IDENTIFICAÇÃO DE REUSO EM DOCUMENTOS DIGITAIS

Fellipe Ribeiro Duarte

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Geraldo Bonorino Xexéo

Rio de Janeiro
Junho de 2011

IDENTIFICAÇÃO DE REUSO EM DOCUMENTOS DIGITAIS

Fellipe Ribeiro Duarte

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

_____

Prof. Geraldo Bonorino Xexéo, D.Sc.

_____

Prof. Jano Moreira de Souza, Ph.D.

_____

Profa. Karin Koogan Breitman, D. Sc.

RIO DE JANEIRO, RJ – BRASIL
JUNHO DE 2011

Duarte, Fellipe Ribeiro

Identificação de Reuso em Documentos Digitais / Fellipe Ribeiro Duarte – Rio de Janeiro: UFRJ/COPPE, 2011.

XII, 115 p.: il.; 29,7 cm.

Orientador: Geraldo Bonorino Xexéo.

Dissertação (Mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2011.

Referências Bibliográficas: p. 105-111.

1. Extração de informação. 2. Engenharia de Documentos. 3.Reuso de documentos. 4. Reuso de conteúdo. I. Xexéo, Geraldo Bonorino II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

À minha família, pelo incentivo, carinho e compreensão.

Agradecimentos

Agradeço a Deus, por me capacitar e guiar em todos os desafios que sou submetido.

À minha mãe, Denizá, pelo incentivo, carinho, apoio e pelos seus paparicos que sempre me motivaram a prosseguir.

Ao meu pai, Carlos, pelo carinho, compreensão e por ser sempre solícito a todos, o que sempre me causou admiração.

À minha irmã Andréa, minha eterna melhor amiga.

Ao meu sobrinho Lucas, por compreender que, em alguns momentos, o tio não podia brincar, pois estava estudando.

Ao meu orientador Prof. Geraldo Bonorino Xexéo, pelo tempo, confiança e compromisso depositados para garantir a qualidade final desta dissertação. Espero que este seja apenas o primeiro de muitos outros trabalhos que possamos realizar juntos.

Aos amigos Abel Carneiro, Alexandre Rabello, Rogério Almeida, Raquel Lemos, Carlos Wagner, Glauber Marcius, Marcelo Bueno, Rafael Espirito Santo, Eliza Chaves e Lucio Paiva, Rodrigo Barboza que acompanharam e apoiaram esta saga, mesmo no momento em que não fui muito presente.

Ao Prof. Jano Moreira de Souza, pela oportunidade concedida para a realização do mestrado na linha de Banco de dados e por aceitar participar da minha banca de defesa de mestrado.

À professora Karin Koogan Breitman, por participar da minha banca de defesa de mestrado.

Aos amigos que fiz no PESC, Itamar, Adilson, Cláudia Prata, Solange, Sônia, Ari, Nathalia, Gutierrez, Juliana, João Victor, Júlio Cesar, Mercedes, Eliah, Taísa, Patrícia Leal, Ana Paula Rabello, Maria de Lourdes, Alexandre Vieira e Thiago da Rocha, pelo apoio durante esta etapa.

Aos amigos do labGCCBD, Luis Orleans, Carlos Mello, Saulo Tavares, Francilei, Pedro Rougemont, Rodrigo Mesquita, Filipe Braida, Ricardo Barros e Bruno Osiek, pelas dicas, conversas e momentos de descontração que facilitaram a caminhada nesta jornada.

Ao CNPq, pelo apoio financeiro durante o mestrado.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

IDENTIFICAÇÃO DE REUSO EM DOCUMENTOS DIGITAIS

Fellipe Ribeiro Duarte

Junho/2011

Orientador: Geraldo Bonorino Xexéo

Programa: Engenharia de Sistemas e Computação

Nesta dissertação apresentaremos um mecanismo de identificação de Reuso de conteúdo de documentos. Para tanto, definimos um modelo teórico para basear a análise e o armazenamento dos documentos e propomos operações e relações possíveis entre documentos. Além disso, apresentamos uma medida que avalie o reuso do conteúdo de um documento e algoritmos para identificação de operações entre dois documentos. Arquiteturas lógicas e físicas para o mecanismo, assim como um protótipo para validar as propostas deste trabalho de pesquisa também são apresentados neste trabalho.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

IDENTIFICATION OF DIGITAL DOCUMENT REUSE

Fellipe Ribeiro Duarte

June/2011

Advisor: Geraldo Bonorino Xexéo

Department: Computer Science and Systems Engineering

This dissertation proposes an engine to identify documents' content reuse. Therefore, a theoretical model is defined to base: Document analysis and storage; Possible operations and relationships between two documents. Furthermore, a measure to estimates document' content reuse and an algorithm to identify operations between documents are proposed. Mechanism' logical and physical architecture and a prototype to validate the proposals of this research are also showed in this work

ÍNDICE

ÍNDICE DE IMAGENS

ÍNDICE DE TABELAS

1 Introdução

Neste capítulo apresentamos o contexto do trabalho, o que motivou esta pesquisa e a questão de investigação. São também apresentados os objetivos, a metodologia de pesquisa adotada e a organização deste texto.

1.1 Motivação e Contexto

Para CAVALCANTI (1995), estamos testemunhando uma nova revolução, a revolução da informação. No contexto dessa revolução MORESI (2000) afirma que a informação pode gerar uma vantagem competitiva que assegura uma posição dominante no mercado potencial, ou mesmo a sobrevivência da organização[1] no mercado (o que gera um valor maior do que apenas o econômico). Nas palavras do teórico Marshall McLuahn[2], um documento é a 'mídia' na qual uma 'mensagem' (informação) é comunicada. Portanto, em uma organização, há outros recursos muito importantes além do próprio capital humano que a compõe. Estes recursos são os documentos que são gerados durante a realização das atividades. Espera-se que no conteúdo dos documentos sejam armazenadas as informações[3] de vital importância para a atividade em questão.

Para PETERS (1993) a informação é o que realmente importa e, as redes de informações serão os fatores decisivos da competitividade em um futuro próximo (CAVALCANTI, 1995).

A informação não é apenas importante para controle, ela também serve para outras funções administrativas como a tomada de decisões e o planejamento. Ela serve de ponto de referencia para que o administrador tire conclusões, planeje, realize e avalie os resultados de decisões tomadas anteriormente (CAVALCANTI, 1995).

---

1 Para o propósito deste documento o termo organização pode ser: empresas, associações, agências governamentais e outras entidades públicas e privadas.

2 http://en.wikipedia.org/wiki/Marshall_McLuhan

3 Apesar da diferença entre o termo "dados" (representação / notação) e "informação" (significado / denotação) neste documento eles são utilizados de forma semelhante.

A informação deve ser captada, em seguida suas implicações são analisadas e com base nessas análises decisões são tomadas, elas produzem uma nova ação e novas informações que deverão ser analisadas (CAVALCANTI, 1995).

A agilização do fluxo de informações nas organizações é de extrema importância e, à medida que o mercado se globaliza mais importância é agregada a tal insumo (CAVALCANTI, 1995).

Para STREITZ et. al. (1989) escrever é uma atividade complexa que envolve a modelagem e a resolução determinado problema com várias restrições. Para tanto, ferramentas computacionais devem ser adequadas, para a atividade que está sendo realizada, de acordo as necessidades dos seus usuários (STREITZ; HANNEMANN; THÜRING, 1989). Além do mais, a estratégia de criação de nova informação usando reuso de informação é bem difundida (HOLLAND, 1992).

Em um mundo em mudança os documentos provêm uma medida de estabilidade (LEVY, 1993). Porém, eles próprios são objetos de mudança, assim como todos os materiais e artefatos sociais (LEVY, 1993). Os novos documentos e as novas versões de documentos freqüentemente compartilham muito material com documentos antigos, assim como os novos estados do mundo compartilham muito com os estados anteriores (LEVY, 1993). Isto significa que os documentos só necessitam ser atualizados de forma incremental e atualizações incrementais são mais bem alcançadas quando existe material para ser reutilizado (LEVY, 1993). Assim, para perpetuar a atividade de Reuso, devem existir ferramentas que possibilitem gerenciar as mudanças. Elas devem assegurar o Reuso e a integridade das informações, bem como garantir a disponibilidade das mesmas, de forma prática e acessível. A engenharia de documentos surgiu e vem evoluindo para atender um conjunto de necessidades que englobam as atividades listadas anteriormente.

A título de compreensão, apresentaremos nas próximas seções as características de texto que desejamos abordar, a engenharia de documentos, o reuso de documentos e os trabalhos e ferramentas relacionadas com o problema que abordaremos.

1.1.1 Características do texto

Os textos são convertidos em representações eletrônicas por vários motivos. As operações de busca e recuperação realizada nesses textos são, geralmente, de vital importância para a manipulação do texto. A maioria das operações é baseada elaborar consultas de texto, cujos resultados podem ser usados para comparações, classificação, computação numérica, impressão, editar, anotar, etc, bem como para operações de recuperação suplementares. Acessar textos por em

modelos baseados apenas em palavras ignoram partes importantes da informação presente no texto original (LOEFFEN, 1994, p 96).

Assume-se que dados textuais são modelados se três aspectos são cobertos: a estrutura do texto, operações que são permitidas e relevantes para manuseamento de texto e restrições que são aplicadas tanto na estrutura quanto no texto (LOEFFEN, 1994, p 96).

Existem algumas características inerentes aos textos eletrônicos que devem ser levadas em consideração quando se pensa em modelos para a representação dos mesmos, por exemplo: a natureza do texto (que muitas vezes não é levada em consideração). LOEFFEN (1994, p 96) Destaca algumas destas características:

- Textos são escritos em linguagem natural[4];
- Em texto eletrônico, as estratégias de codificação utilizadas para a transmissão das características do texto original são sempre da maior importância;
- Extremamente relacionada com os tópicos anteriores, o sistema de texto deve ser capaz de determinar os componentes básicos e esta decisão cabe não só as representações textuais, porém também no âmbito da estrutura do texto;
- Textos e subtextos são normalmente estendidos com meta-informações de diversas naturezas (atributos);
- Podem existir estruturas de texto paralelas entre uma entidade e o seu texto;
- Textos são mutáveis por natureza ou uso[5];

1.1.2 Valor da informação

A informação é um dos recursos mais importantes para as organizações. A gestão e o aproveitamento da informação são diretamente relacionados com o sucesso que uma organização deseja alcançar (MORESI, 2000, p 1).

Algumas organizações utilizam a informação como fator estruturante de seu instrumento de gestão (MORESI, 2000, p 1). Logo, para que a gestão seja bem

---

[4] Portanto, o conjunto de caracteres deve ser diferente entre textos e o conceito de palavra pode ser diferente entre linguagens.

[5] Segundo o autor o texto incorpora a informação da estrutura, das operações do mesmo e as restrições na estrutura e nas operações..

realizada, os valores da informação e do sistema de informação devem ser precisamente avaliados (MORESI, 2000).

CHAUMIER (1986) classifica as informações como as que servem para conhecimento dos ambientes internos e externos de uma organização assim como para a atuação nestes ambientes. MORESI (2000) apresenta uma variação desta classificação em função do papel que a informação desempenha nas atividades de uma organização, ela pode ser: Crítica, mínima, potencial ou sem interesse.

Independente do tamanho e da natureza de uma organização, a informação deve atender a todas as necessidades de todos os níveis administrativos das organizações (CHIAVENATO, 2001).

Para CHIAVENATO (2001) as organizações apresentam três níveis:

- Operacional: Localizado nas áreas inferiores da organização, é o nível no qual as tarefas são executadas e as operações realizadas e é relacionado com os problemas de desempenho eficaz e dirigido para as exigências impostas pela natureza da tarefa técnica;
- Intermediário ou gerencial: Gerencia particularmente as atividades do nível operacional, mediando as fronteiras ambientais e administrando as tarefas técnicas que devem ser desempenhadas, escala de operações etc.;
- Institucional: constitui-se na fonte do significado e da legitimização que possibilita a consecução dos objetivos organizacionais.

Existem quatro tipos nos quais o valor da informação pode ser classificado (CRONIN, 1990):

- Valor de uso: Com base na utilização final da informação;
- Valor de troca: (também pode ser chamado de valor de mercado) Varia de acordo com a lei da oferta e da demanda que o usuário se predispõe a pagar;
- Valor de propriedade: Relativo ao custo de substituição de um bem;
- Valor de restrição: Informação de uso restrito, secreta ou de interesse comercial;

1.2 Engenharia de documentos

A engenharia de documento é uma disciplina que estuda metodologias e técnicas de modelagem que gerem modelos significativos e reutilizáveis para troca de informação entre empresas (GLUSHKO; MCGRATH, 2005). Esta abordagem de utilizar documentos como entrada e saída para processo de negócio é ótima em ambientes tecnologicamente heterogêneos (GLUSHKO; MCGRATH, 2005).

GLUSHKO e MCGRATH (2005) definem a engenharia de documentos como: “um conjunto de análises e técnicas de modelagem que fornecem modelos de processos de negócio, e os seus documentos, reutilizáveis e significativos”.

Uma importante parte da engenharia de documentos lida com a transformação de documentos para facilitar seu processamento e reuso através de ambientes computacionais (LECERF; CHIDLOVSKII, 2006).

1.2.1 Reuso de documentos

Na escrita de obras que não são ficção, é uma prática muito comum utilizar pedaços de materiais existentes para a construção de documentos (BARTA; GIL, 1996).

Para a engenharia de documentos, uma importante linha de pesquisa é a de transformar documentos em formas que favoreçam o reuso de seu conteúdo (LECERF; CHIDLOVSKII, 2006).

Na engenharia de software, na escrita jurídica e no domínio pedagógico o reuso ou adaptação de documentos como manuais, contratos e objetos de aprendizado são práticas essenciais e que tem uma demanda crescente de tecnologia para atender suas respectivas atividades (BRANTING; LESTER, 1996 e DOWNES, 2001 e DUVAL et al., 2001 e VERBERT et al., 2008).

(LEVY, 1993) define o processo da criação de novos documentos e novas versões de velhos documentos em quatro classes:

- Criação – é a produção de novo material. Por exemplo, através de manuscritos ou digitação de textos;
- Coleção – é a identificação e a compilação de material que já existia só que separadamente;
- Combinação – é costurar um material novo e um antigo de forma a criar uma nova unidade;
- Customização – é retrabalhar em cima de um novo material para que ele se adéqüe a uma nova configuração.

Coleção, combinação e customização envolvem reuso enquanto apenas a criação introduz um novo material (LEVY, 1993).

(GUERRIERI, 1998) define alguns tipos de reuso que podem ocorrer em documentos:

I. Reuso de conteúdo de documento – aborda o reuso da informação contida no documento;

II. Reuso da estrutura do documento – aborda o Reuso da estrutura do documento como, por exemplo: título, autor, parágrafos, capítulos, apêndices entre outros;
III. Reuso do estilo do documento – aborda o reuso da informação de estilo como, por exemplo: tipo de fonte, estilo de fonte, espaçamento entre outros;
IV. Reuso de renderização de documento – aborda o reuso pela renderização dele em diferentes dispositivos como, por exemplo, CDROM, navegadores de internet, dispositivos de braile, pdf entre outros.

1.2.2 Trabalhos relacionados

Existem vários tipos de tecnologias que podemos considerar como resultado direto ou indireto da engenharia de documentos. Entre elas destacamos:

- Os editores e processadores de texto - que tem como objetivo facilitar o manusear de documentos textuais;
- Os bancos de dados textuais - que tem como prioridade estruturar a informação textual de documentos. Possibilitando assim, consultas que envolvam tanto o conteúdo quanto os elementos estruturais dos documentos da base;
- Modelos e técnicas que possibilitem a comparação de documentos que levem em consideração tanto a informação contida no texto como a informação contida na estrutura dos mesmos.

1.2.2.1 Editores de texto

De acordo com FRASER (1980, p 1) “editar significa examinar e modificar dados”, então os sistemas de edição de texto exploram as características do computador e as capacidades não numéricas (armazenamento, apresentação e manipulação de dados, resposta em tempo real a comandos e manuseio de estruturas de dados complexas) (GUTKNECHT, 1985, p 1).

A diferença entre editores de texto e processadores de texto é que o foco dos editores de texto está em funções de edição para texto plano e não em se preocupar com a formatação do texto (Text-Editor, 2010).

Exemplos de editores de texto consolidados são: SED[6], VI[7] e EMACS[8].

6 http://sed.sourceforge.net/

1.2.2.2 Banco de dados textuais

GONNET et al. (1987, p 339) usa o termo “Banco de dados dominados por texto” para se referir a coleções de dados estruturados que são predominantemente compostos de caracteres alfabéticos. Bancos de dados textuais estão necessitando de mais a mais atenção, graças a suas múltiplas aplicações: Bibliotecas, automação de escritório, engenharia de software, dicionários automatizados e enciclopédias, e em qualquer problema geral baseado em manter e recuperar informação textual (FRAKES; BAEZA-YATES, RICARDO, 1992).

O propósito de bancos de dados textuais é armazenar documentos de texto, estruturados ou não. Um banco de dados textual é composto por duas partes: conteúdo e estrutura (se apresentá-la) (BAEZA-YATES, R.; NAVARRO, G., 1996, p 67). O conteúdo é o texto por si só, a estrutura relaciona diferentes partes do banco de dados por algum critério (BAEZA-YATES, R.; NAVARRO, G., 1996, p 67 e NAVARRO, GONZALO; BAEZA-YATES, RICARDO, 10995).

1.2.3 Modelos e técnicas de comparação de documentos

Modelos, algoritmos e técnicas foram propostos para possibilitar a comparação de documentos utilizando as informações do seu conteúdo e da sua estrutura. Nesta seção destacaremos alguns exemplos:

- A distância de Levenshtein é uma métrica que explicita o quão diferentes duas sequências podem ser (LEVENSHTEIN, 1966), algumas adaptações dessas medidas são utilizadas para resolver problemas de reconhecimento de caracteres (KHURSHID et al., 2009), avaliar a similaridade entre as árvores de documentos XML (ZHOU et al., 2007), classificação de documentos XML (XING et al., 2006) e reconhecimento de discurso (FISCUS et al., 2006);
- GUERRIERI (1998) aborda o reuso de documentos compostos de artefatos de software (por exemplo: requisitos e especificações funcionais) utilizando XML;
- Um framework para reuso de conteúdo “*on-the-fly*” é proposto em (VERBERT et al., 2008).

---

7 http://ex-vi.sourceforge.net/

8 http://www.gnu.org/software/emacs/

1.3 Objetivo

O objetivo deste trabalho é propor um mecanismo de identificação de Reuso de conteúdo de documentos. Para tanto, definimos um modelo teórico para basear a análise e o armazenamento dos documentos, propomos operações e relações possíveis entre documentos, uma medida que avalie o reuso do conteúdo de um documento, algoritmos para identificação de operações entre dois documentos e arquiteturas lógicas e físicas para o mecanismo, assim como um protótipo para validar as propostas deste trabalho de pesquisa.

1.3.1 Metas

Com o intuito de alcançar o objetivo desejado, algumas metas foram traçadas:

- Apresentar um meta-modelo ontológico no domínio de documentos digitais que abordem as estruturas dos documentos, seus relacionamentos e as operações de manipulação possíveis entre eles;
- Estudar e listar as possíveis relações entre documentos, suas partes e outros documentos e suas respectivas partes;
- Propor uma arquitetura para detecção, gerenciamento e manipulação dos documentos, suas estruturas e relacionamentos;
- Possibilitar consultas em documentos utilizando informações semânticas que abordem o conteúdo e a estrutura;
- Propor uma medida de “similaridade” entre documentos e partes de documentos com outros documentos e suas respectivas partes de documentos;
- Propor um mecanismo de Reuso de documentos;
- Apresentar um protótipo de ferramenta utilizando as propostas anteriores;
- Avaliar os resultados obtidos pela ferramenta.

1.4 Organização do trabalho

Além deste capítulo introdutório, Este trabalho contém mais 6 capítulos. O capítulo 2 apresenta o conceito formal de documento, suas representações e questões de recuperação de informação relacionadas. Ele também apresenta a engenharia de documentos e a atividade de Reuso de documentos dentro desse escopo.

O capítulo 3 propõe uma ontologia no domínio de documentos digitais e seus relacionamentos e uma álgebra de manipulação de documentos.

No capítulo 4 apresentaremos uma medida para calcular o quão similar dois documentos são utilizando, também, a informação semântica encontrada na estrutura dos documentos. Nele também discutimos a identificação das operações e apresentamos os algoritmos propostos para o calculo da medida e a identificação das operações.

O capítulo 5 apresenta a arquitetura para o mecanismo de Reuso de documentos e um protótipo de ferramenta baseado nessa arquitetura.

O capítulo 6 aborda um estudo experimental para avaliar a viabilidade técnica da hipótese desta dissertação. Nele apresentamos os procedimentos executados, a forma como os resultados foram obtidos e apresentamos os resultados do estudo em questão.

No capítulo 7 se encontra a conclusão do nosso trabalho, as nossas contribuições e os respectivos trabalhos futuros.

2 Modelos, Representações e Reuso de Documentos

Esse capítulo apresenta o conceito formal de documento e como podem ser compreendidos e organizados em bases de documentos. Além disso, apresenta de forma geral a questão da recuperação de informação dentro dessas bases e de forma mais específica, os modelos de recuperação de informação baseados na estrutura do texto. Finalmente, a engenharia de documentos é apresentada junto com os conceitos, características e trabalhos relacionados com o reuso de documentos

2.1 O que é um documento?

A denotação da palavra documento é uma classe de artefatos de comunicação muito abrangente, ela inclui formas eletrônicas, representações multimídia e arquivos de áudio e vídeo (LEVY, 1993). Três aspectos que podemos relacionar ao termo documento são: (LEVY, 1993)

I. O repositório de um material comunicativo para propósitos particulares;
II. Instâncias de tipos sociais reconhecíveis;
III. Sua representação é feita de forma estável.

Segundo REITZ (2004) um documento é definido como:

"Um termo genérico para uma entidade física que consiste de qualquer substância na qual é gravado o todo, ou parte, de um ou mais trabalhos com o propósito de transportar ou preservar um conhecimento. Nas palavras do teórico Marshall McLuahn[9], um documento é a 'mídia' na qual uma 'mensagem' (informação) é comunicada."

Para BAEZA-YATES et al. (1999) um documento denota uma simples unidade de informação, tipicamente na forma digital, mas pode incluir outros tipos de mídia. A ação de coletar artefatos e representar suas informações é conhecida como "documentar".

Texto tem estrutura natural, para GROZA e HANDSCHUH (2009, p 44) um documento pode ser dividido em capítulos, páginas, sentenças, parágrafos, seções

9 http://en.wikipedia.org/wiki/Marshall_McLuhan

subseções, livros, volumes, edições, linhas versos ou estrofes. Eles afirmam que um documento pode incluir um título, um prefácio, um resumo, um epílogo, citações, referências, digressões e notas.

Dicas de gênero provêem uma forma bem eficiente para suprir informação do contexto do documento e seu suposto uso (LEVY, 1993). Por exemplo, artigos em jornais ou periódicos tem uma forma que pode informar muito sobre o gênero de documento ao qual eles pertencem (LEVY, 1993).

As características dos documentos variam muito. Um documento pode ter um autor identificado ou ele será anônimo; pode ser precisamente datado ou não; pode ser escrito em russo ou usando caracteres do alfabeto cirílico; pode ser escrito em japonês usando Kanji; pode ser parte de um trabalho maior; pode ser autônomo (GROZA; HANDSCHUH, 2009, p 44). Cada documento é estruturado de forma diferente e a estrutura pode variar mesmo dentro do mesmo documento (GROZA; HANDSCHUH, 2009, p 44).

2.2 O que é um modelo?

A forma como armazenamos, acessamos e comparamos o conteúdo de um documento é de vital importância para o mecanismo de recuperação de informação que irá suportar a atividade.

A informação que pertence a este documento deve ser representada considerando: a sua estrutura, os elementos que o compõem e os significados semânticos de cada elemento no conjunto da obra.

Portanto, é necessário definir os modelos que podem atender e satisfazer esses requisitos. Para isso, devemos primeiro entender o que é um modelo. TAGUE et al. (1991, p 14) esclarece estas questões:

“O propósito de um modelo de sistema formal é descrever as características comuns de um conjunto de sistemas que foram feitos para problemas similares. O modelo explicará a estrutura e os processos deste sistema, e esclarecerá suas características gerais, i.e. características não específicas. Os componentes do modelo devem incluir os tipos de entidades, os relacionamentos, e as transformações ou operações que formam parte do sistema que desejamos descrever. Um modelo completo conterá uma representação de todos os componentes de qualquer sistema do tipo referenciado pelo modelo”.

2.3 Representando um documento

Geralmente, um documento é dividido em vários campos pré-definidos, tipicamente título, autor, data, resumo e corpo. Consultas devem referenciar estes campos (GROZA; HANDSCHUH, 2009, p 44).

Grandes blocos de texto geralmente são divididos em sentenças, parágrafos e outras unidades pré-definidas (GROZA; HANDSCHUH, 2009, p 44). Campos pré-definidos podem ser caracterizados como hierárquicos, contudo, muitos bancos de dados textuais (por exemplo, programas, notícias, patentes, relatórios, documentos SGML em geral) têm uma estrutura descrita por uma gramática que define a hierarquia de regiões aninhadas (CONSENS; MILO, 1995, p 13).

Algumas tentativas foram feitas para integrar busca de texto estruturado em bancos de dados orientados a objeto (BAEZA-YATES, R.; NAVARRO, G., 1996, p 68). Toma-se conhecimento disto quando tentamos modelar aplicação de dados não tradicionais como: CAD-CAM, automação de escritório, bancos de dados orientados a texto ou a multimídia. Algumas alternativas foram propostas (GYSSENS et al., 1989, p 263).

Todas as alternativas compartilham da propriedade de reconhecer que a mais fundamental característica do dado é de apresentar uma estrutura hierárquica. Contudo, não esta claro quando eles podem efetivamente modelar todos os dados para aplicações os quais exibem uma natureza hierárquica (GYSSENS et al., 1989, p 263).

Um elemento estrutural em particular – fronteira do parágrafo, por exemplo – deve ser indexado usando tanto marcações específicas para o formato do documento quanto marcações genéricas para a coleção como um todo (CLARKE; CORMACK; BURKOWSKI, 1995a, p 8). Visualizações na recuperação de texto estruturado devem ser sumarizadas por três princípios independentes de esquemas (CLARKE; CORMACK; BURKOWSKI, 1995a, p 8):

I. Independência de meta-estrutura – Consultar referenciando a estrutura do documento não deve ser expressa em termos da meta-estrutura do documento;
II. Independência de marcação – A marcação descreve o texto, mas não faz parte dele;
III. Independência de elemento de consulta – Elementos estruturais análogos de documentos de diferentes formatos devem ser indexados como elementos pertencentes a um único grupo.

2.4 Representando a estrutura do texto

Existem duas abordagens básicas para representar a estrutura do texto. A primeira é codificar a estrutura no texto diretamente utilizando algum tipo de marcação e a segunda é expressar a estrutura em uma árvore de estrutura de dados separada (CONSENS; MILO, 1995, p 21).

Independência de dados tem, contudo, forte influência no usuário final. Ou seja, em um modelo de dados independente de dados o usuário pode expressar consultas sem conhecer detalhes irrelevantes do esquema de banco de dados. (SCHWARZ, 1978, p 215).

As necessidades para intercâmbio, gerenciamento e publicação de documentos eletrônicos têm motivado a criação dos padrões ODA e SGML. A idéia básica nestes padrões é que um documento é acompanhado pela definição de sua estrutura (SCHWARZ, 1978, p 215).

Um documento SGML contém a descrição de sua estrutura, sua definição de tipo de documento (DTD) além do seu texto (CLARKE; CORMACK; BURKOWSKI, 1995a, p 8). Esta facilidade para explicitamente descrever um esquema de documento ou meta estrutura (estrutura da sua estrutura) é central para o SGML. Outras convenções além do SGML geralmente misturam as especificações da sua estrutura lógica e física e são restritas na variedade de estruturas lógicas que podem ser diretamente expressas (CLARKE; CORMACK; BURKOWSKI, 1995a, p 8). Porém, do ponto de vista de um usuário, uma marcação não é parte do texto por si só, mas é uma parte da representação do texto (CLARKE; CORMACK; BURKOWSKI, 1995a, p 8).

Uma métrica é necessária para especificar a proximidade dos elementos no texto. É desejável visualizar o texto como uma sequência de palavras (ou outra unidade textual básica). Marcações não devem ser tratadas como palavras por questão de proximidade (GROZA; HANDSCHUH, 2009, p 45).

2.5 Texto exclusivamente estrutural

Grandes blocos de texto são geralmente divididos em sentenças, parágrafos ou outras unidades pré-definidas. Documentos devem então ser selecionados com base nas palavras que aparecem na mesma sentença ou parágrafos (GROZA; HANDSCHUH, 2009, p 44).

A estrutura do documento que não pode ser mapeada em campos pré-definidos ou unidades textuais pode ser perdida e não poderá ser referenciada em uma consulta (GROZA; HANDSCHUH, 2009, p 44). Muitas propostas para lidar com

estrutura de documento foram feitas. Geralmente, estas propostas abordam a estrutura do documento como hierárquica (GROZA; HANDSCHUH, 2009, p 44).

Um sistema para capturar a estrutura de um documento deve ser flexível o suficiente para acomodar as variações na estrutura que ocorrem naturalmente. Deve ser possível indexar toda a estrutura em um documento, contudo, é importante ressaltar que, quando o documento é adicionado ao banco de dados a estrutura de documentos nem sempre é estritamente hierárquica, isto é, parágrafos se estendem por páginas, sentenças estendem-se por linhas (GROZA; HANDSCHUH, 2009, p 45). Não obstante, um elemento estrutural estar contido em outro é significante para a estrutura de um documento – Sentenças são, usualmente, totalmente contidas em parágrafos, linhas geralmente estão totalmente contidas em uma página (GROZA; HANDSCHUH, 2009, p 45).

A presença de marcações para marcação nos documentos sugere que a indexação das marcações deve ser uma abordagem efetiva para capturar a estrutura do documento, porém existem vários problemas com esta abordagem. Formatos de documentos diferentes usam diferentes sintaxes para marcações. Além do mais, estas diferenças são sempre desejáveis para que elementos estruturalmente equivalentes sejam indexados juntos (GROZA; HANDSCHUH, 2009, p 45).

2.6 Ontologias baseadas em texto estruturado

Ontologias podem ser utilizadas para modelar domínios nos quais o insumo central é o documento. Este domínio pode cobrir o conteúdo, a estrutura, o conhecimento ou os relacionamentos de documentos. Exemplos de aplicações de ontologias textuais são: busca e recuperação de informação, analise de patentes, sistemas de gerenciamento de documentos, plágio, clusterização de documentos, gestão do conhecimento e websemântica.

2.6.1 Ontologia

Sistemas baseados em conhecimento usam a representação de conhecimento formal para operar e comunicar. Portanto, estes sistemas apresentam requisitos especiais de interoperabilidade (GRUBER, 1995, p 1).

O uso de uma ontologia para representar o conhecimento formal e suas atividades de compartilhamento de conhecimento visa viabilizar as bibliotecas de componentes de conhecimento reusáveis e os serviços baseados em componentes. Estas bibliotecas e serviços podem ser solicitados através de redes (GRUBER, 1995, p 1).

Uma conceitualização é a base do conhecimento formalmente representado e pode ser definida como “Uma abstrata, simplificada vizualização do mundo que nós desejamos representar para algum propósito” (GRUBER, 1995, p 1). Uma especificação explícita de uma conceitualização é uma ontologia (GRUBER, 1995, p 2).

Quatro usos gerais de ontologias no desenvolvimento de sistemas baseados em conhecimento são descritos em (ABU-HANNA; JANSWEIJER, 1994, p 3): Compartilhabilidade – a ontologia determina a definição de termos únicos utilizados em diferentes sistemas; Aquisição de conhecimento; Organização do conhecimento; Raciocínio (“*Reasoning*”) – A ontologia e a base podem ser utilizadas na tarefa de raciocinar;

De acordo com o tipo de aplicação e do contexto a ontologia pode ser classificada de acordo com os níveis de generalidade: ontologias genéricas; ontologias de domínio; ontologias de tarefas e ontologias de aplicação (GUARINO, N., 1998, p 7).

Uma ontologia genérica descreve conceitos mais gerais sem um domínio ou um problema particular tipo: espaço; tempo; substância; objetos, eventos e ações.

Uma ontologia de domínio usa um domínio específico para agrupar os conceitos e os relacionamentos aplicando restrições e regras sobre a estrutura e o conteúdo de um conhecimento.

Uma ontologia de tarefa expressa conceitos independentes de domínio para realizar uma tarefa ou atividade genérica para resolver um problema, por exemplo: acesso de informação.

Uma ontologia de aplicação expressa conceitos de domínios e tarefas específicas a domínio.

O relacionamento entre os tipos de ontologias são expressos Figura 1 (GUARINO, NICOLA, 1997).

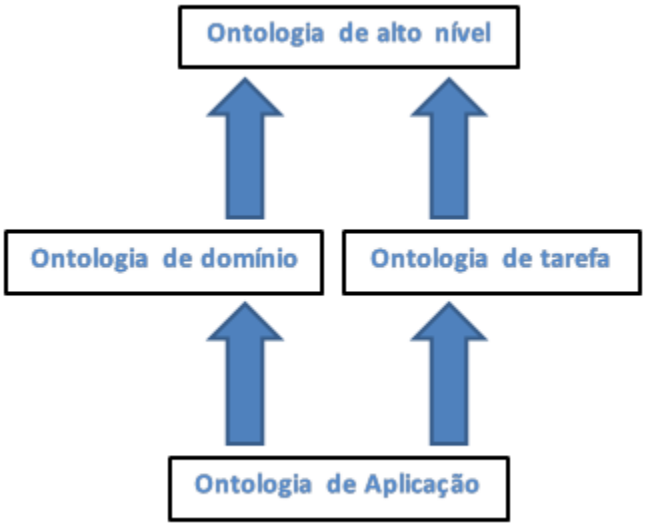


Figura 1: Tipos de ontologias, de acordo com o seu nível de dependência em uma particular tarefa ou ponto de vista (adaptada de (GRUBER, 1995, p 2))

2.6.2 Ontologia SALT

A ontologia SALT modela o domínio da estrutura linear de uma publicação. O objetivo desta ontologia é representar o conteúdo de publicações científicas em diferentes níveis de granularidade (GROZA; HANDSCHUH, 2009). Figura 2 apresenta uma visão geral do modelo.

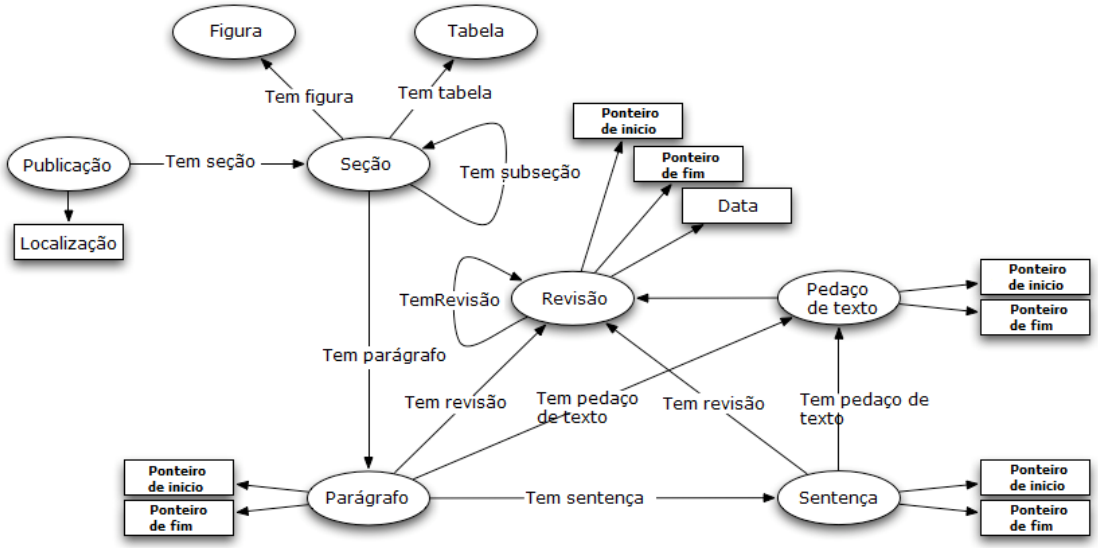


Figura 2: Visão geral da ontologia SALT (adaptada de (GROZA; HANDSCHUH, 2009))

2.6.3 Ontologia CNXML

CNXML destaca o conteúdo da apresentação. A informação de como representar os documentos CNXML são feitas separadamente. O modelo marca a estrutura e o conteúdo semântico de um documento.

Um modelo UML da ontologia pode ser encontrado em (GROZA; HANDSCHUH, 2009).

2.6.4 Outras ontologias de documentos

Existem outras iniciativas de aplicação de ontologias de documentos para resolver um problema. Entre elas queremos destacar:

I. DINOS e VEGA-RIVEROS (2004) Usa uma ontologia de documentos e técnicas de busca e recuperação de informação para: gestão do conhecimento; identificação e gerenciamento de conceitos ou tópicos para facilitar a busca de estudantes;
II. TRAPPEY et al. (2009) Usa a ontologia com uma abordagem de clusterização hierárquica para gerenciamento de documentos focado em análise de patentes.

2.7 Banco de dados textuais

GONNET et al. (1987, p 339) usa o termo “Banco de dados dominados por texto” para se referir a coleções de dados estruturados que são predominantemente compostos de caracteres alfabéticos.

Bancos de dados textuais estão necessitando de mais a mais atenção, graças a suas múltiplas aplicações: Bibliotecas, automação de escritório, engenharia de software, dicionários automatizados e enciclopédias, e em qualquer problema geral baseado em manter e recuperar informação textual (FRAKES; BAEZA-YATES, RICARDO, 1992).

O propósito de bancos de dados textuais é armazenar documentos de texto, estruturados ou não. Um banco de dados textual é composto por duas partes: conteúdo e estrutura (se apresentá-la) (BAEZA-YATES, R.; NAVARRO, G., 1996, p 67). O conteúdo é o texto por si só, a estrutura relaciona diferentes partes do banco de dados por algum critério (BAEZA-YATES, R.; NAVARRO, G., 1996, p 67 e GROZA; HANDSCHUH, 2009, p 93). O texto é visto como uma (longa) sequência de símbolos e a estrutura é organizada como um conjunto de hierarquias independentes (ortogonais)(CIANCARINI, 1996, p 408 e GROZA; HANDSCHUH, 2009, p 95).

Qualquer modelo de informação para bancos de dados textuais deve compreender três partes: texto, estrutura e linguagem de consulta. Deve-se especificar como o texto é visto (isto é, o que pode ser perguntado, o que são as respostas, etc.) (BAEZA-YATES, R.; NAVARRO, G., 1996, p 67 e GROZA; HANDSCHUH, 2009, p 43, 2009, p 93).

Uma linguagem de consulta é dita independente de dado quando a estrutura lógica do dado no banco de dados, isto é, o esquema de banco de dados pode ser modificado sem a necessidade de modificar as aplicações ou as consultas que acessam os dados (SCHWARZ, 1978, p 215).

Tradicionalmente, existem dois mecanismos pelos quais usuários podem achar o que precisam e os bancos de dados textuais permitiram a seus usuários buscar pelo conteúdo (palavras, frases, etc.) ou pela estrutura (olhando através de uma tabela de conteúdos)(GROZA; HANDSCHUH, 2009, p 93).

BURKOWSKI (1992, p 113 e 114) apresenta algumas características de bancos de dados de textos estruturados e as metas desejadas para sistemas de recuperação de texto:

I. Heterogeneidade – espera-se que uma coleção de peças de teatro tenha uma estrutura que é diferente de uma coleção de sonetos;
II. Multi-media;
III. Camadas hierárquicas;
IV. Independência de dados – no motor de busca. Uma meta maior é isolar a complexidade do motor de recuperação encapsulando-o em um módulo que irá responder à simples sentenças de comandos;
V. Utilização de várias camadas de hierarquia – a busca e a navegação do banco de dados podem ser consideravelmente melhoradas quando as hierarquias lógicas, semânticas e de apresentação são corretamente utilizadas;
VI. Funcionalidades de interface extensíveis – a heterogeneidade das coleção de dados irá complicar a interface de usuário, visto que, as técnicas de busca e navegação podem variar de uma coleção de dados para a outra. Existem duas razões para isto: as tags irão diferenciar-se em diferentes coleções de dados e a saída da hierarquia pode variar de uma coleção de dados para a outra.

Nenhum modelo tradicional dependente de dados (por exemplo, o modelo relacional) ou que assume objetos de dados não interpretados ou que depende dos seus atributos é poderoso o suficiente para representar a riqueza da informação contida no texto. A informação tem que ser extraída do texto, porém de uma forma não

tão rígida (CIANCARINI, 1996, p 401). Para GROZA e HANDSCHUH (2009, p 93) a informação esta lá, porém não há uma forma fácil de extraí-la.

Existe um interesse significativo em combinar e estender bancos de dados e tecnologias de busca e recuperação da informação para gerenciar dados textuais (CONSENS; MILO, 1995, p 11). Contudo, as formalizações de relações textuais nesses sistemas de bases de dados factuais receberam pouca atenção pela comunidade de banco de dados (LOEFFEN, 1994, p 96).

2.8 Outros modelos Conceituais

2.8.1 FRBR

De acordo com TILLETT (2003) os requisitos funcionais de para registros bibliográficos, em inglês “Functional Requirements for Bibliographic Records (FRBR),” são “um modelo de entidade-relacionamento como uma visão generalizada do universo, com o objetivo de ser independente de qualquer código de catálogo ou implementação”.

A intenção da IFLA[10] quando desenvolveu o FRBR era de criar um modelo conceitual que relacionasse as tarefas realizadas pelos usuários, quando estão consultando registros bibliográficos, com atributos e relacionamentos específicos (IFLA, 1998, p 3).

As seguintes tarefas genéricas de usuários foram utilizadas como fundamento para a definição do FRBR (IFLA, 1998, p 8):

I. Usar os dados para achar matérias que correspondam com o estado dos critérios da busca do usuário;
II. Usar os dados recuperados para identificar uma entidade;
III. Usar os dados para selecionar uma entidade que é apropriada para as necessidades do usuário;
IV. Usar os dados solicitados para adquirir ou obter acesso a entidade descrita;
V. No FRBR as entidades foram divididas em três grupos (IFLA, 1998, p 11):
    i. Os produtos, i.e. resultados, do esforço intelectual ou artístico foram abrangidos por este grupo;

10 http://www.i fla.org

ii. Entidades responsáveis por conteúdo intelectual ou artístico, produções físicas ou disseminações, ou a custódia de tais produtos estão neste grupo;

iii. Entidades que servem de assunto de esforço intelectual ou artístico estão incluídas neste ultimo grupo.

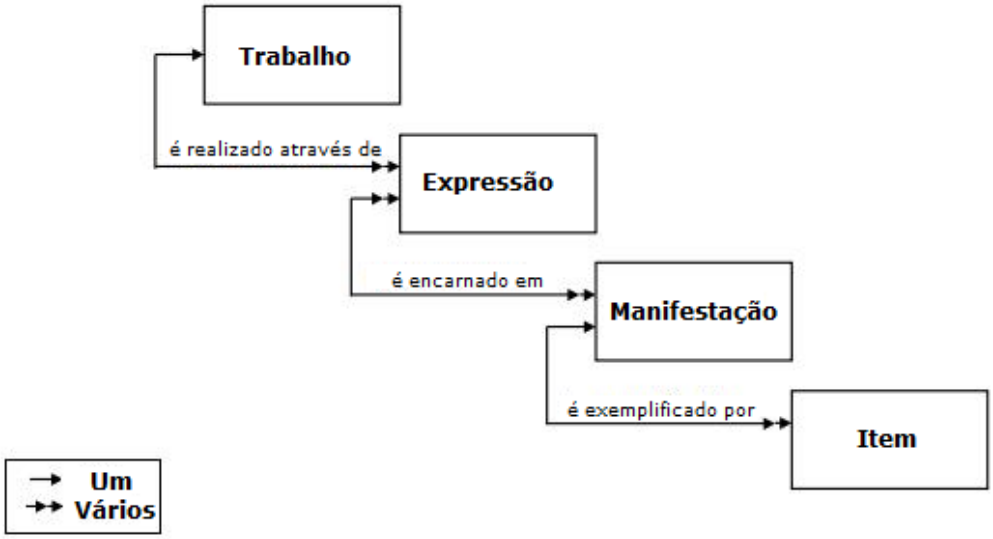


Figura 3: Grupo 1 (adaptada de (TILLETT, 2003))

A Figura 3 apresenta as entidades e os seus relacionamentos que compõem o grupo 1 do FRBR. Estas entidades são: Trabalho, expressão, manifestação e item.

De acordo com IFLA (1998, p 11) “trabalho é uma criação intelectual ou artística distinta”.

Quando uma entidade é criada usando um grau significante de independência intelectual ou esforço artístico, sobre as modificações de um trabalho, esta entidade é um novo trabalho (IFLA, 1998, p 15). Contudo, o fator cultural influencia na decisão de o que compõe um trabalho e a fronteira deste trabalho. Portanto, foi definido no FRBR que a variações de textos que incorporam revises ou atualizações de um texto anterior são encaradas como expressões de um mesmo trabalho (IFLA, 1998, p 15).

Uma expressão é “a forma intelectual ou artística específica que um trabalho toma quando é ‘realizado’” (IFLA, 1998, p 18). As fronteiras de uma expressão eram definidas para não cobrir aspectos das formas físicas, como fonte e formato de página que não pertencem completamente à realização artística ou intelectual do trabalho (IFLA, 1998, p 18).

Uma manifestação é “a encarnação física de uma expressão de um trabalho” (IFLA, 1998, p 20). Ela é uma representação do objeto inteiro que apresenta as mesmas características, tanto no que diz respeito ao conteúdo intelectual quanto na

forma física e estas mudanças incluem mudanças nas características de apresentação, na mídia física e no invólucro (IFLA, 1998, p 20).

Um item é “um simples exemplar de uma manifestação” (IFLA, 1998, p 23). Contudo uma instância de um item é mais do que apenas um único objeto físico.

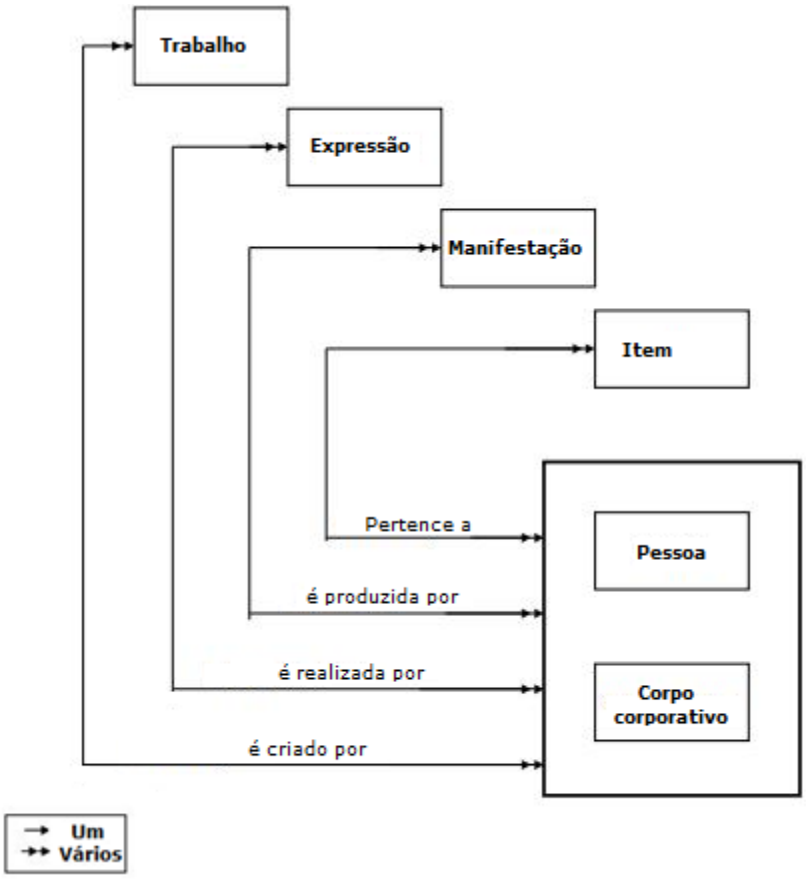


Figura 4: Grupo 2 (adaptada de (TILLETT, 2003))

A Figura 4 apresenta as entidades e seus relacionamentos que compõem o grupo 2 do FRBR. Estas entidades são: pessoa e corpo corporativo.

De acordo com (IFLA, 1998, p 23) “pessoa é um indivíduo” e corpo corporativo é “uma organização ou grupo de indivíduos e/ou organizações agindo como unidade”.

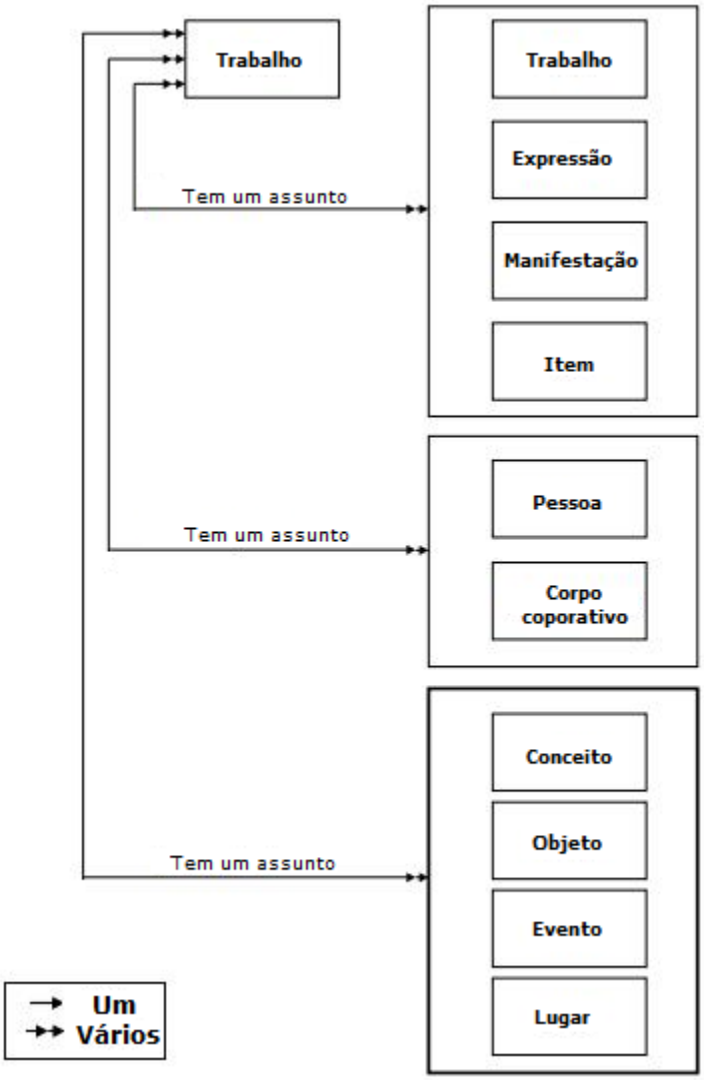


Figura 5: Grupo 3 (adaptada de (TILLETT, 2003))

A Figura 5 apresenta as entidades e seus relacionamentos que compõem o grupo 3 do FRBR. Estas entidades são: Conceito, objeto, evento e lugar.

Segundo IFLA (1998, p 25) “conceito é uma noção abstrata ou idéia”, “objeto é uma coisa material”, “evento é uma ação ou ocorrência” e “lugar é uma localização”.

2.8.2 INDECS

O framework INDECS foi criado para colocar os identificadores de criações diferentes e seus metadados de apoio de forma que eles possam operar juntos, dando suporte ao gerenciamento de direitos de propriedade intelectual(GODFREY; BIDE, 2000, p 4). Seu foco principal é no conteúdo ou na propriedade intelectual.

Embora o modelo possa ser aplicado em vários contextos, os ambientes digitais e da internet são os que os problemas de interoperabilidade de metadados estão ficando especialmente graves (GODFREY; BIDE, 2000, p 4).

O framework posiciona-se sobre axiomas de comércio eletrônico, onde o senso do termo comércio é mais abrangente que apenas lucro (GODFREY; BIDE, 2000, p 4). Estes axiomas são:

I. Metadados são críticos – Para identificar metadados e lidar com propriedade intelectual, os identificadores pertencem a uma cadeia complexa e dinâmica que irá determinar se uma transação irá acontecer ou não (GODFREY; BIDE, 2000, p 4);
II. Coisas são complexas – Uma criação é definida como “um pedaço de coisa na qual se deve carregar os direitos de propriedade intelectual” (GODFREY; BIDE, 2000, p 4);
III. Metadado é modular – metadado é modular porque “coisa” é complexa (GODFREY; BIDE, 2000, p 5) Para realizar a tarefa de achar e usar coisas cada entidade como partes, criações e transações devem ter seus próprios conjuntos de metadados (GODFREY; BIDE, 2000, p 5);
IV. Transação necessita de automação – A escala e a natureza do comercio eletrônico tem tornado imperativo que estes padrões locais e sistemas possam interoperar de formas autonômicas com outros (GODFREY; BIDE, 2000, p 5).

O quarto axioma faz a interoperabilidade um requisito fundamental para o INDECS. Interoperabilidade permite que a informação originada em um contexto seja concedida e utilizada em outro contexto, de forma mais automatizada quanto for possível (GODFREY; BIDE, 2000, p 6).

Interoperabilidade requer suporte em no mínimo seis tipos diferentes: Através de mídias, através de funções, através de níveis de metadados, através de barreiras lingüísticas e semânticas, através de barreiras territoriais e através de plataformas tecnológicas (GODFREY; BIDE, 2000, p 6).

Interoperabilidade semântica será realizada pelo INDECS para evitar confrontação entre padrões (GODFREY; BIDE, 2000, p 7).

Apesar de o INDECS focar no gerenciamento de direitos, i.e. propriedade intelectual, ele se preocupa apenas com os mecanismos para descrever as transações que surgem e é neutro no mérito dos direitos ou da prática (GODFREY; BIDE, 2000, p 8).

De acordo com GODFREY e BIDE (2000, p 8) o framework reconhece:

I. Todos os metadados relacionados com qualquer tipo de criação;
II. A integração de metadados descritivos com transações comerciais e direitos;

III. O metadado deve ser criado uma vez e utilizado várias vezes para diferentes propósitos;
IV. E propõe:
V. Um atributo estrutural genérico para todas as entidades;
VI. Eventos como chave para relacionamentos de metadados complexos;
VII. Um dicionário de metadados para comércio de propriedade intelectual para mutimídia;
VIII. Identificadores únicos (iids) para serem atribuídos a todos os elementos de metadados;
IX. A necessidade de processos de transformação para expressar o mesmo metadado em diferentes níveis de complexidade para requisitos diferentes.

O princípio da identificação única, o princípio da granularidade functional, o princípio da autoridade designada e o princípio do acesso apropriado são os quatro princípios para o desenvolvimento de metadados para suporte efetivo do comércio eletrônico no INDECS (GODFREY; BIDE, 2000, p 8). Nós queremos destacar o princípio da granularidade que é "é possível identificar uma entidade onde quer que seja preciso se distinguir" (GODFREY; BIDE, 2000, p 10).

Granularidade funcional é um conceito que permite identificar partes ou versões de uma entidade quando a ocasião surgir, mas ele não identifica todas as versões e partes possíveis todo o tempo (GODFREY; BIDE, 2000, p 10).

O modelo de metadados do INDECS tem uma definicão geral de metadado "um item de metadado é um relacionamento que alguém reinvindica existir entre duas entidades" (GODFREY; BIDE, 2000, p 11) que serve para distinguir os dados dos metadados.

Os termos básicos de semantica são entidades, atributos e valores que são classificados como tipos de elementos de metadados (GODFREY; BIDE, 2000, p 11).

No INDECS uma entidade pode ser definida através de três distintas, porém sobrepostas, visões de entidades: uma visão geral, uma visão comercial e uma visão de propriedade intelectual.

A Figura 6 apresenta que a visão geral categoriza as entidades em: Percepções, conceitos e relações. Nesta visão a entidade eventos estabelece relacionamentos entre todos os metadados e outros eventos que destacam a importância desta entidade no modelo (GODFREY; BIDE, 2000, p 13).

Percepções são aquelas identificadas pelos sentidos, conceitos são aquelas que existem na mente e entidades interconectadas são relações.

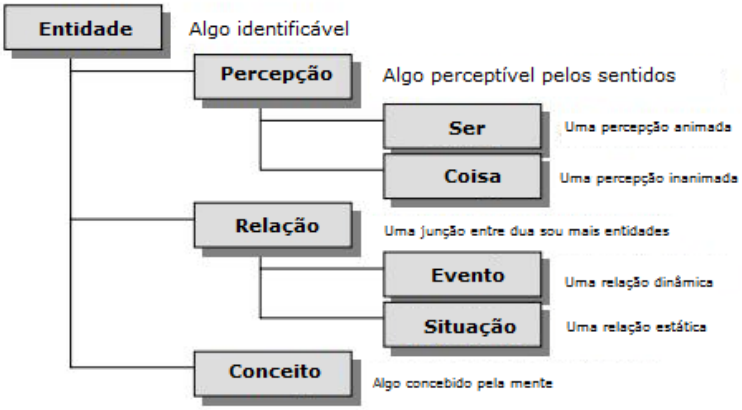


Figura 6: Visão geral, entidades primitivas (adaptada de (GODFREY; BIDE, 2000, p 13)).

A visão comercial é geralmente preocupada com a atividade de fazer coisa, que é representada na Figura 7 e na Figura 8.

Figura 7: Visão Comercial (adaptada de (GODFREY; BIDE, 2000, p 4)).

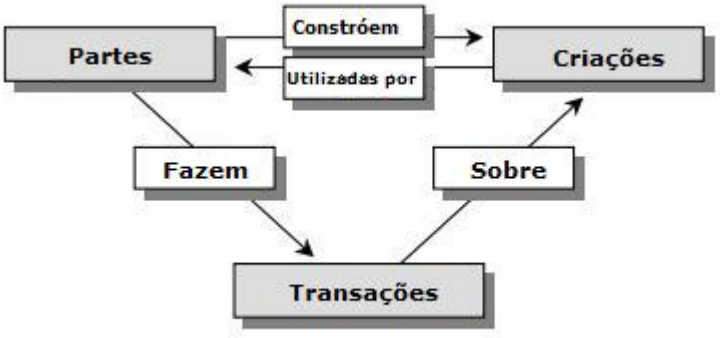


Figura 8: Transações sobre as criações (adaptada de (GODFREY; BIDE, 2000, p 4)).

A visão de propriedade intelectual é representada na Figura 9.

Figura 9: Visão da propriedade intelectual (adaptada de (GODFREY; BIDE, 2000, p 4)).

A Figura 10 apresenta os cinco tipos de atributos de uma entidade que são definidos pelo INDECS (GODFREY; BIDE, 2000, p 16)

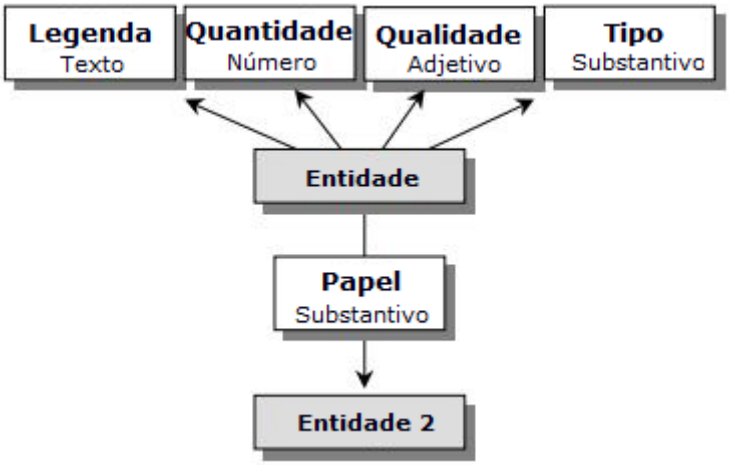


Figura 10: Atributos genéricos (adaptada de (GODFREY; BIDE, 2000, p 4)).

No INDECS existem três níveis de complexidade do relacionamento de metadados:

I. Eventos – “são relações nas quais algo muda”(GODFREY; BIDE, 2000, p 13);
II. Situações – “são relações nas quais algo permanece o mesmo” (GODFREY; BIDE, 2000, p 13);
III. Atributos.

De acordo com Figura 11 relações são compostas de duas ou mais entidades que tem papéis definidos entre elas (GODFREY; BIDE, 2000, p 13). Estes papéis são: Agente, Saída, Entrada e Contexto.

Figura 11: Papéis na relação (adaptada de (GODFREY; BIDE, 2000, p 4)).

Para GODFREY e BIDE (2000, p 13) “um evento pode ser simples ou complexo” um evento pode conter ou sobrepor, em níveis diferentes de granularidade, com outros eventos e relações. Figura 12, Figura 13 e Figura 14 apresentam que as entidades de eventos podem ser agrupadas em três tipos: O evento de criação, Os eventos de utilização e os eventos de transformação.

Figura 12: Evento criação (adaptada de (GODFREY; BIDE, 2000, p 4)).

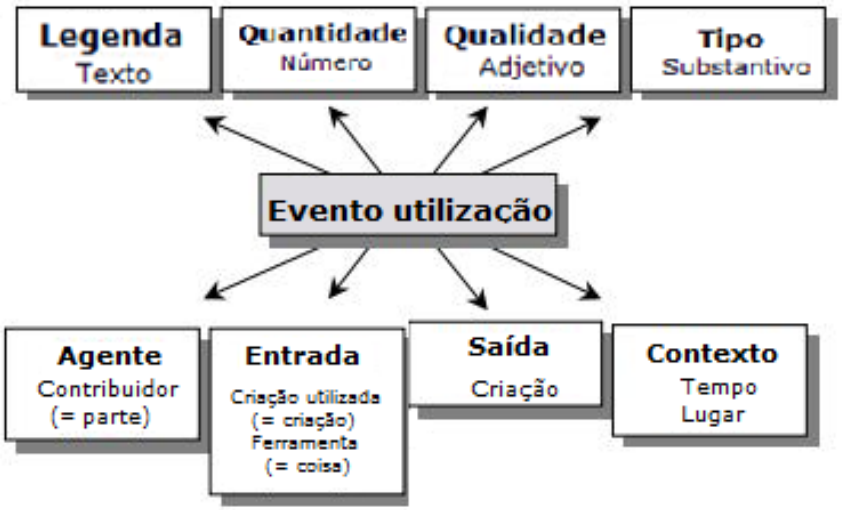


Figura 13: Evento utilização (adaptada de (GODFREY; BIDE, 2000, p 4)).

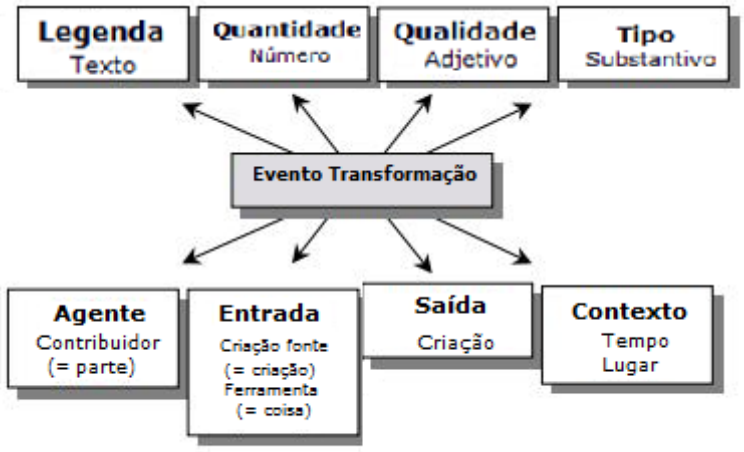


Figura 14: Evento transformação (adaptada de (GODFREY; BIDE, 2000, p 4)).

A estrutura do evento oferece ao menos três atrações principais para interoperabilidade de metadados:

I. É uma forma de criar o máximo de relacionamentos de metadados com o mínimo de duplicação;
II. A estrutura do evento é proposta como uma cola de longo prazo entre termos para interoperabilidade de metadados de comércio eletrônico;
III. A estrutura do evento providencia os meios mais eficientes para rastrear mudanças que relacionam entidades persistentes.

De acordo com GODFREY e BIDE (2000, p 13) existem algumas regras sintáticas que um evento deve respeitar para ser classificado como um evento válido do INDECS:

I. Cada entidade em um evento apresenta ao menos um papel expressado como uma relação entre a entidade e o evento;
II. Cada evento tem no mínimo um agente exercendo ao menos um papel de agente;
III. Entidades exercem mais de um papel em um evento;
IV. Duas ou mais entidades exercem o mesmo papel em um evento;
V. Com qualquer evento, todos os papéis não agentes (entrada, saída e contexto) devem ser aplicados diretamente a todos os papéis de agentes;

Figura 15 apresenta um resumo do processo completo de “fazer coisas”.

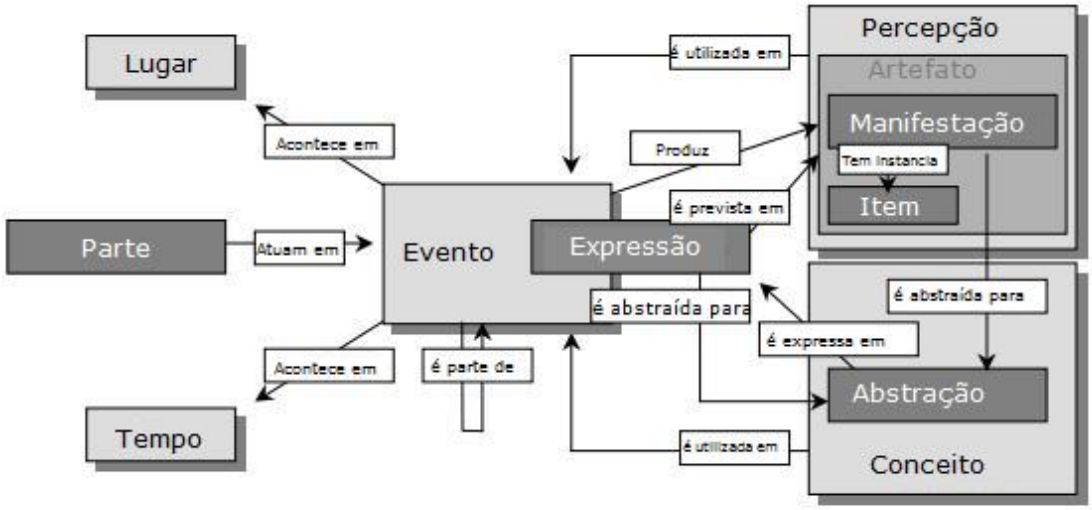


Figura 15 : Um modelo para “fazer coisas” (adaptada de (GODFREY; BIDE, 2000, p 4))

A Figura 16 apresenta todos os tipos de criações e relações que existem entre eles.

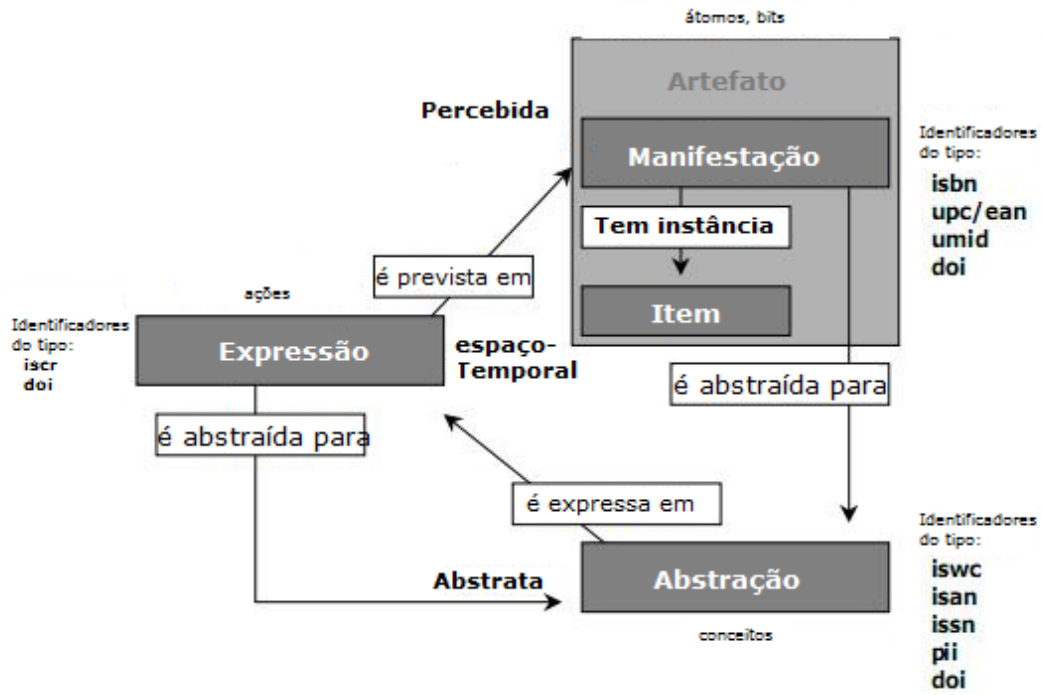


Figura 16: Tipos de criações (adaptada de (GODFREY; BIDE, 2000, p 4))

Popularmente uma abstração é a entidade geralmente chamada de trabalho e uma expressão pode dar origem a um trabalho abstrato, ao mesmo tempo em que pode ser uma expressão de uma obra abstrata já existente (GODFREY; BIDE, 2000, p 13). Contudo enquanto todos os trabalhos são abstrações, todas as abstrações não necessariamente são trabalhos no sentido legal (GODFREY; BIDE, 2000, p 13).

2.8.3 FRBR e INDECS

Enquanto o FRBR foi criado como um modelo preocupado com a perspectiva do usuário, o framework INDECS tem o seu foco na interoperabilidade semântica entre sistemas. Contudo, estes dois aspectos são importantes para nosso trabalho e esta seção mostrará que eles têm características em comum e as definições que podem ser relacionadas.

Figura 17 agrupa as entidades dos frameworks em dois grupos, as entidades conceituais e as entidades físicas. A linha verde representa o relacionamento de equivalência enquanto a azul representa o relacionamento de contingência.

As entidades físicas do FRBR e do INDECS são equivalentes e podem ser mapeadas entre si diretamente.

Através das entidades conceituais as expressões delas são equivalentes e podem ser mapeadas diretamente.

Todas as entidades de trabalho do FRBR podem ser mapeadas como abstrações do INDECS, porém existem abstrações que não podem ser mapeadas como trabalhos do FRBR. Então nos podemos dizer que o grupo de entidades trabalho está contido no grupo de entidades abstração do INDECS.

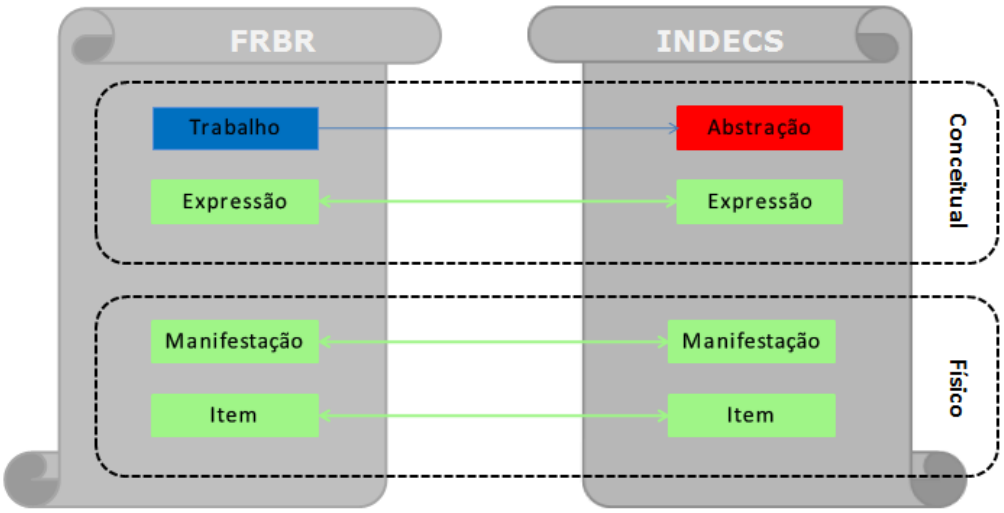


Figura 17: Relacionamentos entre as entidades do FRBR e do INDECS

2.8.4 Metadados

PINHEIRO (2010, p 83) define um modelo de representação de dados e metadados que permite a construção de objetos complexos, de maneira uniforme, a

partir de objetos simples. Este modelo está representado no diagrama da Figura 18 e suas classes são definidas na Tabela 1.

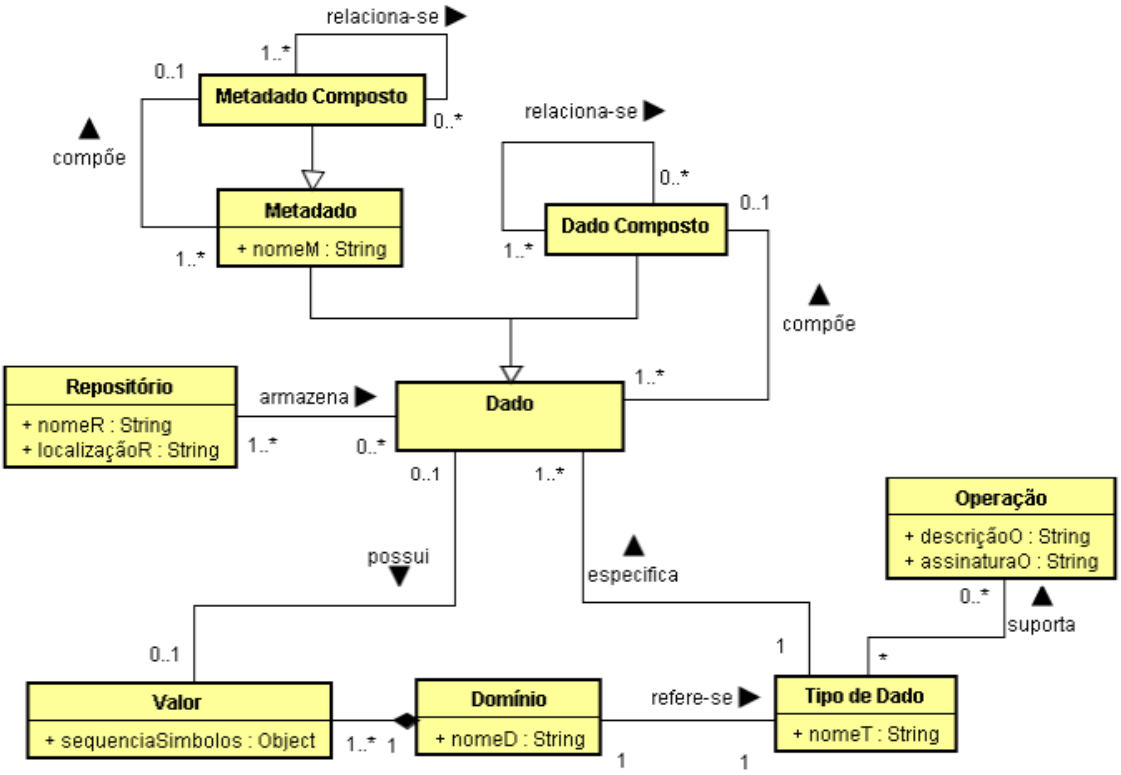


Figura 18: Diagrama do modelo de metadados (extraída de (PINHEIRO, 2010, p 83)).

Classe	Definição
Repositório	local onde ficam fisicamente armazenados os dados e metadados
Dado	é a representação/denotação de um objeto de interesse, podendo ser usado para representar objetos simples de um determinado tipo ou a combinação de objetos simples
Metadado	conjunto de características e atributos que representam ou descrevem um dado. Os metadados geralmente têm a finalidade de localizar, avaliar, descobrir, analisar ou citar o dado a que se referem, não se limitando a essas funções
Valor	é um conjunto de elementos contendo uma sequência de símbolos
Domínio	define um conjunto de valores limitados por um tipo de dado (NationMaster, 2003).
Tipo de dado	classifica dados com características em comum em grupos
Operação	define o conjunto de procedimentos que podem ser realizados sobre certa quantidade de elementos

Tabela 1: Definição das classes do modelo de metadados

Contexto, frequência de acesso, data de atualização e criação são exemplos de características importantes, entre outras, que os metadados podem fornecer ao serem armazenados (PINHEIRO, 2010, p 83). Destaca-se também que, no modelo em questão, existem uma separação entre dados compostos e metadados colocando-o em conformidade com as idéias de MOF[11] (Meta Object Facility) e suas aplicações (PINHEIRO, 2010, p 83).

2.8.5 Dublin metadata core:

Dublin core (DC) pode ser definido como um modelo de metadados para descrever entidades digitais. DC inicialmente foi proposto pela entidade Dublin Core Metadata Initiative (DCMI) como um "cartão de catálogo digital" para a internet, que é simples o suficiente para a compreensão de pessoas que não são especialistas (BAKER, 2009, p 4).

Segundo DEKKERS (2009, p 2) a missão da DCMI é: "Prover padrões simples para facilitar a busca, o compartilhamento e o gerenciamento de informação, desenvolvendo e mantendo padrões internacionais para a descrição de recursos, e assistir uma comunidade de usuários e desenvolvedores pelo mundo e promovendo a disseminação das soluções Dublin Core".

DEKKERS (2009, p 2) apresenta que a ideia inicial do DC era um mecanismo básico de descrição de informação digital que:

I. Pode ser utilizada em todos os domínios;
II. Pode ser utilizada por qualquer tipo de recurso;
III. É simples, porém poderosa;
IV. É extensível e pode trabalhar com soluções específicas.

O primeiro DC proposto em 1995 era composto de 15 elementos básicos: Identificador, Título, Autor, Editor, Assunto, descrição, Segurança, Formato, Tipo, Data, Narrativa, Direitos autorais e idioma.

Desde então a quantidade de elementos e a complexidade dos modelos aumentou ate 2000 quando o modelo era composto de 15 elementos, 36 refinamentos, 17 codificações e 13 tipos (BAKER, 2009, p 13).

DCMI foi apresentado em Andy Powell (2007), como um modelo abstrato que é compostos de outros três modelos:

I. O modelo de recursos (Representado na Figura 19);

11 http://www.omg.org/mof

II. O modelo de conjunto de descrições (Representado na Figura 20);

III. O modelo de vocabulário (Representado na Figura 21);

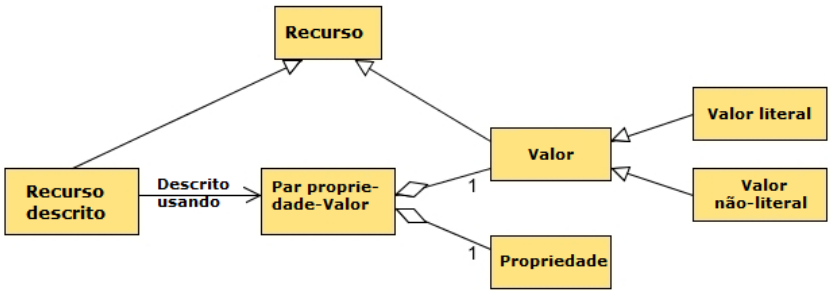


Figura 19: O modelo DCMI de recursos (adaptado de (Andy Powell, 2007)).

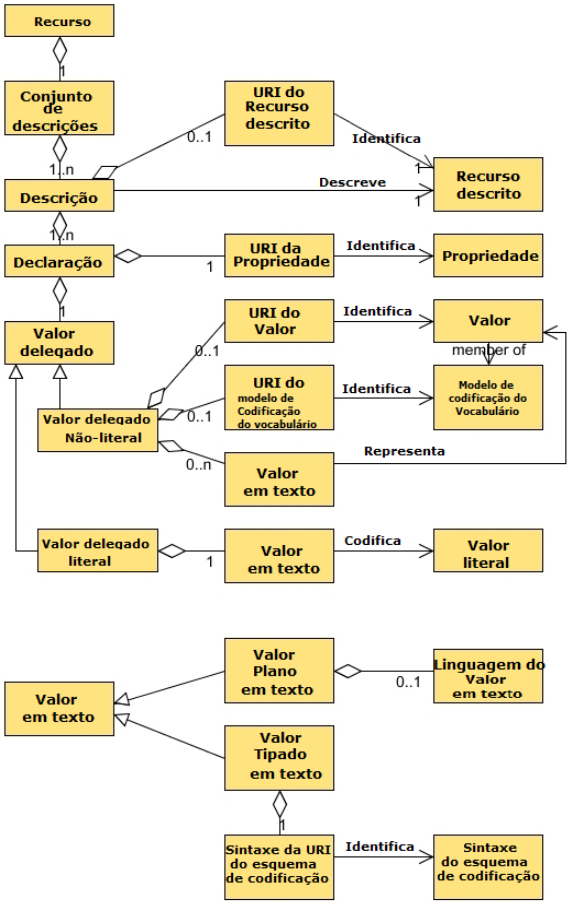


Figura 20: o modelo DCMI do conjunto de descrições (adaptado de (Andy Powell, 2007)).

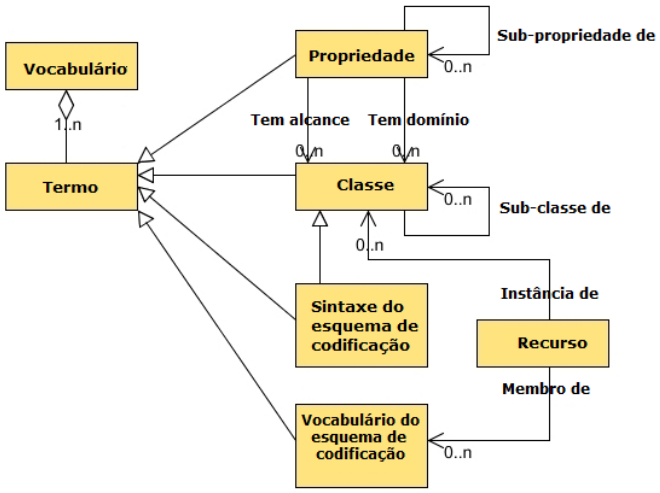


Figura 21: o modelo DCMI de vocabulário (adaptado de (Andy Powell, 2007)).

2.9 Documentos e relacionamentos

Conteúdo é criado, consumido e transformado por pessoas através da execução de atividades de criação intelectual ou artística. A representação e organização deste conteúdo e suas expressões pode ser categorizada por seus relacionamentos.

2.9.1 Relacionamentos bibliográficos

A atividade de catalogar material bibliográfico é feita, analisada e otimizada por bibliotecário por séculos. A meta deles é possibilitar as pessoas a acharem o que precisam e auxiliar na escolha do material por edição ou literatura ou tópico (BEAN; GREEN, 2001, p 19). Organizar materiais por autores, edições, assuntos e outros relacionamentos que podem existir entre eles está incluído nessas atividades (BEAN; GREEN, 2001, p 19).

O relacionamento envolvido na catalogação descritiva de unidades bibliográficas pode ser abrangentemente definido como relacionamento bibliográfico. Desde que esta unidade seja classificada fisicamente ou material como unidade intelectual (BEAN; GREEN, 2001, p 7).

Os relacionamentos bibliográficos podem ser classificados em sete tipos e BEAN e GREEN (2001, p 19) descreve esses tipos:

I. Relacionamentos de equivalência, que estão entre cópias exatas da mesma manifestação de um trabalho ou entre um item original e suas reproduções, desde que o conteúdo intelectual e autoria sejam

preservados. Incluem-se aqui cópias, as questões, réplicas e reproduções, fotocópias, microformas e outras reproduções semelhantes;

II. Relacionamentos derivados, que lidam entre um trabalho bibliográfico e a modificação baseada no trabalho. Eles incluem:
    i. Variações ou versões do mesmo trabalho, como edições, revisões, traduções, sumários, resumos e sumários;
    ii. Adaptações ou modificações que se tornam novos trabalhos mas são baseados em trabalhos antigos;
    iii. Mudanças de gênero, como na dramatização e nas “novelizações”; e
    iv. Novos trabalhos baseados no estilo ou conteúdo temático do trabalho, como as traduções livres, paraphrases, imitações e paródias.

III. Relacionamentos descritivos, os quais estão entre a entidade bibliográfica e uma descrição, uma crítica, uma avaliação ou revisão daquela entidade, como aquelas entre um trabalho e uma revisão de livro descrevendo-o; também incluídas são as edições de anotações, “casebooks”, os comentários, as críticas, etc

IV. Relacionamento parte-todo, os quais estão entre a entidade bibliográfica e uma parte que compõe a entidade, como o caso entre uma antologia e uma seleção individual tirada dela ou entre uma serie e um de seus volumes;

V. Relacionamentos de acompanhamento, os quais estão entre uma entidade bibliográfica e seus materiais de acompanhamento. Em alguns casos uma entidade é predominante e a outra é subordinada a ela, como no caso entre um texto e seus suplementos ou entre uma entidade bibliográfica e outra que provê acesso a ela (isto é, concordâncias, índices bibliográficos, catálogos de bibliotecas). Em outros casos as entidades são de status iguais porem não tem organização cronológica entre elas, como no caso de partes de um kit;

VI. Relacionamentos sequenciais, que estão entre entidades bibliográficas que continuam ou precedem a outra, como entre sucessivos títulos de uma série, seqüências de monografias, ou entre varias partes de uma série numerada;.

VII. Relacionamentos de características compartilhadas, que estão entre entidades bibliográficas que não são outrora relacionadas mas

coincidentemente tem em comum um autor, título, assunto ou outra característica utilizada como ponto de acesso em um catálogo como: linguagem compartilhada, data de publicação ou país de publicação.

Um exemplo de classificação de relacionamentos bibliográficos é representado em (TILLETT, 2003, p 4) e Figura 23 a explicita o exemplo.

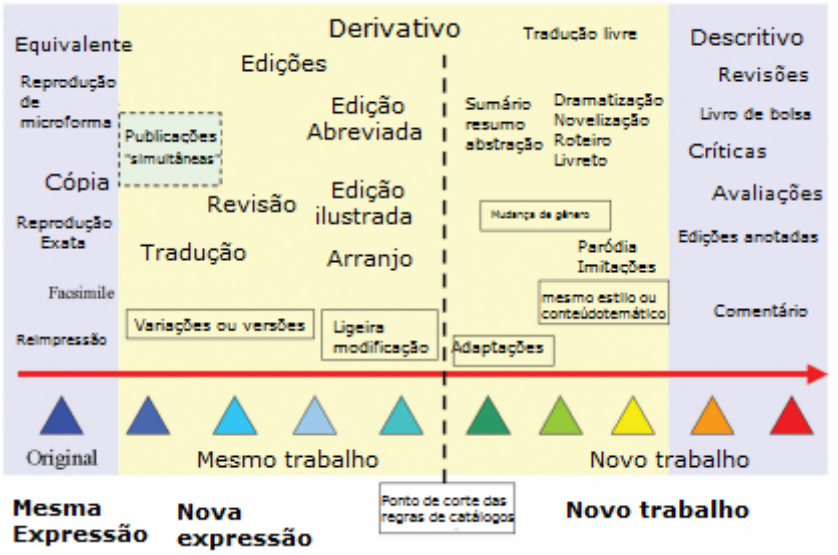


Figura 22: Exemplo de relacionamentos bibliográficos (adaptado de (TILLETT, 2003, p 4)).

2.10 Enterprise content management

O gerenciamento empresarial de conteúdo (em inglês "*enterprise content management*"- ECM) tenta capturar, gerenciar, armazenar, preservar e entregar conteúdo e documentos (AIIM, 2010). Ele permite o gerenciamento de informação não estruturada de uma organização independente da sua localização (AIIM, 2010).

As áreas de aplicação tradicionais de um ECM são gerenciamento de documentos, colaboração, gerenciamento de conteúdo da web, gerenciamento de registros e workflow/gerenciamento de processos de negócios (KAMPFFMEYER, 2004, p 7).

O sistema ECM engloba (AIIM, 2010):

I. Indexação de dados e metadados – Suportando a atividade de recuperação de informação;
II. Gerencia e rastreia documentos, emails e conteúdo da web;
III. Gerenciamento de ativos digitais – Similar ao gerenciamento e rastreamento de documentos salvo que ele foca em documentos ricos em mídia;

Soluções ECM são uma combinação de vários tipos de tecnologias (KAMPFFMEYER, 2004, p 7). Estas tecnologias podem ser utilizadas como soluções independentes de uma plataforma que as incorporem (KAMPFFMEYER, 2004, p 7).

KAMPFFMEYER (2004, p 7) categoriza essas tecnologias como:

I. Capturar,
II. Gerenciar,
III. Armazenar,
IV. Entregar, Disponibilizar em longo prazo
V. Preservar.

Estas categorias estão representadas na Figura 23 e este modelo está fundamentado nas cinco categorias principais do AIIM Internacional.

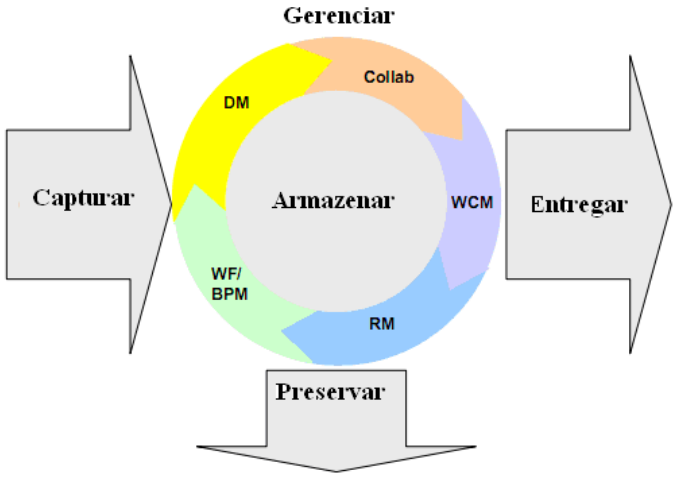


Figura 23: Categorias dos componentes de um ECM. (extraída de (KAMPFFMEYER, 2004, p 7))

A categoria Capturar gera, captura, prepara e processa tanto informação digital quanto analógica através de suas funcionalidades e componentes (KAMPFFMEYER, 2004, p 7).

Vários tipos de informação são abordados pela captura manual, elas estão representadas desde documentos em papel ate documentos eletrônicos, e-mails, formulários e vídeos entre outras formas de representação (KAMPFFMEYER, 2004, p 7).

Documentos XML, aplicações de negócios e ERP ou aplicações específicas são fontes para a captura automática ou semi-automática (KAMPFFMEYER, 2004, p 7).

A categoria Gerenciar incorpora bases de dados para administração e recuperação, sistemas de autorização de acesso para o gerenciamento, processamento e utilização de informação (KAMPFFMEYER, 2004, p 7).

A meta de um sistema ECM é prover estes dois componentes como um único serviço para todas as soluções de gerenciamento como gerenciamento de documentos, Colaboração, gerenciamento de conteúdo da web, gerenciamento de registros e workflow/gerenciamento de processsos de negócios (KAMPFFMEYER, 2004, p 7).

A categoria Armazenar é um armazenamento temporário para informação que não é desejável ou necessária para arquivar (KAMPFFMEYER, 2004, p 7). Os componentes desta categoria podem ser divididos em três categorias: Repositórios (como locais de armazenamento), Serviços de biblioteca (como componentes de administração para repositórios) e “tecnologias” de armazenamento (KAMPFFMEYER, 2004, p 7).

A categoria Preservar lida tanto com o armazenamento, de forma segura, de longo prazo e cópias de segurança da informação estática quanto do armazenamento temporário de informação que não é desejável ou necessária para arquivar (KAMPFFMEYER, 2004, p 7).

A categoria Entregar serve para apresentar a informação das categorias: Gerenciar, Armazenar e Preservar (KAMPFFMEYER, 2004, p 7). Esta categoria compreende três grupos de funções e mídias: Tecnologias de transformação, tecnologias de segurança e distribuição. As tecnologias de transformação e segurança devem pertencer a um único serviço e devem ser disponibilizadas para todos os componentes do ECM de forma igual (KAMPFFMEYER, 2004, p 7).

AMBRIOLA et al. (1990) apresenta algumas características interessantes sobre o versionamento:

I. Controle de versionamento pode ser definido como a atividade de controlar versões dos componentes de um componente em particular;
II. As metas do controle de versões são facilidade, eficiência, recuperação e armazenamento de várias versões do mesmo componente e aplicação de restrições na evolução de um componente para que tal troca seja observável e controlável;
III. Um componente é a unidade básica na qual um sistema é montado. Ela pode ser atômica ou composta de outros componentes, por exemplo, a modelagem de um maquinário, um capítulo de livro e as partes (módulos) de um programa.

Os sistemas de controle de versões descentralizados (DVCS)[12] são apresentados em (DE ALWIS; SILLITO, 2009) como os sistemas que irão atender algumas das lacunas não preenchidas pelos sistemas de controle de versões descentralizados (CVCS)[13].

DE ALWIS e SILLITO (2009) lista alguns dos benefícios que os DCVS tem comparados aos CVCS, se você quer uma comparação mais detalhada dos sistemas de versões você a encontrará em (RAYMOND, 2009).

2.11 Modelos de busca e recuperação de informação

Nesta seção serão apresentados os modelos de BRI (Busca e Recuperação da Informação), como eles podem ser classificados, passando desde os modelos estruturados, definindo e abordando como a hierarquia pode ser endereçada no modelo e, finalmente, comparando alguns modelos estruturados de nosso interesse.

2.11.1 Classificação dos modelos

Com o aumento da capacidade de armazenamento, a aparição de novas tecnologias como a internet e a aplicação de sistemas computacionais com grande quantidade de dados (por exemplo, bioinformática) vários tipos de modelos de BRI foram propostos para tentar suprir um paradigma em particular ou, de forma mais abrangente, todos os paradigmas.

BAEZA-YATES et al. (1999, p 21) propõe uma taxonomia de modelos de BRI, focando na organização destes modelos, a qual é exposta na Figura 24.

12 Exemplos de sistemas de controle de versões descentralizados são: GIT [http://git-scm.com/], MERCURIAL [http://mercurial.selenic.com/] ,BZR [bazaar-vcs.org/] and BITKEEPER [http://bitkeeper.com/].

13 Exemplos de sistemas de controle de versões centralizados são: CVS (BERLINER; PRISMA, 1990 e GRUNE, 1986) e Subversion (COLLINS-SUSSMAN et al., 2007)

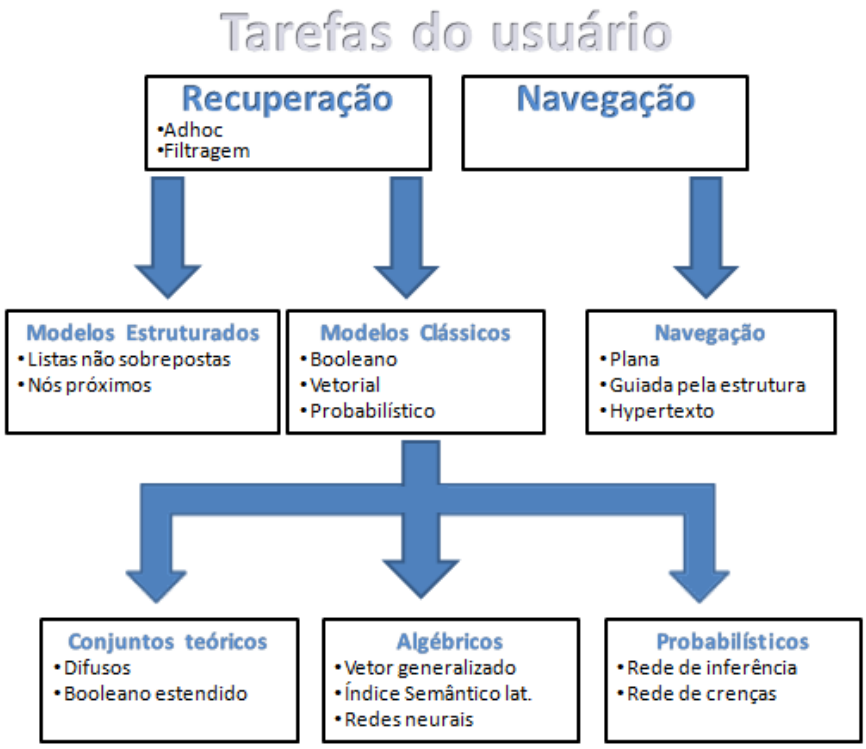


Figura 24: Taxonomia de modelos BRI (adaptada de (BAEZA-YATES, R. et al., 1999, p 21))

De acordo com BAEZA-YATES et al. (1999, p 24) o modelo clássico de BRI considera que cada documento é descrito por um conjunto representativo de palavras chave chamado de índice de termos. Um termo é uma simples palavra na qual a semântica irá lembrar os temas principais do documento. Enquanto os modelos com estrutura combinam informação do conteúdo do texto com a informação na estrutura do documento (BAEZA-YATES, R. et al., 1999, p 62).

2.11.2 Clássicos × estruturados

Recuperação usando índices de termos adota como alicerce fundamental a ideia de que a semântica dos documentos e a informação do usuário devem ser naturalmente expressas através de conjuntos de índices de termos. Claro que isto é uma considerável simplificação do problema, pois, uma grande quantidade de semântica em um documento ou requisição de usuário é perdida quando nós trocamos o texto por um conjunto de palavras. Além disto, a comparação entre documentos e a requisição do usuário é realizada neste espaço impreciso que é um índice de termos (BAEZA-YATES, R. et al., 1999, p 19).

Como clássico compreendemos que o modelo não integra conteúdo e estrutura em consultas. Contudo, eles atendem outros problemas que os novos modelos não

resolvem de forma geral (por exemplo, tuplas e joins, ranking por relevância, e questões de implementação como segurança, tolerância a falhas e concorrência) (BAEZA-YATES, R.; NAVARRO, G., 1996, p 68).

Algumas tentativas foram feitas para integrar a buscar de texto estruturado nos bancos de dados orientados a objetos, que geralmente resultam em expressar a estrutura como uma rede hierárquica, conectada por partes de atributos (BAEZA-YATES, R.; NAVARRO, G., 1996, p 68). Estas abordagens são caracterizadas por uma imposição generalizada de uma estrutura hierárquica no banco de dados e pela mesclagem de consultas no conteúdo e na estrutura. Porém, o problema de mesclagem de conteúdo e estrutura não foi satisfatoriamente resolvido (CIANCARINI, 1996, p 404).

Para os modelos de recuperação de texto estruturado a questão de ranquear não esta bem estabelecida, visto que, não existe a noção de relevância acoplada à tarefa de recuperação (BAEZA-YATES, R. et al., 1999, p 63).

2.11.3 Modelo estruturado

Alguns dos modelos clássicos não permitem expressar estruturas suficientemente ricas, outros são muito orientados ao conteúdo, e outros a estrutura (GROZA; HANDSCHUH, 2009, p 93). Mesclar conteúdo e estrutura em consultas nos permite realizar consultas muito poderosas sendo muito mais expressivo que cada mecanismo por si só (GROZA; HANDSCHUH, 2009, p 93 e BAEZA-YATES, R.; NAVARRO, G., 1996, p 67).

Uma questão importante a se ponderar em um modelo deste tipo é que a informação tem que ser extraída do texto, mas não de uma forma rígida (BAEZA-YATES, R.; NAVARRO, G., 1996, p 68).

Modelos de busca e recuperação que combinam a informação contida no conteúdo do texto com a informação contida na estrutura do documento são chamados de modelos estruturados de recuperação de texto (BAEZA-YATES, R. et al., 1999, p 62).

Tais modelos textuais consistem de três elementos: a forma como as características de texto são abstraídas, as operações que são necessárias para acessar a informação no armazenamento e as restrições na representação do texto e no seu acesso (LOEFFEN, 1994, p 96).

Recuperação de texto estruturado requer que a indexação de elementos estruturais e a linguagem de consulta sejam capazes de referenciar diretamente estes elementos estruturais. Em sistemas tradicionais de recuperação de texto a indexação de estrutura é normalmente limitada a elementos pré-definidos como palavras,

sentenças e parágrafos (CLARKE; CORMACK; BURKOWSKI, 1995a, p 1). Em grandes coleções de texto ou coleções de texto especializadas os documentos têm mais tipos de estruturas do que podem ser capturados por estes três simples elementos (CLARKE; CORMACK; BURKOWSKI, 1995a, p 1).

A estrutura de cada documento deve ser indexada da forma mais apropriada para o documento em particular, na qual, elementos estruturais análogos, em documentos diferentes, são indexados de forma idêntica ou facilmente relacionados entre si (CLARKE; CORMACK; BURKOWSKI, 1995a, p 1).

Modelos estruturados combinam as propriedades de bancos de dados fortemente estruturados e texto livre. Técnicas de consultas para tais documentos necessitam que ambos os aspectos sejam considerados (SCHWARZ, 1978, p 214). Outro aspecto importante de gerenciar dados textuais estruturados e semi-estruturados consiste em suportar a eficiência dos componentes de recuperação de texto baseados tanto em conteúdo quanto em estrutura (CONSENS; MILO, 1995, p 11).

2.11.4 Modelo híbrido

Este modelo conceitualiza uma base de dados textual como um conjunto de documentos que podem conter campos estes campos podem se mesclar e sobrepor, porém não são obrigados a cobrir todo o texto do documento(BAEZA-YATES, R.; NAVARRO, G., 1996). Existem operações de união, intersecção, diferença e complemento definidas para este modelo (BAEZA-YATES, R.; NAVARRO, G., 1996).

Este modelo é bem simples e, por conta deste fato, pode ser implementado de forma bem eficiente. Porém, este modelo é plano e não composicional, visto que não é possível fazer algumas composições de expressões que envolvam seus campos (BAEZA-YATES, R.; NAVARRO, G., 1996).

2.11.5 Expressões PAT

O modelo PAT é conhecido por apresentar uma álgebra para consultar texto que têm um mecanismo de avaliação bem eficiente (BAEZA-YATES, R.; NAVARRO, G., 1996). Para tanto, ele usa um único índice baseado em “match points” que são utilizados para definir regiões (BAEZA-YATES, R.; NAVARRO, G., 1996). Portanto, não há indexação permanente de elementos da estrutura e existe uma linguagem que permite a indexação dinâmica desses elementos. Embora essa abordagem de definição dinâmica seja flexível, ela recai em requisitos especiais de marcação que

devem permitir a definição de regiões através de expressões baseadas em “match points” (BAEZA-YATES, R.; NAVARRO, G., 1996).

2.11.6 Listas sobrepostas

CLARKE et al. (1995b) propõe uma álgebra de consultas que tira partido dos relacionamentos de contingencia entre níveis de uma estrutura hierárquica de documento pré-definida. O modelo de expressões PAT é uma proposta bem próxima do espírito deste modelo, contudo o modelo de listas sobrepostas resolve os problema das expressões PAT permitindo sobreposições, porém ele não permite aninhamento de elementos (CLARKE; CORMACK; BURKOWSKI, 1995b e NAVARRO, GONZALO; BAEZA-YATES, RICARDO, 10995).

A idéia original era gerar listas planas de segmentos disjuntos, mas em (BAEZA-YATES, R.; NAVARRO, G., 1996) a álgebra é aprimorada com capacidade de sobreposição. Logo, cada região é uma lista de segmentos, que são aptos a se sobrepor, originados de buscas de texto ou por regiões previamente marcadas (como capítulos, por exemplo) (NAVARRO, GONZALO; BAEZA-YATES, RICARDO, 10995).

2.11.7 Nós próximos

No modelo de nós próximos elementos aninhados são permitidos na resposta, porém não há sobreposição (BAEZA-YATES, R.; NAVARRO, G., 1996). Neste modelo cada hierarquia é chamada de visão, e este nome é sugerido como uma forma independente de ver o texto. Cada visão tem um conjunto de construtores que denotam o tipo de nós na árvore correspondente (por exemplo, páginas, capítulos ou seções) (NAVARRO, GONZALO; BAEZA-YATES, RICARDO, 10995).

Para cada nó da visão existe um construtor associado e um segmento que é um par de números que representam uma parte contígua do texto. Cada segmento de um nó deve conter os segmentos de seus nós descendentes (NAVARRO, GONZALO; BAEZA-YATES, RICARDO, 10995).

As consultas são modeladas para retornar nós de uma hierarquia de texto em especial. O modelo especifica uma linguagem completamente composicional com três tipos de operadores: Casamento de padrões de texto, recuperação de componentes de estruturais por nome e combinação de resultados (BAEZA-YATES, R.; NAVARRO, G., 1996).

O texto é considerado estático, a estrutura é praticamente estática também. Isto é, embora o modelo permita a construção de novas hierarquias, a alteração e a

remoção delas, a meta não é fazer o uso dessas operações de forma continua e pesada (NAVARRO, GONZALO; BAEZA-YATES, RICARDO, 10995).

2.11.8 Comparação de árvores

A Idéia deste modelo é combinar tanto a estrutura do banco de dados e a consulta, um padrão para estrutura, como árvores para encontrar um padrão embutido no banco de dados que respeitam os relacionamentos hierárquicos entre os nós do padrão (NAVARRO, GONZALO; BAEZA-YATES, RICARDO, 10995).

O uso de variáveis lógicas que este modelo emprega faz o problema se tornar intratável (NP-difícil em alguns casos) e, mesmo sem elas, a inclusão de arvores não-ordenadas é NP-completo (BAEZA-YATES, R.; NAVARRO, G., 1996 e SCHWARZ, 1978).

2.11.9 P-strings

Este modelo utiliza de gramáticas como estrutura e “parsed strings” como instâncias. Os operadores nessas instâncias foram definidos e resultaram numa álgebra de p-string que pode ser utilizada para manipulação de dados e vizualização (GONNET et al., 1987).

No modelo a linguagem utilizada para expressar a base de dados é uma gramática livre de contexto, que é a estrutura da base de dados (10). A estrutura de dado fundamental é a p-string que é composta de derivações da árvore mais o respectivo texto (BAEZA-YATES, R.; NAVARRO, G., 1996).

O problema para este modelo é a eficiência, pois sendo uma abordagem tão dinâmica é difícil de programar-la tão eficientemente (BAEZA-YATES, R.; NAVARRO, G., 1996).

2.11.10 Lista de referencias

Na lista de referências a estrutura do documento pode ser hierárquica (sem sobreposição) mas as respostas a consultas são planas e todos os elementos devem ser do mesmo tipo (por exemplo, apenas seções ou parágrafos). As respostas a consultas são apresentadas como lista de referencias, isto é, ponteiros para o banco de dados. Este modelo é muito poderoso e, por conta disto, difícil de desenvolver de forma eficiente (NAVARRO, GONZALO; BAEZA-YATES, RICARDO, 10995).

2.11.11 Classificação dos modelos estruturados

BAEZA-YATES e NAVARRO (1996) classificam os modelos de acordo com a forma que ele delimita o documento:

I. Modelos fortemente delimitados por estrutura – Apresenta a estrutura tão separada quanto for possível do texto. Eles são: Comparação de árvores e lista de referências;

II. Modelos fortemente delimitados pelo texto – Normalmente põe a estrutura no mesmo texto, i. e., ele utiliza a estrutura implícita. Eles são: Expressões PAT, Modelo Híbrido e Listas Sobrepostas. O modelo de listas sobrepostas tira vantagem completamente da flexibilidade permitida por essas estruturas implícitas;

III. Modelos intermediários – se importam da mesma forma com o texto e a estrutura, i.e., embora a estrutura tenha uma identidade separada e seja possível manipulá-la de forma independente do texto, o texto por si só pode ser recuperado e operado da mesma forma que nós de estruturas. Eles são Nós Próximos e P-strings.

Modelos de estruturas dinâmicas permitem fácil reindexação e as consultas normalmente são rápidas também e as vantagens de estruturação estática está em sua característica relacionada: A estruturação explícita (BAEZA-YATES, R.; NAVARRO, G., 1996).

2.12 Engenharia de documentos

A engenharia de documento é uma disciplina que estuda metodologias e técnicas de modelagem que geram modelos significativos e reutilizáveis para troca de informação entre empresas (GLUSHKO; MCGRATH, 2005). Esta abordagem de utilizar documentos como entrada e saída para processo de negócio é ótima em ambientes tecnologicamente heterogêneos (GLUSHKO; MCGRATH, 2005).

GLUSHKO e MCGRATH (2005) define a engenharia de documentos como: “um conjunto de análises e técnicas de modelagem que fornecem modelos de processos de negócio, e os seus documentos, reutilizáveis e significativos”.

A engenharia de documentos baseia-se no modelo clássico de três camadas adaptado para uma abordagem centralizada em documentos (GLUSHKO; MCGRATH, 2005). As camadas são: (GLUSHKO; MCGRATH, 2005)

I. Representação externa – Descreve coisas, artefatos ou instâncias do mundo, de forma específica;

II. Visão interna – apresenta diferentes modelos de instâncias em determinada tecnologia;
III. Visão conceitual – abstrai as descrições de implementações dando ênfase ao relacionamento semântico entre classes ou instâncias;

GLUSHKO e MCGRATH (2005) apresenta uma forma de classificar os modelos, apresentada na Figura 25, de acordo com: os três níveis de abstração e a granularidade que o modelo atende.

A classificação superior esquerda indica que os modelos são mais abrangentes em escopo e abstratos em perspectiva, permitindo assim, uma melhor compreensão do que o negócio faz e os relacionamentos existentes no negócio (GLUSHKO; MCGRATH, 2005). Esta classificação granular é importante para estabelecer os requisitos e regras para os documentos em cada escopo de acordo com cada granularidade (GLUSHKO; MCGRATH, 2005). Expressar estas regras através de modelos nos permite estruturar os serviços do negócio de forma a facilitar a interoperabilidade entre duas organizações, visto que, ao se compreender o modelo da outra organização garantimos conhecer o contexto da mesma (GLUSHKO; MCGRATH, 2005).

Granularidade	Organização			
	Processo			
	Informação			
		Modelos conceituais	Modelos físicos	Instâncias
	Abstração			

Figura 25: Matriz de classificação de modelos (adaptada de (GLUSHKO; MCGRATH, 2005))

A proposta da engenharia de documento é a de utilizar o conhecimento da análise de processos, documentos, dados e tarefas na criação de modelos para processos de negócios e seus documentos (GLUSHKO; MCGRATH, 2005).

A análise de processos normalmente começa com visões abstratas dos modelos de negócios e processos. Com ela, o contexto para compreender a semântica da informação é estabelecido (GLUSHKO; MCGRATH, 2005).

A análise de tarefas é baseada em elementos e atividades atuais, ela identifica os passos e a informação necessários para realizar uma tarefa (GLUSHKO; MCGRATH, 2005).

A análise de documentos extrai ou esclarece os componentes de apresentação, estrutural e de conteúdo de documentos ou outras fontes de informação (GLUSHKO; MCGRATH, 2005).

A análise de tarefas e a análise de documentos são extremamente relacionadas, pois, a análise de documentos revela os componentes de informação candidatos e a análise de tarefas revela as regras sobre eles e seus usos (GLUSHKO; MCGRATH, 2005).

A análise de dados normalmente começa de uma perspectiva conceitual de um domínio e alcança uma visão abstrata dos componentes revelados pela análise de documentos (GLUSHKO; MCGRATH, 2005).

Cada uma das análises parte de uma extremidade da matriz e segue em direção da parte central onde ocorre um ponto de sobreposição que é o objetivo da engenharia de documentos.

Para a engenharia de documentos, uma importante linha de pesquisa é a de transformar documentos em formas que favoreçam o reuso de seu conteúdo (LECERF; CHIDLOVSKII, 2006). Na próxima seção o reuso de documentos será abordada assim como seus aspectos, teorias, técnicas e ferramentas.

2.12.1 Reuso de documentos

Reuso é um grande interesse na criação e utilização de documentos (LEVY, 1993). O trabalho de criação de novos documentos, ou de novas versões de documentos, envolve o reuso de pedaços de documentos existentes, onde reutilizar aborda encontrar material relevante, modificá-lo para atende a necessidade em questão e compilar os pedaços juntos (LEVY, 1993). Esta estratégia de reutilizar informação para criar nova informação é aplicada na escrita (STREITZ; HANNEMANN; THÜRING, 1989) e é bem difundida (HOLLAND, 1992). Por exemplo, na escrita de obras que não são ficção, é uma prática muito comum utilizar pedaços de materiais existentes para a construção de documentos (BARTA; GIL, 1996). Outros exemplos são a engenharia de software, a escrita jurídica e o domínio pedagógico onde o reuso ou as adaptações de documentos como manuais, contratos e objetos de aprendizado são práticas essenciais e que tem uma demanda crescente de tecnologia para atender suas respectivas atividades (BRANTING; LESTER, 1996 e DOWNES, 2001 e DUVAL et al., 2001 e VERBERT et al., 2008).

Os problemas de manter documentos tendem a aumentar com freqüente mudanças por indivíduos diferentes (BRANTING, K.; LESTER, J. C., 1996). Isto ocorre, principalmente quando não se tem acesso ao autor do documento, pois a intenção com a qual determinada parte do documento foi criada pode se perder, dificultando ou ate inviabilizando a compreensão do documento (BRANTING, K.; LESTER, J. C., 1996). Portanto, o reuso de documentos é benéfica porque além de

reduzir o tempo de elaboração do documento ela provê consistência no estilo e no conteúdo(BRANTING, K.; LESTER, J. C., 1996).

O processo de gerenciamento de conhecimento é composto por quatro passos básicos: encontrar/criar, organizar, compartilhar e usar/reutilizar (BURK, 1999). O passo encontrar/criar preocupa-se com o conhecimento/informação gerado por pesquisa ou especialidade em alguma área. Os passos de organizar e compartilhar inicialmente filtram e organizam o conhecimento, para, em seguida, o conhecimento ser compartilhado permitindo uma ampla disponibilidade (BURK, 1999). A etapa de usar/reutilizar permite a utilização e o reuso do conhecimento compartilhado para minimizar a sobrecarga de informação e maximizar a usabilidade do conteúdo, o que diminui o tempo, o esforço e o custo (BURK, 1999).

Uma proposta parecida no domínio de criação de novos documentos é apresentada em (LEVY, 1993). A proposta define o processo da criação de novos documentos e novas versões de velhos documentos em quatro termos:

I. Criação – é a produção de novo material. Por exemplo, através de manuscritos ou digitação de textos;
II. Coleção – é a identificação e a compilação de material que já existia só que separadamente;
III. Combinação – é costurar um material novo e um antigo de forma a criar uma nova unidade;
IV. Customização – é retrabalhar em cima de um novo material para que ele se adeque a uma nova configuração.

Coleção, combinação e customização envolvem Reuso enquanto apenas a criação introduz um novo material (LEVY, 1993).

“Copiar e colar” é o método mais difundido de Reuso de documentos, porém, quando a informação é copiada desta forma, ela se torna um conteúdo associado a o novo documento sem qualquer rastro de onde ela foi retirada (BARTA; GIL, 1996). Este método de reuso é limitado de varias formas que (BARTA; GIL, 1996) destaca:

I. Duplicação – A informação é duplicada em vários documentos. Se existem varias versões do mesmo documento apenas a mais recente será a atualizada;
II. Trabalho intensivo – A atualização de alterações e retificações da informação tem que ser feita em todos os documentos de forma manual;
III. Propenso a erros – É difícil localizar e atualizar da mesma forma todas as cópias.

A representação da estrutura de documentos de forma consistente é um aspecto importante quando se reutiliza documentos. Quatro características foram identificadas em (LEVY, 1993) e são apresentadas na Tabela 2.

Características	O que é?	Formas de reuso
Intercâmbio	Copiar material de um ambiente para outro	Transporta material possibilitando a coleção
Estrutura composta	Especificar como os pedaços podem ser agrupados para forma uma nova entidade	Possibilita a combinação, a extração e a recombinação
Independência de apresentação	A representação independe das características de apresentação	Base para customização para diferentes realizações físicas
Meta-estrutura	A representação é abstrata e aborda vários esquemas de representação de documentos	Meta customizações

Tabela 2 : Características estruturais para reuso de documentos (adaptada de (LEVY, 1993))

Intercâmbio é uma forma de mover material entre ambientes onde o material pode ser coletado, combinado e, por conseguinte, reutilizado (LEVY, 1993). Estrutura composta é a forma de especificar como os pedaços são combinados e então reutilizados (LEVY, 1993). Independência de apresentação é uma forma de representar documentos para fácil customização, para diferentes tipos de mídia e é considerada outra forma de Reuso (LEVY, 1993). Meta-estrutura envolve uma forma de Reuso diferente das anteriores ela permite o reuso das técnicas de representação do documento ao invés de reutilizar o documento (LEVY, 1993).

GUERRIERI (1998) define alguns tipos de reuso que podem ocorrer em documentos:

I. Reuso de conteúdo de documento – aborda o reuso da informação contida no documento;
II. Reuso da estrutura do documento – aborda o reuso da estrutura do documento como, por exemplo: título, autor, parágrafos, capítulos, apêndices entre outros;
III. Reuso do estilo do documento – aborda o reuso da informação de estilo como, por exemplo: tipo de fonte, estilo de fonte, espaçamento entre outros;
IV. Reuso de renderização de documento – aborda o reuso pela renderização dele em diferentes dispositivos como, por exemplo, CDROM, navegadores de internet, dispositivos de braile, pdf entre outros.

2.12.1.1 Ferramentas de Reuso de documentos

Ferramentas e metodologias foram propostas para atender os requisitos da reuso de documentos. Entre elas destacamos:

I. MicroSystems docXtools software é uma coleção ferramentas de avaliação, limpeza, e resolução de problemas para documentos que ajudam os advogados e as equipes jurídicas a melhorar a produtividade e prevenir problemas relacionados a documentos (Microsystems, 2011). Ela verifica a integridade, corrige estilos e avalia a qualidade de determinado documento de acordo com a sua formatação.

II. RADA (1990) representa o conteúdo dos documentos em uma rede semântica, isto é, a abstração do texto é uma rede semântica na qual os caminhos transversais na rede representam novos documentos.

III. LECERF e CHIDLOVSKII (2006) apresenta um sistema que gera anotações semânticas orientadas a layout em documentos. Nele são apresentados métodos probabilísticos e técnicas de aprendizado ativo[14] para treinar o modelo de anotações de documentos.

IV. BRANTING e LESTER (1996) Elaboram documentos a partir de modelo composto de taxonomias e anotações para representar a estrutura da informação a ser reutilizada. Essas anotações são a base para a busca e recuperação das partes que vão compor o novo documento a ser gerado.

V. TITEMORE et al. (2007) associa regras de processo de negócios a documentos ou a pedaços de documentos permitindo assim o reuso e elaboração de novos documentos através das regras informadas.

VI. Boukottaya et al. (2006) propõe uma ferramenta para transformação de estruturação automática que converte um documento estruturado, a partir de um modelo, em um novo documento com outra estrutura, permitindo assim o reuso do seu conteúdo e outra ferramenta que auxilia e acompanha alterações e a evolução de um documento.

VII. BARTA e GIL (1996) apresentam um modelo para representação de documentos e seu conteúdo. Faz um protótipo de ferramenta em cima de uma biblioteca de documentos e seus pedaços. Um escritor pode

---

14 aprendizado ativo é uma forma de aprendizado de máquina supervisionado na qual o algoritmo de aprendizado capaz de interagir com a consulta do usuário para obter o resultado desejado (SETTLES, 1994).

reutilizar o conteúdo existente na biblioteca durante a elaboração de um novo documento.

Um ponto interessante sobre as propostas anteriores é que elas apresentam como requisito que a estrutura do documento seja apontada pelo usuário de alguma forma para que seu modelo possa armazenar e reutilizar o conteúdo futuramente.

3 *UMA ONTOLOGIA DE DOCUMENTOS DIGITAIS*

Este capítulo apresenta o problema de comparar documentos de texto. Para tanto, exploramos e conceitualizamos as características e a estrutura dos documentos, os relacionamentos entre dois documentos, uma ontologia no domínio da manipulação de documentos, as operações possíveis para transformar um documento no outro e uma álgebra que identificará o relacionamento gerado a partir das operações realizadas.

3.1 Definição da ontologia de documentos digitais

A ontologia de documentos digitais é proposta para conceitualizar o domínio da manipulação de documentos. Para realizar esta tarefa a ontologia captura a estrutura dos documentos, identifica os relacionamentos existentes entre documentos, o conteúdo dos documentos, e suas respectivas estruturas.

A Figura 26 apresenta a representação ontológica da estrutura do documento e seus dados relacionados. Ela apresenta que um documento é composto de uma ou mais partes de documentos e cada parte pode ser composta de subpartes. As partes de documentos e suas respectivas subpartes são representadas pela entidade Conteúdo.

As relações de “tem parte” existentes entre a entidade documento e as entidades conteúdo possibilitam a representação do conteúdo do documento de forma linear. Enquanto as relações de “tem subparte” entre entidades Conteúdo permitem que o conteúdo do documento apresente, além de uma característica linear, uma representação de forma hierárquica.

Para cada entidade Documento ou Conteúdo existe um conjunto de entidades Metadado. Estas entidades representam informações de criação (por exemplo: Autores, data de criação e local de criação), acesso (por exemplo: Permissões e localização da entidade) ou outro metadado relacionado com aquela entidade. O modelo de metadados é apresentado na seção 2.8.4.

Informações de estilo como, por exemplo, tipo, tamanho ou cor de fonte, espaçamento e posicionamento também são representadas como instâncias de entidades Metadados.

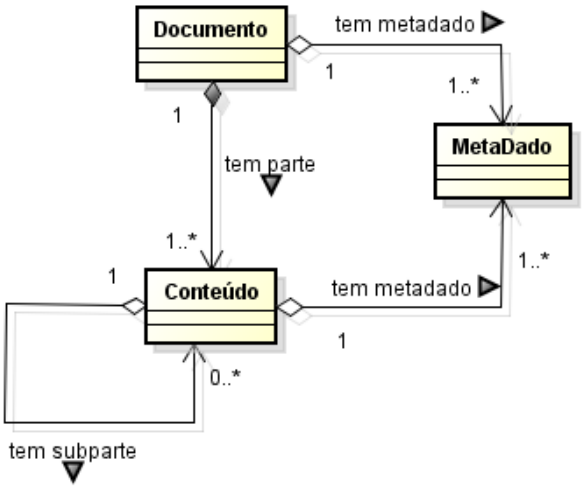


Figura 26: Visão geral do documento

A granularidade da estrutura do texto é uma questão importante. Esta questão determina o escopo e a eficiência do modelo, i.e., quanto mais refinado for o modelo, mais abrangente será sua cobertura, contudo mais custo terá sua computação. A característica hierárquica e linear apresentada na Figura 26 torna o nosso modelo flexível o suficiente para atender este requisito de granularidade. A Figura 27 apresenta como essa granularidade é atendida pelo nosso modelo no caso de entidades conteúdo do texto. Neste caso, a granularidade é apresentada na forma de parágrafos, sentenças e palavras.

A entidade Conteúdo representa uma generalização flexível que atende todas as formas de se representar o conteúdo de um documento. Figura 27 explicita três formas diferentes que o Conteúdo pode ser especializar, elas são:

- Tabela – Ela é composta por um conjunto de Células, onde, uma Célula é uma especialização da entidade Conteúdo. Exemplos de Metadados relacionados às células são o índice da linha da célula e o índice da coluna da célula;
- Imagem – Ela é composta por um conjunto de Pixels, onde, um Pixel é uma especialização da entidade Conteúdo. Exemplos de Metadados relacionados a um Pixel são informações de posicionamento do pixel e Codificação do pixel no sistema RGB;
- Conteúdo de texto – Pode ser especializado em Parágrafos, Sentenças e Palavras de acordo com a granularidade escolhida para a representação. O Metadado relacionado com estas entidades que queremos destacar é o “Pedaço do texto” que representa o trecho do texto do documento que aquela entidade representa.

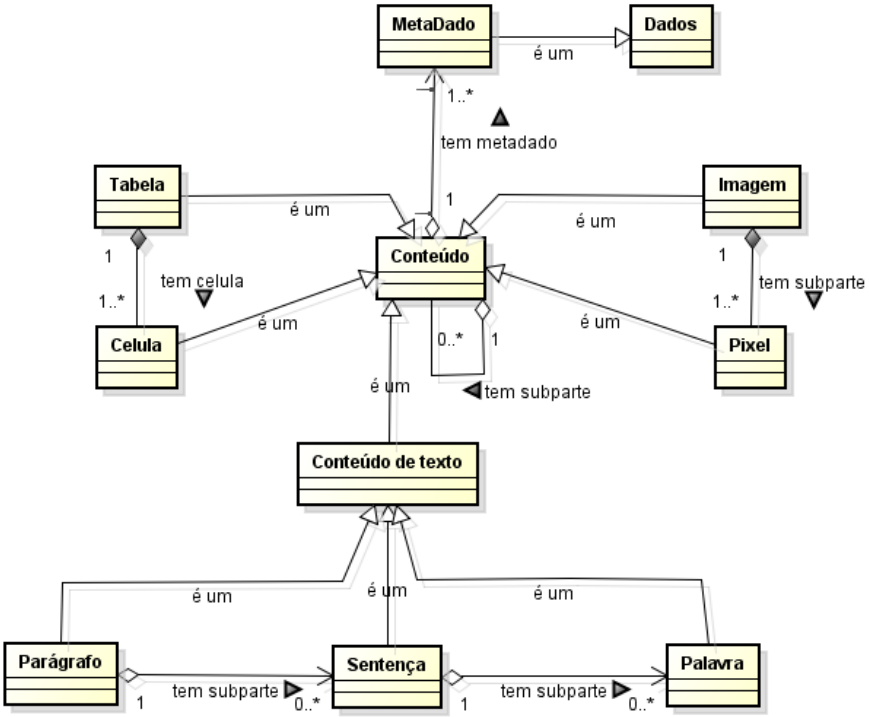
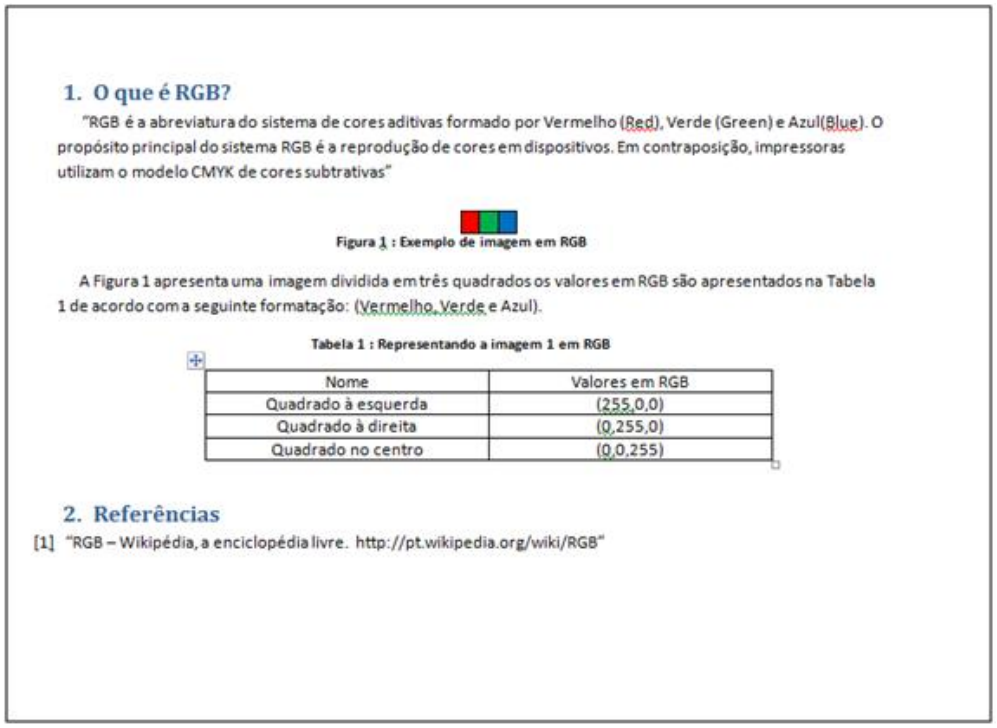


Figura 27: Conteúdo de um documento

A Figura 28 é um exemplo de documento composto de dois capítulos, três parágrafos, uma tabela e uma imagem.

A Figura 29 apresenta a visão geral do documento da Figura 28 ao se representá-lo de acordo com a ontologia proposta. O documento é composto por duas instâncias do Conteúdo (Capitulo 1 e Capitulo 2) e cada instância é composta por um título que contém um Pedaço de texto como Metadado (“O que é RGB?” e “Referências” respectivamente).



1. O que é RGB?

"RGB é a abreviatura do sistema de cores aditivas formado por Vermelho (Red), Verde (Green) e Azul(Blue). O propósito principal do sistema RGB é a reprodução de cores em dispositivos. Em contraposição, impressoras utilizam o modelo CMYK de cores subtrativas"

Figura 1 : Exemplo de imagem em RGB

A Figura 1 apresenta uma imagem dividida em três quadrados os valores em RGB são apresentados na Tabela 1 de acordo com a seguinte formatação: (Vermelho, Verde e Azul).

Tabela 1 : Representando a imagem 1 em RGB

Nome	Valores em RGB
Quadrado à esquerda	(255,0,0)
Quadrado à direita	(0,255,0)
Quadrado no centro	(0,0,255)

2. Referências

[1] "RGB – Wikipédia, a enciclopédia livre. http://pt.wikipedia.org/wiki/RGB"

Figura 28: Exemplo de documento

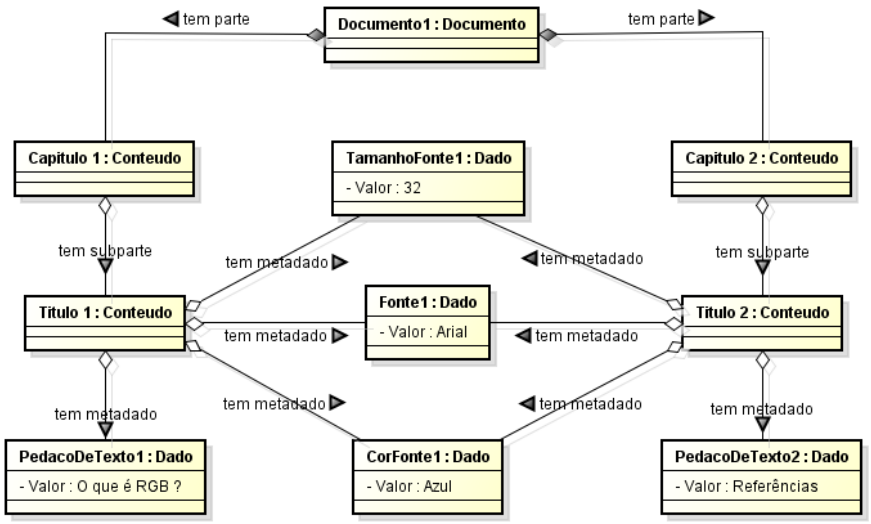


Figura 29: Visão geral do documento da Figura 28 na ontologia

A Figura 30 apresenta a estrutura da entidade Capítulo 1. Esta entidade é composta de duas instâncias da entidade Parágrafo (Parágrafo1 e Parágrafo 2) e têm um metadado Pedaço de texto representando o conteúdo do texto do documento principal que elas abordam. É importante destacar que a granularidade escolhida para o exemplo foi a de parágrafo, porém, nada impede que as entidades Parágrafo 1 ou Parágrafo 2 sejam desmembradas em entidades Sentenças ou Palavras. Por exemplo, na granularidade de palavras, o Parágrafo 2 pode ser composto por um conjunto de vinte e oito entidades Palavras. Uma entidade Imagem e uma entidade Tabela

também compõem o Capitulo 1 e cada uma delas tem um Metadado legenda associado.

A entidade Tabela 1 representada nas Figura 30 e Figura 31 é composta de seis entidades Célula, onde, para cada uma delas existem três Metadados associados: Índice da coluna, Índice da linha e Pedaço de texto.

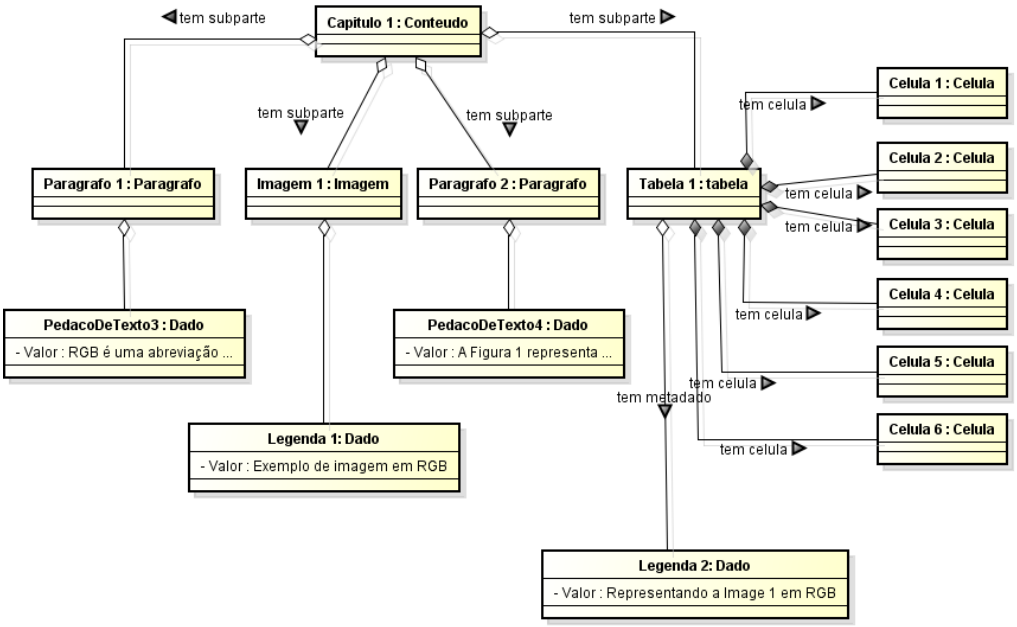


Figura 30: Visão geral do Capítulo 1 da Figura 28 a ontologia

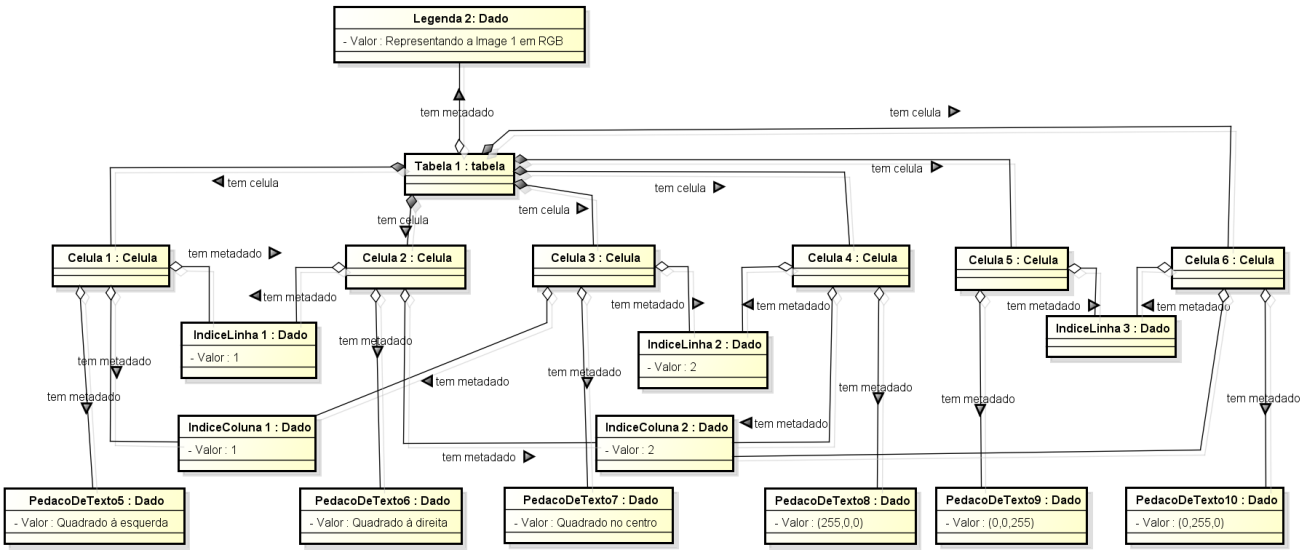


Figura 31: Tabela do documento da Figura 28 na ontologia

3.2 Relacionamentos entre documentos

Durante o processo de confecção do documento, técnicas de reuso geralmente são empregadas, como, por exemplo, copiar e colar, resumir, criticar ou referenciar outros documentos. Os resultados deste processo de confecção são os relacionamentos entre os documentos envolvidos.

A Figura 32 apresenta uma representação gráfica dos relacionamentos que podem existir entre documentos. Os relacionamentos são:

I. Cópia Idêntica: É um documento que têm a mesma estrutura e conteúdo do documento original;
II. Mudanças no estilo: É um documento que têm o mesmo conteúdo e estrutura, como uma Cópia Idêntica, mas têm mudanças nas informações de estilo (por exemplo, mudanças no tamanho ou na cor da fonte em uma palavra);
III. Incremento / Anexo: É um documento no qual mais conteúdo foi adicionado ao documento original;
IV. Decremento / Síntese: É um documento no qual os pedaços do seu conteúdo foram removidos do documento original;
V. Atualização / Versão: É um documento híbrido composto de: Incrementos; Decrementos; Mudanças no estilo;
VI. Avaliação / Referencia: É um documento que aborda (i. e. referencia, critica ou avalia) um documento ou qualquer parte dele.

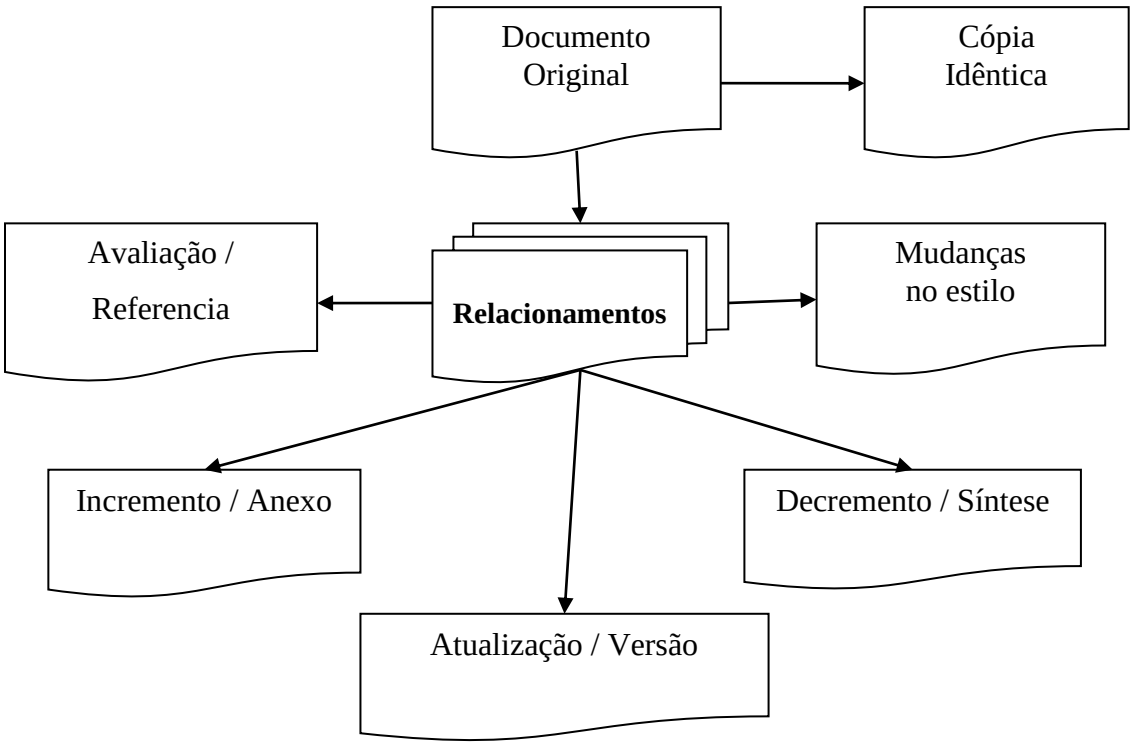


Figura 32: Visão global das relações entre documentos

A Figura 33 representa um mapeamento entre os relacionamentos entre documentos com o grupo de relacionamentos entre entidades bibliográficas.

No grupo de relacionamentos de equivalência estão os documentos que são Cópias Idênticas de um documento. Isto inclui fotocópia, cópia, facsímile e reimpressões.

Nos grupo descritivo e no grupo acompanhamento estão os documentos que são resultados de avaliações, referências e críticas de outros documentos. Os exemplos incluídos nesses grupos são: Livros de bolso, comentários, críticas, índices bibliográficos, catálogos de bibliotecas, edições anotadas, revisões e comentário.

Nos grupo derivativo e no grupo parte-todo estão os documento que são Incrementos, Sínteses ou Versões de outros documentos. Isto inclui edições, revisões, traduções, sumários, resumos e compilações, adaptações ou modificações, dramatizações, traduções livres, paráfrases, imitações e paródias.

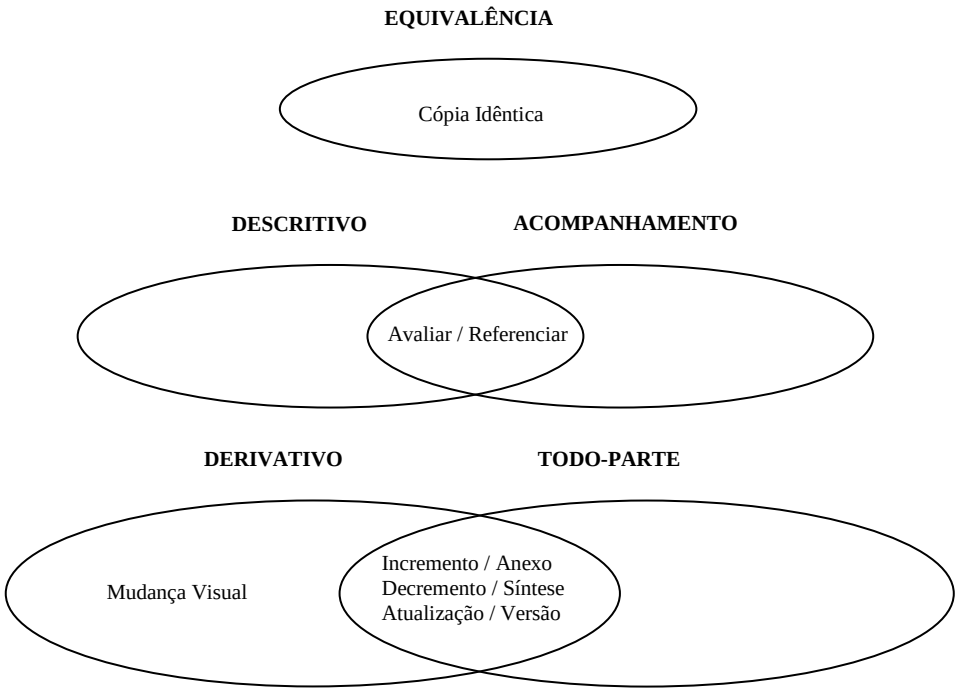


Figura 33: Relacionamentos bibliográficos e de operações

3.3 Eventos em documentos digitais

Os relacionamentos entre documentos podem ser analisados através da perspectiva de eventos. Isto é, cada relacionamento existente entre dois documentos é fruto de um evento que ocorreu. Logo, para cada relacionamento, existe um evento que o criou conforme apresentado na Tabela 3.

Relacionamento	Evento
Cópia Idêntica	Copiar e colar
Atualização	Atualizar
Decremento	Decrementar
Incremento	Incrementar
Referência	Referenciar
Modificação de estilo	Modificar estilo

Tabela 3: Eventos e seus Relacionamentos gerados

Tabela 4 representa a influência dos Eventos nas entidades do modelo FRBR.

I. O evento Atualizar gera novas entidades de trabalho (uma adaptação de trabalho é um novo trabalho) e uma nova entidade de expressão (uma revisão de um trabalho é uma nova expressão do mesmo trabalho);

II. Os eventos Incrementar, Decrementar e Referenciar geram entidades de trabalho (como sumário de um trabalho gerado de outro trabalho) e entidades de expressão (uma pequena modificação gera novas expressões do trabalho);

III. O evento Modificar estilo gera novas entidades de manifestação (Quando elas não afetam o conteúdo do trabalho) e novas entidades de itens;

IV. O evento Copiar e colar gera novas entidades itens (uma reimpressão cria um novo item) e novas entidades manifestações (digitalizar uma entidade de documento cria uma nova manifestação do documento).

Entidade	Copiar e colar	Atualizar	Decrementar	Incrementar	Referenciar	Modificar estilo
Trabalho	Não	Sim	Sim	Sim	Sim	Não
Expressão	Não	Sim	Sim	Sim	Sim	Não
Manifestação	Sim	Não	Não	Não	Não	Sim
Item	Sim	Não	Não	Não	Não	Sim

Tabela 4: Influencia dos eventos nas entidades FRBR

Assim como em (PINHEIRO, 2010, p 85), associaremos o conceito de eventos na representação das relações entre documentos digitais. Utilizaremos também, a definição de eventos do framework INDECS onde, define-se que, tudo pode ser expresso através de eventos.

Para o framework INDECS um evento é realizado por um agente, onde ele transforma os insumos em resultados de acordo com o tipo de evento que será realizado. Para o domínio deste trabalho os insumos podem ser entidades Documentos ou Conteúdo que serão manipulados gerando novos Documentos e relacionamentos.

Os Eventos em documentos digitais serão classificados de acordo com duas especializações do evento do framework INDECS, eles são: Eventos de criação e Eventos de Transformação. A Figura 34 apresenta esta classificação onde os eventos que manipulam um documento de entrada para criar outro documento são classificados como Eventos de Transformação, eles são os eventos Incrementar, Decrementar e Atualizar. Enquanto os eventos que criam um documento que seu conteúdo está relacionado com outro documento, porém não é uma manipulação direta do conteúdo dele, são classificados como Eventos de Criação.

Não abordaremos nesta seção os Eventos Modificar estilo e Copiar e Colar visto que, eles não apresentam manipulações do conteúdo ou na estrutura dos documentos. Os resultados destes eventos serão representados através de alterações nos metadados relacionados às entidades.

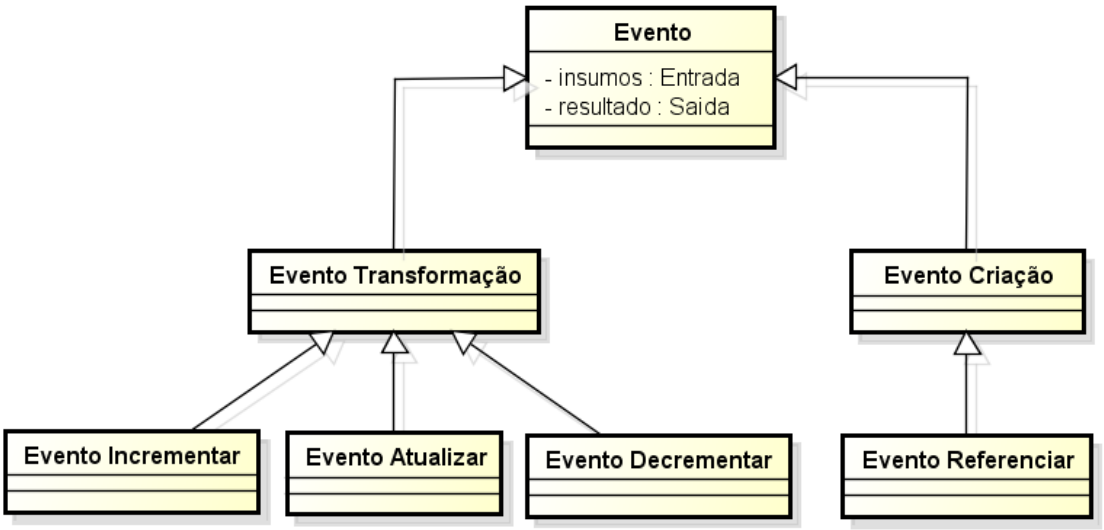


Figura 34: Eventos, sua composição e hierarquia.

3.3.1 Operações dos Eventos em documentos digitais

Quando um documento digital é manipulado, operações são aplicadas para transformar o documento no resultado desejado. Logo, os Eventos em documentos digitais realizam Operações no conteúdo de documentos que resultam em relacionamentos entre os documentos.

Mapear operações aplicadas em documentos e os relacionamentos criados por elas são funcionalidades interessantes, quando, por exemplo, existe a necessidade de trabalho colaborativo, suporte a auditoria ou controle de vazamento de informação.

A representação das Operações aplicadas pelos Eventos e os Relacionamentos gerados por elas é apresentada na Figura 35, e observa-se que:

- Uma Operação relaciona dois conteúdos de documentos;
- Um Relacionamento relaciona dois documentos;
- Um Relacionamento é composto de uma ou mais Operações.

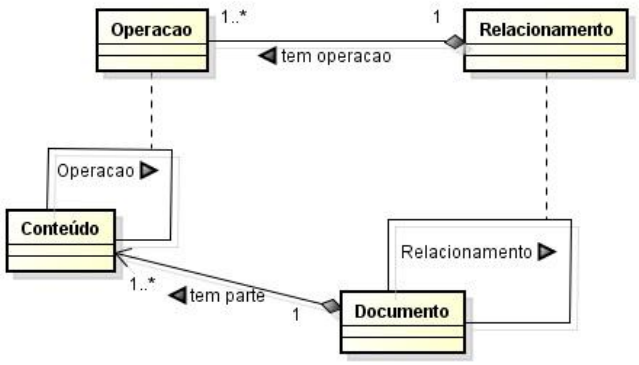


Figura 35: Operações entre Conteúdos × Relacionamentos entre Documentos

As operações que adotaremos para este trabalho estão expressas na Figura 36 e podem ser definidas como:

- Equivalente – As duas entidades Conteúdo em questão são consideradas equivalentes durante a realização do Evento;
- Adicionar – Uma das entidades Conteúdo foi adicionada após a outra no documento que as contém durante a realização do Evento;
- Remover – Uma das entidades Conteúdo foi removida da posição anterior a outra durante a realização do Evento;
- Substituir – Uma entidade Conteúdo foi substituída pela outra durante a realização do Evento;

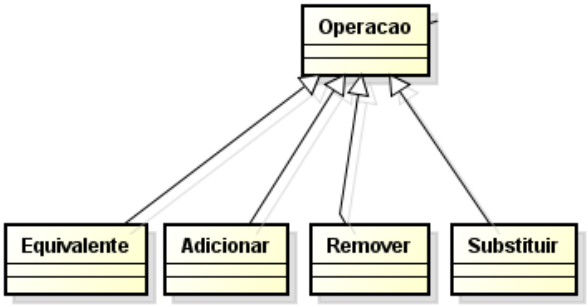


Figura 36: Tipos de Operações possíveis

3.3.2 Operações geradas pelo Evento Copiar e Colar

Durante a realização do Evento de Copiar e Colar um Relacionamento de Cópia Idêntica é gerado, entre entidades documentos, e este Relacionamento é composto por uma ou mais operações Equivalentes entre os Conteúdos dos Documentos do Relacionamento. Conforme está apresentado na Figura 37.

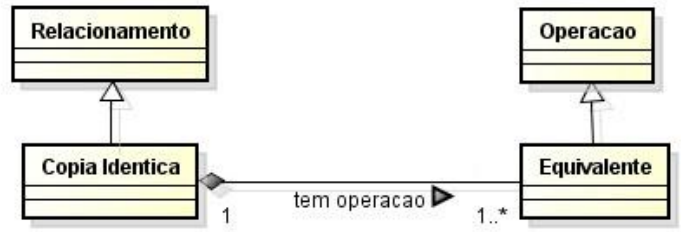


Figura 37: Relacionamento de Cópia Idêntica e suas Operações

3.3.3 Operações geradas pelo Evento Incrementar

A Figura 38 apresenta que o Evento Incrementar é uma generalização do Evento de Adicionar Parte.

A Figura 39 apresenta que, durante a realização do Evento de Incrementar, um Relacionamento de Incremento é gerado, entre entidades documentos, e este Relacionamento é composto por:

- Zero ou mais operações Equivalentes entre os Conteúdos dos Documentos do Relacionamento;
- Uma ou mais operações de Adicionar entre os Conteúdos dos Documentos relacionados.

As manipulações exercidas pelos comandos dos editores de texto de inserir linhas, capítulos, seções, sentenças, parágrafos e etc. são representadas nesta seção.

Figura 38: Especialização do Evento Incrementar

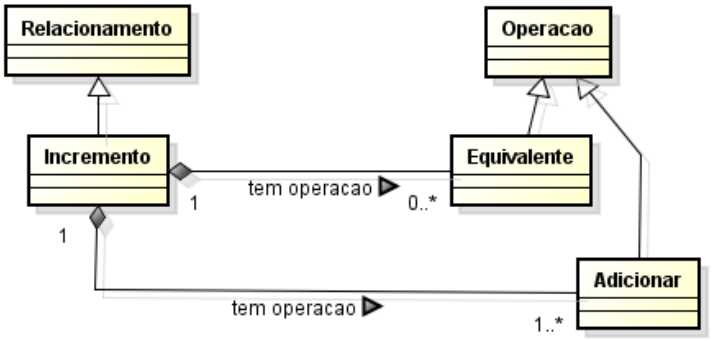


Figura 39: Operações relacionadas com o Evento Incrementar

3.3.4 Operações geradas pelo Evento Decrementar

A **Erro! Fonte de referência não encontrada.** apresenta que o Evento Decrementar é uma generalização do Evento de Remover Parte.

A **Erro! Fonte de referência não encontrada.** apresenta que, durante a realização do Evento de Decrementar, um Relacionamento de Decremento é gerado, entre entidades documentos, e este Relacionamento é composto por:

- Zero ou mais operações Equivalentes entre os Conteúdos dos Documentos do Relacionamento;
- Uma ou mais operações de Remover entre os Conteúdos dos Documentos relacionados.

As manipulações exercidas pelos comandos dos editores de texto de remover linhas, capítulos, seções, sentenças, parágrafos e etc. são representadas nesta seção.

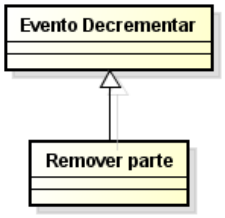


Figura 40: Especialização do Evento Decrementar

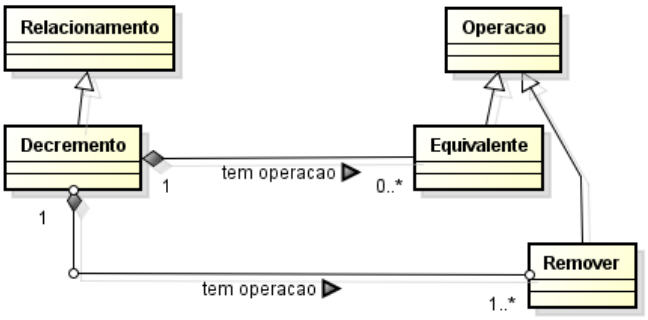


Figura 41: Operações relacionadas com o Evento Decrementar

3.3.5 Operações geradas pelo Evento Atualizar

A Figura 42 apresenta que o Evento Atualizar é uma generalização dos Eventos:

- Promover parte – Gerar um novo parágrafo a partir de uma sentença no documento é um exemplo;
- Rebaixar parte – Transformar um capítulo em seção é um exemplo;
- Deslocar parte – Adicionar uma palavra no meio de outras palavras é um exemplo;
- Inverter partes – Inverter a ordem de dois capítulos, sequenciais, em um documento é um exemplo;
- Juntar partes – Unir frases respeitando as posições nas quais elas apresentam suas palavras é um exemplo;
- Mesclar partes – Unir frases sem respeitar as posições nas quais elas apresentam suas palavras (“embaralhando as palavras”) é um exemplo.

A Figura 43 **Erro! Fonte de referência não encontrada.**apresenta que, durante a realização do Evento de Atualizar, um Relacionamento de Atualização é gerado, entre entidades documentos, e este Relacionamento é composto por:

- Uma ou mais operações Equivalentes entre os Conteúdos dos Documentos do Relacionamento;
- Zero ou mais operações de Remover entre os Conteúdos dos Documentos relacionados;
- Zero ou mais operações de Adicionar entre os Conteúdos dos Documentos relacionados;
- Zero ou mais operações de Substituir entre os Conteúdos dos Documentos relacionados.

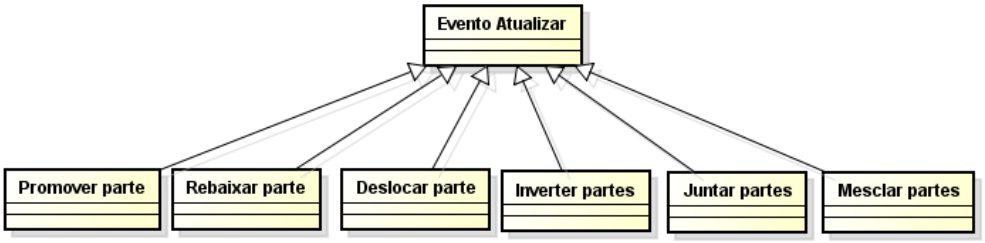


Figura 42: Especializações do Evento Atualizar

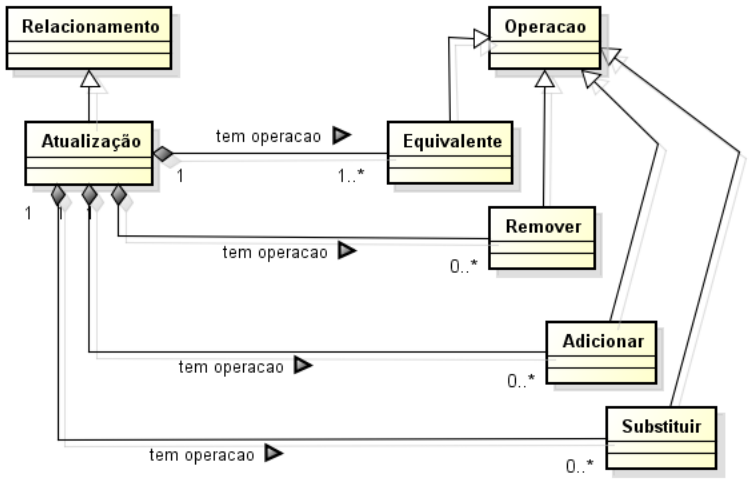


Figura 43: Operações relacionadas com o Evento Atualizar

As manipulações exercidas pelos comandos dos editores de texto de mover e arrastar linhas, capítulos, seções, sentenças, parágrafos e etc. são representadas nesta seção.

3.3.6 Operações do evento Referenciar

A Figura 44 apresenta que o Evento Referenciar é uma generalização dos Eventos:

- Comentar – Descreve alguma informação relacionada ao Documento, ou ao conteúdo deste Documento, referenciado;
- Criticar – Analisa através de comentários a referência;
- Revisar – Agrupa Críticas e Comentários sobre determinada referência;
- Catalogar – Agrupa referências a Documentos de acordo com critérios previamente determinados;
- Indexar – Agrupa as referências para a bibliografia de um Documento específico.

A Figura 45 **Erro! Fonte de referência não encontrada.**apresenta que, durante a realização do Evento de Referenciar, um Relacionamento de Referência é gerado, entre entidades Documentos, e este Relacionamento é composto por uma operação Equivalente entre o Conteúdo do Documento referenciado e o Conteúdo do Documento gerado.

O Evento Referenciar é um evento de criação. Portanto, suas especializações apresentam novas entidades como resultado, estas entidades também são apresentados na Figura 45, elas são:

- Comentário – Composto de referências ao Conteúdo que foi comentado;
- Crítica – É um comentário que tem como objetivo avaliar o Conteúdo que foi referenciado;
- Revisão – Conjuntos de Críticas e Comentários sobre um Documento;
- Catálogo – Agrupamento de Referências de Documentos de acordo com um critério;
- Índice Bibliográfico – Agrupamento de Referências de Documentos relacionados com o assunto de um Documento específico.

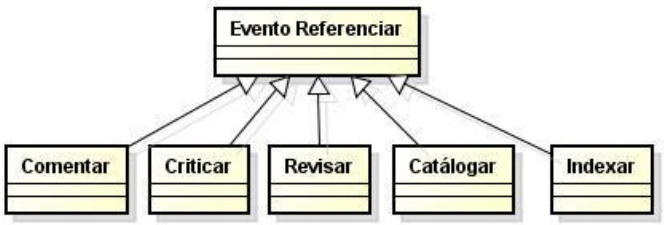


Figura 44: Especializações do Evento Referenciar

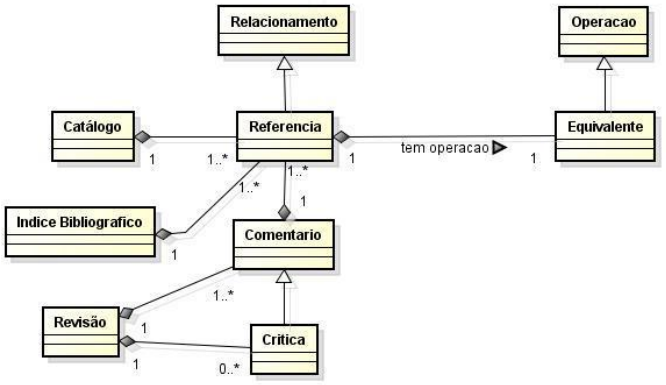


Figura 45: Operações relacionadas com o Evento Referenciar

3.4 Álgebra dos relacionamentos entre documentos

A Figura 46, apresenta o processo de identificação de relacionamentos entre dois Documentos envolvidos em um Evento. Para realizar a identificação dos relacionamentos a partir das Operações apresentaremos, nesta seção, uma Álgebra de identificação de Relacionamentos entre Documentos.

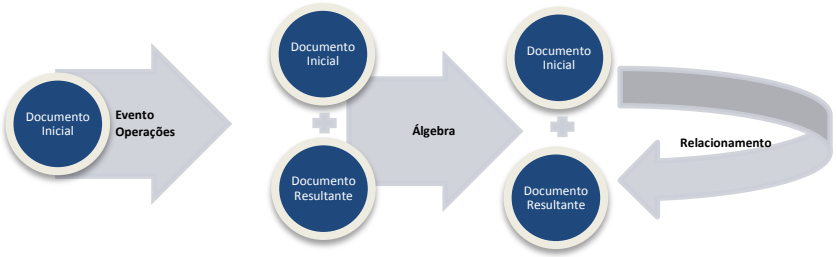


Figura 46: Como identificar os Relacionamentos gerados por um Evento

Definiremos a Álgebra dos Relacionamentos entre Documentos Digitais utilizando uma formalidade parecida com a de lógica de primeira ordem. Contudo, antes de apresentar uma definição formal, alguns conceitos precisam ser esclarecidos.

A partir da visão geral do documento, apresentada na Figura 26 define-se:

- D um conjunto de Documentos, onde D(j) é um documento neste conjunto;

- C um conjunto de Conteúdos, onde C(i,j) é o i-ézimo conteúdo do documento D(j);
- SP(x,y,j) um relacionamento onde C(x,j) é uma subparte de C(y,j) em D(j), isto é, C(x,j) compõe C(y,j) em D(j);
- SE(x,y,j) um relacionamento onde C(x,j) e C(y,j) estão em posições seqüenciais em D(j), isto é, C(x,j) está na posição seguinte a C(y,j) em D(j).

A partir dos tipos de Operações apresentados em Figura 36 define-se:

- eq(x,i,y,j) – Operação Equivalente entre C(x,i) e C(y,j), isto é, o Conteúdo C(x,i) no Documento D(i) é equivalente ao Conteúdo C(y,j) no Documento D(j);
- adc(x,i,y,j) – Operação Adicionar entre C(x,i) e C(y,j), isto é, o Conteúdo C(x,i) foi adicionado após o Conteúdo C(y,i) no Documento D(i) e C(y,i) é equivalente a C(y,j) no Documento D(j);
- rem(x,i,y,j) – Operação Remover entre C(x,i) e C(y,j), isto é, o Conteúdo C(x,i) foi removido de uma posição anterior a do Conteúdo C(y,i) no Documento D(i) e C(y,i) é equivalente a C(y,j) no Documento D(j);
- sub(x,i,y,j) – Operação Substituir entre C(x,i) e C(y,j), isto é, o Conteúdo C(x,i) no Documento D(i) substituiu o Conteúdo C(y,j) no Documento D(j).

A partir dos Relacionamentos apresentados na Tabela 3 define-se:

- CPID(i,j) é o Relacionamento de Cópia Idêntica entre D(i) e D(j), isto é, D(j) é uma Cópia Idêntica de D(i);
- INC(i,j) é o Relacionamento de Incremento entre D(i) e D(j), isto é, D(i) é um Incremento de D(j);
- DEC(i,j) é o Relacionamento de Decremento entre D(i) e D(j), isto é, D(i) é um Decremento de D(j);
- ATU(i,j) é o Relacionamento de Atualização entre D(i) e D(j), isto é, D(i) é uma Atualização de D(j);
- REF(i,j) é o Relacionamento de Referência entre D(i) e D(j), isto é, D(i) é uma Referência de D(j).

O Relacionamento de Atualização pode ser gerado por várias especializações do Evento Atualizar. Logo, classificaremos os Relacionamentos de Atualização de acordo com a Figura 42, da seguinte forma:

- PRO(i,j) Expressa que, algum conteúdo de D(j) foi promovido em D(i);
- REB(i,j) Expressa que, algum conteúdo de D(j) foi rebaixado em D(i);
- DESL(i,j) Expressa que, D(i) tem novo conteúdo entre dois conteúdos de D(j) que eram sequenciais;
- INV(i,j) Expressa que, dois conteúdos de D(j) trocaram de posição em D(i);
- JUN(i,j) Expressa que, algum conteúdo de D(i) é a junção de Conteúdos de D(j);
- MESC(i,j) Expressa que, algum conteúdo de D(i) é a mesclagem de Conteúdos de D(j).

3.4.1 Definição textual

Suponha que existam dois Documentos D(i) e D(j) o Relacionamento entre eles será:

- CPID(i,j) se:
  i. Todo Conteúdo de D(j) tem Conteúdo equivalente em D(i) e vice-versa.
- INC(i,j) se:
  i. Todo Conteúdo de D(j) tem Conteúdo equivalente em D(i) e
  ii. Existe pelo menos um Conteúdo em D(i) que não tem equivalente em D(j).
- DEC(i,j) se:
  i. Todo Conteúdo de D(i) tem Conteúdo equivalente em D(j) e
  ii. Existe pelo menos um Conteúdo em D(j) que não tem equivalente em D(i).
- PRO(i,j) se:
  i. Existe em D(j) um Conteúdo C(x,j) que tem subparte C(y,j) e
  ii. D(i) tem Conteúdos equivalentes a C(x,j) e C(y,j), que são C(x,i) e C(y,i) respectivamente e
  iii. C(x,i) não é subparte de C(y,i) e
  iv. Todos os Conteúdos D(j), que não são C(x,j) e C(y,j), têm partes equivalentes em D(i).
- REB(i,j) se:

i. Existe em D(i) um Conteúdo C(x,i) que tem subparte C(y,i) e
ii. D(j) tem Conteúdos equivalentes a C(x,i) e C(y,i), que são C(x,j) e C(y,j) respectivamente e
iii. C(x,j) não é subparte de C(y,j) e
iv. Todos os Conteúdos D(i), que não são C(x,i) e C(y,i), têm partes equivalentes em D(j).

- DESL(i,j) se:
  i. Todos os Conteúdos de D(j) têm um Conteúdo equivalente em D(i) e
  ii. Existem dois Conteúdos sequenciais C(x,j) e C(y,j) em D(j), isto é, SE(x,y,j) e
  iii. Existe um Conteúdo C(z,i) em D(i) e
  iv. Não existe um Conteúdo equivalente a C(z,i) em D(j);
  v. D(i) tem Conteúdos equivalentes a C(x,j) e C(y,j), que são C(x,i) e C(y,i) respectivamente e
  vi. SE(z,x,i) e
  vii. SE(y,z,i).
- INV(i,j) se:
  i. Todos os Conteúdos de D(j) têm um Conteúdo equivalente em D(i) e
  ii. Existem dois Conteúdos sequenciais C(x,j) e C(y,j) em D(j), isto é, SE(x,y,j) e
  iii. D(i) tem Conteúdos equivalentes a C(x,j) e C(y,j), que são C(x,i) e C(y,i) respectivamente e
  iv. SE(y,x,i).
- JUN(i,j) se:
  i. Existe em D(j) um Conteúdo C(x,j) que tem subpartes C(y,j) e C(z,j) e
  ii. SE(z,y,j) e
  iii. Existe em D(j) um Conteúdo C(x’,j) que tem subpartes C(y’,j) e C(z’,j) e
  iv. SE(z’,y’,j) e
  v. D(i) tem Conteúdos equivalentes a C(y,j), C(z,j), C(y’,j) e C(z’,j), que são C(y,i), C(z,i), C(y’,i) e C(z’,i) respectivamente e
  vi. Não existem Conteúdos equivalentes a C(x,j) e C(x’,j) em D(i);
  vii. Todos os Conteúdos de D(j), que não são C(x,j), C(y,j), C(z,j), C(x’,j), C(y’,j) e C(z’,j), têm um Conteúdo equivalente em D(i) e

viii. SE(z,y,i) e SE(y',z,i) e SE(z',y',i).

- MESC(i,j) se:
  i. Existe em D(j) um Conteúdo C(x,j) que tem subpartes C(y,j) e C(z,j) e
  ii. SE(z,y,j) e
  iii. Existe em D(j) um Conteúdo C(x',j) que tem subpartes C(y',j) e C(z',j) e
  iv. SE(z',y',j) e
  v. D(i) tem Conteúdos equivalentes a C(y,j), C(z,j), C(y',j) e C(z',j), que são C(y,i), C(z,i), C(y',i) e C(z',i) respectivamente e
  vi. Não existem Conteúdos equivalentes a C(x,j) e C(x',j) em D(i);
  vii. Todos os Conteúdos de D(j), que não são C(x,j), C(y,j), C(z,j), C(x',j), C(y',j) e C(z',j), têm um Conteúdo equivalente em D(i) e
  viii. D(i) Não é Junção de D(j).
- ATU(i,j) se:
  i. PRO(i,j) ou REB(i,j) ou DESL(i,j) ou INV(i,j) ou JUN(i,j) ou MESC(i,j).
- REF (i,j) se:
  i. Existe pelo menos um Conteúdo em D(i) que tem equivalente em D(j) e
  ii. D(i) não é Cópia Identica, Incremento, Decremento ou Atualização de D(j).

3.4.2 Definição formal

Símbolos: $\{\vee,\wedge, \leftrightarrow,\rightarrow, \neg,\forall,\exists\}$

Axiomas:

1) $C(x,i) \rightarrow D(i)$;
2) $D(i) \rightarrow \exists x\ C(x,i)$;
3) $SP(x,y,i) \rightarrow C(x,i) \wedge C(y,i)$;
4) $SE(x,y,i) \rightarrow C(x,i) \wedge C(y,i)$;
5) $eq\ (x,i,y,j) \rightarrow (\ C(x,i) \wedge C(y,j)\ )$;
6) $adc\ (x,i,y,j) \leftrightarrow (\ C(x,i) \wedge C(y,j) \wedge \exists y'\ eq\ (y',i,y,j) \wedge \forall x' \neg\ eq\ (x',j,x,i)\ )$;
7) $adc\ (x,i,y,j) \leftrightarrow rem\ (y,j,x,i)$;

8) $sub(x,i,y,j) \leftrightarrow \exists k\ (\ adc\ (x,i,z,k) \wedge rem\ (y,j,z,k)\ ).$

Considere que existe um documento i D(i) e um documento j D(j). O Relacionamento entre estes documentos será:

- $CPID(i,j) \leftrightarrow \forall x\ eq\ (x,i,y,j);$
- $INC(i,j) \leftrightarrow (\forall x\ eq\ (x,j,y,i)) \wedge\ adc\ (z,i,w,j);$
- $DEC(i,j) \leftrightarrow INC(j,i);$
- $PRO(i,j) \leftrightarrow \exists x\ (\ SP(y,x,j) \wedge rem(x,i,x',j) \wedge eq\ (z,i,z',j)\ );$
- $REB(i,j) \leftrightarrow PRO(j,i);$
- $INV(i,j) \leftrightarrow\ \forall x\ eq\ (x,i,y,j) \wedge SE(z,w,j) \wedge SE(w,z,j);$
- $DESL(i,j) \leftrightarrow\ \forall x\ eq\ (x,j,y,i) \wedge \exists k\ (\ SE(z,w,j) \wedge adc(k,i,w,j) \wedge SE(k,w,i) \wedge SE(z,k,i)\ );$
- $JUN(i,j) \leftrightarrow\ SE(x,y,j) \wedge \exists \kappa(\ SP(x,k,i) \wedge SP(y,k,i) \wedge SE(x,y,j)\ );$
- $MESC(i,j) \leftrightarrow\ SE(x,y,j) \wedge \exists \kappa(\ SP(x,k,i) \wedge SP(y,k,i) \wedge \neg\ SE(x,y,j)\ );$
- $ATU(i,j) \leftrightarrow\ \ PRO(x,y,j) \wedge REB(x,y,j) \wedge INV(x,y,j) \wedge DESL(x,y,j) \wedge JUN(x,y,j) \wedge MESC(x,y,j);$
- $REF(i,j) \leftrightarrow\ \exists x\ eq(x,i,x',j) \wedge \neg\ CPID(x,y,j) \wedge \neg\ INC(x,y,j) \wedge \neg\ DEC(x,y,j) \wedge \neg\ ATU(x,y,j);$

3.5 Conclusão

Neste capítulo apresentamos uma ontologia que conceitualiza: o domínio de documentos digitais, a estrutura de um documento digital, os possíveis relacionamentos existentes entre documentos, os eventos que geraram estes relacionamentos e as operações de manipulação de documentos digitais. Também definimos e apresentamos uma álgebra que formaliza a identificação e a representação dos relacionamentos existentes entre dois documentos. Todas as formalidades apresentadas neste capítulo serão utilizadas como base para o trabalho realizado nos próximos capítulos.

4 UMA MEDIDA DE SIMILARIDADE ESTRUTURAL

Neste capítulo apresentamos uma forma de calcular a similaridade estrutural de documentos de texto e suas partes, assim como identificar as operações necessárias para transformar determinado documento em outro. Para tanto, apresentaremos a distância de inspiração para a nossa medida, conceitos de lógica difusa que complementam nossa proposta e um algoritmo que utiliza a nossa medida para calcular os valores. Para finalizar apresentamos o problema de se identificar as operações necessárias para transformar determinado documento em outro, algoritmos relacionados com o problema e a nossa abordagem para resolvê-lo.

A similaridade estrutural é uma métrica para identificar os quão similares dois documentos podem ser. Esta métrica leva em consideração tanto a informação contida no conteúdo quanto a que está contida na estrutura de cada documento. Para tanto será proposta uma nova medida de edição que utiliza conceitos estatísticos para medir a “diferença” entre dois documentos.

A nossa medida tem como inspiração a distância de Levenshtein, que também é conhecida como distância de edição, nela mede-se o quão diferentes duas seqüências de caracteres podem ser (LEVENSHTEIN, 1966).

Assim como distância de Levenshtein entre duas strings é definida como o menor número de operações de remoção, adição ou substituição de caracteres para transformar uma string em outra(NURMI; SUSTRETOV, 2006,, p 207), a nossa medida pode ser definida como:

“o menor número de operações de remoção, adição ou substituição de partes de documentos para transformar um documento em outro”.

Para calcularmos a medida de similaridade proposta, utilizamos valores de variáveis difusas para quantificar uma pontuação de similaridade entre partes de documentos distintos que chamamos de SCORE.

Para efeito de esclarecimento apresentaremos a distância de Levenshtein, uma breve introdução a lógica difusa e em seguida apresentaremos como calcular a similaridade estrutural entre documentos.

4.1 Distância de Levenshtein

A distância de Levenshtein é uma métrica que explicita o quão diferentes duas sequências podem ser (LEVENSHTEIN, 1966). O termo distância de edição também é atribuído a ela. Existem várias áreas de aplicação que esta métrica pode ser aplicada, entre elas destacamos: Bioinformática, lingüística, recuperação de informação, classificação de texto, busca em multi-idiomas, correção e complementação de texto, recuperação de ruídos em análises de sinais, erros de digitação, erros de OCR entre outros.

Adaptações desta medida são muito comuns na área de reconhecimento de discurso como em (FISCUS et al., 2006, p 3).

A distância de Levenshtein entre duas strings é definida como o menor número de operações de remoção, adição ou substituição de caracteres para transformar uma string em outra(NURMI; SUSTRETOV, 2006,, p 207).

Para se calcular a distância mínima, um algoritmo de programação dinâmica pode ser utilizado seguindo os seguintes passos:

I. Preenche-se uma matriz bidimensional D, onde D[i][j] representa a distância entre o prefixo de tamanho i da primeira string S1 (de tamanho m) e o prefixo de tamanho j da segunda string S2 (de tamanho n);
II. Para i de 1 até m e j de 1 até n calcula-se D[i][j] da seguinte forma:
III. Se o caractere i de S1 é igual ao caractere j de S2 então D[i][j] = D[i – 1][j – 1];
IV. Caso contrário é atribuído a D[i][j] o mínimo entre:
    - D[i – 1][j – 1] + custo de substituição;
    - D[i][j – 1] + custo de adição;
    - D[i – 1][j] + custo de remoção;
V. A distância entre S1 e S2 será o valor na posição D[m][n] ao termino da execução do algoritmo;

Este algoritmo tem complexidade de O(m.n) em tempo e espaço(NURMI; SUSTRETOV, 2006,, p 207);

4.2 Lógica difusa

Lógica difusa é uma variante de lógica multi-valorada e concentra-se em abordar princípios formais que lidam com o processo de raciocínio baseado em aproximações ao invés de raciocínio baseado em precisão (ZADEH, 1988, p 1). Ela usa como fundamento os conceitos de conjuntos difusos como base para suas definições (ZADEH et al., 1996 e ZADEH, 1988).

A diferença entre a lógica difusa e o modelo clássico de lógica é que ela tem o objetivo de modelar as formas imprecisas de raciocinar (ZADEH, 1988, p 1). Esta diferença consegue abordar a capacidade do ser humano de tomar decisões racionais de acordo com um ambiente impreciso e incerto (ZADEH, 1988, p 1). Para tanto, necessitamos inferir uma resposta aproximada baseada no conjunto de informações conhecidas que são inexatas ou incompletas ou não são totalmente confiáveis (ZADEH, 1988, p 1).

Para ZADEH (1988, p 1) as características principais da lógica difusa que a diferencia da lógica tradicional é que:

I. Na lógica binária, uma proposição é sempre verdadeiro ou falso. Em sistemas de lógica multivalorada uma proposição pode ser verdadeiro, falso ou ter um intermediário, que pode ser um elemento finito ou infinito de um conjunto de valores T. Na lógica difusa, os valores verdadeiros estão dispostos em um intervalo de subconjuntos difusos de T;
II. Um predicado da lógica binária tem uma restrição de ser um subconjunto não-difuso do universo de discurso já para a lógica difusa, quanto mais difuso for o predicado melhor;
III. Tanto a lógica binária quanto para a lógica muti-valorada permite apenas dois quantificadores, tudo e alguns no caso da lógica muti-valorada. A lógica difusa permite, adicionalmente, a utilização de quantificadores difusos, como por exemplo, alguns, muitos, poucos, frequentemente entre outros. Tais quantificadores devem ser interpretados como números difusos que provêem uma caracterização imprecisa da cardinalidade de um ou mais conjuntos difusos ou não-difusos. Nesta perspectiva, deve-se encarar um quantificador difuso como um predicado difuso de segunda ordem e os quantificadores difusos podem ser utilizados para representar o significado das proposições contendo as probabilidades difusas e assim, possibilitar a manipulação das probabilidades com a lógica difusa;
IV. A lógica difusa provê um método para representar o significado dos modificadores de predicados difusos e não-difusos. Isto leva a um sistema para computação com variáveis lingüísticas, que são variáveis as quais os valores são palavras ou sentenças em uma linguagem natural ou sintética, um exemplo de variável lingüística é idade, que pode assumir os valores de novo, muito novo, velho entre outros;
V. Em um sistema binário, uma proposição p deve ser qualificada, principalmente associando-se com p um valor real como verdadeiro ou

falso; um operador modal como possível ou necessário; ou um operador intencional como conhecido ou crença. A lógica difusa tem três modos principais de qualificação: qualificação verdadeira; qualificação provável; e qualificação possível.

4.3 Cálculo da medida de similaridade estrutural

Antes de se calcular a distância alguns termos precisam ser definidos:

- Um prefixo de tamanho i de um documento é um conjunto que se inicia na primeira parte de documento e termina na i-ézima parte do documento;
- O score entre duas partes de documentos é um valor no intervalo [0,1] que segue uma abordagem vetorial que se utiliza de tf-idf para o calculo de seu valor.

Seja:

- dp(i,d1) a i-ézima parte do documento d1;
- dp(j,d2) a j-ézima parte do documento d2;
- tf-idf(dp(i,d1), dp(j,d2)) o cálculo do tf-idf entre dp(i,d1) e dp(j,d2).

O SCORE(dp(i,d1), dp(j,d2)) é calculado como:

- (1 - tf-idf(dp(i,d1), dp(j,d2))) caso este valor seja positivo;
- 1 caso contrário.

Para se calcular a distância mínima, um algoritmo de programação dinâmica pode ser utilizado seguindo os seguintes passos:

I. Preenche-se uma matriz bidimensional D, onde D[i][j] representa a distância entre o prefixo de tamanho i do primeiro documento d1 (de tamanho m) e o prefixo de tamanho j do segundo documento d2 (de tamanho n);

II. Para i de 1 até m e j de 1 até n calcula-se D[i][j] da seguinte forma:

   i. Se SCORE(dp(i,d1), dp(j,d2)) é igual a zero então:

      - D[i][j] = D[i – 1][j – 1];

   ii. Caso contrário:

      - D[i][j] = SCORE(dp(i,d1), dp(j,d2)) +
        Minimo de : D[i – 1][j – 1] ; D[i][j – 1] ; D[i – 1][j];

III. A distância entre d1 e d2 será o valor na posição D[m][n] ao termino da execução do algoritmo;

Este algoritmo é uma adaptação do algoritmo utilizado por Levenshtein e tem complexidade de O(m.n) em tempo e espaço(NURMI; SUSTRETOV, 2006,, p 207);

4.3.1 Reconhecimento das operações realizadas

Ao final da execução do algoritmo de similaridade estrutural, é possível detectar as operações entre as partes dos dois documentos co mparados. Para tanto, utilizaremos as medidas calculadas como uma distância para chegar ao início do documento a partir de qualquer parte de documento.

Este problema pode ser relacionado ao problema do caminho mínimo, da teoria de grafos, onde precisamos descobrir o menor número de operações para transformar um documento no outro. Para resolver este problema, é preciso levar em consideração que pode existir mais de uma sequência de operações para se transformar um documento no outro, logo, dois ou mais caminhos que representem configurações válidas para o problema podem existir.

O problema do caminho mínimo, ou melhor, o problema do caminho mais curto apresenta a necessidade de se calcular o caminho com menor valor através de um grafo que têm arestas com pesos (que serão utilizadas para calcular o valor do caminho).

Existem vários algoritmos propostos para resolver este tipo de problema (NSIT, 2005), entre eles destacamos:

O algoritmo de Dijkstra (Dijkstra, 1959) – Determina o caminho mínimo para todos os vértices do grafo a partir de um vértice inicial Vi;

O algoritmo A* (DECHTER; PEARL, 1985) – É um algoritmo de busca em profundidade que utiliza como recursos adicionais aproximações heurísticas e a formalidade do algoritmo de Dijkstra.

Utilizamos a medida de similaridade estrutural como heurística em uma adaptação do algoritmo A* (DECHTER; PEARL, 1985) para o reconhecimento das operações. Esta escolha foi feita ao se pensar na abordagem gulosa que o algoritmo A* apresenta potencializada pela medida de similaridade calculada no momento anterior. Apresentaremos o algoritmo de forma textual no Algoritmo 1 e, em seguida, uma definição formal do mesmo no Algoritmo 2.

4.3.1.1 Algoritmo 1: Descrição textual do algoritmo

Seja:

I. D uma matriz bidimensional, onde D[i][j] representa a distância entre o prefixo de tamanho i do primeiro documento d1 (de tamanho Td1) e o prefixo de tamanho j do segundo documento d2 (de tamanho Td2);

II. P(i,j,k) uma entidade, onde P.i representa um índice numérico i , p.j representa um índice numérico j e p.k representa um índice numérico k;

III. Q uma pilha de P(i,j,k), onde:

   i. Q ← P(i,j,k) representa que P(i,j,k) foi empilhado em Q;

   ii. Q → E representa que o elemento no topo da pilha foi removido de lá e associado a variável E;

   iii. E = Q representa que o elemento no topo da pilha foi associado a variável E, porém, não foi removido do topo da pilha;

IV. Mini,j(k) é o K-ézimo menor valor entre D[i][j-1], D[i-1][j] e D[i-1][j-1].

V. OP(i,j) a operação associada aos índices i e j;

VI. QFinal é a pilha contendo o caminho associado as operações identificadas.

```
1.  Q ← P(Td1 - 1, Td2 – 1,1)
2.  Proximo = Q
3.  K= 1
4.  Enquanto Proximo ≠ Ø faça:
5.      Enquanto k<4 faça:
6.          i = Proximo.i
7.          j = Proximo.j
8.          k = Proximo.k
9.          Se Mini,j(k) = D[i-1][j-1] então:
10.             Se D[i-1][j-1] = D[i][j] então:
11.                 OP(i,j) = Equivalente
12.                 Q ← P(i – 1, j – 1,k)
13.                 K = 1
14.             Senão
15.                 OP(i,j) = Substituir
16.                 Q ← P(i – 1, j – 1,k)
17.                 K = 1
18.         Senão
19.             Se Mini,j(k) = D[i][j-1] então:
20.                 OP(i,j) = Adicionar
21.                 Q ← P(i, j – 1,k)
22.                 K = 1
23.             Senão
24.                 Se Mini,j(k)  = D[i -1][j] então:
                        OP(i,j) = Remover
                        Q ← P(i – 1, j,k)
                        K = 1
                    Senão
```

```
                    Se k = 3 então
                        Q → Proximo
                    K = k + 1
                    Próximo.k = k
    Proximo = Q
    Se Proximo.i = 1 e Proximo.j = 1 e OP(Proximo.i, Proximo.j) ≠ Ø
    então
        Q_Final ← Proximo
        Enquanto Q ≠ Ø faça:
            Q → temp
            Q_Final ←temp
    Senão
        K = k + 1
        Próximo.k = k
```

Algoritmo 1: Descrição textual do algoritmo

4.3.1.2 Algoritmo 2: Definição formal do algoritmo

Utilizando as definições do item 4.3, define-se:

I. mínimo(i,j) – o menor valor entre D[i-1][j-1], D[i-1][j] e D[i][j-1], tal que este valor seja maior que D[i][j];

II. operação(i,j,legenda) – uma tupla que representa a operação (Inserir, remover, substituir ou semelhante) entre a parte i do documento d1 e a parte j do documento d2;

III. a_estrela(i,j) um procedimento recursivo que retorne fila de operação(i,j,legenda);

Vamos definir o procedimento a_estrela(i,j):

```
se o mínimo(i,j) = D[i-1][j-1] e mínimo(i,j) = D[i][j] então
    fila1 = a_estrela (i-1,j-1);
    se (fila1 está vazia) então
        fila1 = nova fila;
    adiciona na fila1 operação(i,j, Equivalente);
    retorna fila1
senão
    se o mínimo(i,j) = D[i-1][j-1] e mínimo(i,j) diferente de D[i][j] então
        fila2 = a_estrela (i-1,j-1);
        se (fila2 está vazia) então
            fila2 = nova fila;
        adiciona na fila2 operação(i,j, Substituir);
        retorna fila2
    senão
```

```
13.     se o mínimo(i,j) = D[i-1][j] então
14.         fila3 = a_estrela (i-1,j);
15.         se (fila3 está vazia) então
16.             fila3 = nova fila;
17.         adiciona na fila3 operação(i,j, Remover);
18.         retorna fila3
19.     senão
20.         fila4 = a_estrela (i,j-1);
21.         se (fila4 está vazia) então
22.             fila4 = nova fila;
            adiciona na fila4 operação(i,j-1, Adicionar);
            retorna fila4
```

Algoritmo 2: DEFINIÇÃO FORMAL DO ALGORITMO

Finalmente, executamos o algoritmo a_estrela(índice da ultima parte de d1, índice da ultima parte de d2).

4.3.2 Conclusão

Neste capítulo apresentamos uma medida de similaridade estrutural entre documentos, explicitamos um algoritmo para o cálculo da similaridade utilizando informações estruturais dos mesmos e um algoritmo para o reconhecimento e a apresentação das operações identificadas entre os documentos. Todas as formalidades apresentadas neste capítulo serão utilizadas como base para o trabalho realizado nos próximos capítulos.

5 Um mecanismo para Identificação de Reuso de Documento

Neste capítulo apresentamos a arquitetura lógica, descrevemos uma associação da arquitetura lógica com a física, os agentes propostos para cada atividade e uma descrição detalhada das atividades que compõem o mecanismo proposto. Em seguida, um protótipo de ferramenta de identificação de Reuso de documentos, que batizamos de IReDoc, é proposto, assim como sua arquitetura, suas características tecnológicas utilizadas e um exemplo de sua utilização.

(parei aqui)

Considerando que o objetivo da nossa proposta é a identificação de Reuso de conteúdo de documentos para criação de outro documento, necessitamos de uma estrutura onde os documentos candidatos sejam desmembrados e armazenados de forma a possibilitar a comparação. Por conseguinte, o nosso mecanismo deve atender uma etapa de pré-processamento e armazenamento dos documentos e seus conteúdos.

Em seguida uma etapa de comparação e alinhamento entre os documentos será realizada onde documentos candidatos a possíveis fontes de Reuso de conteúdo são comparados e uma medida para avaliar o quão similares são, com o documento consultado, é utilizada para apresentar os resultados.

A arquitetura proposta leva em consideração tanto os aspectos citados ate o momento quanto o aspecto de custo de processamento e independência das etapas, onde a etapa de pré-processamento e armazenamento de um documento não deve concorrer com o processamento da comparação de outros dois documentos diferentes dele.

A Figura 47 apresenta a arquitetura do mecanismo onde inicialmente temos um conjunto de documentos que estão representados de diversas formas, como: arquivos de processadores de texto, documentos de texto simples ou arquivos XML. Ela é composta de três camadas:

I. Camada de interface com o usuário;

II. Camada de gerenciamento de informações de documentos;

III. Ca de monitoramento de documentos.

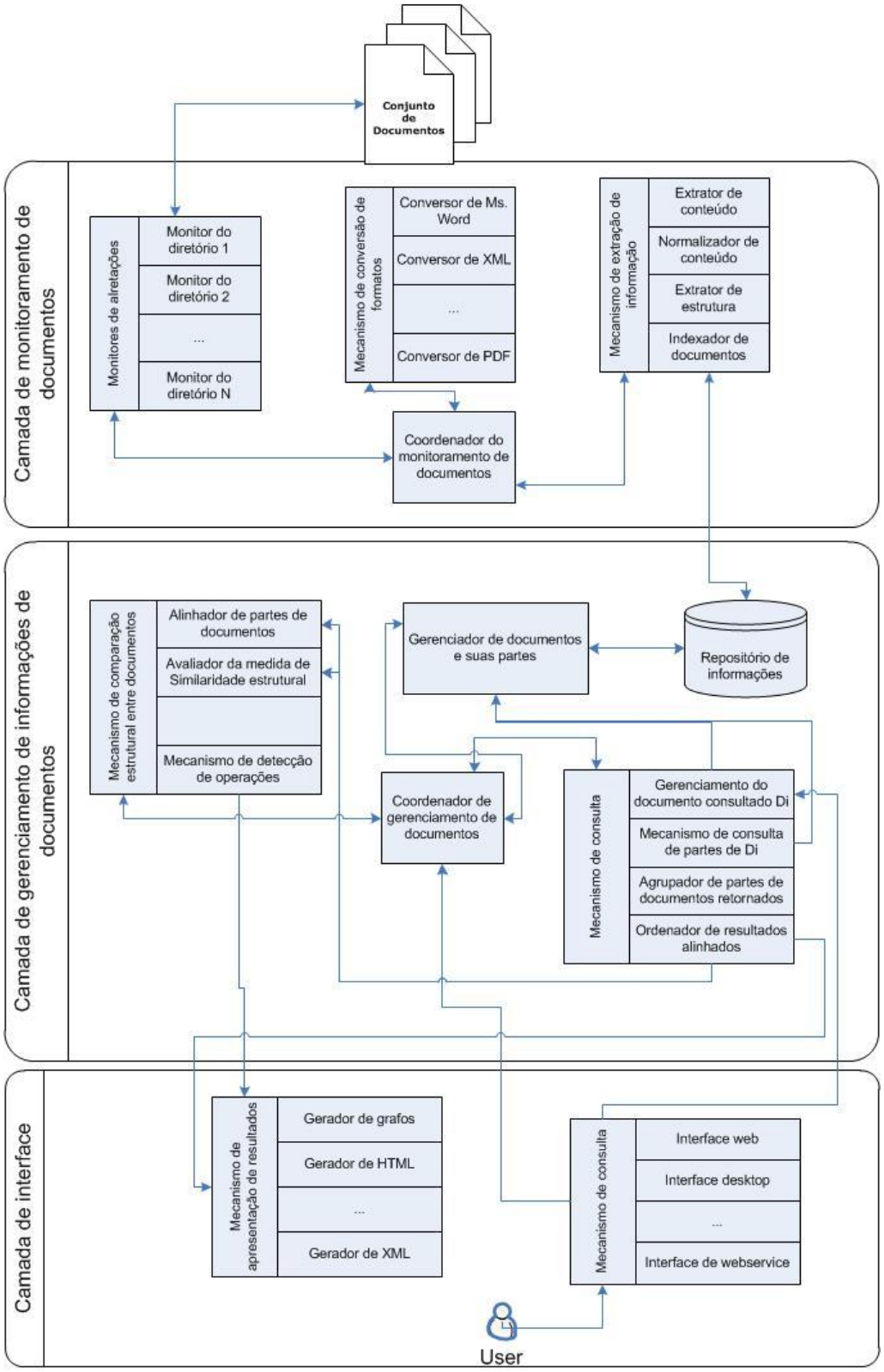


Figura 47: Arquitetura do mecanismo

5.1 Camada de interface com o usuário

O objetivo desta camada é o de servir de interface entre o usuário e o mecanismo. Ela é composta de dois mecanismos:

I. Mecanismo de consultas – Possibilita ao usuário escolher todos os parâmetros necessários para utilização do mecanismo e a apresentação dos resultados do mesmo. Ele é flexível de acordo com os módulos que o compõem. Por exemplo, o usuário pode ter acesso através de uma interface web ou uma interface de um programa instalado em sua maquina;

II. Mecanismo de apresentação de resultados – Permite uma flexibilidade na apresentação do resultado final (por exemplo, pode-se apresentar o resultado final através de um mapa conceitual e forma de árvore ou através de uma página HTML).

5.2 Camada de gerenciamento de informações de documentos

O objetivo desta camada é o de gerenciar todas as informações dos documentos e suas partes armazenadas no repositório de informações, assim como suportar a busca dos documentos e suas partes, o alinhamento entre documentos, o cálculo da medida de similaridade estrutural, a ordenação dos resultados e a identificação das operações entre os documentos do resultado final.

Ela é a camada que contém os mecanismos mais importantes da arquitetura, além de conter o repositório de informações do sistema. Os mecanismos que a compõe são:

I. Gerenciador de documentos e suas partes – gerencia o acesso ao repositório de informações, para resgate ou atualização de documentos e suas partes;

II. Mecanismo de consultas – Consulta documentos, busca partes similares no repositório, prepara os documentos para alinhar e ordena o resultado final;

III. Mecanismo de comparação estrutural entre documentos – alinha os documentos de acordo com suas partes, calcula a medida de similaridade estrutural e identifica as operações para transformar um documento no outro.

5.3 Camada de monitoramento de documentos

Esta camada é responsável por monitorar os repositórios que dispõem de documentos passíveis de Reuso de conteúdo. Nela os documentos são identificados, preparados para entrar no repositório de informações e monitorados para perpetuar, no repositório de informações, possíveis atualizações dos mesmos. Ela é composta de três mecanismos:

I. Monitores de alterações – monitoram os repositórios de documentos para identificar novos documentos ou atualização de documentos antigos. Estes repositórios podem ser pastas de um sistema de arquivos ou até repositórios de documentos na internet (por exemplo, discos virtuais, e-mails ou paginas da web);
II. Mecanismo de conversão de formatos – identifica qual o formato inicial do documento e transforma o mesmo para um formato único que será o padrão para o repositório de documentos;
III. Mecanismo de extração de informações – Extrai as informações de conteúdo e estrutura do documento, normalizando o conteúdo (para UTF-8, por exemplo) e prepara o resultado para armazenar no repositório de informações.

Para atender os requisitos propostos pela arquitetura, o mecanismo é modelado como um sistema de multi-agentes. Os sistemas multi-agentes são constituídos por agentes capazes de trabalhar de forma colaborativa para resolver um problema. Como um sistema complexo pode ser decomposto em subsistemas descentralizados e cooperativos (WOOLDRIDGE, M., 1997), a utilização de sistemas multi-agentes é a solução para atender este requisito.

Os agentes são entidades autônomas que reagem a estímulos de seu ambiente e, com base, nesses estímulos tomam decisões e executam determinado procedimento. Para (BORDINI et al., 2007) os agentes percebem o seu ambiente e agem, de acordo com um conjunto pré-definido de ações, para alterar o estado do ambiente em questão.

Autonomia, pró-atividade, reatividade e habilidade social são características intrínsecas dos agentes (DE LIMA, 2010, p. 30).

As camadas estão associadas à atividade que o mecanismo aborda. Para atender a atividade em questão propomos três agentes:

I. Agente desmembrador – agente que monitora os repositórios de documentos, identifica os documentos contidos nos mesmos e faz o processamento inicial dos documentos. Os documentos serão divididos e instanciarão objetos: documentos, partes de documentos, parágrafos, sentenças e palavras;
II. Agente indexador – agente que prepara as classes anteriores para serem armazenadas no repositório de informações;
III. Agente comparador – agente que verifica os documentos indexados, executa o algoritmo de cálculos de similaridade entre documentos, identifica operações entre partes de documentos e apresenta resultados através de mapas conceituais ou páginas HTML para futura avaliação de resultados.

No contexto do mecanismo a granularidade é de vital importância para futura comparação, para tanto, a arquitetura possibilita a execução de dois tipos distintos: a granularidade do desmembramento do documento e a granularidade da comparação dos documentos. Estas granularidades podem atingir três níveis:

I. Parágrafos;
II. Sentenças;
III. Palavras.

5.4 Comportamento dos agente

O comportamento dos agentes está representado na Figura 48 e na Figura 49 por meio de diagramas de atividades. Na Figura 48 as atividades estão divididas por raias de acordo com o agente que as executam, enquanto as atividades representadas na Figura 49 são executadas pelo agente comparador.

O agente Desmembrador realiza:

- A atividade de “desmembrar documento”, que analisa o tipo (se é um documento do Word ou do PowerPoint) e o conteúdo do documento e desmembra o mesmo em parágrafos, sentenças e palavras de acordo com a granularidade de desmembramento definida.
- A atividade de “resgatar o documento”, que resgata o documento desmembrado que está armazenado no índice da base de conhecimento.

O agente Indexador realiza:

- A atividade de “atualizar alterações”, que verifica, no documento, qual parte foi alterada e prepara o documento desmembrado para refletir essas alterações.
- A atividade de “atualizar o índice” verifica quais são as mudanças que devem ser atualizadas no índice e as realiza.

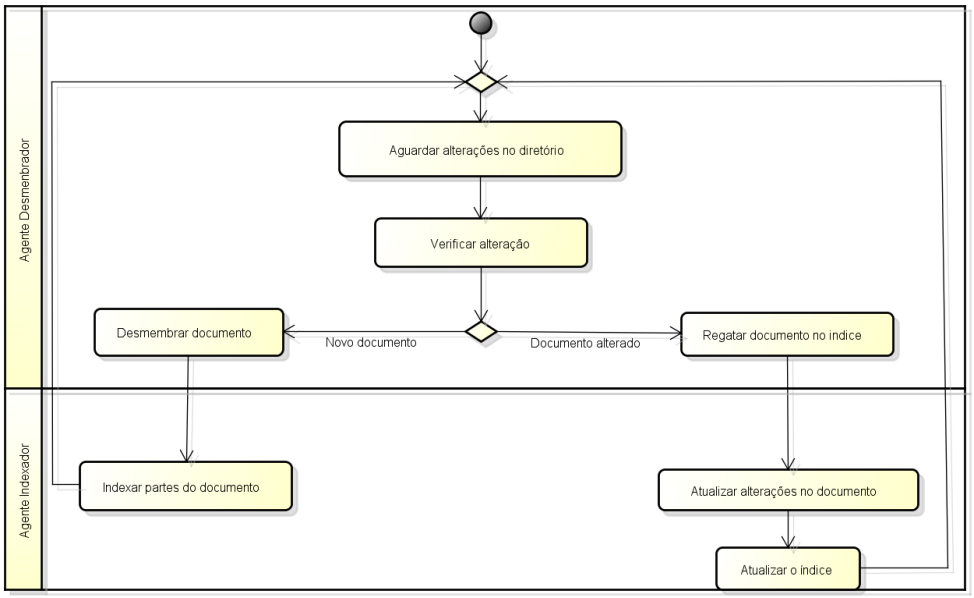


Figura 48: Comportamento dos agentes Desmembrador e Indexador

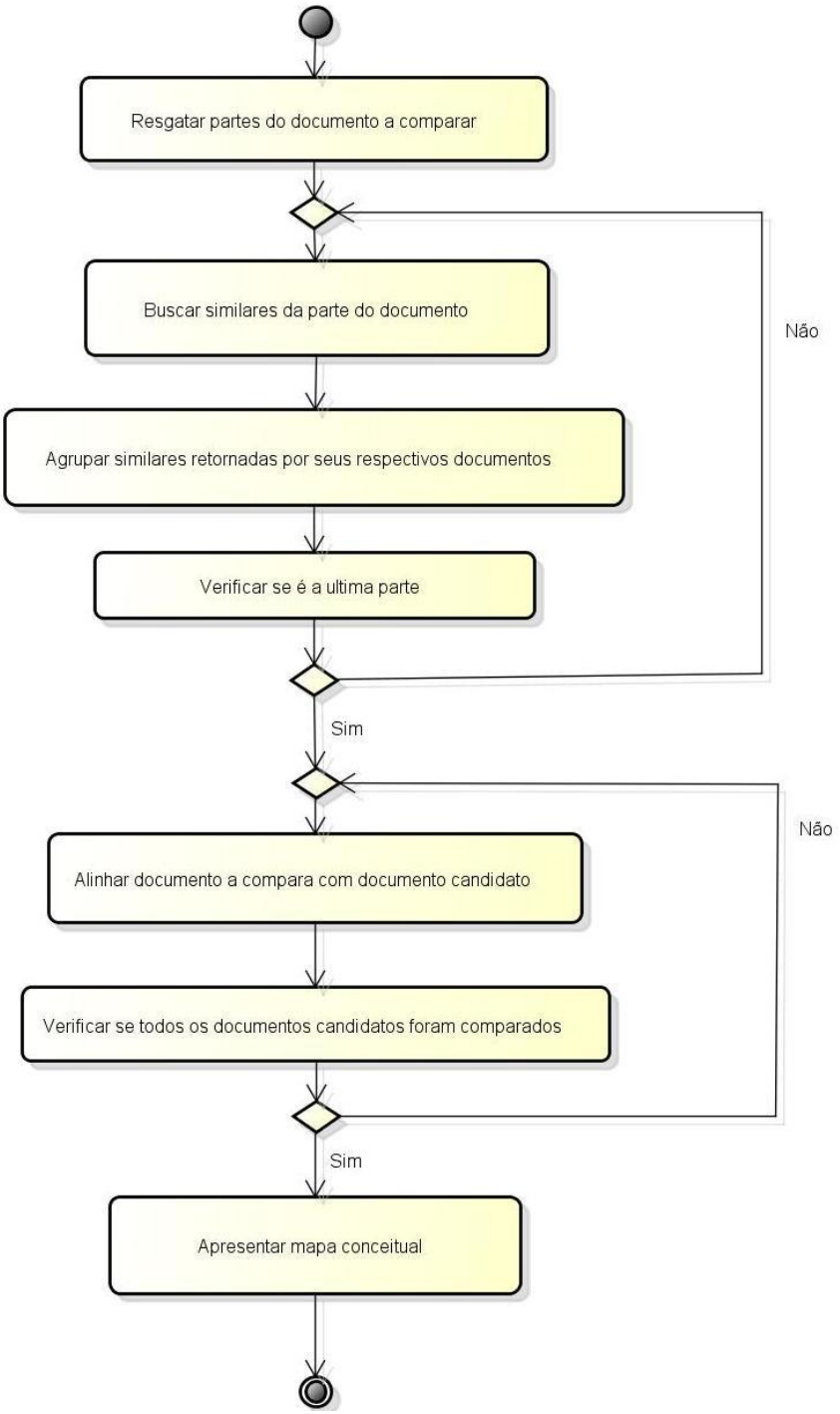


Figura 49: Comportamento do agente Comparador

Na atividade de “buscar similares da parte do documento”, o agente comparador, busca todas as partes de documentos, do mesmo tipo da que estou procurando (parágrafo, sentença ou palavra), e armazena a parte retornada e a medida de similaridade que foi utilizada para encontrar a mesma.

Na atividade de “agrupar partes similares retornadas por seus respectivos documentos” ”, o agente comparador, agrupa todas as partes de um documento em

um único lugar, junto com as suas respectivas pontuações e as partes de documento as quais elas estão relacionadas.

Na atividade de “alinhar o documento a comparar com o documento candidato” ”, o agente comparador, dispara o algoritmo da seção 4.3 e calcula a similaridade estrutural entre os dois documentos. Esta atividade também identifica as operações entre as partes dos dois documentos em questão utilizando o algoritmo da seção 4.3.1.

A atividade de “apresentar o mapa conceitual” organiza todos os documentos candidatos de acordo com a medida de similaridade calculada, na ordem crescente deste valor, em seguida gera um mapa conceitual em forma de grafo para representar os resultados e as relações entre as partes dos documentos.

5.5 Protótipo de ferramenta

O propósito deste protótipo de ferramenta é, a partir de um conjunto de documentos pré-definido, avaliar o grau de Reuso[15] do conteúdo de determinado documento e acompanhar a evolução do mesmo em relação aos outros documentos do conjunto.

A ferramenta mede a afinidade de um documento com o outro no momento da consulta. Para tanto, ela pega todas as partes[16] do documento consultado e verifica quais os documentos candidatos a documentos com seus conteúdos reutilizados. Em seguida, a ferramenta alinha cada documento candidato com o documento consultado para calcular a similaridade estrutural entre eles e, finalmente, a ferramenta escolhe os n[17] documentos com melhor similaridade estrutural para identificar – comparando cada documento com o documento consultado – as operações necessárias para transforma-los no documento consultado.

A apresentação final dos resultados é feita através de um mapa conceitual em forma de grafo no qual os nós representam as partes de documentos e os documentos que as contém, e as arestas representam as relações as partes de documentos e o documento ou as operações entre duas partes de documentos.

Um conjunto de documentos, que podem ser de processadores de texto ou planilhas digitais, é o insumo inicial para a utilização da ferramenta.

---

15 ATRAVÉS DA MEDIDA DE SIMILARIDADE ESTRUTURAL.

16 A PARTE DE DOCUMENTO QUE SERÁ ESCOLHIDA DEPENDERÁ DA GRANULARIDADE ESCOLHIDA, NO CASO DO PROTÓTIPO ESCOLHEMOS A GRANULARIDADE DE PARÁGRAFO.

17 ESTE VALOR É INFORMADO PELO USUÁRIO NO MOMENTO EM QUE A CONSULTA É REALIZADA.

O monitoramento do diretório, a normalização dos documentos e o armazenamento de informação são feitos de forma independente das outras etapas a serem realizadas pela ferramenta, possibilitando que a indexação dos documentos não interfira nas outras etapas da arquitetura. Para tanto, utilizamos as bibliotecas jnotify[18] para o monitoramento das alterações dos documentos, a biblioteca Apache Tika[19] para manipulação e normalização dos documentos digitais e a biblioteca Apache lucene[20] para indexação, busca e recuperação dos documentos e suas partes. Finalmente, um repositório de informações composto de índices do lucene é atualizado com as informações dos documentos processados. A arquitetura física da base do conhecimento da ferramenta é apresentada na Figura 50. Este repositório contém o conteúdo dos documentos em forma de instâncias de objetos da Ontologia de Documentos Digitais e o seu conteúdo foi pré-processado, pelo Standard Analyzer da biblioteca Apache Lucene, para a remoção de espaços em branco indesejados e a tokenização. Também é importante destacar que a granularidade utilizada no contexto deste protótipo de ferramenta é a de parágrafo.

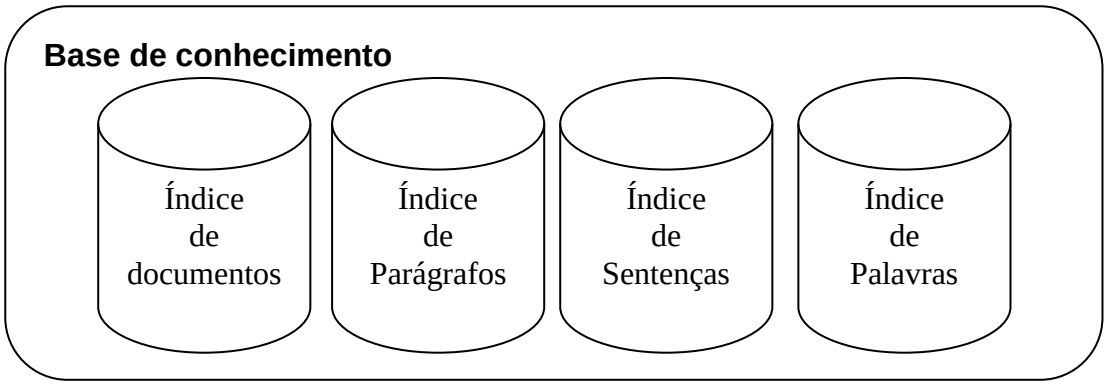


Figura 50: Estrutura física da base de conhecimento

Para a implementação da arquitetura foram utilizadas as seguintes tecnologias:

A biblioteca Apache lucene também foi utilizada como interface de acesso aos dados dos documentos a serem comparados nas etapas de “Resgatar os documentos para comparar” e “Medir a similaridade estrutural”.

18 HTTP://JNOTIFY.SOURCEFORGE.NET/

19 HTTP://TIKA.APACHE.ORG/

20 HTTP://LUCENE.APACHE.ORG/

Para a apresentação final da ferramenta a etapa de “montar mapa conceitual” utiliza a biblioteca jgrapht[21] para a geração do mapa conceitual com os resultados da execução dos algoritmos.

Para atender os requisitos da arquitetura proposta o protótipo de ferramenta foi dividido em dois elementos:

I. O serviço de monitoramento de diretório – é um serviço que monitora um diretório e seus subdiretórios, lista os documentos encontrados dentro dos diretórios, desmembra os documentos em partes de documentos e os armazena nos seus respectivos índices;

II. A ferramenta de comparação – é uma ferramenta que apresenta ao usuário uma listagem dos documentos do diretório que foram desmembrados, possibilita a escolha de um documento para a comparação com os outros e a geração do mapa conceitual do resultado da comparação.

A Figura 51 contextualiza os papéis de cada agente dentro das ferramentas do protótipo.

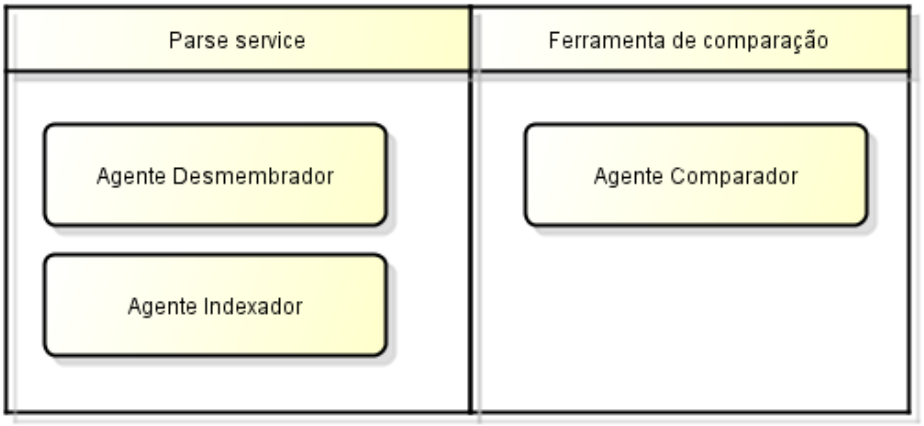


Figura 51: Papéis dos agentes dentro do protótipo

A granularidade de desmembramento definida para este protótipo será a de palavras enquanto a granularidade de comparação será de parágrafos.

A Figura 52 apresenta a tela inicial da ferramenta de comparação. Ela apresenta as informações que o usuário deve apresentar antes que a consulta seja realizada, elas são:

Documento a ser consultado pela ferramenta;

I. Se o usuário deseja que o algoritmo de alinhamento seja executado invertido (do ultimo parágrafo para o primeiro);

21 HTTP://WWW.JGRAPHT.ORG/

II. O número de parágrafos permitidos ao lucene retornar quando se busca um parágrafo no índice de parágrafos;

III. A quantidade de documentos que devem ser apresentados no mapa conceitual.

A ferramenta permite que o usuário escolha quais os documentos ele deseja consultar ou roda para todos os documentos da base (através da opção rodar para todos).

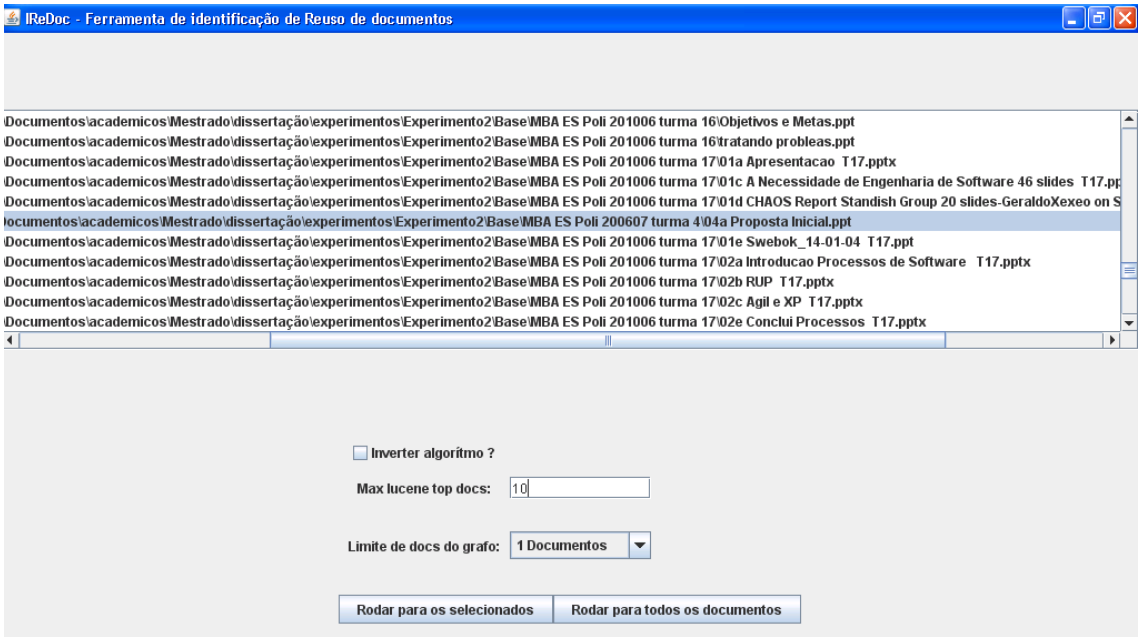


Figura 52: Tela inicial IReDoc

Um mapa conceitual em forma de grafo é gerado ao final da computação da consulta do usuário. A Figura 53 é um exemplo de mapa gerado pela ferramenta, nela o usuário consultou o documento D1 e a ferramenta apresentou que D29 é o documento com melhor similaridade estrutural e, portanto, o melhor candidato a documento que seu conteúdo foi reutilizado em D1.

Os vértices alinhados abaixo do vértice D1 são os parágrafos de D1 enquanto os que estão abaixo de D29 são os parágrafos de D29. As arestas que saem dos vértices de D1 e chegam aos de D29 são as operações que foram identificadas.

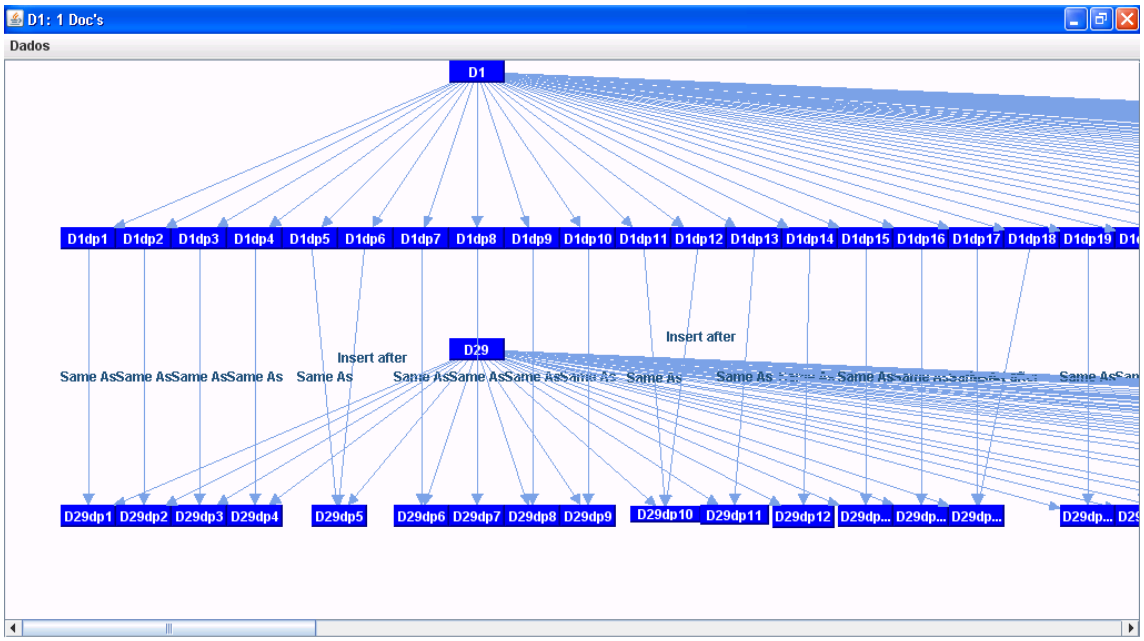


Figura 53: Mapa conceitual gerado por uma consulta ao IReDoc

5.6 Conclusão

Este capítulo apresentou um mecanismo para identificação de Reuso de documentos, nele definimos uma arquitetura, os agentes envolvidos e as atividades que devem ser executadas para que o mesmo funcione. Em seguida apresentamos um protótipo de ferramenta baseado no mecanismo, sua arquitetura física, alguns aspectos de implementação são apresentados e, finalmente, um exemplo da ferramenta sendo utilizada é apresentado. Este capítulo serve como referência para experimentos que serão feitos em capítulos posteriores, com o intuito de avaliar a capacidade das propostas deste trabalho de cumprir com os requisitos desejados assim como os resultados obtidos pelo protótipo apresentado.

6 Uma Ferramenta de Identificação de Reuso de Documentos

Neste capítulo é apresentado um estudo experimental conduzido para avaliar a viabilidade técnica da proposta de identificação de Reuso de documentos digitais. Para tanto, descrevemos os procedimentos executados para a avaliação e apresentamos os resultados obtidos da avaliação do experimento utilizando o IReDoc.

O objetivo do experimento é avaliar a arquitetura que identifica relações entre documentos, de uma base pré-definida, utilizando como métrica a medida de similaridade estrutural definida no capítulo 4 e as operações que o algoritmo identifica no cálculo da mesma medida. Finalmente, os resultados da avaliação com essa medida são analisados.

6.1 Seleção do contexto

Para a realização deste experimento utilizamos, como universo de teste, uma base composta de 384 documentos distintos que são o histórico de apresentações para aulas ministradas para turmas de pós-graduação. O conteúdo dos documentos foi evoluindo a cada turma em que o curso foi ministrado, apresentando até o momento um total de 15 turmas.

Quatro documentos foram escolhidos para identificar o Reuso em seu conteúdo, o objetivo da seleção dos documentos era avaliar a sua evolução com o tempo. Por conveniência a escolha dos documentos foi realizada pelo seu autor e foram selecionados os seguintes documentos da base: D368, D371, D375 e D376[22].

6.2 Participantes

Para avaliar os resultados obtidos escolhemos um grupo de especialistas composto por oito alunos do curso de pós-graduação do PESC[23]. A escolha foi feita por conta do conhecimento nos assuntos que estão no conjunto de documentos utilizado, o que facilitaria a execução do experimento.

22 Uma descrição de cada documento se encontra no Apêndice A.1

23 Programa de Engenharia de Sistemas da Computação, mais informações em www.cos.ufrj.br

6.3 Projeto de estudo

Os quatro documentos selecionados foram consultados na ferramenta preparada com a base escolhida. Para cada consulta um documento foi apresentado como resultado e foi associado um identificador a este resultado. Ele serve para facilitar a associação dos resultados com os documentos. Estes identificadores são apresentados na Tabela 5.

Identificador	Documento consultado	Documento retornado
E1	D368	D312
E2	D371	D210
E3	D375	D228
E4	D376	D90

Tabela 5: Identificador × documentos

A Tabela 6 apresenta uma visão geral dos resultados gerados pela ferramenta para cada consulta realizada. Nela encontramos:

I. O número de parágrafos do documento consultado – são os parágrafos identificados pelo “serviço de monitoramento de diretório”. Para o Experimento E1, por exemplo, o documento D368 apresentou 226 parágrafos;

II. O número de parágrafos do documento retornado – são os parágrafos identificados pelo “serviço de monitoramento de diretório”. Para o Experimento E2, por exemplo, o documento D210 apresentou 139 parágrafos;

III. O valor de similaridade estrutural – Este valor é encontrado ao se alinhar os documentos do experimento parágrafo por parágrafo com o algoritmo que propomos anteriormente. No caso do Experimento E3 o D228 tem uma similaridade estrutural de 183 pontos com o documento D375;

IV. Número de operações encontradas – São o total de operações identificadas pela ferramenta para transformar o documento consultado no documento retornado. Em E4, por exemplo, precisamos de 357 operações para transformar D376 no D90.

Identificador	Paragrafos Dc[24]	Paragrafos Dr[25]	Similaridade Estrutural	Operações encontradas
E1	226	226	115	226
E2	205	139	122	205
E3	410	437	183	448
E4	348	202	177	357

Tabela 6: Parágrafos, similaridade estrutural calculada e operações identificadas pela ferramenta × experimentos

A distribuição das operações em cada experimento é apresentada na Tabela 7, é interessante notar que as porcentagens para as operações de “Adicionar” e “Remover” aumentam quando a diferença entre o número de parágrafos do Dc e do Dr aumentam. Portanto, notamos que, quando o número de parágrafos do Dc é maior do que o do Dr a porcentagem de operações de “Adicionar” aumentará (como é apresentado nos experimentos E2 e E4). No caso do número de parágrafos do Dr ser maior do que o do Dc a porcentagem de operações de “Remover” aumentará (como no caso do experimento E3).

Identificador	Equivalente	Substituir	Inserir	Remover
E1	49,56 %	50,44 %	0,00 %	0,00 %
E2	40,98 %	26,83 %	32,20 %	0,00 %
E3	59,38 %	29,69 %	2,46 %	8,48 %
E4	50,42 %	3,64 %	43,42 %	2,52 %

Tabela 7: Distribuição das operações × experimento

6.4 Instrumentação

O instrumento preparado e usado como apoio ao estudo foi o formulário que foi preenchido pelos avaliadores dos resultados.

O formulário do avaliador está exemplificado na Figura 54. Marcamos o formulário de acordo com as informações que representam, elas são:

I. Vermelho – Apresenta as informações do parágrafo do documento consultado relacionado a operação, nela se encontram o conteúdo do parágrafo e qual parágrafo representa no documento;

II. Amarelo – Apresenta as informações do parágrafo do documento final relacionado a operação, nela se encontram o conteúdo do parágrafo e qual parágrafo representa no documento;

24 Documento consultado

25 Documento retornado

III. Azul – Qual a operação que relaciona os parágrafos;

IV. Verde – Campos onde o usuário irá avaliar a operação de acordo com os critérios de avaliação 1 e avaliação 2 que foram definidos previamente.

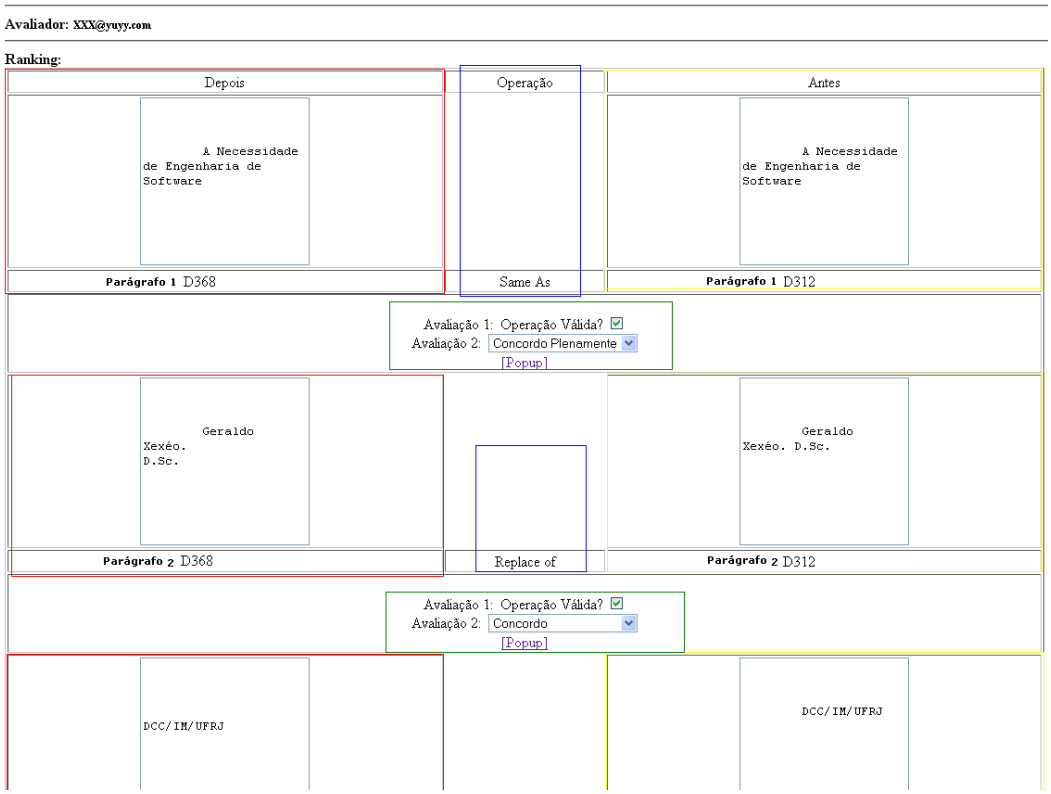


Figura 54: Exemplo de formulário de avaliação de resultados

6.5 Preparação

O especialista avaliará as operações entre as partes de dois documentos sugeridas pela ferramenta. As operações são:

I. “Same As”(Equivalente) – Esta operação identifica que os dois parágrafos, relacionados pela operação nos dois documentos, forma identificados como semelhantes.

II. “Replace of”(Substituir) – Esta operação identifica que o parágrafo do documento final foi relacionado com o outro documento como uma substituição do conteúdo entre os parágrafos.

III. “Insert after”(Adicionar) – Esta operação identifica que o parágrafo do documento final é um novo parágrafo inserido após o parágrafo, do outro documento, relacionado com a operação.

IV. “Delete of”(Remover) – Esta operação identifica que o parágrafo do documento anterior foi removido do documento final.

A Figura 55 Apresenta dois documentos nos quais as operações de “Equivalente” e “Adicionar” aparecem entre eles, a ferramenta gera as seguintes operações para tal exemplo:

I. Parágrafo 1 do documento final “Equivalente” Parágrafo 1 do documento anterior;
II. Parágrafo 2 do documento final “Adicionar” Parágrafo 1 do documento anterior;

Parágrafo 3 do documento final “Equivalente” Parágrafo 2 do documento anterior.

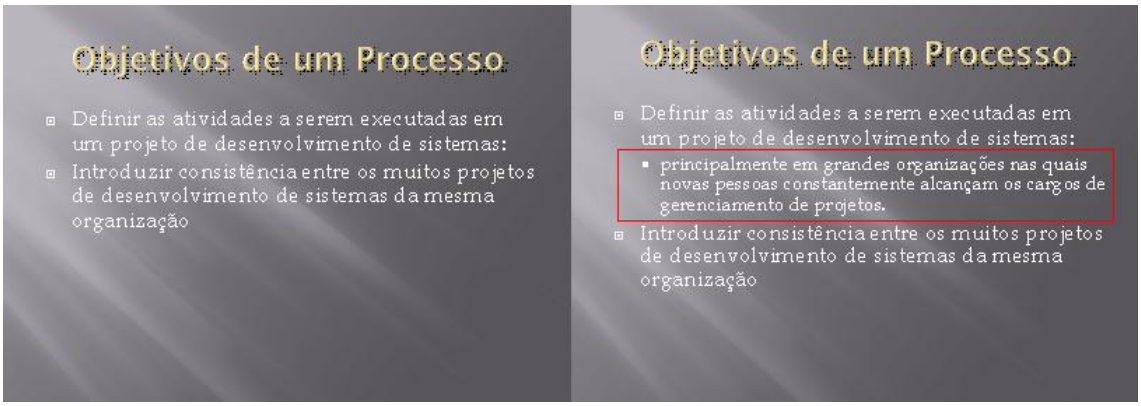


Figura 55: Operação Adicionar

A Figura 56 apresenta dois documentos nos quais as operações de “Equivalente” e “Remover” aparecem entre eles, a ferramenta gera as seguintes operações para tal exemplo:

I. Parágrafo 1 do documento final “Equivalente” Parágrafo 1 do documento anterior;
II. Parágrafo 2 do documento final “Equivalente” Parágrafo 2 do documento anterior;
III. Parágrafo 3 do documento final “Equivalente” Parágrafo 3 do documento anterior;
IV. Parágrafo 3 do documento final “Remover” Parágrafo 4 do documento anterior.

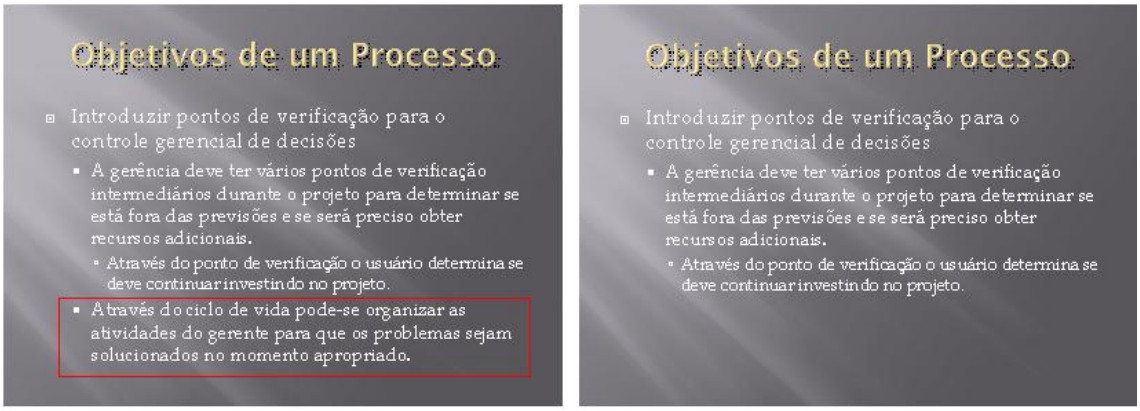


Figura 56: Operação Remover

A Figura 57 apresenta dois documentos nos quais as operações de “Equivalente” e “Remover” aparecem entre eles, a ferramenta gera as seguintes operações para tal exemplo:

I. Parágrafo 1 do documento final “Equivalente” Parágrafo 1 do documento anterior;
II. Parágrafo 2 do documento final “Equivalente” Parágrafo 2 do documento anterior;
III. Parágrafo 3 do documento final “Remover” Parágrafo 3 do documento anterior;
IV. Parágrafo 4 do documento final “Equivalente” Parágrafo 4 do documento anterior;
V. Parágrafo 5 do documento final “Equivalente” Parágrafo 5 do documento anterior;
VI. Parágrafo 6 do documento final “Equivalente” Parágrafo 6 do documento anterior;
VII. Parágrafo 7 do documento final “Equivalente” Parágrafo 7 do documento anterior;
VIII. Parágrafo 8 do documento final “Equivalente” Parágrafo 8 do documento anterior;

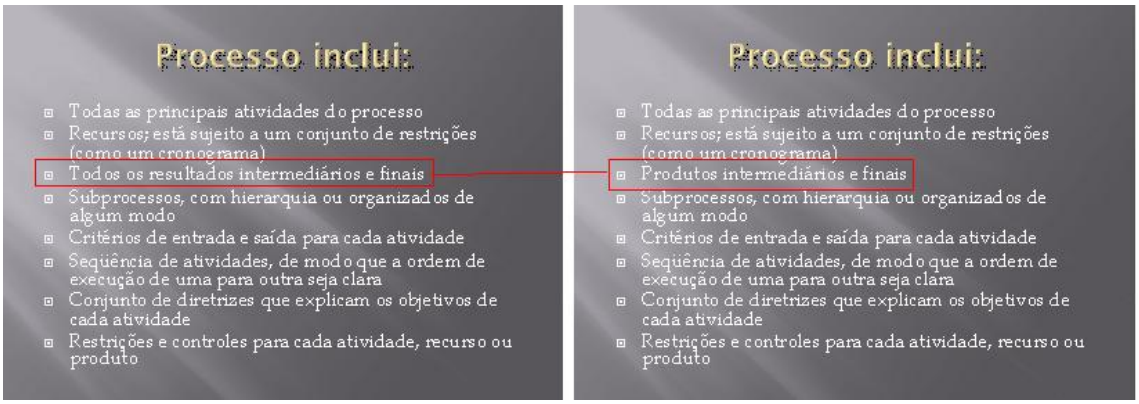


Figura 57: opereção Remover

6.6 Execução

A execução do experimento é representada na Figura 58 da qual destacamos que:

I. A atividade de “Listar documentos que serão avaliados pelo experimento”, representa o momento onde o especialista irá escolher um conjunto de documentos dentro da base para rodar a ferramenta de comparação e avaliar o resultado gerado;

A atividade de “Avaliar resultado” compreende em preencher o formulário de avaliação dos resultados gerados para cada documento escolhido;

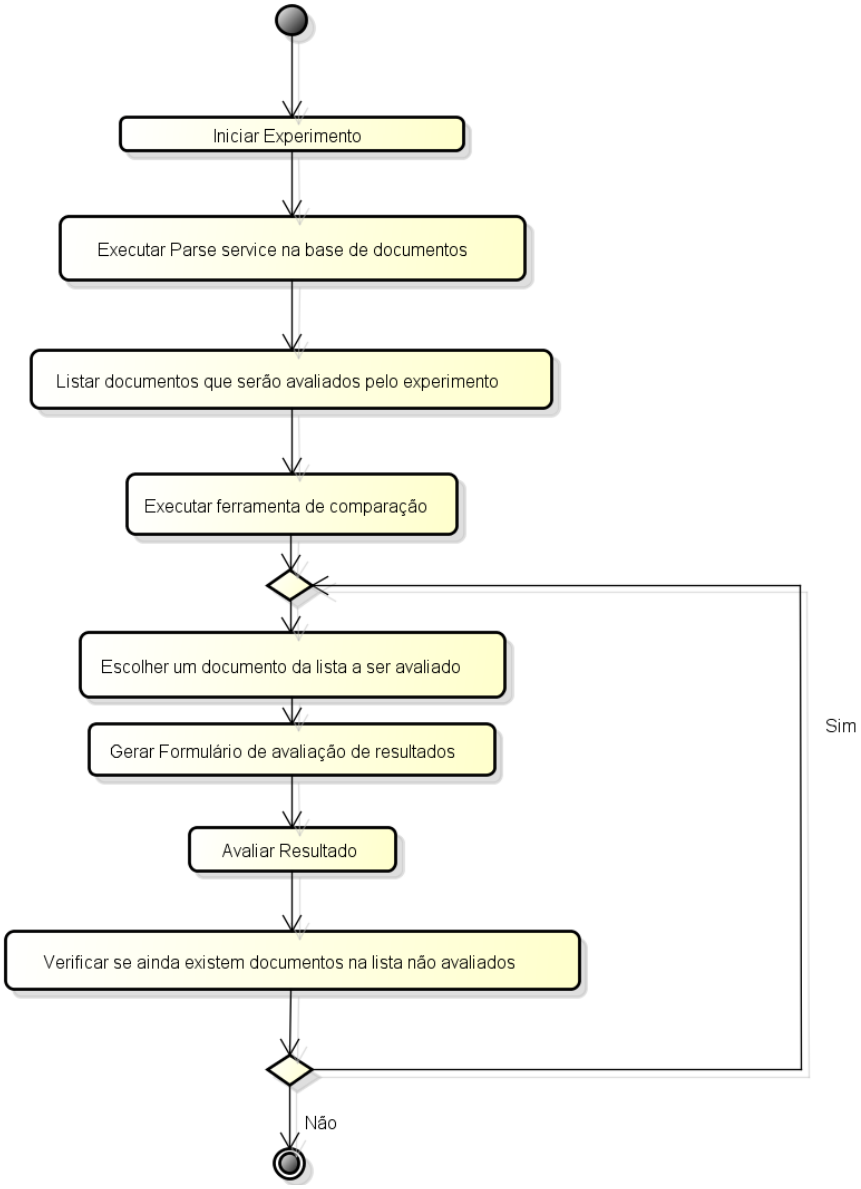


Figura 58: Workflow de execução do experimento

Durante a atividade de “Avaliar resultado” o especialista deve avaliar o resultado obtido de acordo com duas avaliações:

I. Avaliação 1: O especialista deve levar em conta se a operação identificada seria, de forma literal, a que ele identificaria;

II. Avaliação 2: Leva em consideração que existe mais de uma forma para resolver o mesmo problema. Mesmo que a operação não resolva a situação de forma ideal, ela pode levar a um caso próximo do esperado.

Para atender a avaliação 2 foram definidos quatro níveis para avaliar uma operação:

I. Concordo plenamente – o especialista identificaria a operação exatamente da mesma forma;
II. Concordo – o especialista não identificou da mesma forma a operação, porém ela é uma solução válida;
III. Estou em dúvida – o especialista consegue considerar a operação como válida, mas ela não aparenta ser formalmente correta;
IV. Discordo – o especialista afirma que compreende o motivo da operação identificada, porém, não a considera correta;
V. Discordo Plenamente - o especialista afirma que a operação identificada não corresponde à realidade do problema.

6.7 Análise dos resultados

As questões apresentadas no questionário tinham como objetivo a avaliação qualitativa das operações identificadas pela ferramenta de forma a verificar a relevância dos resultados gerados. A seguir são apresentados os resultados mais significativos.

6.7.1 Analise qualitativa

Os resultados obtidos pelas duas avaliações do formulário serão representados aqui em forma de gráficos de barras e tabelas. Neles se encontram percentuais e médias de acordo com as cinco avaliações feitas nos quatro resultados experimentais da ferramenta. Para tanto, observamos os resultados de cada experimento como uma entidade única e, depois, expandimos para a avaliação dos resultados de suas operações.

De uma forma geral, para os quatro casos avaliados, os experimentos apresentaram uma média de aceitação superior a 80 %, conforme o Gráfico 1 apresenta. Este é um bom resultado se levarmos em consideração que, até mesmo para os casos onde obtivemos as piores avaliações individuais, 60,49% para o experimento 2 e 40,30% para o experimento 3, as suas taxas de aceitação médias apresentaram um valor aproximado ou superior do apresentado (90,44% e 83,77% respectivamente).

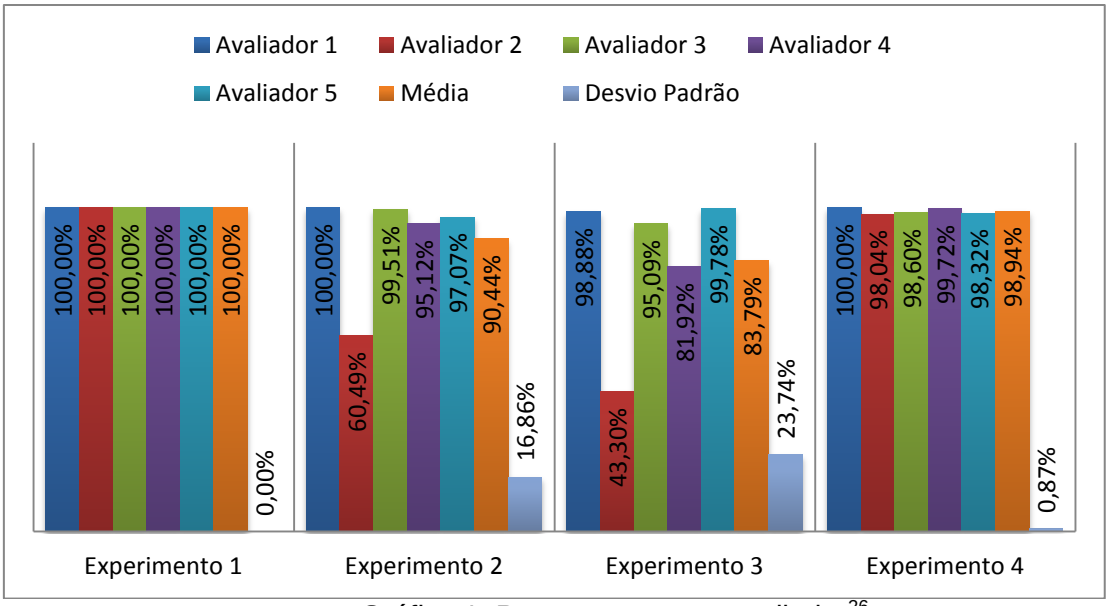


Gráfico 1: Porcentagem por avaliador[26]

O Gráfico 2[27] apresenta a distribuição média da aceitação das operações por experimento, nele observa-se que todas as operações que a ferramenta identifica para os experimentos também foram bem aceitas pelos avaliadores. Observa-se também um comportamento parecido para o maior conjunto de operações identificadas (Experimento 3) e o menor conjunto de operações (Experimento 2). Isto indica que o fator número de operações identificadas não é o único fator que influencia na qualidade dos resultados apresentados. Caso contrario, o comportamento do Experimento 3 deveria se aproximar mais do Experimento 4 do que o que foi apresentado.

26 Representação gráfica dos resultados do Apêndice A.2

27 Grafico gerado dos resultados do Apêndice A.3

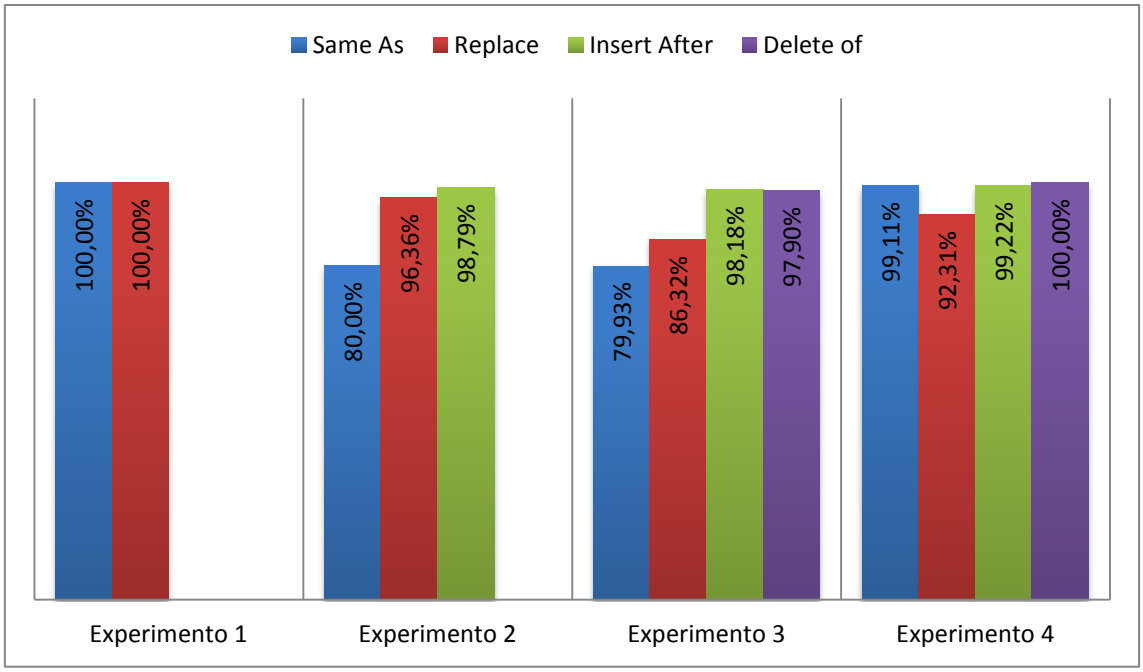


Gráfico 2: Distribuição média das operações corretas por experimento

A identificação de operações é uma atividade na qual o resultado apresentado pode atender um critério pré-determinado. Porém, isso não significa que é o melhor critério para todos os casos, pois, existem situações em que podem existir mais de uma configuração que atente o mesmo proposto. Por conta deste fato, perguntamos aos avaliadores se, no contexto dos experimentos apresentados, as operações atenderiam suas expectativas.

Observando cada experimento como um conjunto de operações sem distinção, conforme representado no

Gráfico 3, nota-se que, por exemplo, os avaliadores do Experimento 1 afirmam que 61,86% das operações estão corretas e de acordo com o que eles apresentariam como solução para aquele caso. 18,05% das operações são classificadas como corretas, porém os avaliadores não identificariam da mesma forma, em 9,91% dos casos eles compreenderam os motivos das operações identificadas, porém não acreditam que sejam formalmente corretas e em 10,18% dos casos eles afirmam que a operação indicada não corresponde com o que ela representa.

A distribuição dessa avaliação por operações é apresentada no Gráfico 4. Nota-se que grande parte da discordância e da dúvida que é apresentada no

Gráfico 3 estão as distribuições das operações de “Remover” dos experimentos avaliados. Contudo, esses valores são inexpressivos ao se comparar esses valores (entre 6,72% e 10,41%) com os do caso de “concordo plenamente” 64,74% é possível avaliar que todas as operações obtiveram uma boa parte de avaliações positivas, onde classificamos como positivas as respostas “concordo” e “concordo plenamente”.

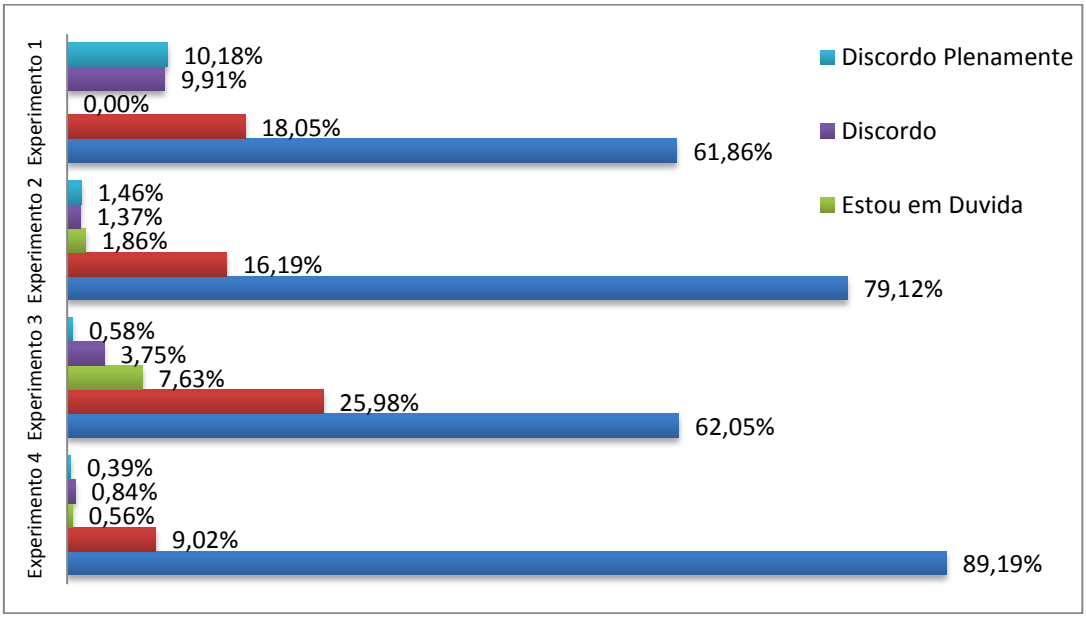


Gráfico 3: Média da distribuição das avaliações por resposta[28]

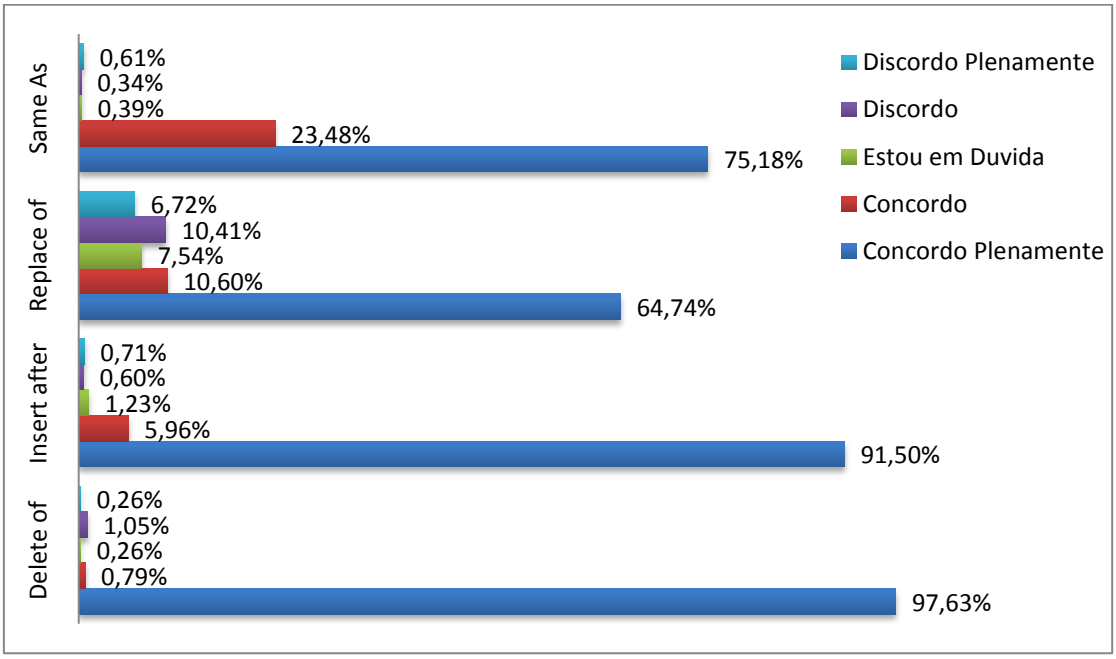


Gráfico 4: Distribuição média das respostas por operação

28 Extraído do Apêndice A.3

7 CONCLUSÃO

Nesta dissertação apresentamos um mecanismo de identificação de Reuso de documentos através do seu conteúdo de forma automática. Para tanto, propomos um modelo ontológico no domínio de documentos, suas partes e relacionamentos possíveis entre os mesmos.

Uma análise dos relacionamentos definidos na seção 3.2 de acordo com os critérios que foram apresentados na seção 1.2.1 foi expressa na Tabela 8. A partir desta tabela concluímos que:

- A criação de novos documentos é observada em Referencia, Incremento, Decremento, Atualização e Mudança de estilo;
- A coleção é observada em Referencia, Atualização e Mudança de estilo;
- A Combinação é observada em Referencia, Incremento, Decremento Atualização e Mudança de estilo;
- A customização apenas não é observada na Cópia Idêntica.
- O Reuso de conteúdo é observado em todos os Relacionamentos excetuando o de Mudança de estilo;
- O Reuso de estilo é observado nos relacionamentos de Incremento, Decremento, Atualização e Mudança de estilo.
- Reuso de estrutura é observados em todos os Relacionamentos;

Classificação	Referência	Incremento	Decremento	Atualização	Mudança de estilo	Cópia Idêntica
Criação	Sim	Sim	Sim	Sim	Sim	Não
Coleção	Sim	Não	Não	Sim	Sim	Não
Combinação	Sim	Sim	Sim	Sim	Sim	Não
Customização	Sim	Sim	Sim	Sim	Sim	Não
-	-	-	-	-	-	-
Reuso de conteúdo	Sim	Sim	Sim	Sim	Não	Sim
Reuso de estilo	Não	Sim	Sim	Sim	Sim	Sim
Reuso de estrutra	Sim	Sim	Sim	Sim	Sim	Sim

Tabela 8: Avaliação dos relacionamentos de acordo com os critérios definidos na seção 1.2.1

Dentro do nosso modelo também definimos uma álgebra de identificação dos relacionamentos que podem surgir entre os documentos. Em seguida, apresentamos uma medida para avaliar o reuso do conteúdo de um documento a partir conteúdo de suas partes e da informação semântica que está agregada a disposição estrutural dessas partes. Apresentamos algoritmos para o cálculo da medida, discutimos e apresentamos um algoritmo para a identificação das operações entre dois documentos.

Propomos também, uma arquitetura para um mecanismo de identificação de Reuso de documentos composta de três camadas e desenvolvemos um protótipo de ferramenta com o objetivo de validar as propostas teóricas apresentadas neste trabalho.

Em todo o conjunto experimental, observou-se uma taxa média de aceitação superior a 80 % tanto para os documentos (como um conjunto de operações) quanto para a avaliação das operações de forma separada. Os resultados obtidos no estudo mostram, de acordo com as nossas expectativas, que:

I. O mecanismo foi bem aceito como ferramenta para identificação de reuso de documentos pelos avaliadores;
II. As operações apresentadas pelo mecanismo também foram bem aceitas.

Inicialmente, acreditávamos que o número de operações identificadas seria um dos fatores preponderantes para a qualidade do resultado da proposta, porém, observamos que ele não influencia de forma tão contundente tal resultado.

O algoritmo de identificação de operações foi proposto utilizando uma abordagem gulosa e necessitávamos de um critério de desempate para casos onde existiam dois ou mais caminhos a escolher com o mesmo peso. O nosso critério priorizou as operações na seguinte ordem decrescente de importância, “Equivalente”, “Remover”, “Substituir” e “Adicionar”.

Por conta do critério de desempate anterior e do fato que a abordagem gulosa sempre procura o caminho de menor custo, observamos que as avaliações para as operações de “Substituir” foram as que mais apresentaram discordância e dúvida com relação ao resultado. Isto acontece, pois existem configurações onde, uma operação de “Substituir” seria mais bem representada por uma operação de “Remover” e outra de “Adicionar”. Apesar disso a operação ainda apresentou uma taxa de concordância significativamente maior comparada com as outras respostas.

7.1 Contribuições

As contribuições deste trabalho são:

- Um meta-modelo ontológico no domínio de documentos digitais que aborda as estruturas dos documentos, seus relacionamentos e as operações de manipulação possíveis entre eles;
- Uma álgebra que representa as possíveis relações entre documentos, suas partes e outros documentos e suas respectivas partes;
- Uma arquitetura para identificação de reuso nos documentos, suas estruturas e relacionamentos;
- Uma medida de similaridade entre documentos e partes de documentos com outros documentos e suas respectivas partes de documentos;
- Um algoritmo fuzzy para o cálculo da medida de similaridade entre documentos e um algoritmo guloso para a detecção de um conjunto mínimo de operações necessárias para se transformar um documento no outro;
- Um mecanismo de identificação de Reuso em documentos;
- Um protótipo de ferramenta que implementa os algoritmos de similaridade e identificação e operações;

7.2 Trabalhos futuros

Como trabalhos futuros podemos destacar:

- Estudo com granularidade diferente;
- Adaptar e utilizar algoritmos de alinhamento genético para a comparação dos documentos e comparar com os resultados obtidos;
- Expandir as operações identificadas para identificar operações compostas pelas operações que a ferramenta já identifica;
- Adaptar os algoritmos com a abordagem de dividir para conquistar;
- Melhorar a interface do protótipo com o usuário, transformando o protótipo em uma ferramenta passível de utilização;
- Expandir o mecanismo de forma a atender as operações de informação visual e agregar o Reuso de renderização ao mesmo;
- Possibilitar a indexação de páginas da web no protótipo para avaliar o reuso do seu conteúdo durante a execução da ferramenta;
- Aplicar a medida e os algoritmos em problemas relacionados a plágio.

- Trocar a granularidade e comparar os resultados com os que já existem;
- Adaptar e utilizar algoritmos de alinhamento genético para a comparação dos documentos e comparar com os resultados obtidos;
- Expandir as operações identificadas para identificar operações compostas pelas operações que a ferramenta já identifica;
- Melhorar a interface do protótipo com o usuário, transformando o protótipo em uma ferramenta passível de utilização;
- Expandir o mecanismo de forma a atender as operações de informação visual e agregar o Reuso de renderização ao mesmo;
- Possibilitar a indexação de páginas da web no protótipo para avaliar o reuso do seu conteúdo durante a execução da ferramenta.

8 BIBLIOGRAFIA

ABU-HANNA, A.; JANSWEIJER, W. Modeling Domain Knowledge Using Explicit Conceptualization. IEEE Expert: Intelligent Systems and Their Applications, [S.l.], 1994. v. 9, n. 5, p. 53-64. . Acesso em: 20 jul. 2010.

Boukottaya, Aida; Charlier, Bernadette; Vanoirbeek, Christine. A Documento Reuse Tool for Communities of Practice. [S.l.], 2006. EC-TEL Workshops. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.142.7832>. Acesso em: 4 mai. 2011.

AIIM. AIIM - What is ECM? What is Enterprise Content Management? Disponível em: <http://www.aiim.org/What-is-ECM-Enterprise-Content-Management.aspx>. Acesso em: 22 out. 2010.

DE ALWIS, B.; SILLITO, J. Why are software projects moving from centralized to decentralized version control systems? Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering. Anais... [S.l: s.n.], 2009. p. 36–39.

AMBRIOLA, V.; BENDIX, L.; CIANCARINI, P. The evolution of configuration management and version control. Software Engineering Journal, [S.l.], 1990. v. 5, n. 6, p. 303–310.

Andy Powell. DCMI Abstract Model. Disponível em: <http://dublincore.org/documentos/abstract-model/>. Acesso em: 21 jul. 2010.

BAEZA-YATES, R.; NAVARRO, G. Integrating contents and structure in text retrieval. ACM Sigmod record, [S.l.], 1996. v. 25, n. 1, p. 67–79.

BAEZA-YATES, R.; RIBEIRO-NETO, B.; others. Modern information retrieval. [S.l.]: Addison-Wesley Reading, MA, 1999.

BAKER, T. Basics of Dublin Core Metadata. ,17 Dez 2009. [S.l: s.n.]. Disponível em: <http://dublincore.org/resources/training/frd_20091217/Tutorial_FRD_baker-1.pdf>. Acesso em: 21 jul. 2010.

BARTA, D.; GIL, J. A System for Documento Reuse. Computer Systems and Software Engineering, 1996., Proceedings of the Seventh Israeli Conference on. Anais... [S.l: s.n.], 1996. p. 83-94.

BEAN, C. A.; GREEN, R. Relationships in the organization of knowledge. [S.l.]: Springer, 2001. ISBN 9780792368137.

BERLINER, B.; PRISMA, I. CVS II: Parallelizing software development. Proceedings of the USENIX Winter 1990 Technical Conference. Anais... [S.l: s.n.], 1990. v. 341, p. 352.

BRANTING, K.; LESTER, J. C. Justification Structures for Documento Reuse. Proceedings of the Third European Workshop on Advances in Case-Based Reasoning. Anais... EWCBR

'96. London, UK: Springer-Verlag, 1996. p. 76–90. ISBN 3-540-61955-0. Disponível em: <http://portal.acm.org/citation.cfm?id=646177.682091>. Acesso em: 3 mai. 2011.

BRANTING, L.; LESTER, J. Justification structures for documento reuse. Advances in Case-Based Reasoning, [S.l.], 1996. p. 76–90.

BURK, M. Knowledge Management: Everyone Benefits by Sharing Information - Vol. 63· No. 3 - Public Roads. Disponível em: <http://www.fhwa.dot.gov/publications/publicroads/99novdec/km.cfm>. Acesso em: 3 mai. 2011.

BURKOWSKI, F. J. Retrieval activities in a database consisting of heterogeneous collections of structured text. Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval. Anais... [S.l: s.n.], 1992. p. 125.

CAVALCANTI, E. P. Revolução da Informação. Caderno de pesquisas em administração, [S.l.], 1995. 1. v. 1.

CHAUMIER, J. Systemes d'information: marché et technologies. [S.l: s.n.], 1986.

CHIAVENATO, I. TEORIA GERAL DA ADMINISTRACAO. [S.l.]: CAMPUS, 2001. ISBN 9788535208498.

CIANCARINI, P. Coordination models and languages as software integrators. ACM Computing Surveys (CSUR), [S.l.], 1996. v. 28, n. 2, p. 302.

CLARKE, C. L. A.; CORMACK, G. V.; BURKOWSKI, F. J. Schema-independent retrieval from heterogeneous structured text. Fourth Annual Symposium on Documento Analysis and Information Retrieval. Anais... [S.l: s.n.], 1995a. p. 279–289.

CLARKE, C. L. A.; CORMACK, G. V.; BURKOWSKI, F. J. An algebra for structured text search and a framework for its implementation. The Computer Journal, [S.l.], 1995b. v. 38, n. 1, p. 43.

COLLINS-SUSSMAN, B.; FITZPATRICK, B. W.; PILATO, C. M. Controle de Versão com Subversion. Disponível em: <http://svnbook.red-bean.com/>. Acesso em: 31 mai. 2010.

CONSENS, M. P.; MILO, T. Algebras for querying text regions (extended abstract). Disponível em: <http://portal.acm.org/citation.cfm?id=212437>. Acesso em: 5 jan. 2011.

CRONIN, B. Esquemas conceituais e estratégicos para a gerênciada informação. Revista da Escola de Biblioteconomia da UFMG, [S.l.], Set 1990. v. 19, n. 2, p. 195-220.

DECHTER, R.; PEARL, J. Generalized best-first search strategies and the optimality of A. Journal of the ACM (JACM), [S.l.], 1985. v. 32, n. 3, p. 505–536.*

DEKKERS, M. History, objectives and approaches of the Dublin Core Metadata Initiative. ,17 Dez 2009. [S.l: s.n.]. Disponível em:

<HTTP://DUBLINCORE.ORG/RESOURCES/TRAINING/FRD_20091217/TUTORIAL_FRD_DEKKERS-1.PDF>. ACESSO EM: 21 JUL. 2010.

DINOS, J. L.; VEGA-RIVEROS, J. F. A DOCUMENTO ONTOLOGY AND AGENT-BASED RDF METADATA RETRIEVAL. NINETEENTH NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE. ANAIS... [S.L: S.N.], 2004.

DOWNES, S. LEARNING OBJECTS: RESOURCES FOR DISTANCE EDUCATION WORLDWIDE. THE INTERNATIONAL REVIEW OF RESEARCH IN OPEN AND DISTANCE LEARNING, [S.L.], 2001. V. 2, N. 1, P. ARTICLE–2.

DUVAL, E.; WARKENTYNE, K.; HAENNI, F.; ET AL. THE ARIADNE KNOWLEDGE POOL SYSTEM. COMMUNICATIONS OF THE ACM, [S.L.], 2001. V. 44, N. 5, P. 72-78. . ACESSO EM: 6 ABR. 2011.

DIJKSTRA, E. A NOTE ON TWO PROBLEMS IN CONNEXION WITH GRAPHS. DISPONÍVEL EM: <HTTP://JMVIDAL.CSE.SC.EDU/LIB/DIJKSTRA59A.HTML>. ACESSO EM: 15 MAR. 2011.

FISCUS, J. G.; AJOT, J.; RADDE, N.; LAPRUN, C. MULTIPLE DIMENSION LEVENSHTEIN EDIT DISTANCE CALCULATIONS FOR EVALUATING AUTOMATIC SPEECH RECOGNITION SYSTEMS DURING SIMULTANEOUS SPEECH. [S.L: S.N.], [S.D.]. DISPONÍVEL EM: <HTTP://WWW.NIST.GOV/SPEECH/PUBLICATIONS/STORAGE_PAPER/LREC06_V0_7.PDF>. ACESSO EM: 15 MAR. 2011.

FRAKES, W. B.; BAEZA-YATES, RICARDO. INFORMATION RETRIEVAL: DATA STRUCTURES AND ALGORITHMS. FACSIMILE ED. [S.L.]: PRENTICE HALL, 1992. ISBN 0134638379.

FRASER, C. W. A GENERALIZED TEXT EDITOR. COMMUNICATIONS OF THE ACM, [S.L.], 1980. V. 23, N. 3, P. 154–158.

GLUSHKO, R. J.; MCGRATH, T. DOCUMENTO ENGINEERING: ANALYZING AND DESIGNING THE SEMANTICS OF BUSINESS SERVICE NETWORKS. PROCEEDINGS OF THE IEEE EEE05 INTERNATIONAL WORKSHOP ON BUSINESS SERVICES NETWORKS. ANAIS... BSN '05. PISCATAWAY, NJ, USA: IEEE PRESS, 2005. P. 2–2. DISPONÍVEL EM: <HTTP://PORTAL.ACM.ORG/CITATION.CFM?ID=1063516.1063518>. ACESSO EM: 6 ABR. 2011.

GODFREY, R.; BIDE, M. THE <INDECS> METADATA FRAMEWORKPRINCIPLES, MODEL AND DATA DICTIONARY. DISPONÍVEL EM: <HTTP://WWW.DOI.ORG/TOPICS/INDECS/INDECS_FRAMEWORK_2000.PDF>. ACESSO EM: 5 JAN. 2011.

GONNET, G. H.; TOMPA, F. W.; SCIENCE, U. OF W. D. OF C. MIND YOUR GRAMMAR: A NEW APPROACH TO MODELLING TEXT. [S.L.]: CITESEER, 1987.

GROZA, T.; HANDSCHUH, S. SALT DOCUMENTO ONTOLOGY (SDO). DISPONÍVEL EM: <HTTP://SALT.SEMANTICAUTHORING.ORG/ONTOLOGIES/SDO.HTML>. ACESSO EM: 21 OUT. 2010.

GRUBER, T. R. TOWARD PRINCIPLES FOR THE DESIGN OF ONTOLOGIES USED FOR KNOWLEDGE SHARING. INT. J. HUM.-COMPUT. STUD., [S.L.], 1995. V. 43, N. 5-6, P. 907-928. . ACESSO EM: 15 JUL. 2010.

GRUNE, D. CONCURRENT VERSIONS SYSTEM, A METHOD FOR INDEPENDENT COOPERATION. IR 113, VRIJE UNIVERSITEIT, [S.L.], 1986. DISPONÍVEL EM: <HTTP://CITESEERX.IST.PSU.EDU/VIEWDOC/SUMMARY?DOI=10.1.1.48.8783>. ACESSO EM: 31 MAI. 2010.

GUARINO, N. FORMAL ONTOLOGY IN INFORMATION SYSTEMS. [S.L.]: IOS PR., 1998.

GUARINO, NICOLA. SEMANTIC MATCHING: FORMAL ONTOLOGICAL DISTINCTIONS FOR INFORMATION ORGANIZATION, EXTRACTION, AND INTEGRATION. INTERNATIONAL SUMMER SCHOOL ON INFORMATION EXTRACTION: A MULTIDISCIPLINARY APPROACH TO AN EMERGING INFORMATION TECHNOLOGY. ANAIS... [S.L.]: SPRINGER-VERLAG, 1997. P. 139-170. ISBN 3-540-63438-X. DISPONÍVEL EM: <HTTP://PORTAL.ACM.ORG/CITATION.CFM?ID=645856.669803>. ACESSO EM: 20 JUL. 2010.

GUERRIERI, E. SOFTWARE DOCUMENTO REUSE WITH XML. PROCEEDINGS OF THE 5TH INTERNATIONAL CONFERENCE ON SOFTWARE REUSE. ANAIS... ICSR '98. WASHINGTON, DC, USA: IEEE COMPUTER SOCIETY, 1998. P. 246–. ISBN 0-8186-8377-5. DISPONÍVEL EM: <HTTP://PORTAL.ACM.ORG/CITATION.CFM?ID=551789.853537>. ACESSO EM: 7 ABR. 2011.

GUTKNECHT, J. CONCEPTS OF THE TEXT EDITOR LARA. COMMUNICATIONS OF THE ACM, [S.L.], 1985. V. 28, N. 9, P. 942–960.

GYSSENS, M.; PAREDAENS, J.; VAN GUCHT, D. A GRAMMAR-BASED APPROACH TOWARDS UNIFYING HIERARCHICAL DATA MODELS. ACM SIGMOD RECORD, [S.L.], 1989. V. 18, N. 2, P. 263–272.

HOLLAND, J. H. ADAPTATION IN NATURAL AND ARTIFICIAL SYSTEMS: AN INTRODUCTORY ANALYSIS WITH APPLICATIONS TO BIOLOGY, CONTROL, AND ARTIFICIAL INTELLIGENCE. [S.L.]: MIT PRESS, 1992. ISBN 9780262581110.

IFLA. FUNCTIONAL REQUIREMENTS FOR BIBLIOGRAPHIC RECORDS - FINAL REPORT - PART 1. DISPONÍVEL EM: <HTTP://ARCHIVE.IFLA.ORG/VII/S13/FRBR/FRBR1.HTM#3.2>. ACESSO EM: 20 MAI. 2010.

KAMPFFMEYER, U. TRENDS IN RECORD, DOCUMENTO AND ENTERPRISE CONTENT MANAGEMENT. PROJECT CONSULT CMBH, [S.L.], 2004.

KHURSHID, K.; FAURE, C.; VINCENT, N. A NOVEL APPROACH FOR WORD SPOTTING USING MERGE-SPLIT EDIT DISTANCE. PROCEEDINGS OF THE 13TH INTERNATIONAL CONFERENCE ON COMPUTER ANALYSIS OF IMAGES AND PATTERNS. ANAIS... CAIP '09. BERLIN, HEIDELBERG: SPRINGER-VERLAG, 2009. P. 213–220. ISBN 978-3-642-03766-5. DISPONÍVEL EM: <HTTP://DX.DOI.ORG/10.1007/978-3-642-03767-2_26>. ACESSO EM: 7 ABR. 2011.

LECERF, L.; CHIDLOVSKII, B. Documento annotation by active learning techniques. Proceedings of the 2006 ACM symposium on Documento engineering. Anais... DocEng '06. New York, NY, USA: ACM, 2006. p. 125–127. ISBN 1-59593-515-0.

LEVENSHTEIN, V. Binary codes capable of correcting deletions, insertions, and reversals. [S.l: s.n.], 1966.

LEVY, D. M. Documento Reuse and Documento Systems. [S.l.], 1993. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.46.6154>. Acesso em: 6 abr. 2011.

LOEFFEN, A. Text databases: A survey of text models and systems. ACM Sigmod Record, [S.l.], 1994. v. 23, n. 1, p. 106.

Microsystems. Microsystems Products: DocXtools for the Legal Industry. Disponível em: <http://www.microsystems.com/products/docxtools-legal.php>. Acesso em: 4 mai. 2011.

MORESI, E., 2000, "Delineando o Valor do Sistema de Informaçao de uma Organizaçao", Revista Ciência da Informação - Instituto Brasileiro de Informação em Ciência e Tecnologia (IBICT), Brasil

NationMaster. StateMaster - Encyclopedia: Data domain. Disponível em: <http://www.statemaster.com/encyclopedia/Data-domain>. Acesso em: 24 dez. 2010.

NAVARRO, GONZALO; BAEZA-YATES, RICARDO. A language for queries on structure and contents of textual databases. Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval. Anais... [S.l: s.n.], 10995. ISBN 0-89791-714-6. Disponível em: <http://portal.acm.org/citation.cfm?id=215206.215336>. Acesso em: 5 jan. 2011.

NSIT. shortest path. Disponível em: <http://xlinux.nist.gov/dads//HTML/shortestpath.html>. Acesso em: 15 mar. 2011.

NURMI, V. V.; SUSTRETOV, D. Twelfth ESSLLI Student Session. [S.l.], 16 Ago 2006

PETERS, T. ROMPENDO AS BARREIRAS DA ADMINISTRAÇAO. [S.l.]: HARBRA, 1993, ISBN 9788529401928.

PINHEIRO, W. A. Arcabouço Autonômico de Padrões para Eliminação de Dados. [S.l: s.n.], 2010. Disponível em: <http://www.cos.ufrj.br/uploadfiles/1274377787.pdf>. Acesso em: 24 dez. 2010.

RADA, R. Hypertext writing and documento reuse: The role of a semantic net. Electronic Publishing—Origination, Dissemination, and Design, ACM ID: 115818, Jul 1990. v. 3, p. 125–140. . Acesso em: 4 mai. 2011.

RAYMOND, E. Understanding Version-Control Systems (DRAFT). Disponível em: <http://www.catb.org/~esr/writings/version-control/version-control.html>. Acesso em: 31 mai. 2010.

REITZ, J. M. Dictionary for library and information science. [S.l.]: Libraries Unlimited, 2004. ISBN 9781591580751.

SCHRIJVER, A. Combinatorial Optimization. Disponível em: <http://www.springer.com/mathematics/applications/book/978-3-540-44389-6>. Acesso em: 9 fev. 2011.

SCHWARZ, G. Estimating the dimension of a model. The annals of statistics, [S.l.], 1978. p. 461–464.

SETTLES, B. Active Learning Literature Survey. Machine Learning, [S.l.], 1994. v. 15, n. 2, p. 201–221.

STREITZ, N. A.; HANNEMANN, J.; THÜRING, M. From ideas and arguments to hyperdocumentos: travelling through activity spaces. Proceedings of the second annual ACM conference on Hypertext. Anais... HYPERTEXT '89. New York, NY, USA: ACM, 1989. p. 343–364. ISBN 0-89791-339-6.

TAGUE, J.; SALMINEN, A.; MCCLELLAN, C. Complete formal model for information retrieval systems. Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval. Anais... [S.l: s.n.], 1991. p. 14–20.

Text-Editor. Text-Editor.org | What is a Text Editor? Disponível em: <http://www.text-editor.org/>. Acesso em: 20 out. 2010.

TILLETT, B. A conceptual model for the Bibliographic Universe. [S.l.]: Technicalities, 2003.

TITEMORE, R. G.; GERVAIS, S.; BOONE, K. W. System and method for auto-reuse of documento text. ,1 Nov 2007. [S.l: s.n.]. Disponível em: <http://www.freepatentsonline.com/y2007/0011608.html>. Acesso em: 4 mai. 2011.

TRAPPEY, A. J. C.; TRAPPEY, C. V.; HSU, F.-C.; HSIAO, D. W. A Fuzzy Ontological Knowledge Documento Clustering Methodology. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), [S.l.], 2009. v. 39, n. 3, p. 806-814.

URDANETA. Gestión de la inteligencia, aprendizaje tecnológico ymodernización del trabajo informacional : retos y oportunidades. ,1992. [S.l.]: Caracas: Universidad Simón Bolivar.

VERBERT, K.; OCHOA, X.; DUVAL, E. The ALOCOM framework: towards scalable content reuse. Disponível em: <https://lirias.kuleuven.be/handle/123456789/158237>. Acesso em: 6 abr. 2011.

XING, G.; MALLA, C. R.; XIA, Z.; VENKATA, S. D. Computing edit distances between an XML documento and a schema and its application in documento classification. Proceedings of the 2006 ACM symposium on Applied computing. Anais... SAC '06. New York, NY, USA: ACM, 2006. p. 831–835. ISBN 1-59593-108-2.

ZADEH, L. A.; KLIR, G. J.; YUAN, B. Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers. [S.l.]: World Scientific Pub Co Inc, 1996. ISBN 9810224214.

ZADEH, L. A. Fuzzy logic. Computer, [S.l.], 1988. v. 21, n. 4, p. 83-93.

ZHOU, C.; LU, Y.; ZOU, L.; HU, R. Evaluate structure similarity in XML documentos with merge-edit-distance. Proceedings of the 2007 international conference on Emerging technologies in knowledge discovery and data mining. Anais... PAKDD'07. Berlin, Heidelberg: Springer-Verlag, 2007. p. 301–311. ISBN 3-540-77016-X, 978-3-540-77016-9. Disponível em: <http://portal.acm.org/citation.cfm?id=1780582.1780616>. Acesso em: 7 abr. 2011.

APÊNDICE A – Resultados do experimento

Este apêndice apresenta os resultados da avaliação da ferramenta proposta nesta dissertação.

A.1. : Tabela de documentos manipulados pelo experimento

Identificador	Nome do documento	Número de parágrafos
D90	MBA ES Poli 200711 turma 7\ARQ17 Event Driven Process Chain 42 slides.ppt	202
D210	MBA ES Poli 200810 turma 10\Aulas\02 ISO_12207_2008Systems_and_Software_Engineering.ppt	139
D228	MBA ES Poli 200810 turma 10\Subsidios\PortugueseScrum.ppt	437
D312	MBA ES Poli 201006 turma 15\01c A Necessidade de Engenharia de Software 46 slides.pptx	226
D368	MBA ES Poli 201006 turma 17\01c A Necessidade de Engenharia de Software 46 slides T17.pptx	226
D371	MBA ES Poli 201006 turma 17\02a Introducao Processos de Software T17.pptx	205
D375	MBA ES Poli 201006 turma 17\03 Scrum T17.pptx	410
D376	MBA ES Poli 201006 turma 17\04 Event Driven Process Chain 42 slides T17.ppt	348

A.2. : Tabela de porcentagem de acertos por avaliador

Identificador	Avaliador 1	Avaliador 2	Avaliador 3	Avaliador 4	Avaliador 5	Média
E1	100,00 %	100,00 %	100,00 %	100,00 %	100,00 %	100,00%
E2	100,00 %	60,49 %	99,51 %	95,12 %	97,07 %	90,44%
E3	98,88 %	43,30 %	95,09 %	81,92 %	99,78 %	83,79%
E4	100,00 %	98,04 %	98,60 %	99,72 %	98,32 %	98,94%

A.3. : Distribuição das avaliações por respostas

Identificador	Avaliador	Concordo Plenamente	Concordo	Estou em Duvida	Discordo	Discordo Plenamente
E1	Avaliador 1	99,56%	0,44%	0,00%	0,00%	0,00%
E1	Avaliador 2	57,52%	42,48%	0,00%	0,00%	0,00%
E1	Avaliador 3	52,65%	47,35%	0,00%	0,00%	0,00%
E1	Avaliador 4	49,12%	0,00%	0,00%	0,00%	50,88%

E1	Avaliador 5	50,44 %	0,00 %	0,00 %	49,56 %	0,00 %
E2	Avaliador 1	98,05%	1,46%	0,49%	0,00%	0,00%
E2	Avaliador 2	54,15%	40,00%	5,37%	0,49%	0,00%
E2	Avaliador 3	91,71%	7,32%	0,49%	0,00%	0,49%
E2	Avaliador 4	88,78%	4,39%	1,95%	2,93%	1,95%
E2	Avaliador 5	62,93%	27,80%	0,98%	3,41%	4,88%
E3	Avaliador 1	85,49%	0,22%	13,62%	0,45%	0,22%
E3	Avaliador 2	26,79%	55,58%	16,74%	0,67%	0,22%
E3	Avaliador 3	75,89%	15,85%	7,14%	1,12%	0,00%
E3	Avaliador 4	80,36%	1,34%	0,67%	16,52%	1,12%
E3	Avaliador 5	41,74%	56,92%	0,00%	0,00%	1,34%
E4	Avaliador 1	99,16%	0,56%	0,00%	0,00%	0,28%
E4	Avaliador 2	97,20%	0,56%	0,56%	1,40%	0,28%
E4	Avaliador 3	96,92%	0,00%	0,28%	2,24%	0,56%
E4	Avaliador 4	54,90%	43,98%	0,56%	0,00%	0,56%
E4	Avaliador 5	97,76%	0,00%	1,40%	0,56%	0,28%

A.4. : Porcentagem de acertos das operações por avaliador

Identificador	Avaliador	Equivalente	Replace	Adicionar	Remover
E1	Avaliador 1	100,00%	100,00%	-	-
E1	Avaliador 2	100,00%	100,00%	-	-
E1	Avaliador 3	100,00%	100,00%	-	-
E1	Avaliador 4	100,00%	100,00%	-	-
E1	Avaliador 5	100,00%	100,00%	-	-
E2	Avaliador 1	100,00%	100,00%	100,00%	-
E2	Avaliador 2	4,76%	98,18%	100,00%	-
E2	Avaliador 3	98,81%	100,00%	100,00%	-
E2	Avaliador 4	97,62%	87,27%	98,48%	-
E2	Avaliador 5	98,81%	96,36%	95,45%	-
E3	Avaliador 1	98,50%	99,25%	100,00%	100,00%
E3	Avaliador 2	6,02%	96,99%	100,00%	100,00%
E3	Avaliador 3	98,87%	87,97%	100,00%	92,11%
E3	Avaliador 4	96,62%	47,37%	90,91%	97,37%
E3	Avaliador 5	99,62%	100,00%	100,00%	100,00%
E4	Avaliador 1	100,00%	100,00%	100,00%	100,00%
E4	Avaliador 2	98,89%	84,62%	98,06%	100,00%
E4	Avaliador 3	98,33%	84,62%	100,00%	100,00%
E4	Avaliador 4	99,44%	100,00%	100,00%	100,00%
E4	Avaliador 5	98,89%	92,31%	98,06%	100,00%

A.5. : Média das avaliações por operação de cada experimento

Identificador	Operação	Concordo Plenamente	Concordo	Estou em Duvida	Discordo	Discordo Plenamente
E1	Substituir	43,16%	17,37%	0,00%	19,47%	20,00%
E1	Equivalente	80,89%	18,75%	0,00%	0,18%	0,18%

E2	Adicionar	79,70%	17,88%	1,22%	0,91%	0,30%
E2	Substituir	79,64%	7,64%	5,45%	3,64%	3,64%
E3	Remover	95,26%	1,58%	0,53%	2,10%	0,53%
E3	Adicionar	96,36%	0,00%	1,82%	0,00%	1,82%
E3	Substituir	53,08%	12,78%	23,16%	10,83%	0,15%
E3	Equivalente	60,38%	37,14%	1,13%	0,60%	0,75%
E4	Remover	100,00%	0,00%	0,00%	0,00%	0,00%
E4	Adicionar	98,45%	0,00%	0,65%	0,90%	0,00%
E4	Substituir	83,08%	4,61%	1,54%	7,69%	3,08%
E4	Equivalente	81,11%	17,56%	0,45%	0,33%	0,56%

A.6. : Distribuição das operações por resposta

Identificador	Avaliador	Operação	Concordo Plenamente	Concordo	Estou em Duvida	Discordo	Discordo Plenamente
E1	Avaliador 1	Substituir	99,12%	0,88%	0,00%	0,00%	0,00%
E1	Avaliador 1	Equivalente	100,00%	0,00%	0,00%	0,00%	0,00%
E1	Avaliador 2	Substituir	16,67%	83,33%	0,00%	0,00%	0,00%
E1	Avaliador 2	Equivalente	99,11%	0,89%	0,00%	0,00%	0,00%
E1	Avaliador 3	Substituir	97,37%	2,63%	0,00%	0,00%	0,00%
E1	Avaliador 3	Equivalente	7,14%	92,86%	0,00%	0,00%	0,00%
E1	Avaliador 4	Substituir	0,00%	0,00%	0,00%	0,00%	100,00%
E1	Avaliador 4	Equivalente	99,11%	0,00%	0,00%	0,00%	0,89%
E1	Avaliador 5	Substituir	2,63 %	0,00%	0,00%	97,37 %	0,00%
E1	Avaliador 5	Equivalente	99,11 %	0,00%	0,00%	0,89 %	0,00%
E2	Avaliador 1	Adicionar	98,48%	0,00%	1,52%	0,00%	0,00%
E2	Avaliador 1	Substituir	96,36%	3,64%	0,00%	0,00%	0,00%
E2	Avaliador 1	Equivalente	98,81%	1,19%	0,00%	0,00%	0,00%
E2	Avaliador 2	Adicionar	93,94%	4,55%	1,52%	0,00%	0,00%
E2	Avaliador 2	Substituir	80,00%	1,82%	18,18%	0,00%	0,00%
E2	Avaliador 2	Equivalente	5,95%	92,86%	0,00%	1,19%	0,00%
E2	Avaliador 3	Adicionar	93,94%	4,55%	1,52%	0,00%	0,00%
E2	Avaliador 3	Substituir	78,18%	21,82%	0,00%	0,00%	0,00%
E2	Avaliador 3	Equivalente	98,81%	0,00%	0,00%	0,00%	1,19%
E2	Avaliador 4	Adicionar	98,48%	0,00%	0,00%	0,00%	1,52%
E2	Avaliador 4	Substituir	74,55%	5,45%	7,27%	10,91%	1,82%
E2	Avaliador 4	Equivalente	90,48%	7,14%	0,00%	0,00%	2,38%
E2	Avaliador 5	Adicionar	13,64%	80,30%	1,52%	4,55%	0,00%
E2	Avaliador 5	Substituir	69,09%	5,45%	1,82%	7,27%	16,36%
E2	Avaliador 5	Equivalente	97,62%	1,19%	0,00%	0,00%	1,19%
E3	Avaliador 1	Remover	94,74%	0,00%	0,00%	5,26%	0,00%
E3	Avaliador 1	Adicionar	100,00%	0,00%	0,00%	0,00%	0,00%
E3	Avaliador 1	Substituir	59,40%	0,00%	40,60%	0,00%	0,00%
E3	Avaliador 1	Equivalente	96,62%	0,38%	2,63%	0,00%	0,38%
E3	Avaliador 2	Remover	100,00%	0,00%	0,00%	0,00%	0,00%
E3	Avaliador 2	Adicionar	100,00%	0,00%	0,00%	0,00%	0,00%
E3	Avaliador 2	Substituir	42,11%	3,76%	53,38%	0,75%	0,00%
E3	Avaliador 2	Equivalente	5,64%	91,73%	1,50%	0,75%	0,38%
E3	Avaliador 3	Remover	92,11%	2,63%	0,00%	5,26%	0,00%
E3	Avaliador 3	Adicionar	90,91%	0,00%	9,09%	0,00%	0,00%

E3	Avaliador 3	Substituir	27,07%	49,62%	21,80%	1,50%	0,00%
E3	Avaliador 3	Equivalente	97,37%	1,50%	0,75%	0,38%	0,00%
E3	Avaliador 4	Remover	89,47%	5,26%	2,63%	0,00%	2,63%
E3	Avaliador 4	Adicionar	90,91%	0,00%	0,00%	0,00%	9,09%
E3	Avaliador 4	Substituir	44,36%	3,01%	0,00%	51,88%	0,75%
E3	Avaliador 4	Equivalente	96,62%	0,00%	0,75%	1,88%	0,75%
E3	Avaliador 5	Remover	100,00%	0,00%	0,00%	0,00%	0,00%
E3	Avaliador 5	Adicionar	100,00%	0,00%	0,00%	0,00%	0,00%
E3	Avaliador 5	Remover	92,48%	7,52%	0,00%	0,00%	0,00%
E3	Avaliador 5	Equivalente	5,64%	92,11%	0,00%	0,00%	2,26%
E4	Avaliador 1	Remover	100,00%	0,00%	0,00%	0,00%	0,00%
E4	Avaliador 1	Adicionar	100,00%	0,00%	0,00%	0,00%	0,00%
E4	Avaliador 1	Substituir	92,31%	7,69%	0,00%	0,00%	0,00%
E4	Avaliador 1	Equivalente	98,89%	0,56%	0,00%	0,00%	0,56%
E4	Avaliador 2	Remover	100,00%	0,00%	0,00%	0,00%	0,00%
E4	Avaliador 2	Adicionar	97,42%	0,00%	0,65%	1,94%	0,00%
E4	Avaliador 2	Substituir	69,23%	15,38%	0,00%	15,38%	0,00%
E4	Avaliador 2	Equivalente	98,89%	0,00%	0,56%	0,00%	0,56%
E4	Avaliador 3	Remover	100,00%	0,00%	0,00%	0,00%	0,00%
E4	Avaliador 3	Adicionar	97,42%	0,00%	0,00%	2,58%	0,00%
E4	Avaliador 3	Substituir	69,23%	0,00%	0,00%	23,08%	7,69%
E4	Avaliador 3	Equivalente	98,33%	0,00%	0,56%	0,56%	0,56%
E4	Avaliador 4	Remover	100,00%	0,00%	0,00%	0,00%	0,00%
E4	Avaliador 4	Adicionar	100,00%	0,00%	0,00%	0,00%	0,00%
E4	Avaliador 4	Substituir	100,00%	0,00%	0,00%	0,00%	0,00%
E4	Avaliador 4	Equivalente	10,56%	87,22%	1,11%	0,00%	1,11%
E4	Avaliador 5	Remover	100,00%	0,00%	0,00%	0,00%	0,00%
E4	Avaliador 5	Adicionar	97,42%	0,00%	2,58%	0,00%	0,00%
E4	Avaliador 5	Substituir	84,62%	0,00%	7,69%	0,00%	7,69%
E4	Avaliador 5	Equivalente	98,89%	0,00%	0,00%	1,11%	0,00%