Interoperabilidade de documentos digitais usando ontologias

Erika Guetti Suca

Dissertação
ao
Instituto de Matemática e Estatística
da
Universidade de São Paulo
para
obtenção do título
de
Mestre

Programa: Ciência da Computação
Orientador: Prof. Dr. Flávio Soares Côrrea da Silva

Durante o desenvolvimento deste trabalho o autor recebeu auxílio financeiro da FUSP

São Paulo, agosto de 2012

Interoperabilidade de documentos digitais usando ontologias

Esta é a versão original da dissertação elaborada pela candidata Erika Guetti Suca, tal como submetida à Comissão Julgadora.

Resumo

Suca G., Erika **Interoperabilidade de documentos digitais usando ontologias**. 2012. 70 f. Dissertação (Mestrado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2012.

As organizações precisam trocar informações de forma simples e eficiente, com custos tão baixos quanto possível. Essas informações às vezes são apresentadas na forma de documentos com formato e conteúdos pré-definidos. Esses documentos podem ser equivalentes ou quase equivalentes, porém bastantes distintos em diferentes organizações. Numa mesma organização, os documentos podem ser diferentes em contextos históricos.

O propósito deste trabalho é facilitar a distribuição dos documentos, superando o problema dos formatos com os quais foram criados. O objetivo é possibilitar a interoperabilidade de documentos e atingir a portabilidade simples e confiável de documentos através da reutilização de formatos e conteúdos, em diferentes combinações plausíveis.

Propomos, usar ontologias como solução ao problema da falta de interoperabilidade nas implementações de formatos de documentos. Como prova de conceito consideramos a portabilidade entre os formatos padrão *ODF* (Open Document Format) e *OOXML* (Office Open XML).
Palavras-chave: Interoperabilidade de Documentos, Portabilidade de Documentos, Ontologias, *OOXML*, *ODF*.

Abstract

Suca G., Erika **Digital Documents Interoperability using Ontologies**. 2012. 70 p. Dissertation (Master) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2012.
Organizations need to exchange information simple and efficient, with costs as low as possible. Such information is sometimes presented as documents and with pre-defined content. These documents may be equivalent or nearly equivalent, but quite distinct in different organizations. In the same organization, documents can be different historical contexts.

The purpose of this work is to enable interoperability of documents and achieve portability of digital documents through the reuse of content and format in different plausible combinations.

We propose the characterization of digital documents using ontologies as a solution to the problem of lack of interoperability in the implementations of document formats. As proof of concept we consider the portability between *ODF* (Open Document Format) and *OOXML* (Office Open XML) document formats.
Keywords: Documents Interoperability, Documents Portability, ontologies, *OOMXL*, *ODF*.

Sumário

Lista de Abreviaturas **vii**

Lista de Figuras **ix**

Lista de Tabelas **xi**

1 Introdução **1**
1.1 Motivação 1
1.2 Objetivos 2
1.3 Trabalhos Correlatos 2
1.4 Organização do Trabalho 3

2 Engenharia de Documentos **5**
2.1 Documentos 5
2.2 Preservação dos Documentos Digitais 5
2.3 Interoperabilidade e Portabilidade de Documentos 6
2.4 A Engenharia de Documentos 6
2.4.1 Modelos, Padrões e Reuso 7
2.4.2 O Desafio da Interoperabilidade 7
2.5 Fases da Engenharia de Documentos 8
2.6 Sumário 11

3 Padrões de Formato de Documentos Digitais **15**
3.1 Introdução 15
3.2 Office Open XML 15
3.3 OpenDocument Format 17
3.4 Portabilidade dos Formatos OOXML e ODF 18
3.5 Sumário 21

4 Ontologias **23**
4.1 Introdução 23
4.2 Ontologias e Ciências da Computação 23
4.3 Componentes de Ontologias 24
4.4 Classificação de Ontologias 24
4.4.1 Classificação quanto ao Tipo de Informação 24
4.4.2 Classificação segundo seu Espectro Semântico 25

4.5 Aplicações, Ventagens e Problemas sobre Ontologias 26
4.6 Linguagens para Representação de Ontologias 27
4.6.1 DARPA Agent Markup Language + Ontology Interface Language (DAML+OIL) 27
4.6.2 Resource Description Framework (RDF e RDF-Schema) 27
4.6.3 Web Ontology Language (OWL) 28
4.6.4 Espaços de Nome 28
4.6.5 Cabeçalho 29
4.6.6 Classes e Subclasses 29
4.6.7 Instâncias 29
4.6.8 Propriedades das Classes 30
4.6.9 Restrições das Propriedades 30
4.7 Ferramentas para a Construção de Ontologias 31
4.8 Ontologias para Modelamento Conceitual 31
4.9 Sumário 31

5 Modelo Conceitual de Documentos 33
5.1 Modelamento Conceitual de Documentos 33
5.1.1 Aplicação da Abordagem da Engenharia de Documentos 33
5.1.2 Construção das Ontologias 33
5.1.3 Caracterização dos Documentos Digitais usando Ontologias 36
5.1.4 Geração do Documento Digital 39
5.2 Resultados 39
5.3 Conclusões 40

6 Conclusões 47
6.1 Considerações Finais 47
6.2 Contribuições desse trabalho 47
6.3 Sugestões para Pesquisas Futuras 47

A Documentos de prova 49

Referências Bibliográficas 53

Lista de Abreviaturas

ACM	Association for Computing Machinery
DAML	DARPA Agent Markup Language
DRS	Digital Rosetta Stone
e-PING	Padrões de Interoperabilidade de Governo Eletrônico
KIF	Knowledger Interchange Format
KSL	Knowledge Systems Laboratory
OOXML	Office Open XML
ODF	Open Document Format
OIL	Ontology Interface Language
OKBC	Open Knowledge Base Connectivity
OWL	Web Ontology Language
RDF	Resource Description Framework
TIC	Tecnologia de Informação e Comunicação
UML	Unified Modeling Language
URI	Universal Resource Identifier
XML	eXtension Markup Language
W3C	World Wide Web Consortium

Lista de Figuras

2.1 O desafio da interoperabilidade. 8
2.2 Níveis de perspectivas de análise nas fases da engenharia de documentos(Glushko e McGrath , 2008). 9

3.1 Componentes principais do *Office Open XML*(Ngo, 2008). 17
3.2 Estrutura dos contêineres dos formatos *OOXML* e *ODF*. 19
3.3 Exemplo de texto básico nas especificações de *OOXML* e *ODF*. 19

4.1 Estrutura do exemplo de RDF 28
4.2 Código de exemplo de uma estrutura RDF 28
4.3 Exemplo de espaços de nome para uma ontologia em *OWL*. 29
4.4 Exemplo de cabeçalho para uma ontologia em *OWL*. 29
4.5 Exemplo de definição de classes para uma ontologia em *OWL*. 29
4.6 Exemplo de definição de uma propriedade tipo **object** para uma ontologia em *OWL*. 30
4.7 Exemplo de definição de uma propriedade tipo **datatype** para uma ontologia em *OWL*. 30
4.8 Exemplo de definição de uma propriedade tipo **datatype** associado a um indivíduo. 30

5.1 Um modelo conceitual para compartilhamento de documentos. 34
5.2 Geração do documento digital a partir da Ontologia de Documento. 35
5.3 Ficha de aluno com a informação pessoal. 35
5.4 Ficha de aluno com a informação acadêmica. 36
5.5 Ontologia de Conteúdo para o domínio *Ficha de aluno*. 36
5.6 Ontologia de Formato. 37
5.7 Instância da classe Estudante do aluno **Vinicius Rezende Oliveira**. 37
5.8 Instância da classe Curso do curso de graduação do aluno **Vinicius Rezende Oliveira**. 38
5.9 Instância da classe Curso do curso de mestrado do aluno **Vinicius Rezende Oliveira**. 38
5.10 Instância da classe Curso do curso de doutorado do aluno **Vinicius Rezende Oliveira** 38
5.11 Instância do Instituto de ensino da graduação. 38
5.12 Instância do Instituto de ensino do doutorado. 38
5.13 Instância Professor. 39
5.14 Instância do Histórico do aluno. 39
5.15 Texto do cabeçalho de documento Ficha de Aluno. 39
5.16 Árvore das instâncias da Ontologia de Formato para o documento Ficha de Aluno . 40

5.17 Instância da classe Tabela que contém os dados pessoais do aluno **Vinicius Rezende Oliveira**. 41
5.18 Fluxo completo de Interoperabilidade de Documentos. 42

A.1 Versão do documento gerado para DOCX 49
A.2 Versão do documento gerado para ODT 50
A.3 Versão do documento gerado para DOCX 51
A.4 Versão do documento gerado para ODT 52

Lista de Tabelas

2.1 Conflitos na interpretação dos componentes de documentos(Glushko e McGrath , 2008). 9
2.2 Artefatos produzidos em cada fase da engenharia de documentos(Glushko e McGrath , 2008). 13

3.1 Pontuação para as implementações de *ODF*(Shah e Kesan, 2009). 20
3.2 Pontuação para as implementações de *OOXML*(Shah e Kesan, 2009). 20

5.1 Conceitos da Ontologia de Conteúdo. 43
5.2 Propriedades da Ontologia de Conteúdo. 43
5.3 Conceitos da Ontologia de Formato. 44
5.4 Propriedades da Ontologia de Formato. 45

Capítulo 1

Introdução

Neste capítulo são descritos o contexto do trabalho, sua motivação, seus objetivos, sua justificativa, a metodologia e finalmente as contribuições esperadas. No final do capítulo é apresentada a organização deste trabalho.

1.1 Motivação

Os governos estão investindo no desenvolvimento de políticas, processos, padrões em Tecnologia de Informação e Comunicação (TIC), montando estruturas dedicadas para obter a interoperabilidade.

O desafio da interoperabilidade além de um problema técnico é um problema social e institucional, porque, principalmente no caso da preservação digital, se depende de instituições que passam por mudanças de direção, missão, administração e fontes de financiamento. Muitos materiais publicados digitalmente são produto de serviços de informação disponibilizados por organizações que adotam alguma infra-estrutura tecnológica. Essas instituições levam em consideração aspectos legais e culturais que afetam a oferta desses serviços orientados a atender as necessidades de determinados usuários. Mas muitas vezes essa preocupação não é a mesma que têm os produtores das novas tecnologias(Arellano, 2004).

Ante isso, o governo brasileiro estabeleceu um conjunto de padrões de interoperabilidade chamado e-PING(Padrões de Interoperabilidade de Governo Eletrônico)[1] para determinar um conjunto mínimo de premissas, políticas e especificações técnicas que regulamentam a utilização da TIC no governo federal, estabelecendo as condições de interação com os demais poderes e esferas de governo e com a sociedade em geral. A respeito da forma de armazenamento da informação, E-PING adota o formato de documento *Open Document Format*(ODF) para transmitir informação governamental entre os setores público e privado mantendo a privacidade e segurança. Os documentos digitais são considerados, atualmente, registros oficiais e são gerenciados segundo leis e padrões que compreendem todo o ciclo de vida desses materiais.

Mas, devido a necessidade de garantir a longevidade dos documentos digitais. Eles requerem superar a obsolescência rápida produzida pelas mudanças tecnológicas. A preservação digital requer procedimentos específicos e técnicas apropriadas para cada tipo de formato e mídia. As pesquisas propõem o uso de técnicas de emulação, migração e preservação da tecnologia, assim como da preservação de metadados e a criação de repositórios que permitam o acesso e a recuperação dos dados(Arellano, 2004).

Então, escolher um formato específico não garante a preservação da integridade e autenticidade dos documentos digitais. Para garantir o acesso a esses documentos a longo prazo é preciso superar as barreiras técnicas associadas aos formatos de documentos. O conteúdo, a estrutura e o contexto dos documentos devem ser associados às funcionalidades do software que preserva suas conexões executáveis ou representações de suas relações, permitindo a sua reconstrução. O propósito deste

[1] http://www.governoeletronico.gov.br/

trabalho é facilitar a distribuição de documentos para superar o problema do formato com os quais foram criados. O objetivo é habilitar a interoperabilidade de documentos e atingir a portabilidade simples de documentos através do reuso de conteúdo e formatos em diferentes formas plausíveis. Neste trabalho se pretende usar ontologias como solução ao problema da falta de interoperabilidade nas implementações de formatos de documentos. A proposta centra-se sobre os padrões *Office Open XML* e *Open Document Format*.

1.2 Objetivos

O objetivo principal do trabalho é:

- Propor um modelo conceitual que possibilite a interoperabilidade e a portabilidade de documentos sobre um determinado domínio.

Os objetivos específicos baseados no objetivo principal são:

- Desenvolvimento de uma ontologia de formato e conteúdo, e seu teste em casos de uso específicos.
- Desenvolvimento de um componente de software que possibilite a tradução da ontologia em formatos específicos.
- Modelagem de documentos a partir do modelo proposto sobre um determinado domínio.

1.3 Trabalhos Correlatos

O modelo Digital Rosetta Stone(DRS) desenvolvido por Heminger e Robertson (1998) descreve três processos necessários para manter o acesso a longo prazo dos documentos digitais: (1) preservação do conhecimento, (2) recuperação dos dados e (3) reconstrução dos documentos. Garantindo o acesso aos repositórios de informação armazenados digitalmente. Os processos e a manutenção dos metadados sugeridos pelo DRS servem para catalogar os diferentes aspectos das tecnologias digitais. A reconstrução dos documentos é o processo mais interessante. Reconstruindo documentos digitais a partir dos arquivos originais, baseada na informação do formato do arquivo reunida durante o processo de preservação do conhecimento. O informação do formato descrevem normas rigidamente definidas para armazenar a informação. Por exemplo, para mídia a ISO 9660 é a norma que especifica como os dados são armazenados em um CD-ROM. Definindo estruturas do volume, arquivo e todos os atributos associados a um CD-ROM. Esse tipo de dado deve ser reunido para cada tipo de mídia e ser incluído no arquivo de meta-conhecimento.

Outro trabalho é o arcabouço CCTS[2](Core Component Technical Specification). CCTS é um arcabouço conceitual europeu que serve para modelar documentos usando uma sintaxes neutral independente da tecnologia. O arcabouço permite manipular configurações diferentes de documentos estabelecidos por legislações distintas. O modelo conceitual do arcabouço é transferido para esquemas XML que servem de base para serviços colaborativos web do governo eletrônico(Vogel *et al.*, 2008).

Schroeder e Mello (2009) desenharam documentos em XML a partir de esquemas conceituais e informações sobre domínios específicos. Eles apresentaram uma abordagem que considera a conversão de dados e consultas para aplicações XML a partir de um esquema conceitual. O trabalho avalia a abordagem através de um estudo de caso realizado em um banco de dados XML nativo. Seus resultados experimentais demonstram que os esquemas XML gerados pela metodologia proposta contribui para um melhor desempenho de consulta que as abordagens relacionadas.

Outro trabalho importante é a biblioteca Universal Business Language (UBL). UBL é uma biblioteca de documentos de negócios electrónicos em XML, tais como ordens de compra e notas

[2]http://www.unece.org/cefact/index.html

fiscais. Foi desenvolvida pela OASIS. UBL foi desenhada para ser uma sintaxe universalmente compreendida e reconhecida de documentos comerciais, está baseada em padrões, como a ISO 15000 (ebXML). Os esquemas UBL são modulares, reutilizáveis e extensíveis. O propósito de UBL foi eliminar a re-digitação de documentos e fornecer um ponto de entrada para o comércio eletrônico. UBL está baseada em um modelo conceitual de componentes de informação conhecido como BIEs(Business Information Entities). Estes componentes são montados em modelos de documentos específicos. E um documento é um conjunto de componentes de informação que são trocados como parte de uma transação comercial, por exemplo, uma ordem de compra. Essa abordagem facilita a criação de tipos de documentos baseados em UBL além dos especificados da versão(Bosak *et al.*, 2011).

Finalmente, Eckert, Ziesing, e Ishionwu (2009) analisa como os formatos *OOXML* e *ODF* especificam as características mais importantes dos documentos e como essas características podem ser traduzidas entre os dois formatos. Eckert, Ziesing, e Ishionwu (2009) apresenta casos de estudo que resumem as características de traduzibilidade compatíveis e não compatíveis entre os formatos *OOXML* e *ODF*. Os casos de uso foram desenvolvidos para documentos de texto, apresentações e planilhas eletrônicas. Esse trabalho concentra-se mais na definição de orientações para a tradução da estrutura da apresentação do documento do que sobre a preservação da estética do documento. Os autores concluíram que a separação da apresentação do documento do seu conteúdo corporativo oferece mais facilidade na manipulação de documentos. Assim, editar os componentes de apresentação e dados do conteúdo corporativo de forma independente confere flexibilidade considerável na criação e edição de documentos.

1.4 Organização do Trabalho

No Capítulo 2 explicamos os problemas encontrados na preservação dos documentos digitais e os conceitos fundamentais da engenharia de documentos. No Capítulo 3 abordamos os principais formatos de padrões abertos de documentos digitais. No Capítulo 4 apresentamos os principais conceitos sobre ontologias. No Capítulo 5 apresentamos nossa solução para interoperabilidade e portabilidade de documentos, baseada em ontologias de formato e de conteúdo. Finalmente, no Capítulo 6 apresentamos uma discussão de nossos resultados, conclusões e propostas de trabalhos futuros.

Capítulo 2

Engenharia de Documentos

Este capítulo tem como objetivo introduzir aos conceitos, características dos problemas relacionados ao compartilhamento de documentos digitais e os princípios básicos da engenharia de documentos. O capítulo está baseado principalmente no livro de Glushko e McGrath (2008).

2.1 Documentos

Os documentos são interfaces entre pessoas e processos de negócios. Eles abrangem uma gama de tipos para atender seu público-alvo dependendo do contexto de uso. Um documento é um peça escrita, impressa ou eletrônica que fornece informação e serve como prova e registro oficial[1]. A Câmara Técnica de Documentos Eletrônicos (Rocha *et al.*, 2004) define documento como a informação registrada, independente da forma ou do suporte, produzida ou recebida no decorrer das atividades de uma instituição ou pessoa, dotada de organicidade, que possui elementos constitutivos suficientes para servir de prova dessas atividades.

Este trabalho está centrado nos documentos digitais. Outro termo usado nesse contexto é documento eletrônico, embora do ponto de vista tecnológico existam diferenças. Um documento eletrônico é acessível e interpretável por meio de um equipamento eletrônico (aparelho de videocassete, filmadora, computador, música em fita cassete), podendo ser registrado e codificado em forma analógica ou digital, e um documento digital é um documento eletrônico caracterizado pela codificação em dígitos binários acesso por meio de sistema computacional. Assim, todo documento digital é eletrônico, mas nem todo documento eletrônico é digital (Rocha *et al.*, 2004).

2.2 Preservação dos Documentos Digitais

Atualmente, fazer negócios intercambiando documentos é natural e intuitivo. No entanto, temos visto que a codificação do documento digital está profundamente ligada ao programa que o criou. Por isso, após um tempo, pode se tornar extremamente difícil de lê-lo sem perda significativa da informação. A vida útil de uma suíte de escritório(*Microsoft Office*, *Open Office.org*, *BrOffice* etc.) não é sempre assegurada. Os documentos gerados hoje devem poder ser recuperados no futuro, independentemente do software que os criou. Os arquivos de documentos, por tanto, não deveriam ser armazenados em formato binário proprietário.

Garantir a possibilidade de recuperar estes documentos significa que se deve preservar suas formatações originais. Para isso, é essencial que os formatos que descrevem como estes documentos estão armazenados eletrónicamente sejam resistentes a mudanças nas tecnologias de *software* e ambientes operacionais. Preservar o investimento financeiro e intelectual desses documentos, existentes e novos, tornou-se uma prioridade urgente(Ngo, 2008).

As organizações precisam disponibilizar documentos independentemente do software com que foram criados, i. e., habilitar o intercâmbio coerente de documentos baseado na adoção de regras e

[1] http://oxforddictionaries.com/definition/document

padrões de comunicação entre sistemas heterogêneos. Para compartilhar informações entre diversos órgãos é preciso superar as incompatibilidades entre os formatos de documentos. Um documento não deveria depender de nenhum *software* específico(Chede, 2009).

No intercâmbio de documentos é importante preservar a autenticidade de documento. Autenticidade é a habilidade de identificar elementos de forma a poder estabelecer se um documento é autêntico. Autêntico significa, em conforme com a verdade e, portanto, digno de fé e confiança. Um documento é considerado autentico se for possível provar que ele contém um conjunto de propriedades, consideradas significativas, que foram corretamente preservadas ao longo do tempo. Para obter garantia de autenticidade é fundamental documentar a origem do documento, contextualizar sua existência, descrever sua história custodial e atestar que sua integridade não foi comprometida(Ferreira, 2006).

2.3 Interoperabilidade e Portabilidade de Documentos

Interoperabilidade(e-Ping, 2010) é a habilidade de transferir e utilizar informações de maneira uniforme e eficiente entre várias organizações e sistemas de informação.

A interoperabilidade e portabilidade de documentos são conceitos pouco diferenciados. A interoperabilidade de documentos é a habilidade, em aplicações de negócios, de extrair informações contidas em vários tipos de documentos e transforma-las em estruturas padronizadas, na forma de arquivos, representados em *XML* que podem ser intercambiados entre vários sistemas e posteriormente processados(Schmidt *et al.*, 2006). A interoperabilidade de documentos procura interpretar corretamente a informação contida no documento, seu objetivo é trocar as informações dos documentos.

Por outro lado, a portabilidade de documentos é o intercambio de documentos com todas as informações que eles contêm, principalmente suas configurações de formato, toda a informação estilística e dados dos gráficos. A portabilidade de documentos considera a questão da fidelidade visual, incluindo requisitos de aparência visual, elementos estilísticos e outras questões(Schmidt *et al.* , 2006). Essa informação pode ser agrupada em quatro componentes: conteúdo, estrutura lógica, design e apresentação (Barron, 1996).

Ao contrário da portabilidade de documentos, a interoperabilidade de documento foca exclusivamente na troca de dados corporativos contidos nos documentos e não faz exigências sobre requisitos de aparência, elementos estilísticos, formato ou questões semelhantes(Schmidt *et al.*, 2006). Com a interoperabilidade de documentos baseada principalmente na tecnologia *XML*, as aplicações podem comunicar-se diretamente com serviços, por exemplo em uma arquitetura orientada a serviços. Os processos de negócio podem ser iniciados a partir de aplicações de documentos e ser integrados nos processos relevantes do sistema orientado a serviços.

Para a portabilidade de documentos é importante a manter fidelidade dos formatos dos documentos. A fidelidade do formato é a medida de quão precisa é uma cópia à sua fonte original, apesar de ser editado em múltiplas aplicações, é a capacidade de manter o formato do documento e seu sentido associado(Ditch, 2007).

2.4 A Engenharia de Documentos

A engenharia de documentos é uma especialidade da ciência da computação que estuda os sistemas de documentos e foca nos princípios, ferramentas e processos que melhoram a nossa capacidade de criar, tratar e manter documentos(ACM, 2001). A engenharia de documentos analisa e desenha métodos que produzem especificações e regras precisas para a informação que os processos de negócio requerem i.e., desenvolve modelos e padrões de intercâmbio de informação que enfatizam os requisitos dos documentos, centrando-se na separação entre conteúdo e apresentação das informações contidas nos documentos.

Como os documentos digitais são ubíquos, a engenharia de documentos enfatiza que um documento deve ser definido numa tecnologia neutra como uma coleção auto-suficiente de informações

construída com objetivos claros e bem definidos.

2.4.1 Modelos, Padrões e Reuso

Um modelo conceitual é uma descrição simplificada de um assunto. O modelo concentra-se no significado, abstraindo a complexidade e removendo características de implementação. A engenharia de documentos é composta por um conjunto de técnicas de desenho e análise que produzem modelos para intercâmbio de documentos. Os modelos definidos permitem automatizar os processos de intercâmbio de documentos. Podemos descrever a semântica dos documentos usando modelos dos conceitos que eles contêm. Os modelos conceituais são independentes de uma implementação física pois eles não estão vinculados a qualquer tecnologia em particular. Assim, se as tecnologias mudam, o modelo de implementação ideal também vai mudar, embora os modelos conceituais subjacentes permaneçam estáveis.

Usualmente as aplicações de negócios demandam suporte a diferentes interfaces de formato de saída do documento, o que implica em mapeamentos muitos-para-muitos entre as interfaces de entrada e de saída para cada aplicação. Esse problema pode ser evitado mapeando todas as interfaces em um modelo conceitual comum. As interfaces do usuário e aplicações baseadas num modelo conceitual comum garantem que os documentos processados sejam interoperáveis.

A engenharia de documentos necessita ser precisa quando define a semântica da informação produzida e consumida por aplicações de negócios. Ela desenvolve modelo de referência, que são modelos que enfatizam os requisitos dos documentos e padrões de intercâmbio de informação. Os modelos de referência são usados para descrever processos que são inerentemente mais abstratos do que os documentos. Um modelo de referência comum ajuda a alinhar diferentes modelos conceituais.

A engenharia de documentos traz a necessidade de padronização da sintaxe, estrutura e semântica dos documentos de negócios e seus componentes reutilizáveis. Os padrões são modelos que são suficientemente gerais, adaptáveis e que podem ser reutilizados. Eles definem um conjunto de especificações cobrindo todos os elementos, processos, formatos ou procedimentos sob sua jurisdição, sendo sobretudo recomendado o uso de padrões abertos(Chede, 2009). Um padrão aberto é uma especificação publicamente validada e livremente implantada. O uso de padrões abertos economiza esforço e produz projetos mais consistentes, compatíveis e com maior probabilidade de ser bem sucedidos.

2.4.2 O Desafio da Interoperabilidade

A melhor maneira de construir a interoperabilidade é garantindo que os participantes compartilham o mesmo modelo conceitual. Interoperabilidade requer que os participantes possam intercambiar informações e usar a informação trocada. Sendo assim, a informação trocada deve ser conceitualmente equivalente. O desafio da interoperabilidade de documentos é entender o significado do conteúdo dos documentos. A figura 2.1 mostra como documentos diferentes sobre transações comerciais posicionam informações que fluem entre eles. As informações entre os documentos não podem interoperar enquanto essas diferenças não forem resolvidas. Quando organizações intercambiam documentos, elas devem concordar no conteúdo semântico, mas não necessariamente precisam concordar na tecnologia que usam. A interoperabilidade exige que todos os membros de uma comunidade usem padrões comuns de sintaxe e semântica de documentos.

Determinar a equivalência de documentos torna-se difícil devido às variações nas estratégias, plataformas tecnológicas, aplicações legadas, processos de negócios e terminologia. Para estabelecer a equivalência semântica, é preciso comparar as representações conceituais e determinar se os diferentes modelos físicos refletem os mesmos requerimentos conceituais. Infelizmente a maioria dos documentos não apresenta uma representação conceitual associada que define inequivocamente a semântica do documento. Sendo assim, é preciso aplicar técnicas da engenharia documentos para determinar se a informação requerida pode ser extraída e transformada a partir do documento recebido. Isso não é tão fácil como parece pois, ao comparar elementos dos esquemas dos documentos, nem sempre os mesmos nomes implicam nos mesmos significados nem vice-versa.

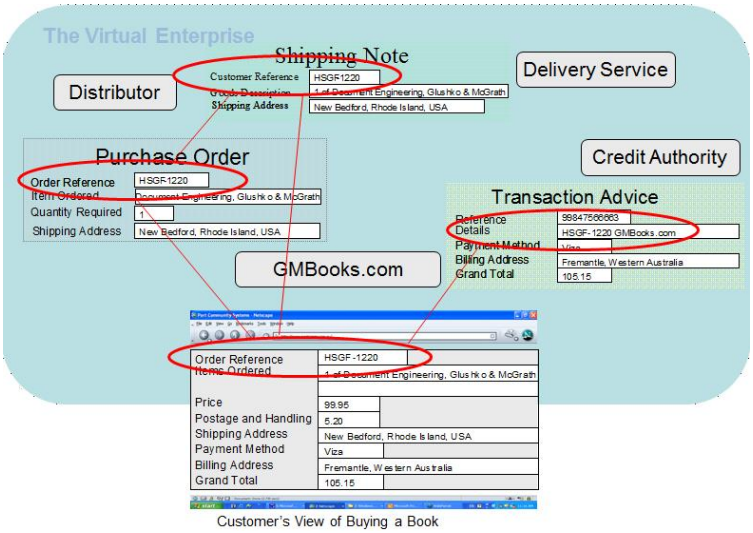


Figura 2.1: *O desafio da interoperabilidade.*

A Tabela 2.1 exemplifica tipos de conflitos que podem surgir na troca de informações. O exemplo mostra as diferentes formas como podemos comunicar o valor de 100 dólares norte-americanos.

2.5 Fases da Engenharia de Documentos

A engenharia de documentos não é uma metodologia, mas sim uma abordagem coerente e pragmática para modelagem de documentos e serviços, fornecendo soluções práticas e eficazes. Não há uma única forma correta de criar estes modelos, e diferentes metodologias têm sido propostas para essa finalidade, as quais compartilham os objetivos comuns de definir regras que reflitam exigências do contexto de uso e comunicar esse entendimento do uso dos documentos em um ou mais artefatos de modelagem.

A abordagem da engenharia de documentos enfatiza a modelagem cuidadosa e a reutilização de modelos, sempre que for possível. Essa abordagem incorpora o princípio de baixo acoplamento no intercâmbio de documentos. A figura 2.2 mostra em resumo a relação dos níveis de perspectivas de análise e fases da abordagem da engenharia de documentos.

Análise do Contexto de Uso

Nenhum documento é uma ilha, sozinho em si mesmo. Uma organização típica pode usar dezenas de diferentes tipos de documentos com complexas interdependências organizacionais e de processos. Por exemplo, os relatórios sobre uma ordem de compra podem ser conectados com os livros contábeis, relatórios de vendas, catálogos, calendários de produção, anúncios, manuais e outros documentos. Não é difícil ver como o escopo se expande rapidamente, então devemos identificar explicitamente o que está dentro e fora do escopo.

Usamos o contexto de uso para organizar e analisar os requisitos e regras. Este análise envolve a identificação de objetivos estratégicos de negócios em termos de requisitos do modelo de negócio e regras que devem ser satisfeitas. Os requisitos formalizam as definições do contexto no qual os documentos serão usados, e também as restrições sobre possíveis soluções que devem ser satisfeitas para que a solução seja aceitável. Estes requisitos são expressos como regras sobre o conteúdo, estrutura, apresentação dos documentos e seus componentes.

Nesta fase, a partir do contexto global, organizamos e analisamos os requisitos e regras que irão conduzir o desenho de processos e seus documentos. Para compreender o alcance dos processos em nosso modelo, precisamos reconhecer as relações de dependência entre processos de negócio. Quanto mais dependências legítimas encontremos mais padrões podemos encontrar, mas ao custo de maior

Diferenças no conteúdo
• <valor>USD 100</valor> • <valor>Cem dólares americanos US</valor> • <valor>$US100</valor>
Diferenças na codificação
• <valor>USD 100</valor> • USD,100 • USD:100
Diferenças na estrutura
• <valor>USD 100</valor> • <moeda>USD</moeda><valor>100</valor> • <valor>100<moeda>USD</moeda></valor>
Diferenças na semântica
• <valor>USD 100</valor> • <valorneto>USD90</valorneto><imposto>USD10</imposto> • <preço>USD 100</preço>

Tabela 2.1: *Conflitos na interpretação dos componentes de documentos(Glushko e McGrath, 2008).*

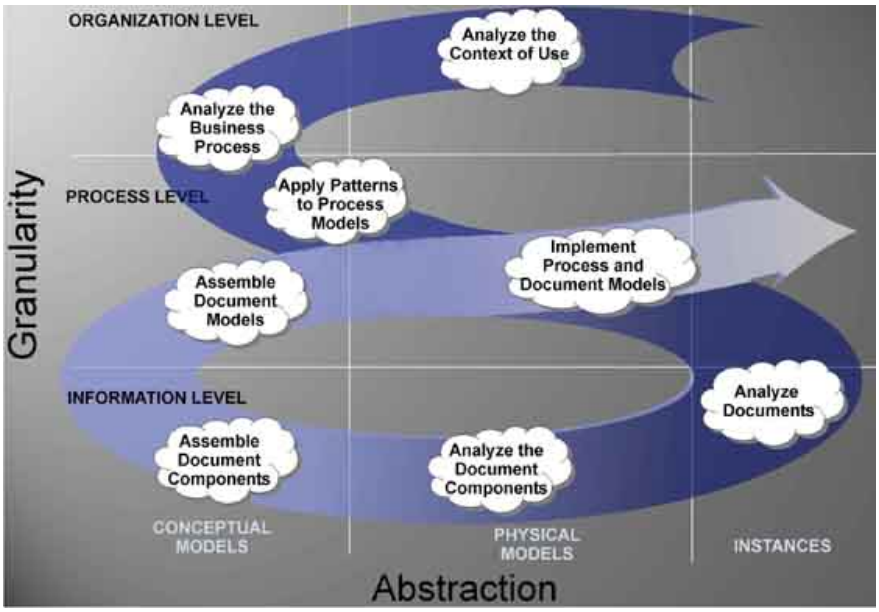


Figura 2.2: *Níveis de perspectivas de análise nas fases da engenharia de documentos(Glushko e McGrath, 2008).*

complexidade. O objetivo principal do escopo do contexto é incluir suficientes dependências para assegurar que seja realista, sem deixar de compreender as interações entre eles.

Análise dos Processos de Negócios

Um processo de negócio é uma cadeia de atividades relacionados e eventos com entradas especificas que agregue valor e produzam um produto ou serviço específico que posse ser a entrada para outro processo.

Alguns processos de negócios são conduzidos inteiramente dentro de uma organização. Eles são chamados de processos privados. Em contraste, os processos de negócios externos ou públicos são realizadas entre duas ou mais organizações. A engenharia de documentos trabalha com ambos tipos de processos, mas é especialmente útil para o último, onde diferentes tecnologias de implementação devam usar arquiteturas de baixo acoplamento para intercâmbio de documentos.

Desenho dos Processos de Negócios

O foco da engenharia de documentos é a criação de processos de negócios usando padrões. A grande preocupação nas organizações é equilibrar as vantagens da criação de documentos de forma otimizada com a necessidade de seus processos serem compreendidos por outras organizações. Isso induz as organizações a reutilizar modelos existentes de processos, documentos e desenhar novos modelos que incentivem a sua reutilização. A tarefa de desenhar novos processos de negócios muitas vezes envolve mais adoção e adaptação de padrões que a invenção de algo novo.

Bibliotecas de processos de negócios constituem repertórios úteis de padrões, precisamente porque não ditam tecnologias específicas para sua implementação. Mesmo se um padrão for apenas um ajuste parcial a um contexto particular, ele ainda pode fornecer informações úteis quanto a identificação de exigências. Por exemplo, alguns aspectos de um padrão de compras se repetem quando uma pessoa está comprando aço, clipes de papel ou cursos universitários.

Análise de Documentos

Nesta fase cria-se o inventário de documentos. Um inventário de documentos é uma coleção de documentos e artefatos relacionados. Ele apresenta informação dos componentes e regras de negócios. O inventário lista as fontes que identificam metadados, propósitos, origens e outros atributos que apoiam na seleção de um subconjunto de dados. Com um inventário mais amplo de documentos, qualquer análise se torna mais eficaz. Assim, quanto mais instâncias de documentos consideremos em nossa analise, maior é precisão reconhecer as regras do negócio que expressem os requisitos, uma vez que novas instâncias expõem nova regras.

O objetivo final é criar um modelo conceitual que inclua toda a informação dos requisitos do contexto de uso apresentados nos documentos.

Análise dos Componentes do Documento

Nesta fase é construído o modelo de componentes. Esta fase é um trabalho de consultoria para ajudar os criadores de documentos e usuários a chegar a um entendimento consensual sobre os componentes de conteúdo em seus domínios.

Os componente descrevem elementos que encapsulam uma série de funcionalidades. Esta fase inicia com o agrupamento dos componentes, identificando os componentes contidos em cada documento ou fonte de informação para elaborar uma listagem dos componentes de informação do contexto e componentes da apresentação da informação. O modelo de componentes descreve a semântica de um domínio. Parte dessa racionalização envolve a escolha de nomes significativos dos componentes, o que é inerentemente uma tarefa contenciosa e iterativa. Por isso, os nomes não são definitivos nesta fase.

A listagem dos componentes começa com a desintegração das estruturas, separando o significado subjacente. Ao analisar esses componentes, procuramos separar componentes de apresentação dos componentes de conteúdo. Desagregamos as estruturas de apresentação em componentes atômicos de informação, até a granularidade necessária para identificar seu significado e encontrar componentes reutilizáveis. Separar a semântica da apresentação leva a reconhecer as convenções estilísticas

e componentes de apresentação aplicados na informação em seus vários formatos. Identificar componentes e estruturas de apresentação nós ajuda a determinar quais características estilísticas são necessárias para entender a informação contida no documento.

Finalmente, a fidelidade e integridade dos componentes de informação são mantidas em base nas regras dos modelos.

Montagem dos Componentes do Documento

A criação do modelo de componentes (ou vista do domínio) nos dá uma apreciação mais profunda das regras e exigências do contexto de uso. Esse análise pode conduzir à melhoria do desenho.

O resultado desta fase é um conjunto de componentes semanticamente únicos. A maneira mais rigorosa para explicar o significado dos componentes é usando uma ontologia formal. As ontologias definem o significado de termos usando uma linguagem formal, de maneira que todas as relações entre os termos são expressos de forma consiste e precisa. As ontologias são explicadas mais adiante nesta dissertação. Por outro lado, os diagramas *UML* são uma alternativa conveniente na formalização dos modelos dos componentes.

Nesta fase organizamos os componentes para construir estruturas mais complexas, ou seja, os documentos. O modelo de componentes apresenta uma visão geral conceitual de todos os componentes de informação necessários para um determinado contexto de uso. O modelo de componentes resultante dessa análise descreve uma rede de relações que criam estruturas de documentos potenciais no contexto de uso.

Montagem dos Modelos de Documento

Um modelo de componentes é útil para o armazenamento de informações, pois é versátil e reutilizável para muitos propósitos, mas não é recomendável para intercambiar informações. Para isso, é necessária uma estrutura hierárquica.

O modelo de documento é um conjunto hierárquico de estruturas aninhadas de componentes. Um modelo hierárquico de um documento é criado selecionando um componente raiz como ponto de entrada. Se for necessário, são adicionadas funções e associações seguindo as regras de negócio e requisitos do contexto de uso.

Ao criar um modelo de documento definimos caminhos de associações na rede do modelo de componentes criada na fase anterior.

Implementando os Modelos do Documento

Esta fase é momento de colocar em prática o modelo conceitual. Qualquer aplicação de software que implemente o modelo conceptual representará as regras e exigências do contexto de uso. Ao fazer isso, as regras sobre informações e processos capturadas nos modelos se tornem explícitas. O software determina seu comportamento a partir do modelo.

A melhor maneira de implementar os modelos físicos a partir dos modelos conceituais é codificá-los em uma linguagem baseada em *XML*.

A tabela 2.2 mostra um resumo dos artefatos criados em cada fase da engenharia de documentos.

2.6 Sumário

Principais conceitos do capítulo:

- Documento é a informação registrada, independente da forma ou do suporte, produzida ou recebida no decorrer das atividades de uma instituição ou pessoa.
- Preservar o investimento financeiro e intelectual desses documentos, existentes e novos, tornou-se uma prioridade urgente.

- As organizações precisam recuperar e disponibilizar documentos independentemente do software com que foram criados.
- No intercambio de documentos é importante preservar a autenticidade dos documentos.
- A interoperabilidade de documentos é a troca das informações dos documentos através de estruturas padronizadas. Por exemplo *XML*.
- A portabilidade de documentos é a troca de documentos mantendo principalmente a fidelidade do formato.
- A engenharia de documentos analisa e desenha métodos que produzem especificações e regras precisas para a informação que os processos de negócio requerem.
- A engenharia de documentos desenvolve modelos e padrões de intercâmbio de informação que enfatizam os requisitos dos documentos, centrando-se na separação entre conteúdo e apresentação das informações contidas nos documentos.
- Um documento deve ser definido usando uma tecnologia neutra, construída com objetivos claros e bem definidos.
- O desafio da interoperabilidade de documentos é entender o significado do conteúdo dos documentos.
- A engenharia de documentos é composta por um conjunto de técnicas de desenho e análise que produzem modelos para intercâmbio de documentos.
- Usamos o contexto de uso para organizar e analisar os requisitos e regras.
- A análise de contexto de uso expressa as regras do conteúdo, estrutura, e apresentação dos documentos e seus componentes.
- A análise dos processos de negócios é especialmente útil nos processos públicos, onde é possível que as organizações usem tecnologias diferentes na implementação dos documentos.
- O desenho de processos de documentos induz as organizações a reutilizar modelos existentes de processos e documentos.
- A análise de documentos cria um inventário de documentos que identificam metadados, propósitos, origens etc., que são atributos representativos do modelo conceitual do contexto.
- A análise de componentes cria um modelo de componentes que descreve a semântica de um domínio, identificando componentes de apresentação e conteúdo.
- A montagem de componentes descreve um modelo de componentes como uma rede. As relações desse modelo expressam as regras e exigências do contexto de uso.
- A implementação do modelo de documento cria um aplicativo cujo comportamento está determinado explicitamente pelo modelo conceitual.
- Os modelos físicos implementam os modelos conceituais usando uma linguagem baseada em *XML*.

No próximo capítulo, discutimos as especificações dos principais formatos de documentos digitais.

Fase	**Artefato**
• Análise do contexto do uso	• Diagrama de casos de Uso *UML*
• Análise/Desenho do processo de negócios – Colaborações de negócios – Transações de negócios – Aplicação de padrões aos processos de negócios	• Vista das planilhas do domínio de negócio • Planilha de negócios das áreas de processos • Vista das planilhas das transações de negócio • Diagrama de casos de uso *UML* • Listagem de documentos • Diagramas de sequencia *UML*
• Análise dos componentes do documento	• Tabela dos componentes do conteúdo
• Montagem dos componentes do documentos	• Diagrama de classes *UML*
• Montagem dos modelos de documento	• Diagrama de classes *UML*
• Implementação de aplicativos baseados nos modelos	• Esquemas *XML* para modelos de documentos • Instancias *XML* para modelos de processos

Tabela 2.2: *Artefatos produzidos em cada fase da engenharia de documentos(Glushko e McGrath, 2008).*

Capítulo 3

Padrões de Formato de Documentos Digitais

Este capítulo apresenta os principais padrões abertos de formatos de documentos digitais, *OOXML* e *ODF* e introduz a problemática na portabilidade de documentos nestes formatos.

3.1 Introdução

Os documentos gerados hoje devem poder ser recuperados no futuro. Garantir a possibilidade de recuperar estes documentos significa que deve-se preservar suas formatações originais. Para isso, é essencial que os formatos destes documentos sejam resistentes a mudanças nas tecnologias de software e ambientes operacionais.

Para garantir a interoperabilidade, os padrões devem ser abertos e não controlados por nenhuma empresa, dessa forma não privilegiando um produto específico em detrimento de outros. Os padrões abertos são publicamente disponíveis e não são controlados por nenhum governo ou corporação, tornando possível que quaisquer empresas, cidadãos e países se conectem e intercambiem informações com autonomia(Chede, 2008).

Para a interoperabilidade de documentos existem atualmente três formatos de arquivo com popularidade difundida: *PDF*, *Portable Document Format* (PDF) da Adobe, *Office Open XML* da *Microsoft Corporation* e *Open Document Format* da OASIS(*Organization for the Advancement of Structured Information Standards*). Todos dão suporte a processamento de dados baseados em *XML* e são padrões abertos.

Os dois principais padrões abertos de formatos *Office Open XML* e *Open Document Format* são incompatíveis e rivais no mercado, gerando a guerra dos formatos abertos, provocando discussões técnicas sobre as vantagens e desvantagens de cada um deles. Por outro lado a *Adobe* alcançou com seu formato PDF aberto ISO 19005 evitar essa guerra, focando só na visualização de documentos para conservação a longo prazo (Ruesta, 2008). Nossa proposta centra-se sobre os padrões *Office Open XML* e *Open Document Format*.

3.2 Office Open XML

Office Open XML (OOXML ou OpenXML) foi desenvolvido pela *Microsoft Corporation* em 2006. A especificação *OOXML* existe em diversas versões. No 2008, uma versão revisada da ECMA-376 (2006) tornou-se um padrão ISO aberto (ISO/IEC 29500:2008). *OOXML* está estruturado em quatro partes independentes, a fim de atender às necessidades de públicos diferentes[1].

OOXML (aproximadamente 6.000 páginas) foi projetado desde o início para ser capaz de representar o corpus preexistente de documentos de processamento de texto, apresentações e planilhas que são codificados pela *Microsoft Corporation*. O padrão especifica seis níveis de conformidade

[1] http://www.iso.org/iso/

do documento ou aplicativo, estrito (estrito) e de transição (transição) para cada idioma (WordprocessingML e SpreadsheetML PresentationML). Especifica, também, as descrições de aplicações, tais como base e cheia (completa).

A Figura 3.1 apresenta os componentes principais da especificação *Open Office XML*:

- Parte 1. Fundamentos e referência do linguagem de marcação (5560 páginas)
  - Contém definições de conformidade
  - Contém material de referência para *WordprocessingML*, *SpreadsheetML*, *PresentationML*, *DrawingML*, outros ML compartilhados e esquemas *XML* personalizados, definindo cada elemento e atributo, incluindo a hierarquia de elementos (relações de parentesco).
  - Esquemas *XML* para as linguagens de marcação, declarados como XSD e (não incluído nos regulamentos), utilizando RELAX NG.
  - Define a unidade de armazenamento de dados como *XML* personalizados.
  - Esquemas *XML* estão incluídos no texto.
- Parte 2. *Open Packaging Conventions* (129 páginas)
  - Define o formato *Open Packaging Conventions* (modelo do pacote, empacotamento físico)
  - Define as propriedades essenciais, miniaturas e assinaturas digitais.
  - Os esquemas *XML* para OPC são declaradas como *XML Schema Definitions* (XSD) e (não incluído nos regulamentos), utilizando RELAX NG (ISO/IEC 19757-2).
  - Declara os esquemas *XML* para as OPC como XSD (*XML Schema Definitions*), em um anexo utilizando RELAX NG (ISO/IEC 19757-2).
- Parte 3. Compatibilidade e capacidade de Extensão das Marcações (40 páginas)
  - Descreve os recursos de extensão de documentos *OOXML*.
  - Específica os elementos e atributos pelos quais aplicativos com diferentes extensões podem interoperar.
  - Expressa regras de capacidade de extensão, utilizando NVDL[2] (*Namespace-based Validation Dispatching Language*) (ISO/IEC 19757-4)
- Parte 4. Características de migração transitórios (1464 páginas)
  - Contém material de referência de interesse de transição.
  - Contém material controverso como as configurações de compatibilidade e linguagem de marcação gráfica *VML*.
  - Contém uma lista de diferenças entre ECMA-376 e ISO/IEC 29500:2008.

Um arquivo em *Office Open XML* pode conter documentos codificados em linguagens de marcação especializadas, para diversas aplicações dentro da linha de produtos da *Microsoft Office*. As principais linguagens de marcação na especificação são:

- *WordprocessingML*: para processamento de texto;
- *SpreadsheetML*: para planilhas.
- *PresentationML*: para apresentações.
- *DrawingML*: usado para desenhar gráficos vetoriais, texto e arte.

[2] http://nvdl.org/

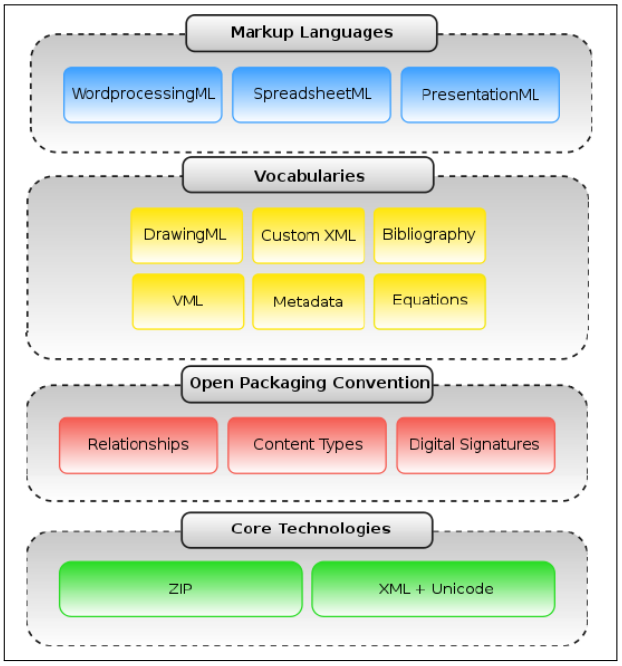


Figura 3.1: *Componentes principais do Office Open XML(Ngo, 2008).*

- *VML*: suportado como uma linguagem de desenho.

O formato *OOXML* usa um contêiner baseado em *Open Packaging Conventions*. Esse formato usa mecanismos herdados do formato de compressão *ZIP*. Além de marcação, o contêiner pode também integrar outros arquivos, como imagens, vídeos ou outros documentos. A Figura 3.2(a) mostra a organização dos arquivos do contêiner de formato *OOXML* para documentos com formato *docx* . Os arquivos principais são:

- *[Content_Types].xml*: Descreve os conteúdos gerais. Também contém estruturas de mapeamento para extensões de arquivos e substituições para URIs específicas.
- *DocProps/app.xml*: Armazena as estatísticas do documento.
- *DocProps/core.xml*: Contém as propriedades básicas de qualquer documento *Office Open XML*.
- *words/document.xml*: Contém o texto base do documento.
- *words/fontTable.xml*: Contém as configurações de fonte do documento.
- *words/styles.xml*: Contém uma lista de acentos disponíveis e detalhes estilísticos que podem ser usados no documento.
- *words/settings.xml*: Contém atributos quanto à ortografia, gramática, matemática, espaçamento.
- *words/webSettings.xml*: Contém uma configuração que especifica como o arquivo é salvo como uma página Web.

3.3 OpenDocument Format

O *OpenDocument Format* (ODF ou OD) foi desenvolvido originalmente pela *Sun Microsystems* em 2000, como formato baseado em *XML* para StarOffice e *OpenOffice.org*. O ODF foi projetado com o objetivo de ser uma especificação de formato de documentos independente de fornecedor

ou de *software*, tornando-o adequado para o armazenamento de documentos a longo prazo(Hiser , 2007). ODF foi publicado em 2005 como padrão OASIS [3] (*Organization for the Advancement of Structured Information Standards*). Em 2006 esse padrão foi aprovado pelas organizações ISO/IEC como ISO/IEC 26300:2006 *Open Document Format for Office Applications* v1.0 e, finalmente em 2012, a OASIS publicou a versão *ODF* v1.2.

Embora a especificação *ODF* seja complexa para os padrões normais (aproximadamente 700 páginas), a reutilização parcial de padrões abertos existentes reduz consideravelmente a complexidade da especificação. Alguns padrões reutilizados no *OpenDocument Format* são: *XML*, *ZIP*, *Dublin Core*, *SVG*, *XSL*, *SMIL*, *XLink*, *XForms*, *MathML* e *HTML*(Ngo, 2008).

O documento da norma *ODF* v1.2 está estruturado em três partes:

- Parte 1: define um esquema *XML* para aplicações de escritório e sua semântica. O esquema é adequado para documentos de escritório, incluindo documentos de texto, planilhas, apresentações, gráficos e documentos com gráficos, desenhos ou apresentações, mas não se restringe a esses tipos de documentos.
- Parte 2: define uma linguagem de fórmulas para ser usada em documentos OpenDocument.
- Parte 3: define um formato de pacote a ser usado nos documentos OpenDocument.

Um arquivo OpenDocument é um arquivo compactado *ZIP* que contém vários arquivos e diretórios. A Figura 3.2(b) mostra a estrutura mínima do contêiner *ODF*, que é a mesma para os documentos de texto, planilhas e apresentações. Os arquivos principais são:

- *context.xml*: Armazena o conteúdo de texto do documento (exceto os dados binários, como imagens).
- *styles.xml*: Armazena a maioria das informações de estilo.
- *meta.xml*: Contém metadados do documento, como por exemplo, o autor, a identificação da pessoa que modificou por último o documento, a data da última modificação etc.
- *settings.xml*: Contém informação sobre propriedades, tais como o fator de zoom ou a posição do cursor (essa informação não afeta a informação do documento).

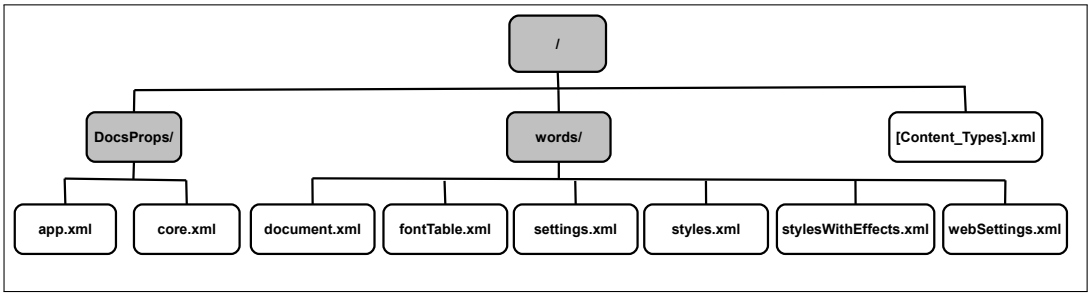
O conteúdo de um arquivo OOXML ou ODF pode ser renomeado para um arquivo *ZIP*. O conteúdo pode ser aberto com qualquer ferramenta *ZIP* e os arquivos reais *XML* podem ser visualizados em um navegador web ou em um editor de texto simples de *XML*. A Figura 3.3 mostra um exemplo de texto básico na especificação dos formatos *OOXML* e *ODF*. O exemplo mostra um extrato de texto do arquivo *word/document.xml* do contêiner *OOXML*, equivalente a um extrato do arquivo *context.xml* do contêiner *ODF*.

3.4 Portabilidade dos Formatos OOXML e ODF

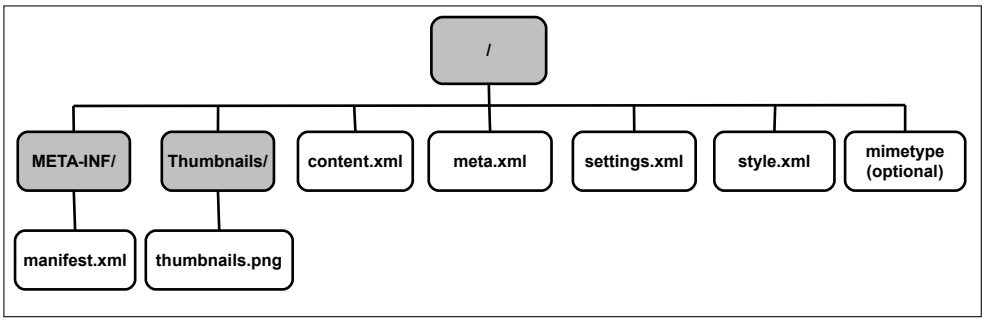
A apresentação de um documento é fundamentalmente regida pela implementação da suíte de escritório, em particular pelo mecanismo de renderização de cada suíte. Assim, usando exatamente o mesmo padrão para descrever um documento não garante que suítes de escritório diferentes produzirão apresentações idênticas.

Shah e Kesan (2009) fizeram um estudo sobre a portabilidade das implementações nas suítes de escritório que implementaram os formatos *Office Open XML* e *Open Document Format*. O objetivo do trabalho foi avaliar as implementações e estimular a busca do aperfeiçoamento da portabilidade e interoperabilidade dos documentos. O estudo também procurou destacar a importância da interoperabilidade usando padrões abertos. Shah e Kesan (2009) trabalharam com cinco tipos de

[3] http://www.oasis-open.org



(a) Estrutura do contêiner *OOXML* para documentos com formato *docx*.



(b) Estrutura do contêiner *ODF*

Figura 3.2: *Estrutura dos contêineres dos formatos OOXML e ODF.*

```
<w:p>
<w:r>
    <w:t>Este é um documento</w:t>
</w:r>
<w:r>
    <w:rPr>
        <w:b/>
    </w:rPr>
    <w:t>básico</w:t>
</w:r>
<w:r>
    <w:t>com alguma</w:t>
</w:r>
<w:r>
    <w:rPr>
        <w:b/>
    </w:rPr>
    <w:t>formatação</w:t>
</w:r>
<w:r>
    <w:t>e um</w:t>
</w:r>
<w:hyperlink r:id="rId5">
<w:r>
    <w:rPr>
        <w:rStyle w:val="Hyperlink"/>
    </w:rPr>
    <w:t>hiperlink</w:t>
</w:r>
</w:hyperlink>
</w:p>
```

Office Open XML

Este é um documento **básico** com alguma **formatação** e um hiperlink

Open Document Format

```
<text:p text:style-name="P1">
Este é um documento
<text:span text:style-name="T7">básico</text:span>
com alguma
<text:span text:style-name="T7">formatação</text:span>
e um
<text:a xlink:type="simple" xlink:href="https://www.google.com.br/">hiperlink</text:a>
</text:p>
```

Figura 3.3: *Exemplo de texto básico nas especificações de OOXML e ODF.*

testes diferentes. O primeiro teste foi focado em características usadas na formatação. O segundo teste foi na utilização de imagens. O terceiro teste foi focado no uso de tabelas. Os cabeçalhos e rodapés foram testados no quarto teste. O teste final continha uma tabela de conteúdos, notas de rodapé, comentários e rastreamento de alterações.

Os resultados do estudo se baseiam em pontuações de quão bem uma implementação lê e armazena um documento criado em outro software que implementou o mesmo formato. Cada documento de teste é aberto em cada aplicação e em seguida é armazenado. O documento armazenado é aberto

Implementação	Leitura	Leitura e escrita	Metadata
OpenOffice.org	100%	100%	100%
StarOffice	99%	100%	100%
Sun Plug-in for Word	94%	94%	93%
Clever Age/MS Plug-in for word	92%	89%	100%
WordPerfect	81%	N/A	N/A
Koffice	80%	82%	67%
Google Docs	77%	86%	20%
TextEdit	36%	34%	0%
AbiWord	32%	37%	0%

Tabela 3.1: *Pontuação para as implementações de ODF(Shah e Kesan, 2009).*

Implementação	Leitura	Leitura e escrita	Metadata
Office 2007	100%	100%	100%
Office 2003	100%	100%	100%
Office 2008 (Mac)	99%	100%	100%
OpenOffice.org	95%	97%	80%
Pages	96%	N/A	N/A
Wordperfect	77%	N/A	N/A
Thinkfree Office	68%	76%	53%
TextEdit	35%	38%	0%

Tabela 3.2: *Pontuação para as implementações de OOXML(Shah e Kesan, 2009).*

na implementação de referência que o criou. O documento resultante é então pontuado. A pontuação é a quantificação das alterações com relação ao conteúdo real. Assim, a pontuação está principalmente focada na capacidade de preservação dos metadados dos documentos. Os metadados são atributos de estilos, números de páginas, tabelas de conteúdos ou cabeçalhos, informações do documento (tempo, ou o número de palavras em documentos) e controle de alterações.

A Tabela3.1 mostra o resultado do trabalho para as implementações do padrão *ODF*. Os documentos de testes foram criados em OpenOffice 2.3. Os resultados indicam que não existem implementações independentes que ofereçam 100% de compatibilidade no formato *ODF*. Apesar de *StarOffice* usar o mesmo código fonte que *OpenOffice*, os resultados mostram que existem diferenças entre eles na leitura. Sobre os pontos perdidos, eles são atribuídos a que *StarOffice* não tem o número correto de páginas. No entanto, nos testes de leitura e escrita, os resultados mostram que esta questão desaparece quando o documento é reaberto em *OpenOffice*.

A tabela 3.2 ilustra os resultados para as implementações do formato *OOXML*. Os documentos de teste foram criados no *Microsoft Office 2007*. O formato *OOXML* teve resultados semelhantes aos resultados do *ODF*, sem 100% de compatibilidade com outras implementações além da *Microsoft Office* para Windows (2003 ou 2007). A versão da *Novell* de OpenOffice, com seu *plugin* tradutor para *OOXML*, fornece boa compatibilidade ($> 90\%$). *Apple´s Pages*, também proporciona uma boa compatibilidade, mas não é interoperável. *Pages* e *WordPerfect* só podem ler documentos *OOXML*. *ThinkFree Office* oferece pouca compatibilidade ($> 65\%$) no teste de leitura, mas melhora na interoperabilidade com o teste de escrita. *TextEdit* oferece um desempenho ruím para *OOXML*. O teste dos metadados também mostra que todas as outras implementações além da Microsoft perdem informação. *OpenOffice* perdeu um pouco de metadados, enquanto *ThinkFree* teve uma perda mais significativa e *TextEdit* não manteve nenhum metadado.

Microsoft Office 2007 mostrou 100% de compatibilidade com o padrão *OOXML*, e *OpenOffice.org* mostrou 100% de compatibilidade com *ODF*. O estudo ressalta que, a pesar de usar exata-

mente o mesmo formato padrão, não existe 100% de portabilidade dos documentos digitais. Suítes de escritório diferentes não garantem a 100% a conservação da informação do formato do documento. O estudo evidência a urgência da melhoria nas implementações dos padrões de formato de documentos *OOMXL* e *ODF*.

3.5 Sumário

Principais conceitos do capítulo:

- Na interoperabilidade, os padrões são abertos e não controlados por nenhuma empresa, não privilegiando um produto específico em detrimento de outro.
- Os padrões abertos são publicamente disponíveis e não são controlados por nenhum governo ou corporação, tornam possível que quaisquer empresas, cidadãos se conectem e troquem informações com autonomia.
- Os principais padrões abertos de formatos *Office Open XML* e *Open Document Format* são incompatíveis e rivais.
- As suítes de escritório embora implementando o mesmo formato não garantem que produzirão apresentações idênticas na leitura do mesmo documento. Como também não garantem a conservação da informação do formato na escrita do documento.

No próximo capítulo, explicamos os principais conceitos das ontologias. As ontologias são utilizadas para compartilhar o conhecimento e solucionar problemas de falta interoperabilidade.

Capítulo 4

Ontologias

Nesse capítulo introduziremos a discussão sobre as ontologias dentro do domínio utilizado nesse trabalho. Descreveremos as principais vantagens do uso de ontologia, suas áreas de aplicação, suas classificações, ferramentas e metodologias de desenvolvimento.

4.1 Introdução

O termo ontologia tem origem no grego *ontos*, ser, e *logos*, palavra. No sentido filosófico o objetivo é o estudo do ser como tal. Do ser como ser, ou dos princípios e das causas do ser e de seus atributos essenciais. O termo original vem da palavra *categoria* definida no livro de Aristóteles chamado Categorias, termo utilizado no sentido de classificação. Aristóteles usa as categorias para classificar qualquer entidade e introduz o termo *differentia* para propriedades que distinguem diferentes espécies do mesmo gênero. Ontologias são utilizadas em diversas áreas para organizar informação permitindo compartilhar conhecimento de forma inequívoca e inferir conhecimento oculto através da definição de componentes reutilizáveis de conhecimento. A ventagem fundamental das ontologias é melhorar a comunicação entre humanos e ou entre computadores, superando o problema de interoperabilidade.

4.2 Ontologias e Ciências da Computação

A definição mais frequente é de Gruber (1993): ontologia é uma especificação formal e explícita de uma conceitualização compartilhada. Conceitualização representa um modelo abstrato de algum fenômeno que identifica os conceitos relevantes para o mesmo. Explícita significa que os elementos e suas restrições estão claramente definidos; formal significa que a ontologia deve ser passível de processamento automático, e compartilhada reflete a noção de que a ontologia captura conhecimento consensual, aceito por um grupo de pessoas.

O consórcio *W3C* define uma ontologia como a definição dos termos utilizados na descrição e na representação de uma área do conhecimento. Guarino e Giaretta (1995) publicaram as seguintes definições para o conceito ontologia:

1. Ontologia é uma disciplina da Filosofia.
2. Ontologia é uma conceitualização informal de um sistema.
3. Ontologia é um relato semântico formal.
4. Ontologia é uma especificação de uma conceitualização compartilhada.
5. Ontologia é uma representação de um sistema conceitual a través de uma teoria da Lógica.
6. Ontologia é um vocabulário utilizado por uma teoria Lógica.

7. Ontologia representa o meta nível da especificação de uma teoria da Lógica.

Ontologia representa o entendimento compartilhado de algum domínio de interesse que pode ser usado como uma estrutura unificadora para resolver problemas de interoperabilidade e falta de comunicação(Uschold e Gruninger, 1996). As ontologias permitem um alto nível de reúso, usando um vocabulário comum no meio de plataformas de softwares heterogêneos. Elas tem sido utilizadas por várias áreas da ciência da computação: Inteligência Artificial, Representação do Conhecimento, Processamento de Linguagem Natural, Web Semântica, Engenharia de Software entre outras, por isso a grande variação na definição desse artefato. Uschold (1998) propõe uma definição mais abrangente: Uma ontologia pode assumir vários formatos, mas necessariamente deve incluir um vocabulário de termos e alguma especificação do seu significado. Esta deve abranger definições e uma indicação de como os conceitos estão inter-relacionados, o que resulta na estruturação do domínio e nas restrições de possíveis interpretações de seus termos.

4.3 Componentes de Ontologias

Maedche e Staab (2001) explícitam em alto grau de formalismo os componentes de uma ontologia, a pesar de isso a descrição dada por Gómez-Pérez, Fernández-López, e Corcho (2004a) é mais completa por considerar a parte das instâncias como componente das ontologias. Sendo assim, uma ontologia é formada pelos seguintes componentes:

- Um conjunto de conceitos e uma hierarquia entre esses conceitos, ou seja, uma taxonomia. Os conceitos podem ser abstratos (ex. força) ou concretos (ex. livro), elementares (ex. átomo) ou compostos (ex. molécula), reais ou fictícios. Um exemplo de taxonomia é o conceito homem ser um sub conceito do conceito pessoa.
- Um conjunto de relacionamentos entre esses conceitos. Um exemplo de relacionamento entre os conceitos carro e pessoa é o relacionamento de pertence.
- Um conjunto de funções. Uma função é um relacionamento em que um conjunto de elementos tem uma relação única com um outro elemento. Um exemplo de função é *ser_pais_biológicos*, onde o conceito homem e um conceito mulher estão relacionados a um conceito pessoa.
- Um conjunto de axiomas, ou seja, regras que são sempre verdade. Um exemplo de axioma é afirmar que toda pessoa tem uma mãe.
- Um conjunto de instâncias que são um conhecimento prévio existente na ontologia.

4.4 Classificação de Ontologias

O trabalho Breitman, Casanova, e Truszkowski (2007) resumem a classificação de ontologias da seguente maneira:

4.4.1 Classificação quanto ao Tipo de Informação

Gómez-Pérez, Fernández-López, e Corcho (2004a) propõem uma classificação centrando-se no tipo de informação a ser modelada, identificando os seguentes tipos:

- Ontologias para representação do conhecimento: capturam primitivas de representação do conhecimento. Esse tipo de ontologia fornece as primitivas para modelagem de linguagens baseadas em frames, amplamente utilizados na Inteligência Artificial, tais como classes, subclasses, atributos, axiomas e valores. Um exemplo desta ontologia está em Ontolingua(KSL, 2008), e são OKBC(KSL, 1995) e *Frame Ontology*(KSL, 1994).

- Ontologias gerais e de uso comum: são utilizadas para representar conhecimento de senso comum que pode ser utilizado em vários domínios. Essas ontologias incluem um vocabulário relacionado a suas classes, eventos, espaço, causalidade, comportamento, entre outros. Um exemplo é a ontologia de tempo, disponível na biblioteca pública de ontologias do site DAML[1], a ontologia *time.daml*.

- Ontologias de nível superior: descrevem conceitos muito gerais. Algumas propostas para esse tipo de ontologia são *SENSUS*, OpenCyc[2] e MikroKosmos.

- Ontologias de domínio: são ontologias reutilizadas dentro de um domínio específico (médico, farmacêutico, direito, financeiro entre outros). Os termos de uma ontologia de domínio são obtidos a través da especialização de conceitos de uma ontologia de topo. O mesmo é verdade para suas propriedades.

- Ontologias de tarefas: descrevem o vocabulário ligado a uma tarefa ou atividade específica.

- Ontologias de domínio-tarefa: são ontologias de tarefas que podem ser reutilizados em domínios similares.

- Ontologias de métodos: fornecem definições para os conceitos e relacionamentos relevantes para um processo de modo a se atingir um objetivo.

- Ontologias de aplicação: são ontologias dependentes de uma determinada aplicação. Contêm toda a informação que se precisa para modelar o conhecimento necessário à aplicação em particular. Esse tipo de ontologia é utilizado para especializar e estender ontologias de domínio ou tarefa para uma dada aplicação.

4.4.2 Classificação segundo seu Espectro Semântico

Lassila e Mcguinness (2001) propõem uma classificação baseada na estrutura interna e no conteúdo das ontologias. A classificação segue uma lista em que as ontologias são dispostas desde a mais leve até a mais pesada.

- Vocabulários controlados: Lista finita de termos. Por exemplo NAICS[3](*North American Industry Classification System*) que lista produtos e serviços oferecidos em diversas áreas, como por exemplo agricultura, finanças, varejo entre muitas outras.

- Glossários: Lista de termos com significado em linguagem natural. Por exemplo NetGlos (*The Multilingual Glossary of Internet Terminology*), que reúne terminologia relacionada a recursos na Internet.

- Tesauros: Lista de termos e suas definições que padroniza a utilização de palavras para indexação. Além das definições, um tesauro fornece relacionamentos entre os termos. Por exemplo *IEEE Web Thesauros Keywords* que contém vocabulário associado à engenharia elétrica e eletrônica.

- Hierarquias tipo-de-informais: Hierarquias que utilizam o relacionamento de generalização(tipo-de) de maneira informal. Nesse tipo de hierarquia, os conceitos relacionados podem ser agregados a uma categoria, mesmo que não respeitem integralmente o relacionamento de generalização. Por exemplo é a hierarquia apresentada pelo *Yahoo*(Labrou e Finin, 1999).

- Hierarquias tipo-de-formais: Hierarquias que incluem instâncias de um domínio. Nessas hierarquias os relacionamentos de generalização são respeitadas integralmente. Um exemplo é a taxonomia dos seres vivos.

[1] http://www.daml.org/
[2] http://www.opencyc.org/
[3] http://www.census.gov/eos/www/naics/

- *Frames*: modelos que incluem classes (or *Frames*) e propriedades. As classes possuem propriedades, chamadas atributos (*slots*). Atributos não têm escopo global, se aplicam apenas nas classes para as quais foram definidas. Cada *Frame* fornece o contexto para modelar um aspecto de um domínio.

- Ontologias que experimentam restrições de valores: ontologias que oferecem subsídios para restringir os valores assumidos pelas propriedades de suas classes. Por exemplo, na propriedade *ano de nascimento* tem que ser um número inteiro.

- Ontologias que exprimem restrições lógicas: ontologias que permitem que sejam expressas restrições em lógica de primeira ordem.

4.5 Aplicações, Ventagens e Problemas sobre Ontologias

Guizzardi (2000), no seu trabalho descreve as principais ventagens de uso de ontologias, elas são:

- Comunicação: As ontologias possibilitam a comunicação entre pessoas acerca de determinado conhecimento, pois permitem raciocínio e entendimento sobre um domínio. Auxiliando na obtenção de um consenso, principalmente sobre termos técnicos, entre comunidades profissionais, de pesquisa, etc.

- Formalização: A especificação da ontologia permite eliminar contradições e inconsistências na representação de conhecimento, além de não ser ambígua. Além disso, essa especificação pode ser testada, validada e verificada.

- Representação de Conhecimento e Reutilização: As ontologias formam um vocabulário de consenso que permite representar conhecimento de um domínio em seu nível mais alto de abstração, possuindo, desta forma, potencial de reutilização.

O trabalho de (Guizzardi, 2000), também explora as principais problemas ao usar ontologias:

- Criação e Evolução das Ontologias: Ontologias precisam evoluir. Poucos trabalhos, entretanto, têm enfocado este aspecto. Alguns exemplos de ontologias podem ser encontrados em (Clark, 2013).

- Bibliotecas de Ontologias: A noção de biblioteca de ontologias está relacionada à independência entre elas. Desta forma, a interface entre estas ontologias se constitui em um problema, uma vez que cada uma delas pode ter sido desenvolvida em um contexto diferente.

- Metodologia de Desenvolvimento: Este é considerado o principal problema relacionado às ontologias, principalmente pela falta de trabalhos descrevendo metodologias para seu desenvolvimento. Existe ausência de atividades padronizadas, ciclos de vida e métodos sistemáticos de desenvolvimento, assim como um conjunto de critérios de qualidade, técnicas e ferramentas. Isso faz com que o processo de criação de ontologias seja uma atividade quase *artística*.

Na ciência da computação, as ontologias são utilizadas em áreas como:

- Recuperação de informações na Internet: Por exemplo, OntoSeek(Guarino *et al.*, 1999), recupera informações de catálogos de produtos on-line utilizando um sistema de agentes inteligentes. Outro exemplo é TOVE[4]. TOVE é uma ontologia organizacional para interoperabilidade dos sistemas de Recursos humanos e marketing e Swoogle[5] permite buscar ontologias disponíveis na web.

[4]http://www.eil.utoronto.ca/enterprise-modelling/tove/
[5]http://swoogle.umbc.edu/

- Processamento de Linguagem Natural: O conhecimento modelado em uma ontologia pode servir de análise semântica de textos(da Camara Junior, 2013). *Ontology Learning* a partir de textos consiste em extrair ontologias aplicando técnicas de linguagem natural e linguística em textos. O mapeamento de Ontologias ou *Ontology Mapping* pode ser definido como uma função que associa termos e expressões definidas em uma ontologia fonte com termos e expressões definidos em uma ontologia alvo.

- Gestão do Conhecimento: Os trabalhos baseados em ontologias propõem o armazenamento da memória corporativa da empresa através das ontologias. CHAR[6], é uma ferramenta para aquisição de conhecimentos por anotações, construindo uma ontologia para armazenar fatos sobre histórias corporativas para depois ser consultadas.

- Web-Semântica: As ontologias compartilham informação com significado explícito(Berners-Lee *et al.* , 2001). Um exemplo é Disco[7], Disco permitem a navegação semântica, i.e., navegar nas propriedades dos recursos de arquivos *RDF*. A indexação é feita pelos metadados e é útil para reuso de ontologias existentes.

- Educação: Por exemplo as ontologias dos trabalhos de Aroyo e Dicheva (2002) descrevem arquiteturas baseadas em ontologias para treinar estudantes e modelar ambientes de aprendizagens.

4.6 Linguagens para Representação de Ontologias

Para propósitos deste trabalho exploraremos com mais profundidade as linguagens de ontologias para Web.

4.6.1 DARPA Agent Markup Language + Ontology Interface Language (DAML+OIL)

A DAML+OIL consiste de uma linguagem de marcação semântica para recursos Web, estende os padrões *RDF* e *RDFS*. A semântica formal de DAML+OIL é fornecida através do mapeamento da linguagem para a linguagem KIF(*Knowledger Interchange Format*).

DAML+OIL é dividida em duas partes, domínio dos objetos, que consiste nos objetos que são membros de classes definidas na ontologia *DAML*, e domínio dos tipos de dados, que consiste nos valores importados do modelo *XML*. O objetivo da separação é permitir a implementação de mecanismos de inferência, já que realizar inferências sobre tipos concretos de dados não seria possível. *DAML* é composta por elementos de classe, expressões de classe e propriedades.

4.6.2 Resource Description Framework (RDF e RDF-Schema)

A linguagem *RDF*(RDF Working Group, 2004), desenvolvida pela *W3C*, representa o conhecimento por meio de redes semânticas. *RDF* é uma extensão *XML* e serve como base para processamento de metadados. *RDF* facilita a transmissão de informações entre aplicativos via Web proporcionando um modelo muito simples para grafos de dados, mas não especifica semântica complexa para relações ou modelos de dados avançados.

A estrutura de descrição de recursos da *RDF* é composta por três tipos de objetos: recursos, propriedades e valores. Uma tripla é formada por um recurso, uma propriedade e um valor para a propriedade do recurso. Um recurso é a informação que podemos identificar usando uma URI (*Universal Resource Identifier*). Assim, uma descrição em *RDF* é uma coleção em triplas. Por exemplo:

- Recurso (ou Sujeito) (http://www.w3c.es/Pessoal/Martin)

[6] http://www.ontoprise.com
[7] http://www.linguatools.de/disco/disco_en.html

- Propriedade (ou predicado) (Criador)
- Objeto - Literal ("Martín Álvarez")

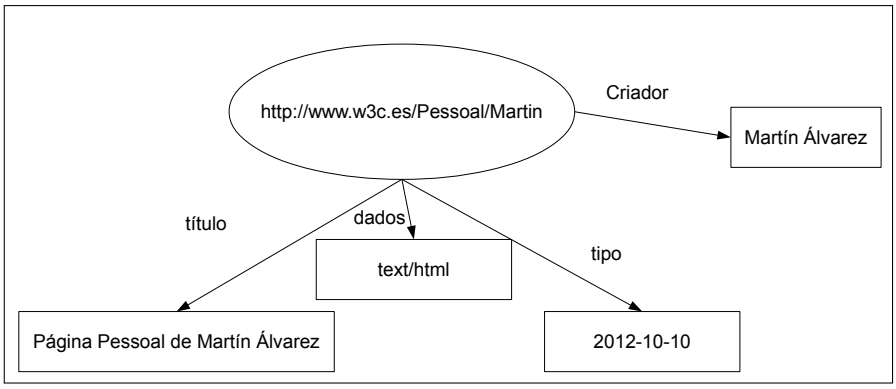


Figura 4.1: *Estrutura do exemplo de RDF*

```
<rdf:RDF>
    <rdf:Description about="http://www.w3c.es/Pessoal/Martin">
        <s:tipo>text/html</s:tipo>
        <s:dados>2012-10-10</s:dados>
        <s:título>Página Pessoal de Martín Álvarez</s:título>
    </rdf:Description>
</rdf:RDF>
```

Figura 4.2: *Código de exemplo de uma estrutura RDF*

4.6.3 Web Ontology Language (OWL)

OWL é uma extensão do modelo de dados *RDF*, para fornecer um conjunto mais abundante de semântica que permita a construção de modelos de dados complexos, vocabulários, lógicas de software(Bechhofer *et al.*, 2004).

OWL fornece um tipo de estrutura orientada a objetos que vincula triplos. *RDF* as classes, associações e outras relações complexas.

OWL é dividida em três sub-linguagens: *OWL Lite*, *OWL DL* e *OWL Full*. *OWL Lite* é uma versão simplificada da *OWL* oferecendo hierarquias e restrições simples. *OWL DL* aumenta a expressividade mantendo a decibilidade, é um subconjunto da *OWL Full* e faz uso da lógica descritiva, de onde vem a extensão *DL* (*Description Logic*). A *OWL Full* mantém a expressividade, porém sem garantias de decibilidade, uma vez que instncias e propriedades podem ser especificadas como classes, diferentemente da *OWL DL*.

OWL representa conceitos e relacionamentos e é constituída de:

4.6.4 Espaços de Nome

Espaços de nome explicita um conjunto de padrões que serão utilizados durante a construção da ontologia. Na Figura 4.3 as duas primeiras linhas declaram o *namespace* da própria ontologia. Na segunda declaração identificamos o nome da ontologia com o prefixo de *vin*, esse exemplo refere se a uma ontologia sobre vinhos, um exemplo clássico[8]. A terceira declaração identifica a *URI* da ontologia. Na quarta declaração identifica outra ontologia importada, a de comida (*food*), da qual alguns conceitos foram importados. A quinta declaração contêm o prefixo *owl*, a qual serve para introduzir vocabulário suportado por *OWL*. Finalmente, *OWL* depende de primitivas definidas nas camadas inferiores, ou seja, tipos de dados nativos de *RDF* e *RDFS* e do *XML Schemas*.

[8]http://www.daml.org/ontologies/76

```
<rdf:RDF
    xmlns       = "http://www.w3.org/2004/owl-guide-2004/wine#"
    xmlns:vin   = "http://www.w3.org/2004/owl-guide-2004/wine#"
    xml:base    = "http://www.w3.org/2004/owl-guide-2004/wine#"
    xmlns:food  = "http://www.w3.org/2004/owl-guide-2004/food#"
    xmlns:owl   = "http://www.w3.org/2002/07/owl#"
    xmlns:rdf   = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs  = "http://www.w3.org/2000/01/rdf-schema#"
    xmlns:xsd   = "http://www.w3.org/2001/XMLSchema#">
```

Figura 4.3: *Exemplo de espaços de nome para uma ontologia em OWL.*

4.6.5 Cabeçalho

Os cabeçalhos são uma coleção de sentenças de etiquetas em que se colocam os metadados referentes ao documento, as quais registram informações como comentários, controle de versões e inclusão de outras ontologias definidos através do elemento <*owl:Ontology*>. Na Figura 4.4 mostra um exemplo.

```
<owl:Ontology rdf:about="">
    <rdfs:commet>Este é um exemplo de comentario dentro de uma ontologia escrita em OWL</rdfs:commet>
    <owl:priorVersion rdf:resource="http://www.w3.org/owl-guide-2004/wine"/>
    <owl:imports rdf:resource="http://www.w3.org/owl-guide-2004/food"/>
    <rdfs:label>Ontologia dos Vinhos</rdfs:label>
```

Figura 4.4: *Exemplo de cabeçalho para uma ontologia em OWL.*

4.6.6 Classes e Subclasses

Uma classe representa um conjunto ou coleção de indivíduos(objetos, pessoas, coisas) que compartilham de um grupo de características que os distinguem dos demais. Classes e subclasses são definidas por elementos *owl:Class* e *rdfs:subClassOf*. Na Figura 4.5 mostra um exemplo de definição de classes. *rdfs:subClassOf* define uma hierarquia de classes a através de relacionamentos de generalização (tipo-de). Esses tipos de relacionamento são transitivos.

```
<owl:Class rdf:ID="Animal_De_Estimacao"/>

<owl:Class rdf:ID="Cachorro">
    <rdfs:subClassOf rdf:resource="#Animal_De_Estimacao"/>
    ...
</owl:Class>

<owl:Thing rdf:ID="Rin_Tin_Tin">
    <owl:Thing rdf:about="#Rin_Tin_Tin">
        <rdf:type rdf:resource="#Cachorro"/>
    </owl:Thing>
</owl:Thing>
```

Figura 4.5: *Exemplo de definição de classes para uma ontologia em OWL.*

4.6.7 Instâncias

As instâncias ou indivíduos são os objetos do mundo. Pertencem à classes e são relacionados a outros indivíduos (e classes) através de propriedades. Todos os indivíduos em uma ontologia *OWL* pertencem a uma classe *owl:Thing*. Essa conceitualização faz com que exista sempre uma única

raiz para qualquer taxonomia. Na Figura 4.5 mostra um exemplo de definição de uma instância para uma ontologia em *OWL*.

4.6.8 Propriedades das Classes

As propriedades das classes servem para descrever fatos em geral. Existem dois tipos de propriedades:

- Propriedades do tipo *object*: definem relacionamentos entre classes. São definidas através da etiqueta *owl:Property* que utiliza os elementos *rdfs:domain* para especificar a qual classe a propriedade está sendo atribuída e *owl:range* para especificar a qual classe ela será aplicada. A Figura 4.6 mostra um exemplo.
- Propriedades do tipo *datatype*: indicam um relacionamento entre instâncias de classes e literais expressos em RDF e datatypes do XML Schema[9]. A Figura 4.7 mostra um exemplo de uma propriedade do tipo *datatype* que relaciona todos os seres vivos com um ano de nascimento. A Figura 4.8 mostra um exemplo de indivíduo chamado Totó, da classe cachorro associado com seu ano de nascimento 2001.

```
<owl:ObjectProperty rdf:ID="come_racao">
    <rdfs:domain rdf:resource="#Cachorro"/>
    <rdfs:range rdf:resource="#Racao"/>
</owl:ObjectProperty>
```

Figura 4.6: *Exemplo de definição de uma propriedade tipo* ***object*** *para uma ontologia em OWL.*

```
<owl:Class rdf:ID="SerVivo">

<owl:DatatypeProperty rdf:ID="anoNascimento">
    <rdfs:domain rdf:resource="#SerVivo"/>
    <rdfs:range rdf:resource="&xsd;positiveInteger"/>
</owl:DatatypeProperty>
```

Figura 4.7: *Exemplo de definição de uma propriedade tipo* ***datatype*** *para uma ontologia em OWL.*

```
<owl:Class rdf:ID="Cachorro">

<Cachorro rdf:ID ="Totó">
    <anoNascimento rdf:datatype="&xsd;positiveInteger">2001</anoNascimento>
</Cachorro>
```

Figura 4.8: *Exemplo de definição de uma propriedade tipo* ***datatype*** *associado a um indivíduo.*

4.6.9 Restrições das Propriedades

As restrições das propriedades são utilizadas para definir alguns limites para indivíduos que pertencem a uma classe. Em *OWL* podem ser de três tipos:

- Restrições que utilizam quantificadores: O quantificador existencial, o quantificador universal.

[9] http://www.w3.org/2001/XMLSchema

- Restrições de cardinalidade: cardinalidade mínima, cardinalidade máxima, relação inversa entre outras propriedades.
- Restrições do tipo, ”tem valor de”.

4.7 Ferramentas para a Construção de Ontologias

O trabalho de (Gómez-Pérez *et al.*, 2004b) classifica as ferramentas relacionadas à ontologias como: ferramentas para construção de ontologias, avaliação de ontologias, alinhamento e combinação de ontologias, manutenção de instâncias, para consulta e motor de inferência, e por último ferramentas de aprendizagem de ontologias.

Para propósitos deste trabalho nos focamos nas ferramentas para construção de ontologias, as mais reconhecidas são:

- *Protegé*: Ambiente iterativo para projeto de ontologias, de código aberto, e oferece uma interface gráfica para edição de ontologias e uma arquitetura para criação de ferramentas baseadas em conhecimento. As ontologias são armazenadas em *OWL*. A arquitetura é modulada e permite a inserção de novos recursos(Noy *et al.*, 2000).
- *OntoEdit*: Ambiente gráfico para edição de ontologias que permite inspeção, navegação, codificação e alteração de ontologias. O modelo de conceitual é armazenado usando um modelo de ontologia que pode ser mapeado em diferentes linguagens de representação. As ontologias são armazenadas em bancos relacionais e podem ser implementadas em *XML*, F-Logic, *RDF* e *DAML+OIL*.
- *WebODE*: Ambiente para engenharia ontológica que dá suporte à maioria das atividades de desenvolvimento de ontologias. As ontologias são armazenadas em *OWL*. A integração com outros sistemas é possível, importando e exportando ontologias de linguagens de marcação.
- *WebOnto*: Ambiente para navegação, criação e edição de ontologias, representadas na linguagem de modelagem *OCML*. Permite o gerenciamento de ontologias por interface gráfica, inspeção de elementos, verificação da consistência da herança e trabalho cooperativo. Possui uma biblioteca com mais de cem ontologias.

4.8 Ontologias para Modelamento Conceitual

Podemos evitar a produção de modelos semânticamente sem sentido ao estabelecer um mapeamento semântico dos elementos do modelo para a ontologia.

(Sugumaran e Storey, 2002), (da Camara Junior, 2013)

4.9 Sumário

Principais conceitos do capítulo:

- Ontologia é uma especificação formal e explícita de uma conceitualização compartilhada.
- As ontologias permitem um alto nível de reúso, usando um vocabulário comum por meio de plataformas de softwares heterogêneos.
- Uma ontologia é formada por um conjunto de conceitos, de relacionamentos, de funções, de axiomas e instâncias.
- *RDF*, *RDFS*, *SHOE*, *XOL*, *DAML+OIL* e *OWL* são alguns dos mais importantes linguagens de ontologias na internet.

- A metodologia *Methontology* apresenta uma maior madurez principalmente pela sua clareza nos artefatos gerados na descrição conceitual da ontologia.
- As ontologias auxiliam na necessidade de caracterizar diferentes contextos.

No próximo capítulo, explicamos como viabilizar a interoperabilidade e atingir a portabilidade documentos usando ontologias.

Capítulo 5

Modelo Conceitual de Documentos

Este capítulo explica como viabilizar a interoperabilidade e atingir a portabilidade confiável de documentos através da reutilização de formatos e conteúdos, em diferentes combinações plausíveis.

O trabalho concentra-se nos dois principais padrões abertos de formatos de documentos, *Office Open XML* e *Open Document Format*, como prova de conceito.

5.1 Modelamento Conceitual de Documentos

O modelo de conceitual de documentos será criado seguindo a abordagem da engenharia de documentos.

5.1.1 Aplicação da Abordagem da Engenharia de Documentos

A engenharia de documentos produz modelos para possibilitar o intercâmbio de documentos. Um dos produtos principais da engenharia de documentos é a criação do modelo de componentes dos documentos. O modelo de componentes é o produto mais importante das fases de análise e montagem de componentes, usualmente apresentado num diagrama de classes *UML*. No entanto, neste trabalho os modelos de componentes são representados usando ontologias. Uma classe da ontologia corresponde a um componente do modelo de componentes.

A relação entre o modelo orientado a objetos (por exemplo um diagrama de classes *UML*) e as ontologias é clara na medida em que ambos buscam representar semânticamente um domínio de conhecimento. O modelo orientado a objetos busca classificar o domínio através de classes e objetos, possibilitando o reuso dos mesmos em outras aplicações. Esses princípios também são empregados nas ontologias. Entretanto, o modelo orientado a objetos não pode ser considerado um formalismo de representação devido à ausência de declaratividade, de um motor de inferência e de uma semântica formal. Assim, ele é empregado mais para modelagem estrutural e comportamental do que para a modelagem conceitual, para a qual são utilizadas as ontologias(da Silva *et al.*, 2009). Qualquer modelo orientado a objetos pode ser representado com ontologias, mas caso contrário nem sempre é possível.

5.1.2 Construção das Ontologias

Um dos objetivos da engenharia de documentos é apresentar o documento final num modelo de formato adequado ao usuário. Seguindo a abordagem da engenharia de documentos, o modelo conceitual de documentos deve ser representado numa tecnologia neutra, focando na separação entre o conteúdo e a forma da apresentação do conteúdo do documento. O conteúdo do documento é independente da sua forma da apresentação, pois diferentes estruturas de apresentações podem existir para um mesmo conteúdo de informação. Como vários formatos podem ser criados a partir da mesma base de conteúdo.

Propomos um modelo conceitual para o compartilhamento de documentos. A Figura 5.1 ilustra nossa proposta. Nosso modelo conceitual é uma Ontologia de Documento. Ela está baseada na

integração de dois módulos de ontologias. Um módulo caracteriza a estrutura da apresentação da informação e outro módulo caracteriza a informação do contexto de negócio, a primeira é a Ontologia de Formato e a segunda é a Ontologia do Conteúdo. Elas são independentes entre si e foram criados baseadas no análise e montagem de componentes da engenharia de documentos.

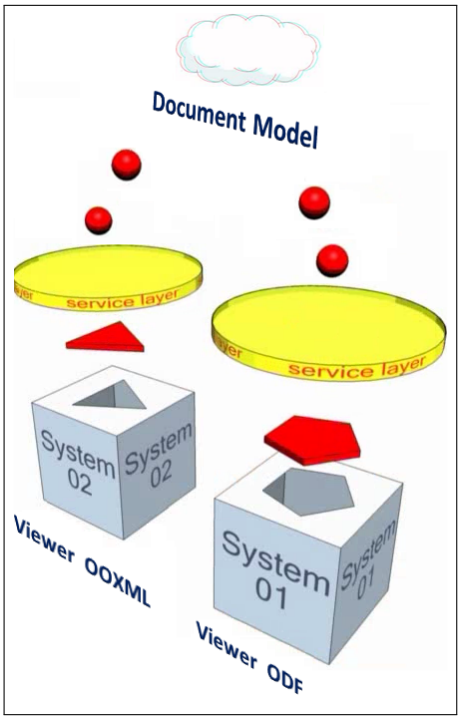


Figura 5.1: *Um modelo conceitual para compartilhamento de documentos.*

A finalidade dessas ontologias é a representação do formato e conteúdo de determinados contextos de documentos digitais. As regras semânticas e estruturais do documento são definidas dependendo do domínio. O propósito da Ontologia de Conteúdo é possibilitar a interoperabilidade de documentos em um determinado domínio. O objetivo da construção da Ontologia de Formato é atingir a portabilidade simples, abordando os principais conceitos da estrutura de apresentação de documentos. A Figura 5.2 mostra a interação entre as Ontologias de Formato e Conteúdo e finalmente a geração do documento final. O Renderizador de Texto combina as duas ontologias e gera um objeto de entrada para os tradutores. O mapeamento entre a Ontologia de Documento a suas interfaces físicas é feito através de tradutores.

Trabalharemos sob o domínio de gestão acadêmica, nosso documento de caso de prova será Ficha de Aluno. O documento Ficha de Aluno mostra os dados pessoais e acadêmicos de um estudante matriculado num curso de alguma determinada universidade. As Figuras 5.3 e 5.4 mostram por exemplo do documento Ficha de Aluno com a informação preenchida do estudante. As Ontologias de Formato e Conteúdo modelam a informação contida no documento Ficha de Aluno da seguinte forma: a Ontologia de Conteúdo foca se na informação do aluno, modelando a informação pessoal e acadêmica de um estudante matriculado em um curso, por outro lado a Ontologia de Formato centra se no modelamento da estrutura da apresentação visual do documento.

As ontologias de Conteúdo e Formato foram construídas usando a ferramenta *Protegé* na linguagem *OWL*. A Figura 5.5 mostra as classes e propriedades da Ontologia de Conteúdo. Para maior detalhe, a Tabela 5.1 descreve as classes da Ontologia de Conteúdo e na Tabela 5.2 são descritas as propriedades das classes da Ontologia do Conteúdo. As classes e suas propriedades da Ontologia de Formato são mostradas na Figura 5.6. Nas Tabelas 5.3 e 5.4 são explicadas com mais detalhe as classes e propriedades da Ontologia de Formato. A Ontologia de Formato foi construída para ser independente da Ontologia da Conteúdo. Ela modela os componentes de apresentação, as características visuais dos documentos digitais. Baseada nessa representação podemos recriar um documento, sem o compromisso de compreender ou possuir a semântica da informação do docu-

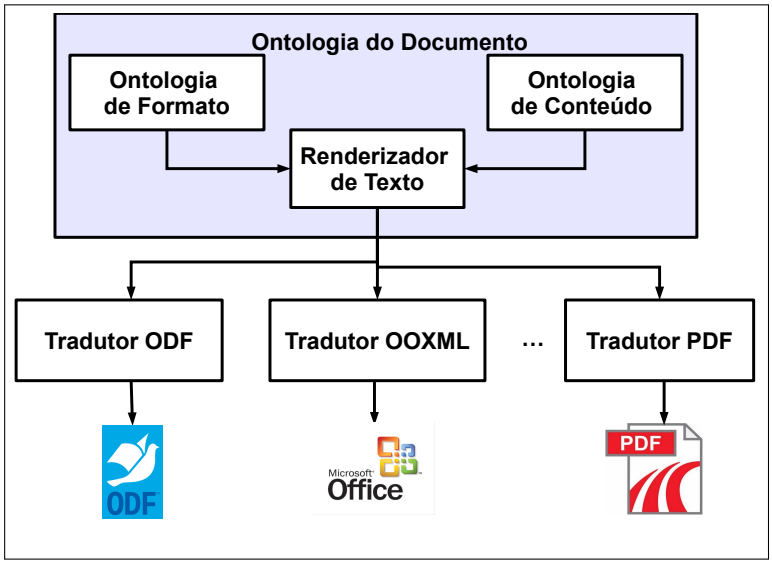
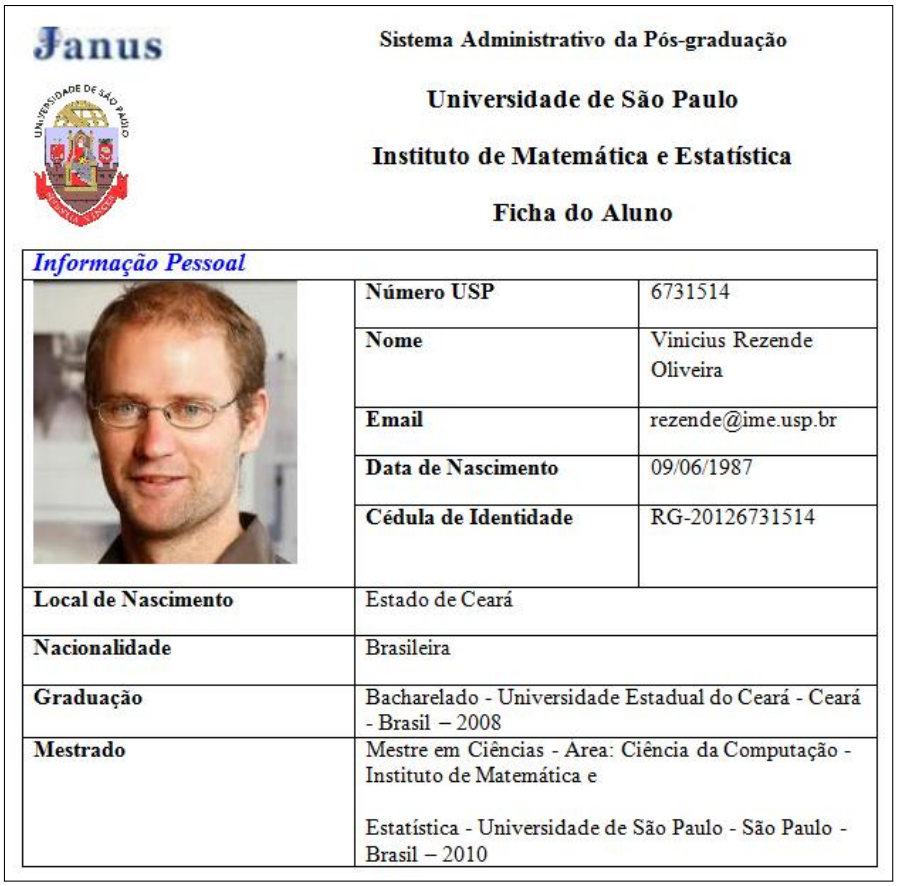


Figura 5.2: *Geração do documento digital a partir da Ontologia de Documento.*



Janus

Sistema Administrativo da Pós-graduação

Universidade de São Paulo

Instituto de Matemática e Estatística

Ficha do Aluno

Informação Pessoal		
	Número USP	6731514
	Nome	Vinicius Rezende Oliveira
	Email	rezende@ime.usp.br
	Data de Nascimento	09/06/1987
	Cédula de Identidade	RG-20126731514
Local de Nascimento	Estado de Ceará	
Nacionalidade	Brasileira	
Graduação	Bacharelado - Universidade Estadual do Ceará - Ceará - Brasil – 2008	
Mestrado	Mestre em Ciências - Area: Ciência da Computação - Instituto de Matemática e Estatística - Universidade de São Paulo - São Paulo - Brasil – 2010	

Figura 5.3: *Ficha de aluno com a informação pessoal.*

mento. Por outro lado, a Ontologia de Conteúdo nos permite entender semânticamente a informação visualizada em uma Ficha de Aluno. Assim, a Ontologia de Formato e de Conteúdo poderiam ser utilizadas em outras aplicações. A combinação das Ontologias de Formato e de Conteúdo chamamos de Ontologia de Documento. Uma instância da Ontologia de Documento é um objeto que serve de entrada a tradutores, os quais criam um documento em um formato específico, por exemplo: *ODF* ou *OOXML*.

Informação Acadêmica	
Curso	Doutorado
Área	Ciência da computação
Data de matrícula	11/01/2011
Inicio da contagem do prazo	11/01/2011
Data limite	11/01/2016
Orientador	Prof. Flávio Soáres Corrêa da Silva fcs@ime.usp.br
Prazo máximo para aprovação no exame de proficiência em línguas	11/07/2013
Prazo máximo para inscrição no exame de qualificação	11/01/2014
Data de aprovação no exame de qualificação	
Data do depósito do trabalho	
Título do trabalho	
Data máxima para aprovação da banca	
Data de aprovação da banca	
Data máxima para defesa	
Data da defesa	
Resultado da defesa	
Histórico de ocorrências	Ingressou no Doutorado em 11/01/2011 Matricula Regular em 18/07/2011 Aluno matriculado nas normas vigentes a partir de 01/07/2009
Última ocorrência	Matricula Regular em 18/07/2011
Impresso em	29/07/12 15:13

Figura 5.4: *Ficha de aluno com a informação acadêmica.*

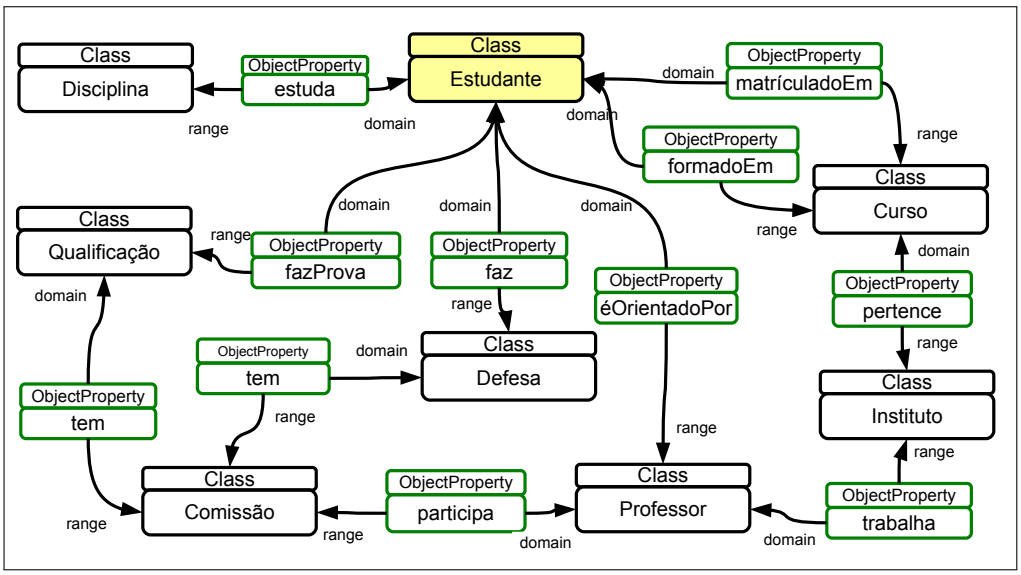


Figura 5.5: *Ontologia de Conteúdo para o domínio Ficha de aluno.*

5.1.3 Caracterização dos Documentos Digitais usando Ontologias

Para caracterizar os documentos digitais precisamos criar instâncias das classes e das propriedades respectivas em conformidade com a informação do domínio a representar. Explicaremos o fluxo completo para a criação de um documento digital a partir das ontologias. Como cenário, usaremos o documento Ficha de Aluno do Estudante *Vinicius Rezende Oliveira* mostrado nas Figuras 5.3 e 5.4. Esse documento foi recriado a partir das Ontologias de Conteúdo e Formato.

A informação da Ontologia de Conteúdo está baseada na informação do documento Ficha de

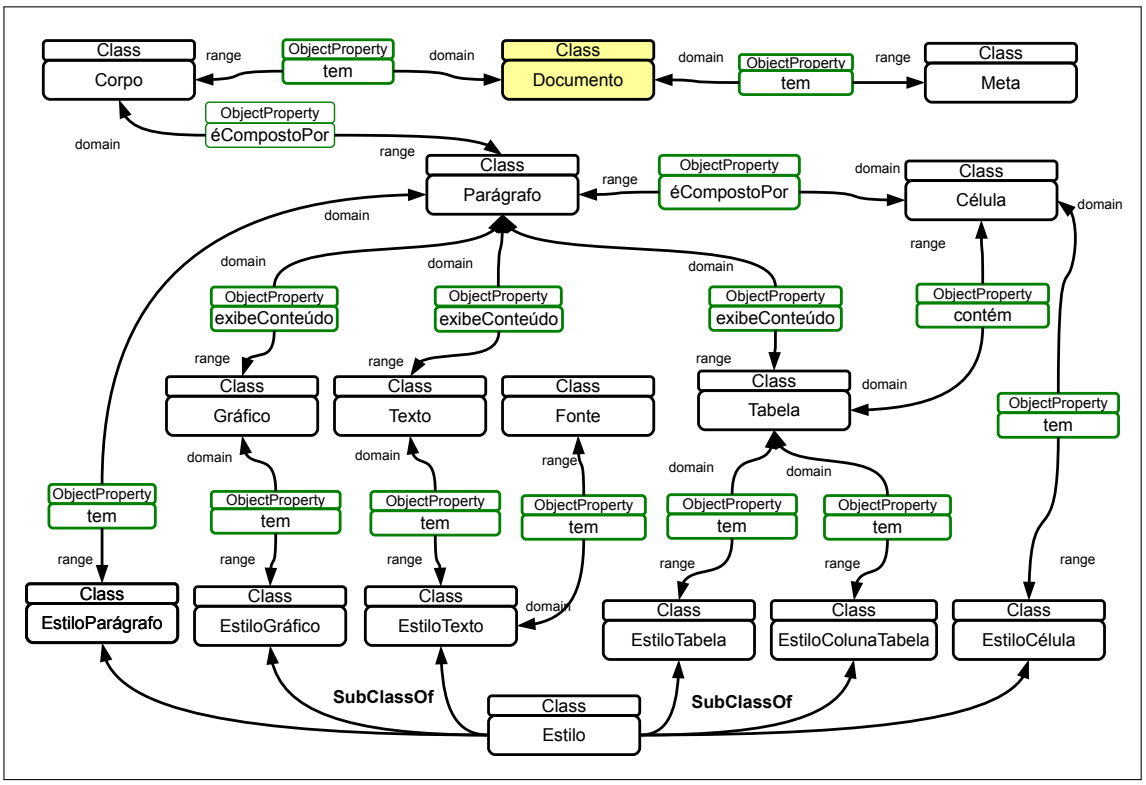


Figura 5.6: *Ontologia de Formato.*

Aluno. Para iniciar, construímos uma instância da classe Estudante com a informação do aluno da Figura 5.3. A Figura 5.7 mostra a informação da instância da classe Estudante para o aluno *Vinicius Rezende Oliveira*. A informação das instâncias *graduaçãoVinicius*, *mestradoVinicius* e *doutorado-Vinicius* são mostradas nas Figuras 5.8, 5.9 e 5.10. No documento de exemplo, o aluno está matriculado no Curso de doutorado, como se mostra na Figura 5.10. Também, a Figura 5.11 mostra a informação da Instituição de ensino de graduação que e é a mesma do mestrado e finalmente, na Figura 5.12 mostra a Instituição do doutorado. Continuando com a informação acadêmica, um Estudante é orientado por um Professor, o orientador do aluno *Vinicius Rezende Oliveira* é o Professor *FlavioSoares*, a Figura 5.13 mostra a informação da instância Professor *FlavioSoares*. Cada aluno mantém um histórico das atividades desde o inicio do Curso, sendo assim a Figura 5.14 mostra a informação do Histórico das atividades do aluno desde seu ingresso no doutorado.

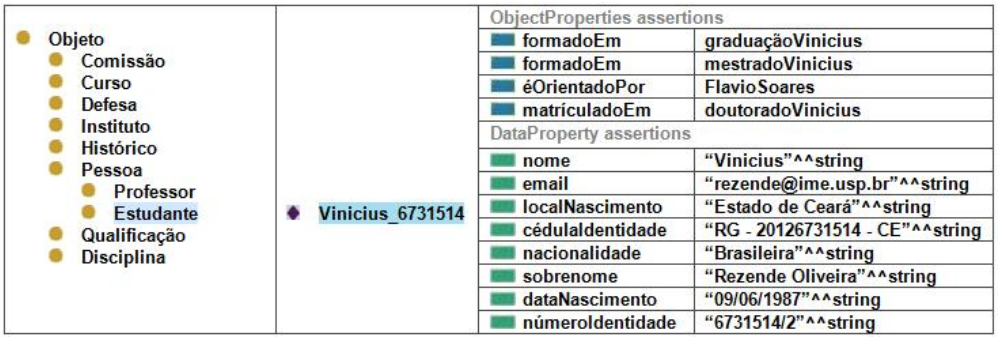


Figura 5.7: *Instância da classe Estudante do aluno* ***Vinicius Rezende Oliveira****.*

A Figura 5.16 mostra a árvore de instâncias que caracterizam o texto da Figura 5.15 que é o cabeçalho do documento Ficha de Aluno. Seguindo com o exemplo do documento da Figura 5.3 e Figura 5.4. A Figura 5.17 mostra as instâncias que são criadas para representar a tabela com dados pessoais do aluno *Vinicius Rezende Oliveira*. Nosso caso, a arvore de instâncias geradas para representar essa tabela é bem maior que o mostrado na Figura 5.16. O número de instâncias geradas

Objeto Comissão Curso Defesa Instituto Histórico Pessoa Professor Estudante Qualificação Disciplina	graduaçãoVinicius	ObjectProperties assertions	
		pertence	institutoGraduação
		DataProperty assertions	
		dataConclusão	"2008"^^string
		área	"Ciência da computação"^^string

Figura 5.8: *Instância da classe Curso do curso de graduação do aluno* ***Vinicius Rezende Oliveira****.*

Objeto Comissão Curso Defesa Instituto Histórico Pessoa Professor Estudante Qualificação Disciplina	mestradoVinicius	ObjectProperties assertions	
		pertence	institutoMestrado
		DataProperty assertions	
		dataConclusão	"2010"^^string
		área	"Ciência da computação"^^string

Figura 5.9: *Instância da classe Curso do curso de mestrado do aluno* ***Vinicius Rezende Oliveira****.*

Objeto Comissão Curso Defesa Instituto Histórico Pessoa Professor Estudante Qualificação Disciplina	doutoradoVinicius	ObjectProperties assertions	
		pertence	institutoDoutorado
		DataProperty assertions	
		dataLimite	"11/01/2016"^^string
		dataMatrícula	"11/01/2011"^^string
		área	"Ciência da computação"^^string
		inícioContagemPrazo	"11/01/2011"^^string

Figura 5.10: *Instância da classe Curso do curso de doutorado do aluno* ***Vinicius Rezende Oliveira***

Objeto Comissão Curso Defesa Instituto Histórico Pessoa Professor Estudante Qualificação Disciplina	institutoGraduação	DataProperty assertions	
		país	"Brasil"^^string
		cidade	"Ceará"^^string
		nome	"Universidade Estadual do Ceará"^^string

Figura 5.11: *Instância do Instituto de ensino da graduação.*

Objeto Comissão Curso Defesa Instituto Histórico Pessoa Professor Estudante Qualificação Disciplina	institutoDoutorado	DataProperty assertions	
		país	"Brasil"^^string
		cidade	"São Paulo"^^string
		nome	"Universidade de São Paulo"^^string

Figura 5.12: *Instância do Instituto de ensino do doutorado.*

para representar um documento geralmente é grande, sendo um processo trabalhoso e repetitivo.

Objeto Comissão Curso Defesa Instituto Histórico Pessoa Professor Estudante Qualificação Disciplina	FlavioSoares	DataProperty assertions	
		nome	"Flávio"^^string
		sobrenome	"Soares Corrêa da Silva"^^string
		email	"fcs@ime.usp.br"^^string
		título	"Prof."^^string

Figura 5.13: *Instância Professor.*

Objeto Comissão Curso Defesa Instituto Histórico Pessoa Professor Estudante Qualificação Disciplina	ocorrência_1	DataProperty assertions	
		descrição	"Ingressou no Doutorado em 11/01/2011"^^string
	ocorrência_2	DataProperty assertions	
		descrição	"Aluno matriculado nas normas vigentes a partir de 01/07/2009"^^string
	ocorrência_3	DataProperty assertions	
		descrição	"Matrícula Regular em 18/07/2011"^^string

Figura 5.14: *Instância do Histórico do aluno.*

Figura 5.15: *Texto do cabeçalho de documento Ficha de Aluno.*

5.1.4 Geração do Documento Digital

Os tradutores para o documento final foram implementados em java. Para o armazenamento e manipulação das ontologias foi usada a biblioteca *owlapi 3.2.4*[1]. O objeto de entrada dos tradutores é gerado da combinação das Ontologias de Formato e Conteúdo. A geração dos documentos para o formato *ODF* foi implementada usando *odfdom 0.8.7*[2] e a geração dos documentos para o formato *OOXML* foi usando *poi 3.8*[3].

5.2 Resultados

No Capítulo 2 explicamos a importância de preservar a autenticidade do documento. Um documento é considerado autêntico se for possível provar que ele contém um conjunto de propriedades, consideradas significativas, que foram corretamente preservadas ao longo do tempo. Um documento é autêntico quando está em conforme com a verdade e, portanto, é digno de fé e confiança. O conjunto de propriedades significativas é definido dependendo do propósito do documento. Baseados nesses princípios, o propósito da Ficha do Aluno é informar os dados pessoais e acadêmicos do aluno. Sendo assim, o documento das Figuras 5.3 e 5.4 está no formato *DOCX*. O documento foi recriado e não sofreu perdida ou modificação da informação pessoal e acadêmica do aluno. O

[1] http://owlapi.sourceforge.net/
[2] http://incubator.apache.org/odftoolkit/odfdom/index.html
[3] http://poi.apache.org/

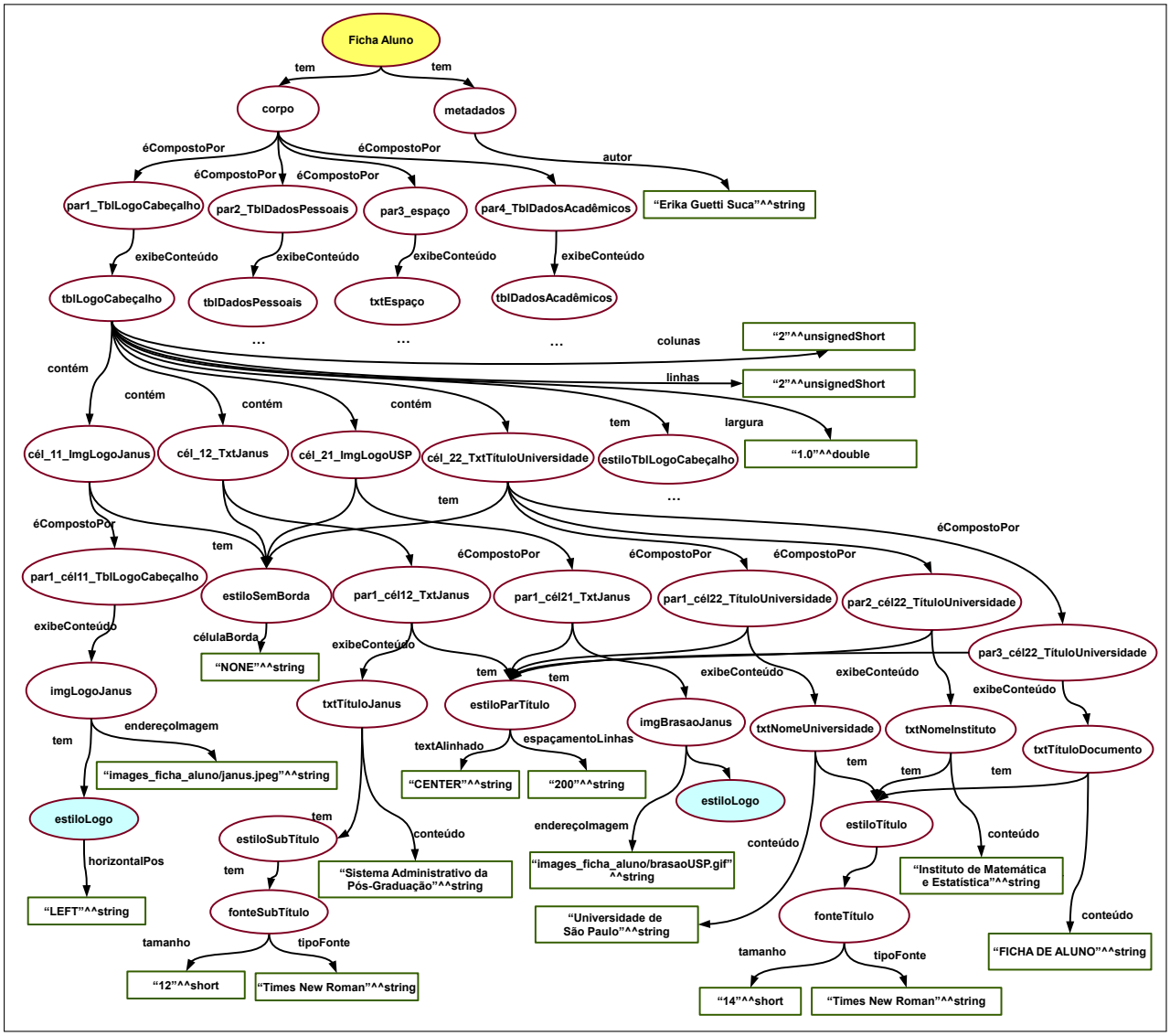


Figura 5.16: *Árvore das instâncias da Ontologia de Formato para o documento Ficha de Aluno*

documento gerado a partir das ontologias manteve seu propósito, então continua sendo autêntico a seus propósitos.

Os elementos estilísticos de um documento são importantes. Mas, são mais difíceis de reproduzir e dificilmente são recriados ao 100%. No apêndice A são mostrados documentos gerados para os formatos *DOCX* e *ODT* a partir da Ontologia de Documentos proposta. Nesses documentos, algumas características estilísticas não puderam ser preservadas ao 100%. Mas a perdida não comprometeu a compreensão da informação e não inválido a informação do aluno. A informação foi mantida em todos os casos.

Para atingir a interoperabilidade de documentos, a Figura 5.18 ilustra a iteração de um fluxo completo de interoperabilidade, isto implicaria que além de gerar documentos a partir de nosso modelo (que pode estar centralizado e compartilhado), também precisariamos possibilitar a entrada de documentos e o preenchimento automático das ontologias de Formato e Conteúdo com a informação dos documentos.

5.3 Conclusões

Principais conclusões do capítulo:

- O modelo conceitual proposto, está baseado na abordagem da engenharia de documentos, e

- Objeto
  - Corpo
  - Célula
  - Documento
  - Fonte
  - Gráfico
  - Meta
  - Parágrafo
  - Estilo
  - Tabela
  - Texto
- tblDadosPessoais

ObjectProperties assertions	
contém	célula_11_TblEtiquetaTítulo
contém	célula_12_TblEtiquetaTítulo
contém	célula_13_TblEtiquetaTítulo
contém	célula_14_TblEtiquetaTítulo
contém	célula_21_PrgFotoAluno
contém	célula_22_PrgFotoAluno
contém	célula_23_PrgEtqtaNúmero
contém	célula_24_PrgTxtNúmero
contém	célula_31_PrgFotoAluno
contém	célula_32_PrgFotoAluno
contém	célula_33_PrgEtqtaNome
contém	célula_34_PrgTxtNome
contém	célula_41_PrgFotoAluno
contém	célula_42_PrgFotoAluno
contém	célula_43_PrgEtqtaEmail
contém	célula_44_PrgTxtEmail
contém	célula_51_PrgFotoAluno
contém	célula_52_PrgFotoAluno
contém	célula_53_PrgEtqtaDataNac
contém	célula_54_PrgTxtDataNac
contém	célula_61_PrgFotoAluno
contém	célula_62_PrgFotoAluno
contém	célula_63_PrgEtqtaIdentidade
contém	célula_64_PrgTxtIdentidade
contém	célula_71_PrgEtqtaLocalNasc
contém	célula_72_PrgEtqtaLocalNasc
contém	célula_73_PrgTxtLocalNasc
contém	célula_74_PrgTxtLocalNasc
contém	célula_81_PrgEtqtaNacionalidade
contém	célula_82_PrgEtqtaNacionalidade
contém	célula_83_PrgTxtNacionalidade
contém	célula_84_PrgTxtNacionalidade
contém	célula_91_PrgEtqtaGraduação
contém	célula_92_PrgEtqtaGraduação
contém	célula_93_PrgTxtGraduação
contém	célula_94_PrgTxtGraduação
contém	célula_10_1_PrgEtqtaMestrado
contém	célula_10_2_PrgEtqtaMestrado
contém	célula_10_3_PrgTxtMestrado
contém	célula_10_4_PrgTxtMestrado
tem	estiloTblDadosPessoais
DataProperty assertions	
colunas	"4"^^unsignedShort
linhas	"10"^^unsignedShort
tblLargura	"1.0"^^double

Figura 5.17: *Instância da classe Tabela que contém os dados pessoais do aluno* ***Vinicius Rezende Oliveira****.*

está representado em dois modelos independentes, um modelo do conteúdo e um modelo de apresentação das informações contidas no documento.

- O modelo conceitual proposto foi implementado usando ontologias. Nosso modelo é uma Ontologia de Documentos. O modelo foi implementado em base das ontologias principalmente porque elas permitem o compartilhamento de conhecimento possibilitando soluções ao problema de falta de interoperabilidade.

- A ontologia de Documentos está composta por duas ontologias. A Ontologia de Formato implementa o modelo de apresentação das informações do documento e a Ontologia de Conteúdo implementa o modelo de conteúdo.

- As Ontologias de Formato e Conteúdo são independentes entre si mesmas. Desta forma, cada uma poderia ser usada por aplicações diferentes.

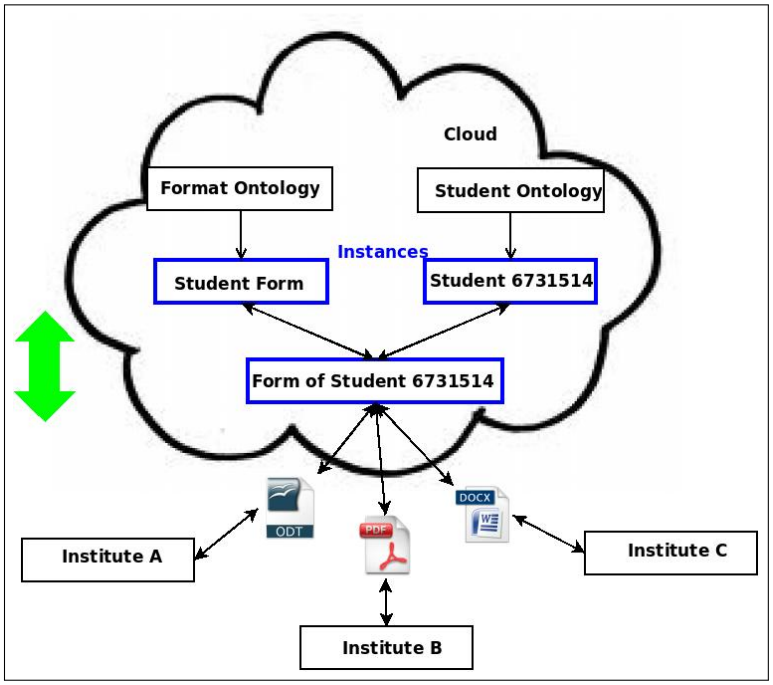


Figura 5.18: *Fluxo completo de Interoperabilidade de Documentos.*

- O objetivo da Ontologia de Formato é possibilitar a portabilidade simples de documentos. No caso da Ontologia de Conteúdo é atingir a interoperabilidade de documentos de um determinado domínio.
- A partir da combinação das Ontologias de Formato e Ontologia de Conteúdo é gerado um objeto que serve de entrada a tradutores que finalmente geram documentos num formato apropriado para o usuário.
- A preservação da autenticidade do documento depende do domínio de contexto a representar. A avaliação das propriedades consideradas significativas dependerá do propósito do documento.

Finalmente no próximo capítulo, apresentamos as conclusões finais e sugestões de trabalhos futuros.

Classe	Descrição
Pessoa	A classe Pessoa é uma classe abstrata. A entidade Pessoa representa a um indivíduo.
Professor	A classe Professor é uma classe concreta. A entidade Professor representa uma Pessoa que ensina alguma Disciplina ou orienta a um estudante.
Estudante	A classe Estudante é uma classe concreta. A entidade Estudante representa uma Pessoa matriculada no Curso que pertence a uma Instituição.
Curso	A classe Curso é uma classe concreta. A entidade Curso representa a um conjunto regras, políticas e Disciplinas que habilitam a um Estudante para um diploma profissional de graduação, mestrado ou doutorado.
Disciplina	A classe Disciplina é uma classe concreta. A entidade Disciplina representa uma matéria de ensino num Curso.
Instituto	A classe Instituto é uma classe concreta. A entidade Instituto representa uma organização destinada ao ensino.
Qualificação	A classe Qualificação é uma classe concreta. A entidade Qualificação representa um exame que avalia o estágio de desenvolvimento do projeto de pesquisa de um Aluno.
Defesa	A classe Defesa é uma classe concreta. A entidade Defesa representa um exame que avalia o trabalho final de um Curso de doutorado ou mestrado.
Comissão	A classe Comissão é uma classe concreta. A entidade Comissão representa a um conjunto de Professores que avaliam uma Defesa ou Qualificação.
Histórico	A classe Histórico é uma classe concreta. A entidade Histórico representa o histórico de ocorrências das atividades do Aluno num Curso.

Tabela 5.1: *Conceitos da Ontologia de Conteúdo.*

Propriedade	Descrição
formadoEm	Um Estudante é formado em um mestrado ou doutorado.
matrículadoEm	Um Estudante está matriculado num Curso.
temComissão	Uma Qualificação ou Defesa tem uma Comissão avaliadora.
fazProva	Um Estudante faz a exame de Qualificação.
faz	Um Estudante defende um Curso de graduação, mestrado ou doutorado.
pertence	Um Curso pertence a uma Instituição.
éOrientadoPor	Um Estudante é orientado por um Professor.
participa	Um Professor participa numa Comissão.
trabalha	Um Professor trabalha numa Instituição de ensino.

Tabela 5.2: *Propriedades da Ontologia de Conteúdo.*

Classe	Descrição
Documento	A classe Documento é uma classe concreta. A entidade Documento representa a um documento digital composto por um Corpo e Meta dados.
Corpo	A classe Corpo é uma classe concreta. A entidade Corpo representa uma contenedor de parágrafos.
Meta	A classe Meta é uma classe concreta. A entidade Meta contém os metadados do documento. Por exemplo, o autor do documento, a identificação da pessoa que modificou por último, a data da última modificação, etc.
Parágrafo	A classe Parágrafo é uma classe concreta. A entidade Parágrafo representa a um contenedor de Texto, imagem ou tabela.
Tabela	A classe Tabela é uma classe concreta. A entidade Tabela representa uma matriz constituída por Células.
Célula	A classe Célula é uma classe concreta. A entidade Célula representa um contenedor de Textos e/ou Gráficos.
Gráfico	A classe Gráfico é uma classe concreta. A entidade Gráfico representa uma imagem digital.
Texto	A classe Texto é uma classe concreta. A entidade Texto define o texto a ser mostrado em um Parágrafo.
Fonte	A classe Fonte é uma classe concreta. A entidade Fonte elige um estilo de apresentação do Texto. Por exemplo seleciona um tipo de fonte tipográfica: *Times New Roman, Arial Black, MS Sans Serif*, etc.
Estilo	A classe Estilo é uma classe abstrata. Estilo por ser uma classe abstrata não é instanciada, mas serve como modelo para as outras classes.
EstiloTexto	A classe EstiloTexto é uma classe concreta. A entidade EstiloTexto define as propriedades estilísticas de um Texto. Por exemplo um tipo de Fonte: *Bold, BoldItalic, Italic, Regular*
EstiloTabela	A classe EstiloTabela é uma classe concreta. A entidade EstiloTabela define as propriedades estilísticas de uma Tabela. Por exemplo: *Width, Height*, etc.
EstiloColunaTabela	A classe EstiloColunaTabela é uma classe concreta. A entidade EstiloColunaTabela define as propriedades estilísticas de um Coluna de Tabela. Por exemplo: *ColumnWidth.*
EstiloCélula	A classe EstiloCélula é uma classe concreta. A entidade EstiloCélula define as propriedades estilísticas de uma célula. Por exemplo define um estilo de borda: *All_Four, None, Bottom, Left, Top*, etc.
EstiloGráfico	A classe EstiloGráfico é uma classe concreta. A entidade EstiloGráfico define as propriedades estilísticas de um Gráfico. Por exemplo define a posição Horizontal do Gráfico: *Center, FromInside, FromLeft, Inside, Left, Outside, Right.*
EstiloParágrafo	A classe EstiloParágrafo é uma classe concreta. A entidade EstiloParágrafo define as propriedades estilísticas de um Parágrafo. Por exemplo define a posição horizontal de um Texto: *Right, Left, Justify, Filled.*

Tabela 5.3: *Conceitos da Ontologia de Formato.*

Propriedade	**Descrição**
tem	Um Documento tem um Corpo e Meta dados.
tem	Um Texto, Gráfico ou Tabela tem um Estilo de apresentação. Um Texto tem um estilo de Fonte.
éCompostoPor	O Corpo do Documento está composto Parágrafos. Uma Célula está composta por Texto ou Gráficos.
exibeConteúdo	Um Parágrafo exibe o conteúdo de um Texto, Gráfico ou Tabela.
contém	Uma Tabela contém Células.

Tabela 5.4: *Propriedades da Ontologia de Formato.*

Capítulo 6

Conclusões

Este capítulo resume as conclusões e contribuições deste trabalho.

6.1 Considerações Finais

- A interoperabilidade e portabilidade de documentos podem ser atingidas usando uma Ontologia de Documento.
- A Ontologia de Documento está composta por duas ontologias: uma Ontologia de Conteúdo e uma Ontologia de Formato.
- A Ontologia de Conteúdo representa a informação de domínio, neste trabalho a Ontologia de Conteúdo caracterizou a informação de um aluno estudante de uma universidade.
- A Ontologia de Formato representa a estrutura de apresentação de componentes de um documento.

6.2 Contribuições desse trabalho

- Relacionar os princípios da engenharia de documentos e a teoria dos formatos padrões de documentos digitais para propor um modelo conceitual de intercambio de documentos.
- A principal contribuição do trabalho foi propor e implementar um modelo conceitual que possibilite a interoperabilidade e a portabilidade de documentos sobre um determinado domínio.

6.3 Sugestões para Pesquisas Futuras

A criação das instâncias da Ontologia de Formato e Conteúdo é trabalhoso e muitas vezes repetitivo. Um trabalho futuro pode ser a automatização da captura da informação dos documentos digitais. Torres (2012) trabalha sobre geração automática de ontologias a partir de texto. Seguindo essa linha, outro trabalho futuro pode ser automatizar o processo da combinação das ontologias para geração do documento final.

Como trabalho futuro, poderá ser desenvolvido um caso de uso de interoperabilidade de documentos aplicada ao governo eletrônico, em que a Ontologia de Documentos tenha papel comprovado na preservação e distribuição eficientes de documentos digitais.

Apêndice A

Documentos de prova

Os documentos gerados apesar de compartilhar o mesmo modelo conceitual diferem levemente na visual do documento. Por exemplo, nas Figura A.1 e Figura A.2 a imagem não mantém o mesmo tamanho.

Para o caso das Figura A.3 e Figura A.4, a tabela do documento gerado em *ODT* perde algumas linhas do lado direito. Os documentos também apresentam o problema do tamanho da imagem. Para ambos casos, o conteúdo da informação das alunas não sofreu perdida.

Student's Information Form

Personal Information

Name	Marina Santana Perez		
Age	23	Sex	F
Institute	University Institute of Technology		
Father's Name	Julio Perez		
Permanent Address	Street Los Angeles 123		
City	Los Angeles	Phone	94735344754

Figura A.1: *Versão do documento gerado para DOCX*

Student's Information Form
Personal Information

Name	Marina Santana Perez		
Age	23	Sex	F
Institute	University Institute of Technology		
Father's Name	Julio Perez		
Permanent Address	Street Los Angeles 123		
City	Los Angeles	Phone	9473534 4754

Figura A.2: *Versão do documento gerado para ODT*

University Institute of Technology, RGPV

Student Academic Detail			
Enrollment No	0101IT10101		
Name	Marina Santana Perez		
Institute	University Institute of Technology		
Area	Networks		
Admission Date	25/09/2010	Course	Information Technology
Email	mariana_perez@gmail.com	Mobile Number	93745243

Student Personal Detail			
Father's Name	Julio Perez	Mother's name	Amy Santana
Sex	F	Date of Birth	04/08/1983
Category		Religion	Catholic
Address	Street Los Angeles 123		
State	California	City	Los Angeles
Postal Code	03847	Telephone Number	84847463

Figura A.3: *Versão do documento gerado para DOCX*

University Institute of Technology, RGPV

Student Academic Detail			
Enrollment No	0101IT10101		
Name	Marina Santana Perez		
Institute	University Institute of Technology		
Area	Networks		
Admission Date	25/09/2010	Course	Information Technology
Email	mariana_perez@ gmail.com	Mobile Number	93745243

Student Personal Detail			
Father's Name	Julio Perez	Mother's name	Amy Santana
Sex	F	Date of Birth	04/08/1983
Category		Religion	Catholic
Address	Street Los Angeles 123		
State	California	City	Los Angeles
Postal Code	03847	Telephone Number	84847463

Figura A.4: *Versão do documento gerado para ODT*

Referências Bibliográficas

ACM (2001) ACM. The acm symposium on document engineering. http://www.documentengineering.org/, 2001. Association for Computing Machinery (ACM) - Último acesso em 16/08/2012. Citado na pág. 6

Arellano (2004) Miguel Angel Arellano. Preservação dos documentos digitais. *Departmento de Ciências da Informação Universidade de Brasilia*, 33:15–17. Citado na pág. 1

Aroyo e Dicheva (2002) Lora Aroyo e Darina Dicheva. Summary of workshop on concepts and ontologies in web-based educational systems. International Conference on Computers in Education, Dec 2002. Último acesso em 01/03/2013. Citado na pág. 27

Barron (1996) David W. Barron. Portable documents: problems and partial solutions. *Department of Electronics and Computer Science University of Southampton*, 8:343–367. Citado na pág. 6

Bechhofer *et al.* **(2004)** Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider e Lynn Andrea Stein. Owl web ontology language. http://www.w3.org/TR/owl-ref/, Feb 2004. Último acesso em 15/08/2012. Citado na pág. 28

Berners-Lee *et al.* **(2001)** Tim Berners-Lee, James Hendler e Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43. URL http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21. Citado na pág. 27

Bosak *et al.* **(2011)** Jon Bosak, Tim McGrath e G. Ken Holman. Universal business language v2.1. Organization for the Advancement of Structured Information Standards (OASIS), Standard, May 2011. Citado na pág. 3

Breitman *et al.* **(2007)** Karin Koogan Breitman, Marco Antonio Casanova e Walter Truszkowski. Ontology in computer science. Em *Semantic Web: Concepts, Technologies and Applications*, NASA Monographs in Systems and Software Engineering, páginas 17–34. Springer London. ISBN 978-1-84628-710-7. Citado na pág. 24

Chede (2008) Cezar Taurion Chede. Padrões abertos, interoperabilidade e interesse público. Em *PoliTICs*, páginas 29–35. PoliTICs. Citado na pág. 15

Chede (2009) Cezar Taurion Chede. Adotando o odf como padrão aberto de documento, 2009. Citado na pág. 6, 7

Clark (2013) Peter Clark. Some ongoing kbs/ontology projects and groups. http://www.cs.utexas.edu/users/mfkb/related.html, 2013. Último acesso em 28/02/2013. Citado na pág. 26

da Camara Junior (2013) Auto Tavares da Camara Junior. *Processamento de Linguagem Natural para Indexação Automática Semântico-Ontológica*. Tese de Doutorado, Universidade de Brasília. Citado na pág. 27, 31

da Silva *et al.* **(2009)** Daniela Lucas da Silva, Eliana Antonia Demarques, Renato Rocha Souza e Gercina A. B. de Oliveira Lima. Ontologias e unified modeling language: uma abordagem para

representação de domínios de conhecimento. *Revista de Ciência da Informação v.10 n.5*. Citado na pág. 33

Ditch (2007) Walter Ditch. *XML-based Office Document Standars*. JISC: Bristol, UK, United Kingdom. Citado na pág. 6

e-Ping (2010) e-Ping. O que é interoperabilidade? http://www.governoeletronico.gov.br/acoes-e-projetos/e-ping-padroes-de-interoperabilidade/o-que-e-interoperabilidade, 2010. Último acesso em 16/08/2012. Citado na pág. 6

Eckert *et al.* **(2009)** Klaus-Peter Eckert, Jan Ziesing e Ucheoma Ishionwu. *Document Operabilitiy Open Document Format and Office Open XML*. Fraunhofer Verlag, Germany, fokus ed. ISBN 978-3-8396-0047-4. Citado na pág. 3

Ferreira (2006) Miguel Ferreira. *Introdução e preservasão digital: Conceitos, estratégias e actuais consensos*. Escola de Engenharia da Universidade do Minho, edição electrônica ed. ISBN 978-972-8692-30-8. Citado na pág. 6

Glushko e McGrath (2008) Robert J. Glushko e Tim McGrath. *Document Engineering: Analyzing and Designing, Documents for Business, Informatics and Web Services*. Massachusetts Institute of Technology. ISBN 0-262-07261-0. Citado na pág. ix, xi, 5, 9, 13

Gruber (1993) Tom R. Gruber. A translation approach to portable ontology specification. Em *Knowlegde Acquisition*, páginas 199–220. Citado na pág. 23

Guarino e Giaretta (1995) Nicola Guarino e Pierdaniele Giaretta. Ontologies and knowledge bases: Towards a terminological clarification. *Towards Very Large Knowledge Bases*. Citado na pág. 23

Guarino *et al.* **(1999)** Nicola Guarino, Claudio Masolo e Guido Vetere. Ontoseek: Content-based access to the web. *IEEE Intelligent Systems*, 14(3):70–80. ISSN 1541-1672. Citado na pág. 26

Guizzardi (2000) GIANCARLO Guizzardi. Desenvolvimento para e com reuso: Um estudo de caso no domínio de vídeo sob demanda. Dissertação de Mestrado, Universidade Federal do Espírito Santo, Centro Tecnológico. Citado na pág. 26

Gómez-Pérez *et al.* **(2004a)** Asunción Gómez-Pérez, Mariano Fernández-López e Oscar Corcho. Theoretical foundations of ontologies. Em *Ontological Engineering*, Advanced Information and Knowledge Processing, páginas 1–45. Springer Berlin Heidelberg. ISBN 978-1-85233-840-4. Citado na pág. 24

Gómez-Pérez *et al.* **(2004b)** Asunción Gómez-Pérez, Mariano Fernández-López e Oscar Corcho. Ontology tools. Em *Ontological Engineering*, Advanced Information and Knowledge Processing, páginas 293–362. Springer Berlin Heidelberg. ISBN 978-1-85233-840-4. Citado na pág. 31

Heminger e Robertson (1998) Alan R. Heminger e Steven B. Robertson. Digital rosetta stone: A conceptual model for maintaining long-term access to digital documents. Em *HICSS (2)*, páginas 158–167. Citado na pág. 2

Hiser (2007) Sam Hiser. Achieving openness: a closer look at ODF & OOXML, 2007. Citado na pág. 18

KSL (1994) KSL. Resource description framework. www-ksl.stanford.edu/knowledge-sharing/ontologies/html/frame-ontology, Jul 1994. Knowledge Systems Laboratory (KSL) - Último acesso em 15/08/2012. Citado na pág. 24

KSL (1995) KSL. Open knowledge base connectivity. http://www.ai.sri.com/~okbc/, 1995. Knowledge Systems Laboratory (KSL) - Último acesso em 15/08/2012. Citado na pág. 24

KSL (2008) KSL. Ontolingua. http://www.ksl.stanford.edu/software/ontolingua/, Jan 2008. Knowledge Systems Laboratory (KSL) - Último acesso em 15/08/2012. Citado na pág. 24

Labrou e Finin (1999) Yannis Labrou e Tim Finin. Yahoo! as an ontology: using yahoo! categories to describe documents. Em *Proceedings of the eighth international conference on Information and knowledge management*, CIKM '99, páginas 180–187, New York, NY, USA. ACM. ISBN 1-58113-146-1. Citado na pág. 25

Lassila e Mcguinness (2001) Ora Lassila e Deborah Mcguinness. The role of frame-based representation on the semantic web. Relatório técnico, Knowledge Systems Laboratory - Stanford University. Citado na pág. 25

Maedche e Staab (2001) A. Maedche e S. Staab. Ontology learning for the semantic web. *Intelligent Systems, IEEE*, 16(2):72 – 79. ISSN 1541-1672. Citado na pág. 24

Ngo (2008) Tom Ngo. Visão geral do office open xml. Relatório técnico, ECMA TC45. Citado na pág. ix, 5, 17, 18

Noy *et al.* **(2000)** Natalya Fridman Noy, Ray W. Fergerson e Mark A. Musen. The knowledge model of protégé-2000: Combining interoperability and flexibility. *Lecture Notes in Computer Science*, 1937:69–82. Citado na pág. 31

RDF Working Group (2004) RDF Working Group. Resource description framework. http://www.w3.org/RDF/, Oct 2004. Último acesso em 15/08/2012. Citado na pág. 27

Rocha *et al.* **(2004)** Claudia Lacombe Rocha, Márcia Helena de Carvalho Ramos e Margareth da Silva e Rosely Cury Rondinelli e Revisão por Alba Gisele Gouget. *Gestão Arquivística de Documentos Eletrônicos*. Cámara Técnica de Documentos Eletrônicos - Conselho Nacional de Arquivos(CONARQ), Rio Janeiro. ISBN 85-225-0396-6. Citado na pág. 5

Ruesta (2008) Carlota Bustelo Ruesta. Documentos electrónicos: Reflexiones sobre las relaciones con la tecnología de los profesionales de la información. parte i. *Anuario ThinkEPI*. Citado na pág. 15

Schmidt *et al.* **(2006)** Kay-Uwe Schmidt, Oliver Fox, Lutz Henckel, Uwe Holzmann-Kaiser, Philipp Martin e Michael Tschichholz. *Document Interoperability for Use in eGovernment. Integration of XML-based Document Content in Public Administration Processes*. Fraunhofer Verlag, fokus ed. Citado na pág. 6

Schroeder e Mello (2009) Rebeca Schroeder e Ronaldo Dos Santos Mello. Designing xml documents from conceptual schemas and workload information. *Multimedia Tools Appl.*, 43(3): 303–326. ISSN 1380-7501. Citado na pág. 2

Shah e Kesan (2009) Rajiv Shah e Jay Kesan. Interoperability challenge for open standars: Odf and ooxml as examples. *The proceedings of the 10th International Digital Government Research Conference*. Citado na pág. xi, 18, 20

Sugumaran e Storey (2002) Vijayan Sugumaran e Veda C. Storey. Ontologies for conceptual modeling: their creation, use, and management. *Data Knowl. Eng.*, 42(3):251–271. Citado na pág. 31

Torres (2012) Carlos Eduardo Atencio Torres. Uso de informação linguística e análise de conceitos formais no aprendizado de ontologias. Dissertação de Mestrado, Instituto de Matemática é Estatística - Universidade de São Paulo. Citado na pág. 47

Uschold (1998) Mike Uschold. Knowledge level modelling: concepts and terminology. *The Knowledge Engineering Review*, 13:1:5–29. Printed in the United Kingdom. Citado na pág. 24

Uschold e Gruninger (1996) Mike Uschold e Michael Gruninger. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11. Citado na pág. 24

Vogel *et al.* **(2008)** Tobias Vogel, Alexander Schmidt, Alexander Lemm e Hubert Österle. Service and document based interoperability for european ecustoms solutions. *J. Theor. Appl. Electron. Commer. Res.*, 3(3):17–37. ISSN 0718-1876. Citado na pág. 2