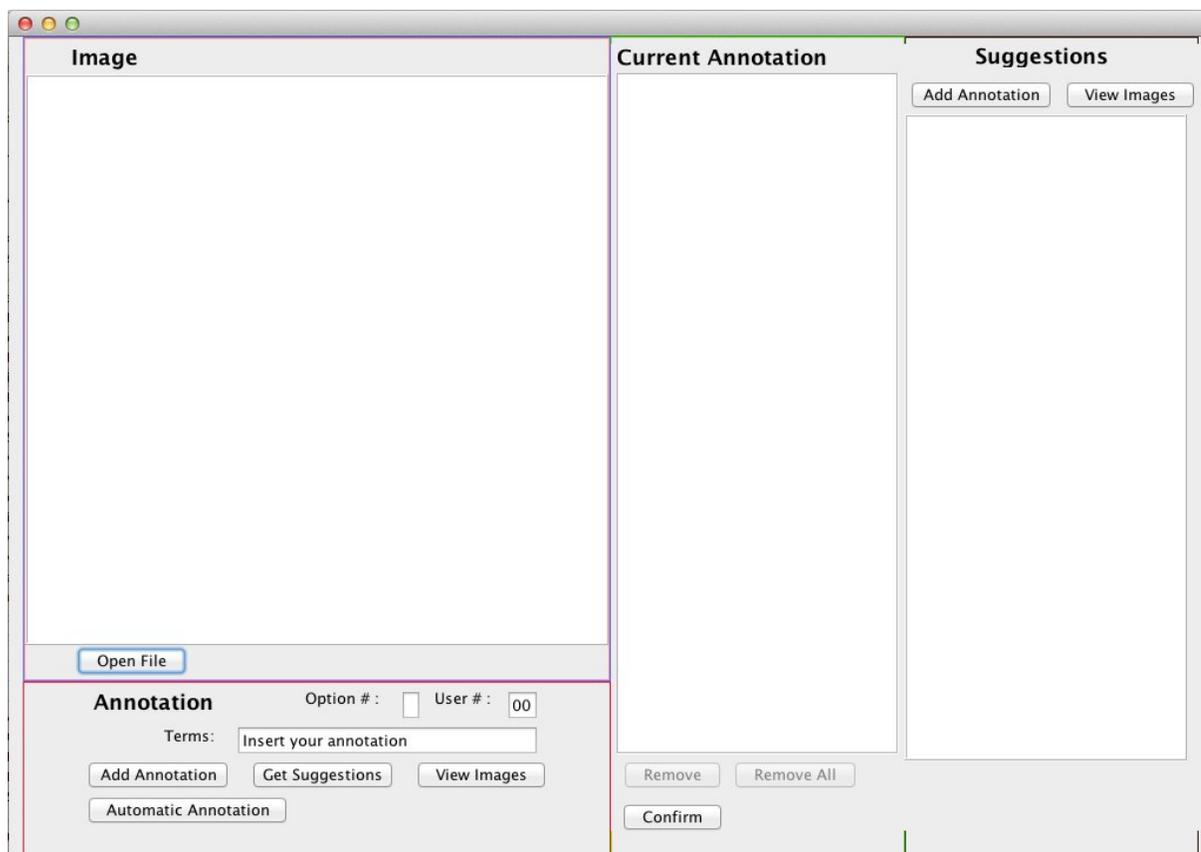


Figura 4.5: Interface do OntoSAIA com o ajuste do arcabouço de anotação.

apenas uma imagem a ser anotada, ou uma imagem e mais alguns termos para anotação. Esses cenários serão chamados de *estudo de caso apenas com imagem* e *estudo de caso com imagem e anotações*. Para realizar os estudos de casos, foram escolhidas imagens de espécies aquáticas, pois o OntoSAIA já possuía uma base de dados com imagens anotadas para esse domínio.

A Figura 4.5 mostra a interface do OntoSAIA, já com o ajuste do arcabouço de anotação: inclusão do botão *Automatic Annotation*. Nesta figura podemos ver no canto superior esquerdo uma janela para carregar e visualizar a imagem. Na parte direita da interface, temos duas colunas: *Current Annotation* e *Suggestions*. *Current Annotation* mostra as anotações escolhidas para aquela imagem, permitindo remover alguma que não seja de interesse ou finalizar o processo de anotação, cadastrando as anotações na base de dados. A coluna *Suggestions* mostra as sugestões resultantes das buscas do OntoSAIA, e permite buscar imagens a partir desses termos ou inclui-los na coluna de *Current Annotation*. A parte inferior esquerda da interface é onde podemos definir a opção de busca por sugestões (*Option #*), se identificar por um número (*User #*), inserir os

termos para busca ou anotação (*Terms*), ou disparar alguma ação utilizando-se os botões disponíveis: *Add Annotation*, *Get Suggestions*, *View Image* ou *Automatic Annotation*. A seguir, são apresentados os dois estudos de casos e no final é realizada uma comparação dos resultados obtidos.

### 4.2.1 Estudo de Caso Apenas com Imagem

O primeiro passo desse cenário consiste em carregar uma imagem de peixe ainda não existente na base, utilizando-se o botão *Open File*, conforme mostra a Figura 4.6.

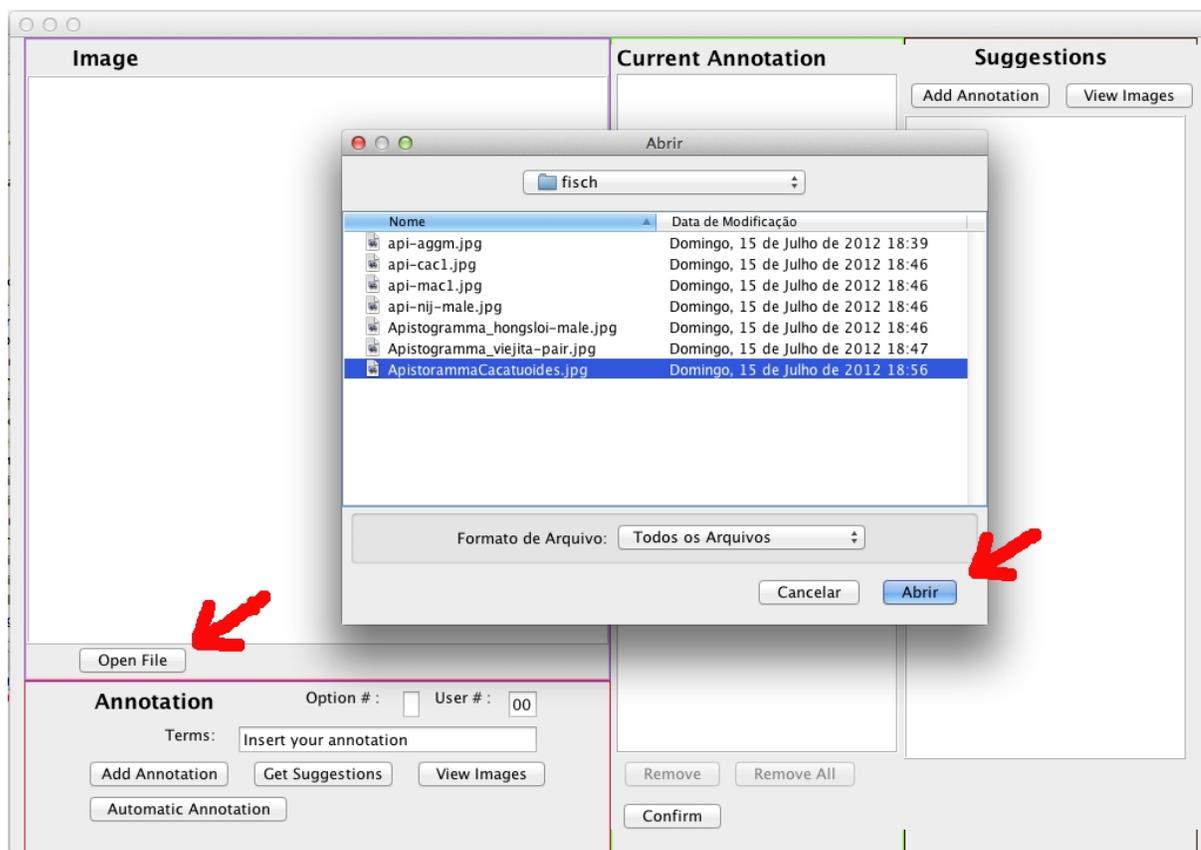


Figura 4.6: Interface OntoSAIA: carregando imagem.

A Figura 4.7 mostra a imagem carregada e o próximo passo desse estudo de caso que é clicar no botão *Automatic Annotation*, para que comece o processo de anotação da imagem e geração do DCC.

No cenário desse estudo de caso não deve ser inserido qualquer termo para anotação, pois as buscas serão realizadas levando em consideração apenas o conteúdo da imagem. Essa busca compara o conteúdo dessa imagem com as demais imagens anotadas anterior-

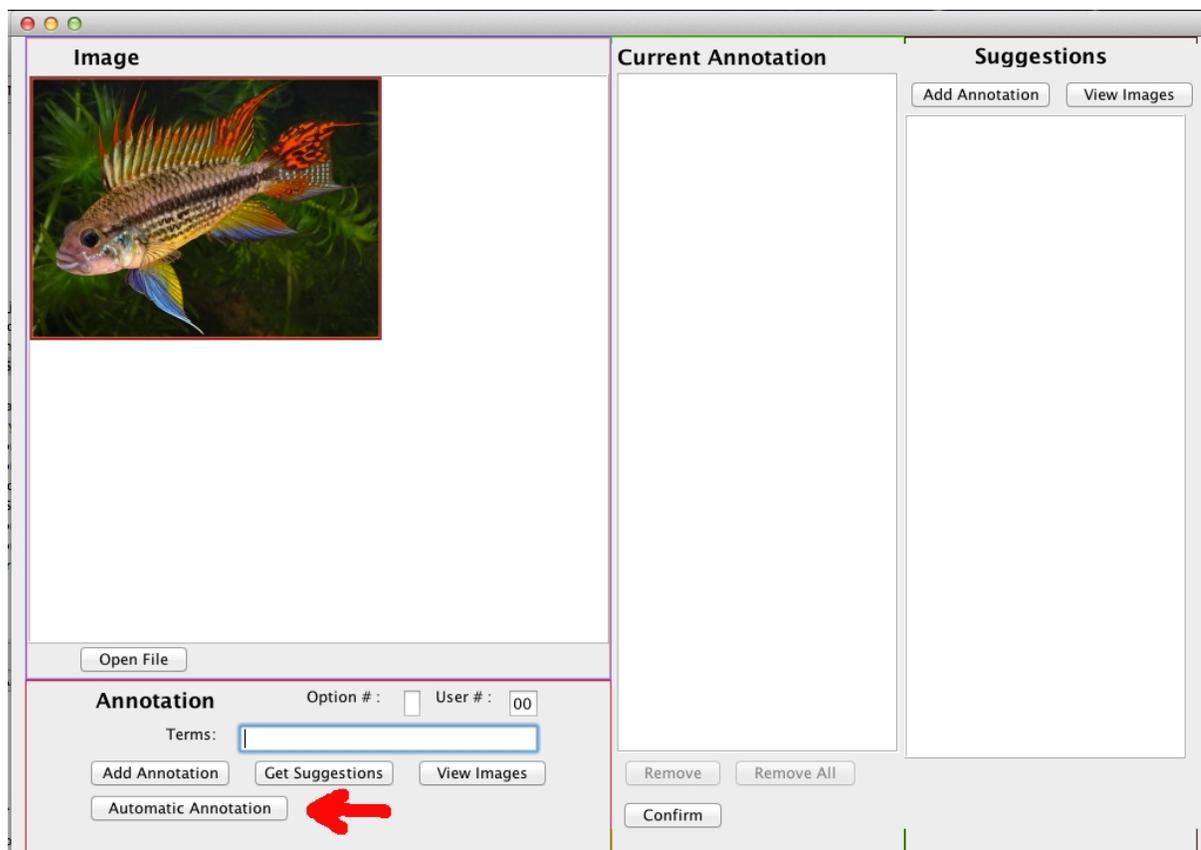


Figura 4.7: Interface OntoSAIA: disparando o processo de anotação de imagem e geração de DCC.

mente na base de dados do OntoSAIA, retornando os termos das imagens que atendem aos critérios de comparação do descritor utilizado, que é o de histograma de cores. Os termos sugerido pela busca por conteúdo são encaminhados para os seguintes fluxos compostos de anotação: fluxo composto de busca conteúdo e busca por texto, e fluxo composto de busca por conteúdo e busca semântica (em ontologias). Como não foram inseridos termos para anotação, o módulo de atribuição de peso não será executado.

Obtidas as listas de termos sugeridos, o processo segue para o módulo de fusão de dados, que utiliza o algoritmo *Borda Count* para combinar os pesos das listas, retornando uma única lista com o resultado consolidado. A lista com os termos sugeridos é filtrada, escolhendo apenas os termos com peso maior que 0.7 (este valor é parametrizado e pode ser ajustado nos arquivos de configuração). Após essa filtragem, é realizada uma busca com cada um dos termos que ainda não faz referência a uma ontologia, no qual o termo esteja presente.

Com isso, temos o resultado final da sugestão: (*South America*, 1.0), (*carnivo-*

*rous*, 0.99775404), (*male*, 0.90038794), (*water*, 0.8933357), (*silver*, 0.87886256), (*small*, 0.7959696) e (*Apistogramma macmasteri*, 0.7108307). Esse resultado é exibido no OntoSAIA na área de sugestões, como mostrado na Figura 4.8. A última etapa é a criação do DCC da imagem carregada e das suas anotações e ontologias sugeridas, finalizando o processo de anotação de imagem e geração do DCC. A tela do OntoSAIA com o resultado final é mostrada na Figura 4.8.

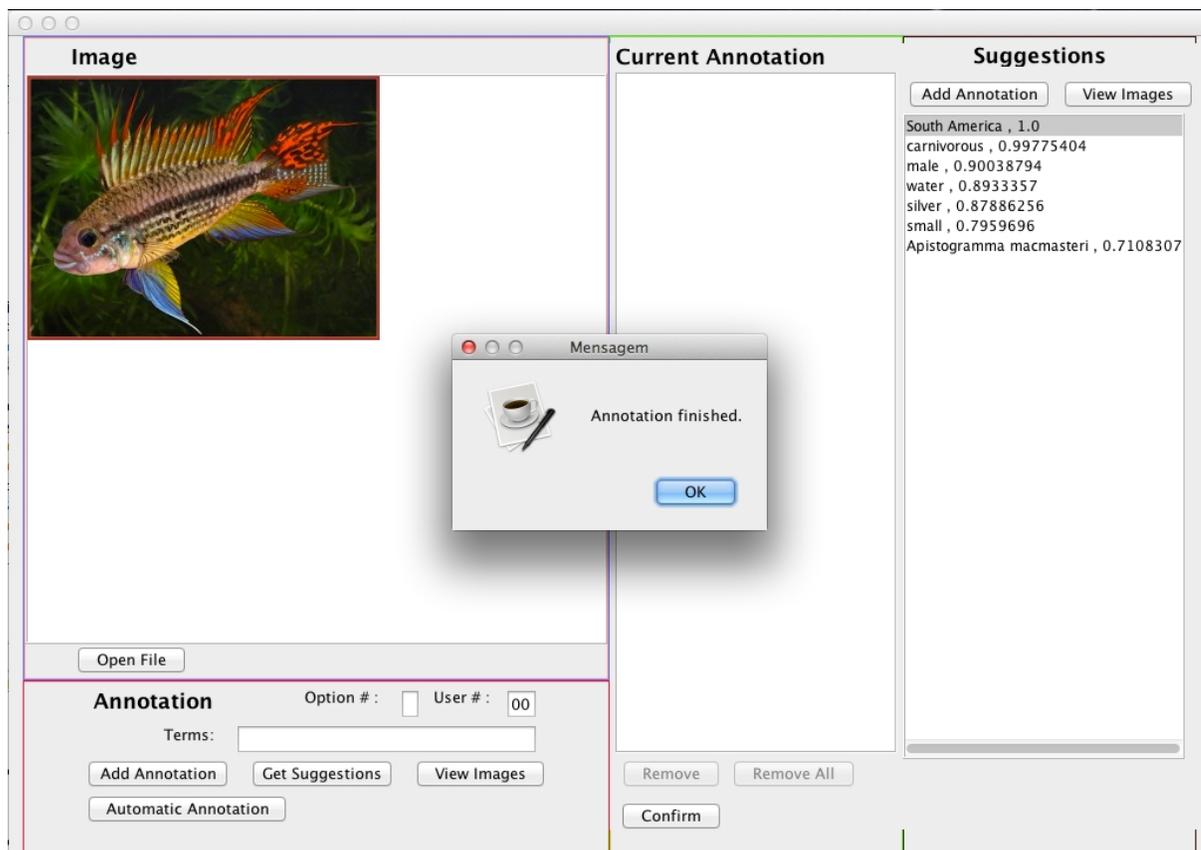


Figura 4.8: Interface OntoSAIA: final do processo de anotação, mostrando os termos anotados e seu pesos.

### 4.2.2 Estudo de Caso com Imagem e Anotação

Na execução desse estudo de caso utiliza-se a mesma imagem do estudo de caso anterior. A primeira etapa, carregamento da imagem, é igual a anterior, mostrada na Figura 4.6. Após o carregamento da imagem são inseridos os seguintes termos na caixa de texto com rótulo *Term: Apistogramma cacatuoides, male*. Em seguida, clica-se no botão *Automatic Annotation* e o processo de anotação de imagem e geração do DCC tem sua execução iniciada. Esses passos de inserir os termos e clicar no botão são exibidos na Figura 4.9.

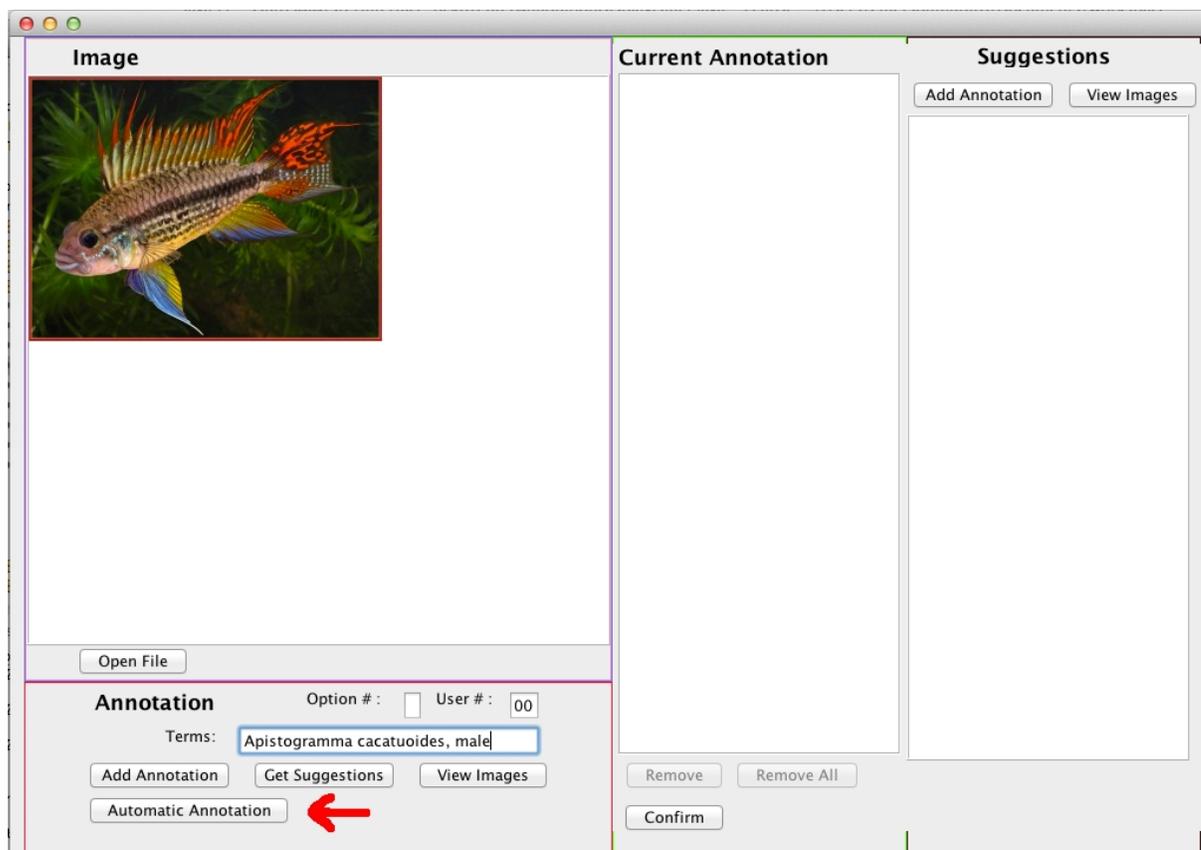


Figura 4.9: Interface OntoSAIA: inserção dos termos e iniciando o processo de anotação de imagem e geração do DCC.

No cenário desse estudo de caso, o processo de busca de termos e anotação de imagem executa todas as buscas e fluxos de anotação do arcabouço de anotação: busca por conteúdo, busca por texto, busca por imagem e texto, busca semântica, e todos os fluxos compostos de busca de termos. Neste estudo de caso, o módulo de atribuição de peso é utilizado para classificar os termos sugeridos pelo usuário. Com as listas de termos sugeridos, o processo segue com a fusão de dados, filtro pelo peso dos termos, busca de ontologias e associação aos termos.

Com isso temos o resultado final da sugestão de termos: (*South America*, 1.0), (*carnivorous*, 0.99775404), (*water*, 0.91877321), (*male*, 0.89787291), (*silver*, 0.87886256), (*small*, 0.7959696), (*Apistogramma macmasteri*, 0.7108307) e (*Apistogramma cactuoides*, 0.70067483), como mostra a Figura 4.10 na janela de *Suggestions*. Por fim, é criado o DCC da imagem carregada e das suas anotações e ontologias sugeridas, finalizando o processo de anotação de imagem e geração do DCC. A tela do OntoSAIA com o resultado desse processo é mostrada na Figura 4.10.

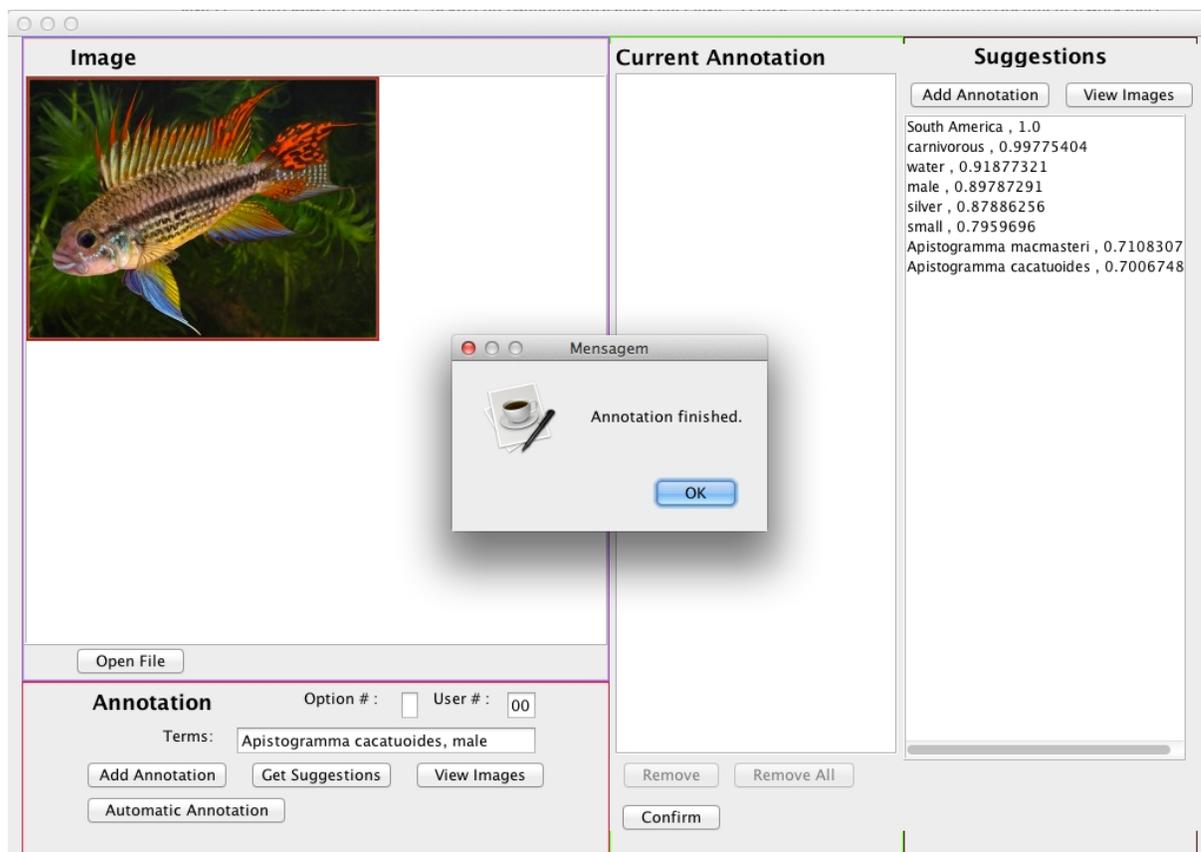


Figura 4.10: Interface OntoSAIA: final do processo de anotação, mostrando os termos anotados e seu pesos.

A Figura 4.11 mostra de uma forma simplificada o DCC gerado neste último estudo de caso, no qual podemos ver a imagem, o local da estrutura XML, uma parte da interface e uma parte dos metadados desse DCC.

A interface mostrada na Figura 4.11 mostra alguns dos recursos disponíveis: *getAnnotation*, *listOntologies*, *getDescription* e *getImage*. Com relação ao recurso *getAnnotation* é detalhado um pouco mais a sua interface, mostrando qual é a sua saída: uma anotação semântica contendo o termo *Apistogramma cactuoides* e o link da ontologia associada a esse termo.

A parte de metadado mostra com detalhes um dos metadados do DCC, que é na verdade a sua anotação semântica. Ela é semelhante a anotação semântica descrita na interface e ainda possui o identificador da sua imagem (*id.imagem*) para consultas e referências futuras.

O resultado do primeiro estudo de caso mostrado na Figura 4.8 trouxe sete termos: (*South America*, 1.0), (*carnivorous*, 0.99775404), (*male*, 0.90038794), (*water*, 0.8933357),

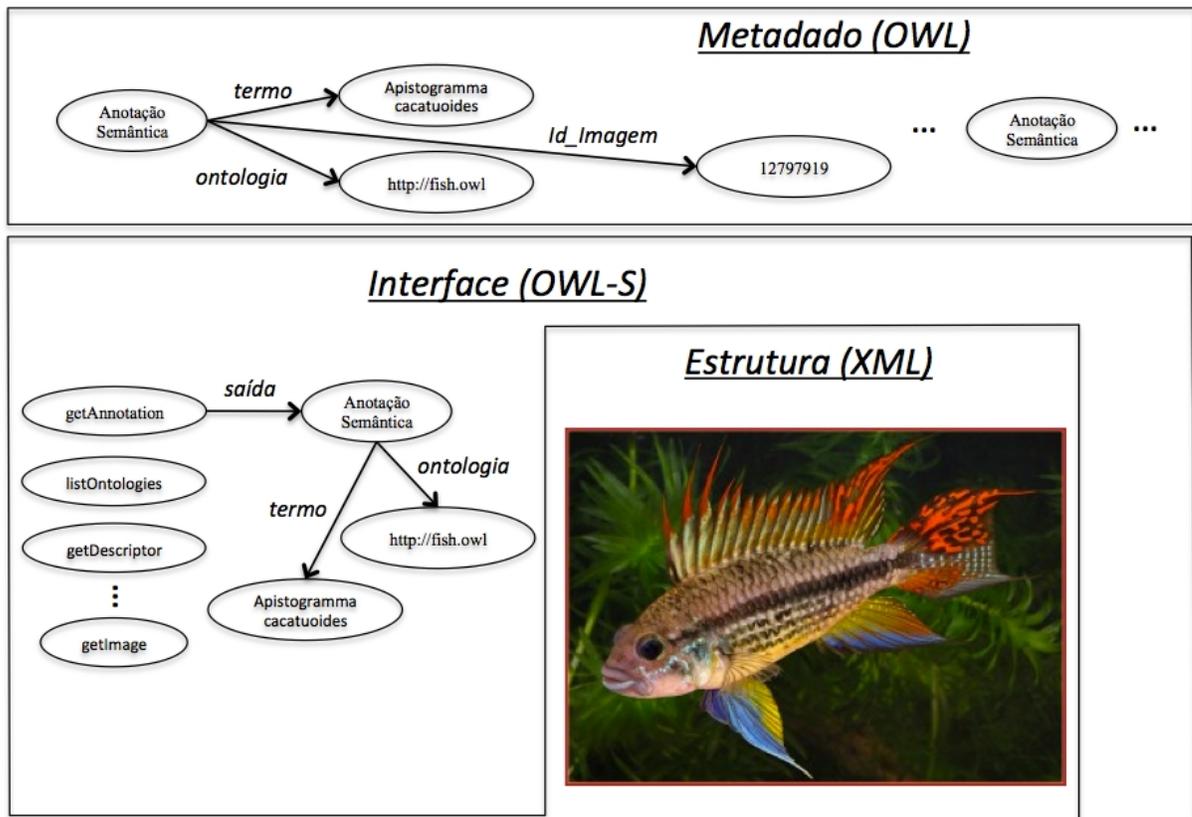


Figura 4.11: Representação simplificado do DCC de imagem anotado semanticamente gerado pelo arcabouço de anotação.

(*silver*, 0.87886256), (*small*, 0.7959696) e (*Apistogramma macmasteri*, 0.7108307). Esses termos refletem uma base de dados com poucas imagens anotadas, pois é a partir das anotações das imagens com cores semelhantes, armazenadas na base de dados, que são disparadas as demais busca por sugestões. Isso faz com que o peso desses termos caiam ao serem combinados pelo algoritmo de fusão de dados. Conseqüentemente, são retornados menos termos devido ao filtro de peso mínimo (maior que 0.7).

Por outro lado, temos o resultado do segundo estudo de caso: (*South America*, 1.0), (*carnivorous*, 0.99775404), (*water*, 0.91877321), (*male*, 0.89787291), (*silver*, 0.87886256), (*small*, 0.7959696), (*Apistogramma macmasteri*, 0.7108307) e (*Apistogramma cactuoides*, 0.70067483). Esse resultado ainda está associado a uma base de dados com poucas anotações em imagens. Contudo, considerando-se os termos sugeridos pelo usuário, o arcabouço de anotação pode utilizar todos os seus recursos de busca por termos.

A classificação dos termos sugeridos pelo usuário e a contribuição dos termos sugeridos pelas buscas, por texto e semântica, a partir dos termos do usuário conseguem influenciar o resultado final. Se as duas listas de termos resultantes são comparadas, percebe-se uma

inversão de ordem entre os pesos *male* e *water*. Apesar do usuário ter sugerido *male*, o termo *water* deve aparecer bastante nas anotações e ontologias, e nos resultados de mais de um dos termos sugeridos pelo usuários, fazendo com que ele tenha preferência em relação ao outro termo. Uma análise desse fato indica que mais pessoas utilizaram esse termo em anotações e na criação das ontologias (próximo aos outros termos do usuários), sinalizando que esse termo pode ser mais relevante que um termo sugerido pelo usuário.

No primeiro estudo de caso, não havia sido retornado o termo *Apistogramma cactuoides*. Essa é mais uma influência da ação do usuário, confirmando a sua expectativa do termo *Apistogramma cactuoides* fazer sentido como anotação. Isso é ilustrado a partir da elevação do seu peso obtida pela fusão de dados das seguintes sugestões para o termo *Apistogramma cactuoides*: sugerido pelo usuário, pelas buscas por texto, pela busca por conteúdo, e pela busca semântica.

# Capítulo 5

## Conclusões e Trabalhos Futuros

### 5.1 Contribuições

Grandes coleções de imagem vêm sendo criadas recentemente. Imagens são comumente combinadas com outros tipos de dados, permitindo a criação de objetos complexos. Este cenário faz surgir a demanda por novos serviços de busca que explorem diferentes modalidades (por exemplo, conteúdo visual e anotações textuais), assim como descrições enriquecidas semanticamente (baseadas em ontologias).

Nessa pesquisa, apresenta-se um arcabouço para anotação semântica automática de imagens encapsuladas em componentes. O arcabouço proposto está dividido em quatro camadas: interface, controle, serviços e repositórios.

A camada de interface permite acesso as funcionalidades do serviço: anotações de imagens, avaliação das anotações, busca e visualização de componentes de dados.

A camada de controle é responsável por controlar o fluxo de execução do processo automatizado de anotação semântica de DCCs de imagens. Esta camada realiza e controla todas as chamadas para serviços e métodos do arcabouço, permitindo a realização de: buscas por termos como sugestões para anotações, atribuição de peso aos termos do usuários (se eles existirem), fusão dos dados das lista de termos sugeridos, filtragem de termos de acordo com sua importância (pesos), e por fim, criação e armazenamento do DCC de imagem semanticamente anotado.

A terceira camada é a dos serviços utilizados pelo arcabouço. Dentre eles, estão os serviços das duas ferramentas utilizada como base na construção do arcabouço: OntoSAIA e Aondê. Com relação ao OntoSAIA, são utilizados os serviços relacionados ao gerenciamento de imagens: busca por conteúdo e textual, comparação de imagens e armazenamento. Da outra ferramenta, Aondê, são utilizados os serviços de busca, *ranking* e armazenamento de ontologias.

A última camada do arcabouço de anotação refere-se aos repositórios utilizados: re-

positório do Aondê para ontologias, repositório do OntoSAIA para armazenamento de imagens, anotações e metadados, repositório para os DCCs de imagens criados e os repositórios na Web de ontologias.

Esse trabalho apresenta as seguintes contribuições:

- especificação do processo de anotação semântica automática em DCCs de imagens, cujo os fluxos estão divididos em dois tipos:
  - Fluxos de Busca, responsável por obter as sugestões de termos para anotação, dividido em:
    - \* Fluxos Básicos de Busca, que consistem em quatro algoritmos, respectivamente: Busca por Texto, Busca por Conteúdo e Texto e Busca Semântica;
    - \* Fluxos Compostos de Busca, que são algumas combinações dos fluxos básicos de busca:
      - Fluxos de Busca por Conteúdo e Busca Semântica;
      - Fluxos de Busca por Texto e Busca Semântica;
      - Fluxos de Busca por Conteúdo e por Texto e Busca Semântica;
      - Fluxos de Busca semântica e Busca por texto;
      - Fluxos de Busca Semântica e Busca por Conteúdo e por Texto.
  - Fluxo de Processamento das Anotações, responsável por realizar as etapas de Atribuição de Peso, Fusão de Dados e Geração do DCC de imagem;
- especificação de um arcabouço para anotação de componentes de imagem, com o intuito de atender o processo de anotação semântica automática de DCCs de imagem, contribuindo com:
  - especificação do módulo de recomendação de termos configurável e flexível, ou seja, cada um dos serviços (incluindo os que compõem os fluxos compostos) pode ser trocado por serviços semelhantes, ou por algoritmos mais apropriados;
  - especificação do módulo *Ranking Aggregation*, que realiza a Fusão de Dados. Embora apenas o método *Borda Count* tenha sido utilizado nessa dissertação, a especificação desse serviço é genérica, o que permite o uso de métodos diferentes de fusão de dados [50];
  - especificação do módulo Gerador de DCC, responsável por anotar semanticamente a imagem, encapsulando-os como um DCC.
- implementação parcial do arcabouço de anotação.

## 5.2 Trabalhos Futuros

Esta seção apresenta algumas opções de trabalhos futuros para o arcabouço de anotação semântica de objetos complexos de imagens encapsulados em DCCs. A seguir, são listadas essas opções.

- Migração do OntoSAIA e os métodos implementados para serviços Web e finalizar a implementação do arcabouço. Para a interface, o ideal seria uma aplicação Web implementada com mecanismos de controle de usuário. Atualmente, o arcabouço possui um mecanismo de controle de usuário, herdado do OntoSAIA, que pode ser facilmente burlado, afetando a qualidade das métricas extraídas a partir dessa informação. O objetivo é permitir a validação do arcabouço de anotação com um grande número de usuários e com acessos simultâneos.
- Estender o serviço Web Aondê para realizar busca em mais repositórios na Web além do seu e do *Swoogle*<sup>1</sup>, caracterizando uma extensão do Aondê e validada pelo arcabouço de anotação.
- Evoluir a busca por conteúdo, incluindo mais descritores de imagem. Ao final do processo, o resultado passaria por uma fusão de dados para consolidar a lista de termos sugeridos. O ideal seria incluir os descritores e realizar estudos de caso (ou com usuários) para validar sua eficiência e eficácia.
- Realizar uma pesquisa por técnicas mais recentes de fusão de dados. Os trabalhos [49–52] mostram outras técnicas promissoras de Fusão de Dados, com as quais poderiam ser criados serviços Web de fusão de dados e utilizar o arcabouço para validar esses serviços.
- Realizar estudos com algoritmos de busca em ontologias e com técnicas para recomendação de termos baseadas em ontologias. Pesquisar como fazer isso de forma a aproveitar as diferentes associações e conceitos definidos nas ontologias.
- Investigar o uso do arcabouço em aplicações voltadas para a criação de bibliotecas digitais de objetos complexos. Um ponto de partida seria investigar o uso do arcabouço dentro da infraestrutura proposta em [36].
- As anotações são representadas por um termo associado a uma ontologia. Uma melhoria seria realizar um estudo de como utilizar padrões de metadados para representação de anotações semânticas. Pro exemplo, como utilizar o padrão *Dublin Core*<sup>2</sup> para representação de uma anotação semântica.

---

<sup>1</sup><http://swoogle.umbc.edu/> (último acesso em junho de 2012).

<sup>2</sup><http://dublincore.org/documents/dces/> (último acesso em junho de 2012).

- Criar um serviço Web para obter as imagens e suas anotações a partir de uma base de dados, e executar o processo de anotação semântica automática de DCCs de imagem para cada uma delas. Esse serviço seria incorporado ao fluxo de trabalho do processo de anotação.

Considerando-se as sugestões acima, verifica-se que o arcabouço funciona, de certa maneira, como um sistema integrado que permite indiretamente a realização de comparações e validações de alguns tipos de recursos: mecanismos de busca, técnicas de fusão de dados, estruturas de componentes de dados. Nesse sentido, sugere-se a investigação de como evoluir o arcabouço para se tornar uma ferramenta de validação desses tipos de recursos citados anteriormente.

A proposta inicial desta pesquisa foi publicada no Workshop de Teses e Dissertações de Banco de Dados (WTDBD) do Simpósio Brasileiro de Banco de Dados (SBBD) de 2010.

# Referências Bibliográficas

- [1] M. Agosti and N. Ferro. A formal model of annotations of digital content. *ACM Transactions on Information Systems (TOIS)*, 26(1):1–57, November 2007. Article No. 3.
- [2] H. Alani, C. Brewster, and N. Shadbolt. Ranking ontologies with aktiverank. In *Proceedings of the 5th international conference on The Semantic Web, ISWC'06*, pages 1–15, Berlin, Heidelberg, 2006. Springer-Verlag.
- [3] S. André and M. Claudia. Managing dynamic repositories for digital content components. In Wolfgang Lindner, Marco Mesiti, Can Türker, Yannis Tzitzikas, and Athena Vakali, editors, *Current Trends in Database Technology - EDBT 2004 Workshops*, volume 3268 of *Lecture Notes in Computer Science*, pages 368–373. Springer Berlin / Heidelberg, 2005.
- [4] M. Araújo and M. A. G. V. Ferreira. Construindo uma ontologia para pesquisa de materiais e objetos de aprendizagem baseada na web semântica. In *XV Simpósio Brasileiro de Informática na Educação, SBIE '04*, pages 388–397, Manaus, Brazil, 2004.
- [5] J. A. Aslam and M. Montague. Models for metasearch. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '01*, pages 276–284, New York, NY, USA, 2001. ACM.
- [6] D. Beckett and J. Broekstra. SPARQL query results XML format. Technical report, World Wide Web Consortium - W3C, April 2006.
- [7] T. Berners-Lee. The semantic web. *Scientific American*, May 2001.
- [8] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau. *Extensible Markup Language (XML) 1.0*. W3C, fifth edition, November 2008. URL: <http://www.w3.org/TR/REC-xml/>.

- [9] Erik Christensen, Francisco Curbera, Greg Meredith, and Sanjiva Weerawarana. *Web Services Description Language (WSDL) 1.1*. W3C, 1.1 edition, March 2001. URL: <http://www.w3c.org/TR/wsdl>.
- [10] L. Clement, A. Hatley, C. Riegen, and T. Rogers. *UDDI Version 3.0.2*. OASIS, October 2004. URL: [http://uddi.org/pubs/uddi\\_v3.htm](http://uddi.org/pubs/uddi_v3.htm).
- [11] W. W. Cohen, P. Ravikumar, and S. E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *Proceedings of IJCAI-03 Workshop on Information Integration*, pages 73–78, August 2003.
- [12] R. D., D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (CSUR)*, 40(2):1–60, 2008.
- [13] R. da S. Torres and A. X. Falcão. Content-based image retrieval: Theory and applications. *Revista de Informática Teórica e Aplicada (RITA)*, 13(2):161–185, 2006.
- [14] R. da S. Torres, C. B. Medeiros, M. A. Goncalves, and E. Fox. A digital library framework for biodiversity information systems. *International Journal on Digital Libraries*, 6(1):3–17, 2006. Springer Berlin / Heidelberg.
- [15] J. Daltio. Aondê: Um serviço web de ontologias para interoperabilidade em sistemas de biodiversidade. Master’s thesis, Instituto de Computação - Unicamp, August 2007.
- [16] J. Daltio and C. B. Medeiros. Um servidor de ontologias para apoio a sistemas de biodiversidade (an ontology server to support biodiversity information systems). In *V Workshop Thesis and Dissertations on Databases, Proc. XXI Brazilian Symposium on Databases - SBBD*, SBBD ’06, pages 71–76, Florianópolis, Brazil, October 2006.
- [17] H. C. de Souza, A. M. de C. Moura, and M. C. Cavalcanti. Ontologias emergentes: uma nova abordagem para integração de ontologias. In *Proceedings of the 23rd Brazilian symposium on Databases*, number 15 in SBBD ’08, pages 165–179, Porto Alegre, Brazil, Brazil, 2008. Sociedade Brasileira de Computação.
- [18] C. G. do N. Macário, J. A. dos Santos, C. B. Medeiros, and R. da S. Torres. Annotating data to support decision-making: a case study. In *Proceedings of the 6th Workshop on Geographic Information Retrieval*, GIR ’10, pages 1–7, New York, NY, USA, 2010. ACM.
- [19] Carla Geovana do Nascimento Macário. *Anotação Semântica de Dados Geoespaciais*. PhD thesis, Instituto de Computação - Unicamp, December 2009.

- [20] J. A. dos Santos, P. H. Gosselin, S. P. Foliguuet, R. da S. Torres, and A. X. Falcão. Multi-scale classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 50(10):3764–3775, 2012.
- [21] P. Dourado, P. Ferreira, and A. Santanchè. Representação Unificada de Objetos Digitais Complexos: confrontando o RAS com o IMS CP. In *III Workshop on Digital Libraries*, pages 123–129, October 2007.
- [22] C. Dwork, R. Kumar, Moni M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th international conference on World Wide Web, WWW '01*, pages 613–622, New York, NY, USA, 2001. ACM.
- [23] D. C. Fallside and P. Walmsley. *XML Schema Part 0: Primer Second Edition*. W3C, second edition, October 2004. URL: <http://www.w3.org/TR/xmlschema-0/>.
- [24] F. F. Faria, A. Veloso, H. M. Almeida, E. Valle, R. da S. Torres, M. A. Gonçalves, and W. Meira. Learning to rank for content-based image retrieval. In *Proceedings of the international conference on Multimedia information retrieval, MIR '10*, pages 285–294, New York, NY, USA, 2010. ACM.
- [25] R. Fileto. *The POESIA Approach for the Integration of Data and Services in the Semantic Web*. PhD thesis, Instituto de Computação - Unicamp, December 2003.
- [26] R. B. Freitas and R. da S. Torres. Ontosaia: An ontology-based image search and semi-automatic annotation tool. In *I Workshop in Digital Libraries, Proc. XX Brazilian Symposium on Databases*, pages 60–69, 2005.
- [27] J. H. Gennari, M. A. Musen, R. W. Fergerson, W.E. Grosso, M. Crubézy, H. Eriksson, N. F. Noy, and S. W. Tu. The evolution of Protégè: an environment for knowledge-based systems development. *International Journal of Human-Computer Studies*, 58(1):89–123, January 2003.
- [28] F. B. Gil. Serviço web para anotação de dados geográficos vetoriais e sua aplicação em sistemas de informação de biodiversidade. Master's thesis, Instituto de Computação - Unicamp, December 2009.
- [29] F. B. Gil, N. P. Kozievitch, and R. da S. Torres. Geonote: A web service for geographic data annotation in biodiversity information systems. *JOURNAL OF INFORMATION AND DATA MANAGEMENT - JIDM*, 2(2):195–210, 2011.
- [30] L. C. Gomes. Uma arquitetura para consultas a repositórios de biodiversidade na web. Master's thesis, Instituto de Computação - Unicamp, May 2007.

- [31] T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies*, 43(5-6):907–928, 1995.
- [32] M. Gudgin, M. Hadley, N. Mendelsohn, J. J. Moreau nad H. F. Nielsen, A. Karmarkar, and Y. Lafon. *SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)*. W3C, 1.2 edition, April 2007. URL: <http://www.w3.org/TR/soap12-part1/>.
- [33] K. Hirata and T. Kato. Query by visual example - content based image retrieval. In *Proceedings of the 3rd International Conference on Extending Database Technology: Advances in Database Technology, EDBT '92*, pages 56–71, London, UK, UK, 1992. Springer-Verlag.
- [34] G. Z. Pastorello Jr, C. B. Medeiros, and A. Santanchè. Applying scientific workflows to manage sensor data. In *XXII Brazilian Symposium on Databases - SBBD*, pages 9–18, October 2007.
- [35] J. Kim, S. Kwon, and Y. Park. Enhanced search method for ontology classification. In *Proceedings of the 2008 IEEE International Workshop on Semantic Computing and Applications, IWSCA '08*, pages 12–18, Washington, DC, USA, 2008. IEEE Computer Society.
- [36] N. P. Kozievitch. *Objetos complexos em bibliotecas digitais : analisando o gerenciamento de componentes de imagens*. PhD thesis, Instituto de Computação - Unicamp, July 2011.
- [37] N. P. Kozievitch, J. Almeida, R. da S. Torres, A. Santanchè, and N. J. Leite. Reusing a compound-based infrastructure for searching video stories. In *Proceedings of the IEEE International Conference on Information Reuse and Integration*, pages 222–227, 2011.
- [38] N. P. Kozievitch, R. da S. Torres, F. Andrade, U. Murthy, E. A. Fox, and E. Halerman. A teaching tool for parasitology: Enhancing learning with annotation and image retrieval. In *14th European Conference on Research and Advanced Technology for Digital Libraries*, pages 466–469, 2010.
- [39] N. P. Kozievitch, R. da S. Torres, S. H. Park, E. A. Fox, N. Short, L. Abbott, S. Misra, and M. Hsiao. Rethinking fingerprint evidence through integration of very large digital libraries. In *Third Workshop on Very Large Digital Libraries, 14th European Conference on Research and Advanced Technologies on Digital Libraries*, page 8, Glasgow, Scotland, 2010.

- [40] N. P. Kozievitch, R. da S. Torres, A. Santanchè, D. C. G. Pedronette, R. T. Calumby, and E. A. Fox. An infrastructure for searching and harvesting complex image objects. *The Information - Interaction - Intelligence (I3) Journal*, 11(2):39–68, 2011.
- [41] D. B. Krafft, A. Birkland, and E. J. Cramer. Ncore: architecture and implementation of a flexible, collaborative digital library. In *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '08, pages 313–322, New York, NY, USA, 2008. ACM.
- [42] Y. Liu, T. Liu, T. Qin, Z. Ma, and H. Li. Supervised rank aggregation. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 481–490, New York, NY, USA, 2007. ACM.
- [43] C. G. N. Macário, S. R. de Sousa, and C. B. Medeiros. Annotating geospatial data based on its semantics. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '09, pages 81–90, New York, NY, USA, November 2009. ACM.
- [44] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan, and K. Sycara. *OWL-S: Semantic Markup for Web Services*. W3C, November 2004. URL: <http://www.w3.org/Submission/OWL-S/>.
- [45] D. L. McGuinness and F. van Harmelen. *OWL Web Ontology Language Overview*. W3C, February 2004. URL: <http://www.w3.org/TR/owl-features/>.
- [46] U. Murthy, L. Tzy Li, E. Hallerman, E. A. Fox, M. A. P. Quiñones, L. M. L. Delcambre, and R. da S. Torres. Use of subimages in fish species identification: a qualitative study. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, JCDL '11, pages 185–194, New York, NY, USA, 2011. ACM.
- [47] Ontotext Lab. *The KIM Platform: Knowledge & Information Management*, 2007. <http://www.ontotext.com/kim/semantic-annotation>.
- [48] G. Z. Pastorello. Publication and integration of scientific workflows on the web. Master's thesis, Instituto de Computação - Unicamp, April 2005.
- [49] D. Pedronette and R. Torres. Exploiting contextual information for image re-ranking and rank aggregation. *International Journal of Multimedia Information Retrieval*, 1(2):115–128, 2012.

- [50] D. C. G. Pedronette. *Exploiting Contextual Information for Image Re-Ranking and Rank Aggregation in Image Retrieval Tasks*. PhD thesis, Instituto de Computação, Universidade Estadual de Campinas, 2012.
- [51] D. C. G. Pedronette and R. da S. Torres. Exploiting pairwise recommendation and clustering strategies for image re-ranking. *Information Sciences: an International Journal*, 207(0):19–34, 2012.
- [52] D. C. G. Pedronette, R. da S. Torres, and R. T. Calumby. Using contextual spaces for image re-ranking and rank aggregation. *Multimedia Tools and Applications*, 65(0):1–28, May 2012.
- [53] T. Pilioura and A. Tsalgatidou. Unified publication and discovery of semantic Web services. *ACM Transactions on the Web (TWEB)*, 3(3):1–44, June 2009.
- [54] E. Prud'hommeaux and A. Seaborne. SPARQL query language for RDF. Technical report, World Wide Web Consortium - W3C, January 2008.
- [55] M. Pujari and R. Kanawati. Supervised rank aggregation approach for link prediction in complex networks. In *Proceedings of the 21st international conference companion on World Wide Web, WWW '12 Companion*, pages 1189–1196, New York, NY, USA, 2012. ACM.
- [56] M. E. Renda and U. Straccia. Web metasearch: rank vs. score based rank aggregation methods. In *Proceedings of the 2003 ACM symposium on Applied computing, SAC '03*, pages 841–846, New York, NY, USA, 2003. ACM.
- [57] A. Santanchè. *Self Describing Components: Searching for Digital Artifacts on the Web*. PhD thesis, Instituto de Computação - Unicamp, 2006.
- [58] A. Santanchè and C. B. Medeiros. Geographic digital content components. In *VI Brazilian Symposium on GeoInformatics - GeoInfo*, pages 281–290, November 2004.
- [59] A. Santanchè and C. B. Medeiros. Self describing components: Searching for digital artifacts on the web. In *XX Brazilian Symposium on Databases*, pages 10–24, October 2005.
- [60] A. Santanchè and C. B. Medeiros. A component model and an infrastructure for the fluid web. *IEEE Transactions on Knowledge and Data Engineering*, 19(2):324–341, February 2007.
- [61] A. Santanchè, C. B. Medeiros, and G. Pastorello. User-author centered multimedia building blocks. *Multimedia Systems*, 12(4):403–421, 2007.

- [62] W. Siberski, J. Z. Pan, and U. Thaden. Querying the semantic web with preferences. In *Proceedings of the 5th international conference on The Semantic Web, ISWC'06*, pages 612–624, Berlin, Heidelberg, 2006. Springer-Verlag.
- [63] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, November 1991.
- [64] W3C. *RDF/XML Syntax Specification (Revised)*, revised edition, February 2004. URL: <http://www.w3.org/TR/rdf-syntax-grammar/>.
- [65] B. Zhou and T. Huang. Semantic web service discovery search with ontology learning. *Computer Science and Software Engineering, International Conference on*, 5(0):1048–1051, December 2008.