

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Informática

Modelagem e Descrição Formal das Fontes do
Direito como Infraestrutura de um Sistema de
Gerenciamento de Conhecimento Legal

Halley Ferreira Solano de Freitas

Dissertação submetida à Coordenação do Curso de Pós-Graduação em
Ciência da Computação da Universidade Federal de Campina Grande -
Campus I como parte dos requisitos necessários para obtenção do grau
de Mestre em Ciência da Computação.

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Sistemas de Informação e Bancos de Dados

Dr. Ulrich Schiel
(Orientador)

Campina Grande, Paraíba, Brasil

©Halley Ferreira Solano de Freitas, 6 de Agosto de 2010

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

F866m Freitas, Halley Ferreira Solano de.
Modelagem e descrição formal das fontes do direito como infraestrutura de um sistema de gerenciamento de conhecimento legal /Halley Ferreira Solano de Freitas. — Campina Grande, 2010.
139 f.: il. col.

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática.
Orientador: Prof. Dr. Ulrich Schiel.
Referências.

1. Sistemas de Gerenciamento de Bases de Dados. 2. Fontes do Direito Positivo. 3. Modelagem Textual Temporal. I. Título.

CDU 04.65:340.130(043)

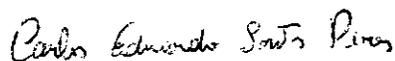
**"MODELAGEM E DESCRIÇÃO FORMAL DAS FONTES DO DIREITO COMO
INFRAESTRUTURA DE UM SISTEMA DE GERENCIAMENTO DE CONHECIMENTO
LEGAL"**

HALLEY FERREIRA SOLANO DE FREITAS

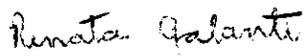
DISSERTAÇÃO APROVADA EM 17.06.2010



**ULRICH SCHIEL, Dr.
Orientador(a)**



**CARLOS EDUARDO SANTOS PIRES, Dr.
Examinador(a)**



**RENATA DE MATOS GALANTE, Drª
Examinador(a)**

CAMPINA GRANDE - PB

*Quando se quer fazer algo, arruma-se um jeito.
Quando não se quer, arruma-se uma desculpa.*

(Provérbio Árabe)

Resumo

Dentre as muitas áreas de conhecimento que utilizam a Ciência da Computação como meio para otimizar sua aplicação, este trabalho explora a interdisciplinaridade entre a Computação e o Direito. Os operadores do Direito utilizam diversas fontes na pesquisa normativo-jurisprudencial, dentre as principais, a legislação e os documentos jurídicos. Este trabalho propõe auxiliar os especialistas que lidam com essas fontes, através de um Sistema de Gerenciamento de Conhecimento Legal (SGCL). Por se tratarem de fontes ricas em informação - tanto em sua estrutura quanto na semântica do seu conteúdo - um SGCL é capaz de extrair esse conhecimento e processá-lo, poupando tempo e esforço humano. Como parte fundamental de sistemas dessa natureza, foi desenvolvido um modelo das fontes do Direito, a partir da especificação de um modelo genérico textual-temporal, com ênfase na descrição estrutural, temporal e interrelações dos dispositivos brasileiros. Operações avançadas para automatizar tarefas realizadas pelos operadores também foram descritas. Além da modelagem, os aspectos estruturais dos documentos legais e jurídicos foram formalizados através da Forma Estendida de Backus-Naur, visando a uma posterior Extração de Informação, executada pelo sistema gestor do conhecimento legal. A avaliação do modelo foi feita a partir da implementação de um protótipo, utilizando como estudo de caso as normas brasileiras descritas no artigo 59 da Constituição Federal de 1988 e acórdãos provenientes do Superior Tribunal Federal e Superior Tribunal de Justiça. Os documentos do estudo de caso também foram base para a descrição formal das fontes do Direito.

Abstract

Among knowledge areas that use scientific computing to optimize their applications, this work explores the interdisciplinarity between Law and Computer Science (CS). Law specialists use several sources during their normative and jurisprudential research, especially legislation and judgement documents. The present study aims to help Law specialists to deal with those sources, proposing a Legal Knowledge Management System (LKMS). As these sources are rich in structural and semantical information, a SGCL can save human time and effort by extracting and processing this knowledge. As a fundamental part of it, a model of Law sources was developed, based on a generic textual-temporal model, emphasizing structural, temporal and interrelational description of brazilian sources. Advanced operations to automate specialists' tasks were also described. Besides modeling, structural aspects of legal and judgement documents were also formalized with the Extended Backus-Naur Form, aiming a posterior Information Extraction, to be performed by the LKMS. The model evaluation was made with a prototype, using brazilian norms described in article 59 of Federal Constitution of 1988 and judgement documents of Federal Superior Court and Justice Superior Court. The formal description of Law sources were also based on those documents.

Agradecimentos

Por terem sido fundamentais na realização deste trabalho, agradeço:

Primeiramente a Deus, pela saúde, paz e força para concluir essa longa jornada.

À minha família. Meus pais, Harlan e Moyra, pelo apoio incondicional em todos os momentos de minha vida e pelo exemplo diário de como é possível vencer na vida de forma honesta, através de muito esforço e dedicação. Aos meus irmãos Michel e Mayra, pelo convívio harmônico e momentos de descontração.

À minha namorada Lalinha, a qual esteve presente e acompanhou meus passos desde o início. Pelos momentos ímpares de amor e cumplicidade. Por ser um dos principais motivos de minha alegria e vontade de continuar sempre progredindo.

Ao meu orientador Ulrich, pelos ensinamentos e paciência em quase cinco anos de trabalho juntos. Agradeço por ter me dado o que às vezes é tudo que alguém precisa: uma oportunidade. E não foi apenas uma, mas várias ao longo de todo esse tempo, as quais me esforcei sem descanso para aproveitá-las da melhor forma possível.

Aos amigos que fiz durante o mestrado, por terem tornado mais divertida a minha estada em Campina Grande. Aos que conviveram comigo: Marcos Henrique (Gaúcho), Guilherme (Guiga), Danilo (Mãozinha), Rafael e Tomás. Aos membros da QI: Yuri (Crato), Danillo (Bob), Saulo, Gilson (Negola), Felipe (Farinha), Carlos (Zéo), Flávio (Barata), Welflen (Boss), Taciano, Bruno Alexandre e Pablo. Aos colegas de laboratório Hugo e Tiago.

A todos os funcionários e professores que compõem a Pós-Graduação em Ciência da Computação da UFCG, em especial a Ana Lúcia e Vera Lúcia.

À CAPES pelo apoio financeiro.

Conteúdo

1	Introdução	1
1.1	Descrição do Problema	3
1.2	Relevância	5
1.3	Objetivos	7
1.3.1	Objetivos Específicos	7
1.4	Metodologia	7
1.5	Estrutura da Dissertação	8
2	Fundamentação Teórica	9
2.1	Direito	9
2.1.1	Norma Jurídica	10
2.1.2	Jurisprudência	16
2.1.3	Doutrina	18
2.2	Banco de Dados Temporal	18
2.2.1	BDS X BDT	18
2.2.2	Tempo válido e tempo transacional	19
2.3	Extração de Informação	20
2.3.1	Fases	21
2.3.2	Métricas de Avaliação	23
2.4	Forma Estendida de Backus-Naur	24
2.4.1	Metalinguagem Sintática	24
2.4.2	Sintaxe	25
2.4.3	Exemplos	26
2.5	Sistema de Gerenciamento de Conhecimento Legal	28
2.5.1	Aquisição	28

2.5.2	Modelagem e Armazenamento	28
2.5.3	Recuperação e Disseminação	29
2.5.4	Manutenção	29
2.6	Considerações Finais	30
3	Trabalhos Relacionados	31
3.1	Modelagem e SGCLs	32
3.1.1	Modelagem Normativo-Jurídica	32
3.1.2	Dinâmica das Fontes	42
3.1.3	Modelos e SGCLs Pátrios	50
3.2	Descrição Formal das Fontes	63
3.3	Análise Comparativa	66
3.4	Considerações Finais	71
4	Modelo ITMJudLaw	72
4.1	TTMDoc	72
4.1.1	Estrutura	73
4.1.2	Dinâmica	74
4.2	Visão geral	75
4.2.1	Fontes	77
4.2.2	Referências	81
4.3	Operações	82
4.3.1	Operações de Manutenção	82
4.3.2	Operações para o Usuário Final	84
4.4	Avaliação	89
4.5	Considerações Finais	89
5	Descrição Formal das Fontes do Direito	90
5.1	Dispositivos Legais	91
5.2	Documentos Jurídicos	95
5.2.1	STF	95
5.2.2	STJ	96
5.2.3	Referências	97

5.3	Avaliação	99
5.4	Considerações Finais	103
6	JudLaw	104
6.1	Arquitetura	104
6.1.1	Coleta de Dados	106
6.1.2	Extração de Informação	109
6.1.3	Recuperação de Informação	109
6.1.4	Interface	110
6.2	Protótipo	111
6.3	Considerações Finais	113
7	Considerações Finais e Trabalhos Futuros	114
7.1	Considerações Finais	114
7.2	Trabalhos Futuros	116
A	Fontes do Estudo de Caso	127
B	Aplicação do Modelo ITMJudLaw	129
C	Descrição Formal Completa das Fontes	134

Lista de Símbolos

AN - *AKOMA NTOSO*

BDS - *Banco de Dados Snapshot*

BDT - *Banco de Dados Temporal*

CCD - *Ciência da Computação e Direito*

CDD - *Change Description Document*

CML - *CEN MetaLex*

DTD - *Document Type Definition*

EBNF - *Forma Estendida de Backus-Naur*

EI - *Extração de Informação*

GCL - *Gerenciamento de Conhecimento Legal*

IAD - *Inteligência Artificial e Direito*

ITMJUDLAW - *Integrated Temporal Model JudLaw*

LCP107 - *Lei Complementar Nº 107, de 26 de abril de 2001*

LCP95 - *Lei Complementar Nº 95, de 26 de Fevereiro de 1998*

LEXMLBR - *LexMLBR*

MS - *Metalinguagem Sintática*

OWL - *Web Ontology Language*

PNJ - *Pesquisa Normativo-Jurisprudencial*

RBC - *Raciocínio Baseado em Casos*

RDF - *Resource Description Framework*

STF - *Superior Tribunal Federal*

STJ - *Superior Tribunal de Justiça*

XML - *Extensible Mark-up Language*

Lista de Figuras

1.1	Modelo mental do operador do Direito	4
2.1	Desdobramentos do artigo	13
2.2	Agrupadores do elemento artigo	14
3.1	Ontologia de domínio, relativa ao acórdão	37
3.2	Esquema XML de uma norma	44
3.3	Rede semântica simplificada de citações	49
3.4	Metadados do AKOMA NTOSO	54
3.5	Linha de tempo de uma norma	56
3.6	Geração de fragmentos através do CDD	57
3.7	Art 1º da CF/88 em XML	61
3.8	Gramática das expressões de alterações em EBNF	66
3.9	Tabela Comparativa 1 - Trabalhos Nacionais	70
3.10	Tabela Comparativa 2 - SGCLs Pátrios	70
4.1	Modelo ITMJudLaw.	76
4.2	Navegação estrutural e temporal nos elementos da norma	85
5.1	Interpretador para documentos do STF	100
5.2	Extração de atributos de um acórdão do STF - Parte 1	101
5.3	Extração de atributos de um acórdão do STF - Parte 2	102
5.4	Extração de atributos de um acórdão do STF - Parte 3	102
6.1	Arquitetura do JudLaw	105
6.2	Esquema do motor de busca remota	107
6.3	Arquivo de configuração do sítio do STJ	108

6.4 Protótipo da interface gráfica do módulo CD 112

Lista de Tabelas

2.1	Tipos de normas e suas definições	11
2.2	Atualização de informações armazenadas em um BDT	19
2.3	Preenchimento dos campos da epígrafe	23
2.4	Símbolos do EBNF	26
3.1	Estrutura de um acórdão	34
3.2	Localização dos atributos de um acórdão	37
3.3	Tabela de modificações nas normas da Tasmânia	47

Lista de Códigos Fonte

3.1	Descrição das partes iniciais e finais de uma norma em DTD	38
3.2	XML resultante da descrição em DTD	39
3.3	Descrição da parte normativa de uma norma em DTD	39
3.4	Elemento de referência <refrecord>	40
3.5	LCP107 altera o inciso II da LCP95	40
3.6	LCP107 inclui o parágrafo 1º no artigo 12 da LCP95	41
3.7	Reconstrução temporal em XQuery	45
3.8	Consulta temporal em XQuery	45
4.1	Revogação recursiva de elementos da norma	83
4.2	Modificação de um elemento da norma	83
4.3	Reconstrução temporal da norma	85
4.4	Validação recursiva de elementos a partir de uma raiz	86
4.5	Validação das referências	86
4.6	Verificação da consistência temporal original de citações feitas por documentos jurídicos	87
4.7	Verificação da consistência temporal expandida de citações feitas por documentos jurídicos	88

Capítulo 1

Introdução

É notório o crescimento do volume de informações a serem processadas pelos seres humanos, nas mais diversas áreas de conhecimento. Felizmente, com o advento dos computadores, está sendo possível transferir grande parte desse trabalho para as máquinas, sem as quais seria praticamente inviável realizar uma vasta gama de operações: cálculos avançados, buscas envolvendo milhões de dados, processamento de som e imagem, entre outras. Dentre as muitas áreas de conhecimento que utilizam a Ciência da Computação como meio para otimizar sua aplicação, este trabalho explora sua interdisciplinaridade com o Direito.

A aplicação da Computação no campo do Direito teve início de forma bastante simples, através da transição do papel para o meio digital. Na segunda metade da década de 90, mais de cinquenta países já possuíam textos relativos às respectivas legislações publicados na Internet, dentre eles: Estados Unidos, Austrália, Noruega, Alemanha, Reino Unido, França, Espanha, Turquia, Índia, México e África do Sul [35]. Em decisão recente¹, a Suprema Corte de Delhi decidiu dar início ao projeto de se tornar a primeira *e-corte* do mundo, abolindo totalmente o uso do papel. Todas as ações dos juízes ao analisarem os processos serão feitas via computador e os documentos gerados pelo Corte (provas, acórdãos e despachos) serão documentados e arquivados digitalmente.

No Brasil, a própria Constituição Federal de 1998, artigo 5º, inciso XIV, define que “é assegurado a todos o acesso à informação e resguardado o sigilo da fonte, quando necessário ao exercício profissional”. A fim de assegurar esse direito, atualmente os três poderes - nas três esferas - já possuem sítios onde são publicados documentos referentes a informações legislativas e jurídicas [50]. Dentre os principais, podem ser citados: Presidência da República do

¹<http://www.conjur.com.br/2009-dez-02/suprema-corte-indiana-abolir-totalmente-uso-papel-anos> (Acesso em Junho/2010)

Brasil², Senado Federal³, Superior Tribunal Federal⁴, Superior Tribunal de Justiça⁵ e alguns Tribunais Estaduais - a exemplo do TJ-RS⁶ e TJ-SC⁷.

Após a transição das fontes do Direito para o meio digital, percebeu-se a riqueza de informações das quais elas são dotadas, tanto em sua estrutura quanto em seu conteúdo. Pesquisadores começaram a estudar quais e como os paradigmas da Computação poderiam ser aplicados a esses documentos, dando origem a uma nova área interdisciplinar intitulada *Computer Science and Law*, ou *Ciência da Computação e Direito* (CCD). Um dos principais centros de referência na área é o *Leibniz Center for Law*⁸ da Universidade de Amsterdã. Dentre as subáreas da CCD, Riveret [64] destaca duas: *Gerenciamento de Conhecimento Legal e Inteligência Artificial e Direito*.

O *Gerenciamento de Conhecimento Legal* (GCL) trata da aquisição, modelagem, distribuição e manutenção do conhecimento legal. A principal entidade interessada na área é a *JURIX: The Foundation for Legal Knowledge Based Systems*, que já conta com mais de vinte anos de experiência em conferências sobre o assunto. A área de *Inteligência Artificial & Direito* (IAD) tem como objetivo o desenvolvimento de ferramentas inteligentes que simulem o raciocínio jurídico. A principal conferência na área é a *International Conference on Artificial Intelligence and Law (ICAAIL)*, promovida pela *Associação Internacional de Inteligência Artificial & Direito*. O presente trabalho tem como área de interesse o **GCL**.

Através do gerenciamento do conhecimento legal, o interesse dos pesquisadores não é apenas publicar as fontes do Direito, mas também representá-las através de modelos computacionais, visando ao seu armazenamento estruturado, requisito fundamental para a realização de consultas e operações avançadas [30]. Governos - principalmente europeus - estão investindo na idéia, com do desenvolvimento de *modelos/esquemas* e *Sistemas de Gerenciamento de Conhecimento Legal* (SGCLs) [50]: *Akoma Ntoso* [1] (África), *eLaw* (Áustria), *LexDania* [47] (Dinamarca), *BWB SDU* (Holanda), *Norme in Rete* [56] (Itália), *CHLexML* [63] (Suíça), *EnAct* [7] (Tasmânia, Austrália), *LexML Brasil* [50] e *CEN Meta-Lex* [19][21] (Europa).

²<http://www.presidencia.gov.br> (Acesso em Junho/2010)

³<http://www.senado.gov.br> (Acesso em Junho/2010)

⁴<http://www.stf.jus.br> (Acesso em Junho/2010)

⁵<http://www.stj.gov.br> (Acesso em Junho/2010)

⁶<http://www.tjrs.jus.br> (Acesso em Junho/2010)

⁷<http://www.tjsc.jus.br> (Acesso em Junho/2010)

⁸<http://www.leibnizcenter.org> (Acesso em Junho/2010)

Apesar da proposta de alguns desses trabalhos ser a extensão das fontes do Direito ao alcance de todos, é indiscutível que o principal público-alvo ainda são os especialistas na área. Estes lidam diretamente com as fontes no âmbito profissional, utilizando-as de acordo com o interesse envolvido, em suas pesquisas normativas e jurisprudenciais. No Brasil, esses especialistas são conhecidos como *operadores do Direito*⁹, a exemplo de advogados, juízes, desembargadores, promotores e até mesmo estudantes de Direito, estes requerendo as fontes para a produção de trabalhos acadêmicos.

A *pesquisa normativo-jurisprudencial* (PNJ) constitui um processo fundamental no âmbito laboral de todo operador do Direito. Após tomar conhecimento do caso a ser resolvido, o operador realiza essa pesquisa para fundamentar o seu ponto de vista acerca da matéria, seja o advogado para preparar a defesa do seu cliente, o juiz para proferir uma sentença, o desembargador para definir seu voto, etc. Como o próprio nome sugere, a parte *normativa* da pesquisa está relacionada às fontes normativas (legislativas), enquanto a parte *jurisprudencial* diz respeito a decisões proferidas dos magistrados ou tribunais acerca de casos julgados.

Ainda há uma terceira fonte, conhecida como *doutrina*. Esta corresponde ao posicionamento de estudiosos do Direito (doutrinadores) acerca de uma matéria, geralmente encontrada em livros especializados relativos às subáreas do Direito. Apesar do foco deste trabalho ser as duas primeiras fontes supracitadas, a *doutrina* também será contemplada, por ser utilizada por alguns operadores em suas pesquisas.

1.1 Descrição do Problema

Em um cenário ideal, os operadores do Direito teriam acesso à informação de forma centralizada em uma ferramenta, na qual todas as fontes estariam armazenadas de forma estruturada e interrelacionada, permitindo a realização de operações avançadas sobre elas. No entanto, a realidade é bem diferente. O processo de *PNJ* requer tempo e esforço do operador, pois via de regra as fontes estão espalhadas, muitas vezes nem mesmo em formato digital, gerando a primeira problemática apresentada por Francesconi & Peruginelli [38]: “como coletar esses dados?”

⁹Há uma discussão entre os doutrinadores acerca da corretude do termo, mas para fins deste trabalho, ele será empregado devido à sua ampla utilização.

Além de dispersas, a única operação disponível na maioria das ferramentas que lidam com esse tipo de informação é a busca simples, por palavras-chave. Nesse contexto, é retornado um número significativo de informações não desejadas, por não haver um refinamento em sua recuperação, como por exemplo definir o grau de relevância dos documentos para o interesse do operador [24].

As fontes do Direito possuem ainda uma característica peculiar, que dificulta seu processamento: um alto índice de referências a outras fontes. Como elucida Sutton [70], a pesquisa jurisprudencial pode se tornar bastante complexa, devido às suas interrelações referenciais com outros documentos - relativos a casos semelhantes e normas envolvidas -, formando uma verdadeira “rede semântica de citações” [76] a outras fontes, a serem processadas pelo operador:

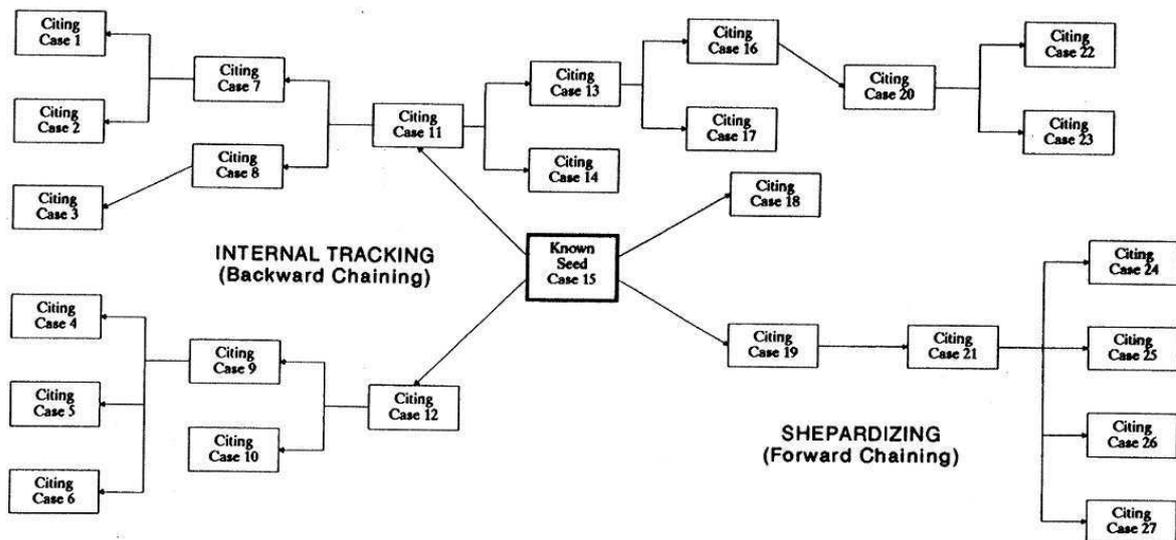


Figura 1.1: Modelo mental do operador do Direito (Sutton [70])

A Figura 1.1 mostra apenas parte do problema, visto que não foram consideradas as referências normativas. A maioria dos repositórios de fontes do Direito nem sequer leva em consideração essa **integração**, provendo apenas buscas a documentos de um tipo. Isso significa trabalho extra para os operadores, que precisam acessar diversas fontes em busca da informação desejada. Ao se referenciar normas jurídicas, a terceira dimensão precisa ser levada em consideração: **tempo**. Essas fontes têm a característica de serem alteradas constantemente, seja por modificações, inclusões ou revogações.

Para que a fundamentação de um operador do Direito seja juridicamente consistente, é

necessário que ele esteja atento à *vigência* das normas associadas ao caso. Como supracitado, o documento jurídico é composto de referências a vários tipos de fontes, dentre as principais, normas e outros documentos jurídicos. Caso o operador tenha interesse em utilizar como base para sua fundamentação a decisão contida em um documento jurídico, ele precisa se assegurar que as fontes referenciadas nessa decisão estejam em consonância com o atual ordenamento jurídico.

Um exemplo prático disso é um juiz querer fundamentar uma sentença acerca de uma matéria a partir do entendimento advindo de acórdão, emitido pelo Superior Tribunal Federal em relação a um caso semelhante. Porém, um dos votos do acórdão se baseou em uma lei que foi revogada ou ainda está em período de *vacatio legis*. Nesse contexto, caso o juiz fundamente seu ponto a partir desse voto, provavelmente um recurso será impetrado pela parte prejudicada alegando tal situação. Averiguado o erro em uma instância superior, a decisão do juiz provavelmente será desconsiderada.

Em resumo, os operadores do Direito precisam processar **informações complexas** (estruturalmente e semanticamente) que se encontram em **fontes espalhadas e não-relacionadas**, tornando a pesquisa normativo-jurisprudencial uma tarefa dispendiosa.

1.2 Relevância

As ferramentas que se dispõem a resolver os problemas supracitados são conhecidas como *Sistemas de Gerenciamento de Conhecimento Legal* (SGCLs). O desenvolvimento de sistemas desse porte, contudo, não é trivial. São necessárias várias etapas, com a primeira sendo a **modelagem das fontes** utilizadas pelo sistema [30]. Nesse contexto, a importância da revisão bibliográfica de trabalhos estrangeiros se torna limitada, pois a estrutura das fontes de cada país difere dos demais. Quanto aos trabalhos brasileiros, alguns modelos se mostraram incompletos ou com focos diferentes, tendo em vista as funcionalidades desejadas neste trabalho.

Após a análise dos modelos existentes, foi desenvolvido o modelo *Integrated Temporal Model JudLaw* (ITMJudLaw), com o objetivo de ser simples, tão funcional quanto os demais, mas com operações não encontradas em nenhum dos modelos estudados. Para uma especificação mais completa do modelo, foi desenvolvido um modelo genérico de documentos textuais-temporais utilizado como base para o *ITMJudLaw*, o *TTMDoc*. Os modelos

TTMDoc e *ITMJudLaw* constituem o primeiro ponto relevante desta dissertação.

Apesar da disponibilização das normas e documentos jurídicos na Internet, eles apresentam um formato semi-estruturado, compreendido pelos seres humanos mas não pelos computadores [59]. Para resolver esse problema, a *Ciência da Computação e Direito* se utiliza de uma subárea da Computação, a *Extração de Informação* (EI), visando à extração automática de trechos de um documento, pré-requisito para o armazenamento estruturado das fontes. Para aplicação de técnicas da EI no contexto do gerenciamento do conhecimento legal, vários autores apontam a **descrição formal** das fontes como passo fundamental nesse processo [13] [32] [35] [48] [62] [75].

Baseado na mesma problemática da modelagem, a descrição formal de fontes estrangeiras não se aplica ao âmbito brasileiro, por isso foi desenvolvida uma descrição formal das fontes nacionais. Essa descrição foi especificada na *Forma Estendida de Backus-Naur* (*Extended Backus-Naur Form*, EBNF), não sendo encontrado nenhum trabalho brasileiro relacionado com esse objetivo, tornando-se o segundo ponto relevante desta dissertação e ponto de partida para trabalhos futuros na área de *EI* aplicada à *CCD*.

O terceiro ponto de relevância do trabalho é a proposta da arquitetura do *SGCL JudLaw*, em conjunto com a implementação de sua infraestrutura, refletida em um protótipo cuja base de dados é composta por normas e documentos jurídicos. Além de servir como base do *JudLaw*, o protótipo foi desenvolvido para validar o modelo proposto e testar a implementação de suas operações.

Atualmente a área de *CCD* está expandindo suas aplicações. Além das já mencionadas neste trabalho, ela está sendo aplicada na *Engenharia de Software*, em sistemas que lidam com informações sujeitas a leis como *HIPAA*¹⁰ (Health Insurance Portability and Accountability Act) e *SOX*¹¹ (Sarbanes-Oxley) [67] [68] [69]. O prospecto é que no futuro seja possível interconectar sistemas desse tipo com sistemas de gerenciamento de conhecimento legal.

¹⁰<http://www.hipaa.org> (Acesso em Junho/2010)

¹¹<http://www.sarbanes-oxley.com> (Acesso em Junho/2010)

1.3 Objetivos

O principal objetivo deste trabalho é propor uma ferramenta de apoio a operadores do Direito, a fim de poupar tempo e esforço na execução da pesquisa normativo-jurisprudencial. Devido ao tamanho e complexidade de um sistema de gerenciamento de conhecimento legal, o escopo deste trabalho foi determinado como a definição e implementação de um modelo estrutural e comportamental para o SGCL *JudLaw*. Os demais módulos serão desenvolvidos em trabalhos futuros.

1.3.1 Objetivos Específicos

1. Desenvolver um modelo integrado temporal das fontes do Direito brasileiro, a partir de um modelo genérico de documentos textuais-temporais;
2. Desenvolver operações estruturais e temporais para o modelo proposto;
3. Descrever formalmente as estruturas dos dispositivos legais e documentos jurídicos;
4. Propor a arquitetura do SGCL *JudLaw*;
5. Implementar a infraestrutura do *JudLaw* para validar o modelo proposto.

1.4 Metodologia

Após a determinação da *Ciência da Computação e Direito* como área interdisciplinar de interesse do trabalho, a metodologia adotada baseou-se na revisão bibliográfica dos principais trabalhos semelhantes na área em busca de problemas e lacunas existentes. Após essa análise, foi levantada a necessidade de um *Sistema de Gerenciamento de Conhecimento Legal* para o âmbito brasileiro e foi delimitado o escopo deste trabalho, supracitado na seção anterior.

Em seguida, foi desenvolvido um protótipo no qual foram feitos os estudos de caso de validação do modelo, envolvendo normas jurídicas e acórdãos do STF e STJ, coletados em Novembro de 2009. Foram implementadas e testadas as operações de CRUD¹², estruturais e temporais através de testes de unidade, garantindo a corretude das especificações.

¹²Acrônimo que representa *Create, Retrieve, Update e Delete*.

1.5 Estrutura da Dissertação

O restante deste trabalho está estruturado da seguinte forma:

- Capítulo 2, *Fundamentação Teórica*: introduz os conceitos associados a *Direito*, *Banco de Dados Temporal*, *Extração de Informação*, *Forma Estendida de Backus-Naur* e *Sistema de Gerenciamento de Conhecimento Legal*;
- Capítulo 3, *Trabalhos Relacionados*: trata da apresentação e discussão dos principais trabalhos relacionados na área, além de uma análise comparativa com este trabalho;
- Capítulo 4, *Modelo ITMJudLaw*: apresenta o modelo *Integrated Temporal Model JudLaw* e suas operações, base do *Sistema JudLaw*. Também é descrito o *TTMDoc*, modelo genérico de documentos textuais-temporais, base do *ITMJudLaw*;
- Capítulo 5, *Descrição Formal das Fontes*: descreve formalmente as fontes do Direito Brasileiro. Nessas estão compreendidas as normas brasileiras e os documentos jurídicos (com ênfase nos documentos do STF e STJ);
- Capítulo 6, *Sistema JudLaw*: discorre sobre a arquitetura proposta do *JudLaw* e o seu protótipo implementado;
- Capítulo 7, *Considerações Finais e Trabalhos Futuros*: apresenta as considerações finais acerca do trabalho e as perspectivas de trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Neste capítulo serão apresentados alguns conceitos fundamentais para um melhor entendimento da dissertação. Na Seção 2.1 serão introduzidos os conceitos do Direito e suas fontes. Na Seção 2.2 será descrita a noção do que é e como funciona um Banco de Dados Temporal. Em seguida, na Seção 2.3 será discorrido sobre a Extração de Informação, suas fases e métricas de avaliação. Posteriormente, na Seção 2.4 será apresentada a sintaxe da Forma Estendida de Backus-Naur (EBNF), utilizada para descrever formalmente o padrão estrutural das fontes do Direito, o qual será utilizado em uma das fases da Extração de Informação. Finalmente, serão definidos na Seção 2.5 o conceito e as operações realizadas pelos Sistemas de Gerenciamento de Conhecimento Legal.

2.1 Direito

O conceito de Direito é inerente a qualquer agrupamento de indivíduos que interagem entre si. Para que haja uma convivência pacífica, são necessárias regras que visem ao bem comum em detrimento dos desejos individuais de cada ser, e isso está presente não apenas nas relações humanas. Com a formação de sociedades cada vez maiores e com interações sociais complexas, surgiu a necessidade da organização dessas regras em *códigos*, inclusive definindo sanções caso houvesse o descumprimento dessas regras. Esse foi o ponto de partida para a edição das fontes do Direito como são conhecidas atualmente, cada uma com sua finalidade específica.

Para a compreensão dessas fontes, é preciso primeiro entender a discussão proposta por Kelsen [46], sobre *atos jurídicos*. De um modo geral, um *ato* pode ser qualquer coisa: andar, falar, escrever, etc. Quando um ato passa a ser denominado *ato jurídico*, quer dizer que agora

esse ato é interessante para o mundo do Direito. Nesse contexto, de um *ato jurídico* ou de qualquer fato que tenha conexão com o Direito, podem ser distinguidos dois elementos [46]: a manifestação externa da conduta humana e seu significado jurídico.

Para ilustrar esse paralelo, pode-se citar dois exemplos:

- Um executivo lê um documento e assina-o, passando para outro executivo que faz o mesmo. Vendo da perspectiva da conduta humana, eles apenas colocaram seus nomes em um documento. Da perspectiva jurídica, firmaram um **contrato**.
- Um homem ao encontrar seu desafeto, desfere três tiros em seu peito, vindo este a falecer. Juridicamente falando, a conduta do primeiro pode ser classificada como **homicídio**, tipificada no Código Penal.

Das três fontes descritas nas próximas seções, duas se referem a atos jurídicos: as *normas jurídicas* e a *jurisprudência*. A primeira é o registro escrito dos atos jurídicos lícitos e ilícitos, advindos dos Poderes Executivo e Legislativo. A segunda é a aplicação das normas visando à resolução de problemas jurídicos, pelo Poder Judiciário. A terceira fonte é a *doutrina*, que é a interpretação das normas por parte dos estudiosos do Direito.

2.1.1 Norma Jurídica

Como supracitado, uma norma jurídica é o registro escrito de um ato jurídico - lícito ou ilícito -, enquanto um conjunto de normas é denominado *ordenamento jurídico* [15]. Também conhecida por *lei*, é definida como “a norma geral de direito formulada e promulgada, por modo autêntico, pelo órgão competente de autoridade soberana e feita valer pela proteção-coerção, exercida pelo Estado” segundo Ráo [65].

A norma mais importante do ordenamento jurídico brasileiro é a Constituição Federal de 1988, à qual todas as demais leis estão subordinadas. No seu artigo 59 são estabelecidos os tipos de normas jurídicas compreendidas no processo legislativo. A Tabela 2.1 relaciona essas normas com seus conceitos definidos por Lima [54] adaptado de Mendes & Júnior [60].

Tipos de Normas	Definições
Emenda Constitucional	Altera a Constituição; Aprovada por três quintos de cada Casa em dois turnos; Promulgada pelo Congresso Nacional.
Lei Complementar	Leis previstas no texto constitucional, aprovadas pela maioria absoluta de cada Casa.
Lei Ordinária	Ato normativo primário, contendo, em regra, normas gerais e abstratas.
Lei Delegada	Ato normativo elaborado e editado pelo Presidente da República em virtude de autorização do Poder Legislativo, expedida mediante resolução e dentro dos limites nela traçados.
Medida Provisória	Ato normativo com força de lei que pode ser editado pelo Presidente da República em caso de relevância e urgência.
Decreto Legislativo	Atos destinados a regular matérias de competência exclusiva do Congresso Nacional que tenham efeitos externos a ele. Não se sujeita à sanção presidencial, sendo promulgada pelo Congresso Nacional.
Resolução	Atos destinados a regular matérias de competência exclusiva do Congresso Nacional ou de suas Casas com efeitos internos.

Tabela 2.1: Tipos de normas e suas definições (Lima [54] adap. de Mendes [60])

Estrutura

A estruturação das normas segue uma *técnica legislativa*, definida por Carvalho [26] como “modo correto de elaborar as leis (...) envolve um conjunto de regras e de normas técnicas que vão desde a necessidade de legislar até a publicação da lei”. A estrutura do ordenamento jurídico atual se baseia na Lei Complementar Nº 95, de 26 de Fevereiro de 1998 (LCP95), posteriormente alterada pela Lei Complementar Nº 107, de 26 de abril de 2001 (LCP107).

Do capítulo II da LCP95 - *DAS TÉCNICAS DE ELABORAÇÃO, REDAÇÃO E ALTERAÇÃO DAS LEIS* são destacadas suas duas primeiras seções: *Da Estruturação das Leis* e *Da Articulação e da Redação das Leis*. Ambas terão seus pontos mais relevantes discutidos, necessários para a posterior compreensão deste trabalho. Em seu *artigo 3º*, a LCP95 define as partes de uma lei:

I - parte preliminar, compreendendo a epígrafe, a ementa, o preâmbulo, o enunciado do objeto e a indicação do âmbito de aplicação das disposições normativas;

II - parte normativa, compreendendo o texto das normas de conteúdo substan-

tivo relacionadas com a matéria regulada;

III - parte final, compreendendo as disposições pertinentes às medidas necessárias à implementação das normas de conteúdo substantivo, às disposições transitórias, se for o caso, a cláusula de vigência e a cláusula de revogação, quando couber.

As definições dos elementos das partes da norma são descritos nos artigos subsequentes da LCP95. Além das regras estruturais, são apresentadas regras relativas ao conteúdo das normas, a exemplo da disposição do *inciso I do artigo 7º*: “excetuadas as codificações, cada lei tratará de um único objeto”. Dada a proposta deste trabalho, não serão discutidas as regras que possuam essa finalidade, sendo focadas as regras associadas à estrutura das leis.

No *artigo 4º*, é definido o conceito de **epígrafe**:

Art. 4º A epígrafe, grafada em caracteres maiúsculos, propiciará identificação numérica singular à lei e será formada pelo título designativo da espécie normativa, pelo número respectivo e pelo ano de promulgação.

Apesar dessa disposição, essa instrução não é seguida pelos legisladores, pois nas leis atuais não é definido apenas o ano de promulgação, e sim a data completa com dia, mês por extenso e ano. A norma LCP95 foi a principal referência na modelagem e descrição formal das normas jurídicas, mas em situações como essa foi considerado o modo atual de estruturação das normas, tendo como base as fontes citadas no Apêndice A.

A **ementa** trata do objeto ao qual a lei se destina (*artigo 5º*) e o **preâmbulo** “indicará o órgão ou instituição competente para a prática do ato e sua base legal” (*artigo 6º*). Como elucida e ilustra Lima [54], cada tipo de norma possui um texto específico para o preâmbulo. Nas leis ordinárias, por exemplo, o preâmbulo expressa: “Faço saber que o Congresso Nacional decreta e eu sanciono a seguinte Lei”.

No *artigo 10* são apresentados os elementos que compõem a norma. Em seu primeiro inciso, a forma de estruturação do elemento **artigo** é introduzida:

I - a unidade básica de articulação será o artigo, indicado pela abreviatura “Art.”, seguida de numeração ordinal até o nono e cardinal a partir deste;

No inciso seguinte são introduzidos os desdobramentos do **artigo**, e dos seus elementos: “os artigos desdobrar-se-ão em parágrafos ou em incisos; os parágrafos em incisos, os incisos em alíneas e as alíneas em itens”.

No *inciso III* do mesmo artigo são definidos os identificadores do elemento **parágrafo**. Via de regra, ele é representado pelo símbolo “§” seguido do mesmo padrão de numeração utilizado pelo artigo. Quando só existe um parágrafo, é utilizada a expressão “Parágrafo único”.

No inciso posterior são descritos os identificadores dos **incisos** (algarismos romanos), **alíneas** (letras minúsculas) e **itens** (algarismos arábicos). Segue uma ilustração de Lima [54] que representa os desdobramentos de um artigo:

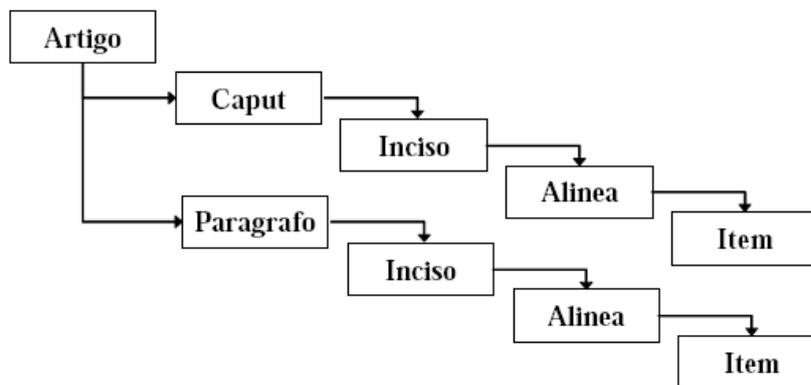


Figura 2.1: Desdobramentos do artigo (Lima [54])

Nos *incisos V e VIII* são apresentados os agrupadores do elemento **artigo**:

V - o agrupamento de artigos poderá constituir Subseções; o de Subseções, a Seção; o de Seções, o Capítulo; o de Capítulos, o Título; o de Títulos, o Livro e o de Livros, a Parte;

VIII - a composição prevista no inciso V poderá também compreender agrupamentos em Disposições Preliminares, Gerais, Finais ou Transitórias, conforme necessário.

No *inciso VI* é definido que o elemento **Parte** pode “desdobrar-se em Parte Geral e Parte Especial ou ser subdivididas em partes expressas em numeral ordinal, por extenso”. Assim como nos desdobramentos do artigo, são descritos nos *incisos VI e VII* os identificadores dos

agrupadores do artigo. Os **Capítulos**, **Títulos**, **Livros** e **Partes** são representados em letras maiúsculas e as **Subseções** e **Seções** em letras minúsculas, todos identificados por algarismos romanos.

Segue uma ilustração de Lima [54] que contempla apenas os agrupadores do artigo, porém que não representa a situação disposta no *inciso VIII*:

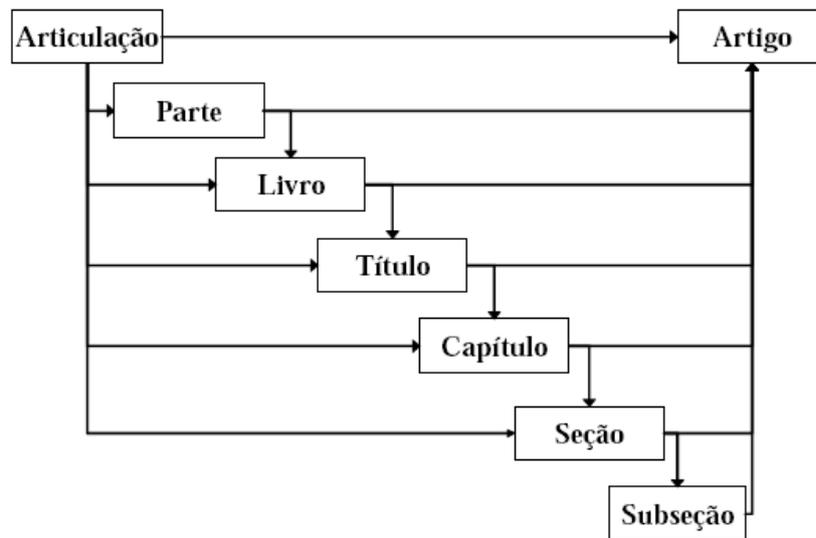


Figura 2.2: Agrupadores do elemento artigo (Lima [54])

Aspectos temporais das normas.

Uma característica comum aos atos jurídicos é a obrigatoriedade do cumprimento das suas disposições. No caso dos documentos jurídicos, essa obrigatoriedade depende apenas de sua publicação, o que não acontece com as leis, devido ao período de *vacatio legis*. como explica França [37]:

O lapso de tempo que vai da publicação da lei até o início de sua eficácia chama-se vacatio legis, cuja finalidade é a de propiciar a mais ampla divulgação dos textos legais, devendo ser assinalado que, enquanto êsse tempo não escoar, continuam em plena vigência as leis antigas ainda que venham a ser revogadas pela lei nova, já publicada.

O período de *vacatio legis* é regulado pelo *caput* de *artigo 8º* da LCP55 e pelos dois parágrafos adicionados pela LCP107:

Art. 8º A vigência da lei será indicada de forma expressa e de modo a contemplar prazo razoável para que dela se tenha amplo conhecimento, reservada a cláusula “entra em vigor na data de sua publicação” para as leis de pequena repercussão.

§ 1º A contagem do prazo para entrada em vigor das leis que estabeleçam período de vacância far-se-á com a inclusão da data da publicação e do último dia do prazo, entrando em vigor no dia subsequente à sua consumação integral.

§ 2º As leis que estabeleçam período de vacância deverão utilizar a cláusula ‘esta lei entra em vigor após decorridos (o número de) dias de sua publicação oficial’

Caso não seja expressa a cláusula de vigência na norma ou a republicação devido a erros, o Decreto-Lei 4.657 de 04 de Setembro de 1942 (Lei de Introdução ao Código Civil) regula:

Art. 1º Salvo disposição contrária, a lei começa a vigorar em todo o país quarenta e cinco dias depois de oficialmente publicada.

§ 1º Nos Estados estrangeiros, a obrigatoriedade da lei brasileira, quando admitida, se inicia três meses depois de oficialmente publicada.

...

§ 3º Se, antes de entrar a lei em vigor, ocorrer nova publicação de seu texto, destinada a correção, o prazo deste artigo e dos parágrafos anteriores começará a correr da nova publicação.

§ 4º As correções a texto de lei já em vigor consideram-se lei nova.

Diferentemente dos documentos jurídicos, a vigência de uma norma pode não se iniciar após sua publicação, caso esteja em período de *vacatio legis*. Como explica Lima [54], essa vigência se estende até a revogação da norma - explícita ou tácita - ou até que se finde por conta própria, caso se trate de uma lei com vigência temporária.

A revogação *explícita* ocorre quando uma norma é expressamente revogada por outra, enquanto a *tácita* resulta da incompatibilidade ou divergência entre normas. Caso a norma

seja revogada completamente, essa revogação é chamada de *ab-rogação*; caso apenas alguns elementos da norma sejam revogados, é chamada de *derrogação*.

Quando se trata de aspectos temporais das normas, três conceitos são fundamentais: *validade*, *vigência* e *eficácia*. Couto Filho [29] promove uma extensa discussão sobre os três conceitos, diferenciando inicialmente *validade* e *vigência*:

A validade não se confunde com a vigência, posto que pode haver uma norma jurídica válida sem que esteja vigente, isso ocorre claramente quando se vislumbra a vacatio legis ou quando o dispositivo legal é revogado, embora continue vinculante para os casos pretéritos. A vigência representa a característica de obrigatoriedade da observância de uma determinada norma, ou seja, é uma qualidade da norma que permite a sua incidência no meio social.

Na conclusão do artigo, é definida a equivalência dos conceitos de *vigência* e *eficácia* para o ordenamento jurídico do Brasil, posicionamento adotado nesta dissertação:

Após esta argumentação, a conclusão indubitável que pode ser retirada do presente estudo e do ordenamento jurídico nacional é que vigência e eficácia são institutos conceitualmente homogêneos, sem qualquer diferença significativa, seja sob o aspecto pragmático, seja sob o aspecto teórico.

2.1.2 Jurisprudência

De acordo com a definição de Maximiliano [58]:

Chama-se Jurisprudência, em geral, ao conjunto das soluções dadas pelos tribunais às questões de Direito; relativamente a um caso particular, denomina-se jurisprudência a decisão constante e uniforme dos tribunais sobre determinado ponto de Direito.

A importância da jurisprudência está na **interpretação** e **aplicação** das leis. Segundo Bergel [12], “Já em sua promulgação, a lei começa a envelhecer (...) A jurisprudência tenta então manter-lhe a atualidade, às vezes com acrobacias ou com artifícios”. Na visão de Maximiliano [58], a jurisprudência aplica a lei levando em consideração os costumes da sociedade na qual ela está inserida, além de preencher suas lacunas.

Quando um juiz emana uma decisão acerca de um caso, esse julgado é conhecido como *sentença*. Quando a decisão parte de um tribunal, ela é composta do voto de três ou mais magistrados, geralmente desembargadores, nesse caso o julgado é chamado *acórdão*. Tanto sentenças quanto acórdãos são tipos de *documentos jurídicos*.

Estrutura

A estrutura de um acórdão definida por Vanzin *et al.* [74] foi adaptada e utilizada como base para o desenvolvimento da modelagem dos documentos jurídicos. Segundo os autores, esses documentos são compostos das seguintes subestruturas:

1. *Cabeçalho*: informações para identificação do documento jurídico;
2. *Ementa*: resumo do conteúdo de um acórdão;
3. *Pessoas Envolvidas*: tipo de recurso e o nome das partes envolvidas no processo;
4. *Relatório*: registro dos principais fatos ocorridos no andamento do processo;
5. *Voto*: análise do magistrado referente às questões de fato e de direito relatadas no processo, além da fundamentação da decisão;
6. *Sentença*: resultado do pedido feito por uma das partes do processo judicial. Indica se os magistrados aceitaram o pedido feito no recurso e se a decisão foi unânime;
7. *Encerramento*: informações sobre os magistrados responsáveis pela elaboração da decisão, local e data do julgamento.

Ainda em Vanzin *et al.* [74] são definidos os elementos de cada subestrutura, porém no modelo proposto esses elementos foram estruturados a partir da observação de acórdãos do Superior Tribunal Federal (STF) e do Superior Tribunal de Justiça (STJ). O detalhamento dos elementos de cada subestrutura será apresentado em paralelo com o modelo *ITMJudLaw*, no capítulo 4.

2.1.3 Doutrina

A doutrina é a interpretação das leis na visão dos estudiosos do Direito, considerada uma fonte auxiliar na pesquisa normativo-jurisprudencial. Apesar do seu atual papel, a doutrina era a principal fonte do Direito Romano, como elucida Bergel [12].

Segundo Lima [54], o motivo da diminuição da importância das fontes doutrinárias se deu devido à falta de lacunas deixadas pela abundância de informações legislativas e jurisprudenciais. Nesse contexto, o número crescente de jurisprudências é um dado relevante, pois além de serem atos jurídicos também são entendimentos acerca do ordenamento jurídico aplicados a problemas concretos.

A doutrina é representada no modelo proposto neste trabalho como fonte passível de citação pelos documentos jurídicos, relação modelada a partir da observação de documentos dessa natureza. As citações a doutrinas são feitas geralmente na forma de referências a trechos de livros que tratam da matéria relacionada, escritos por autores consagrados.

2.2 Banco de Dados Temporal

Apesar da sua simplicidade e sólida base matemática, o modelo relacional não é adequado para a manipulação de dados temporais [71]. As informações armazenadas em um banco de dados tradicional representam apenas um ponto no tempo - o atual -, por isso ele também é conhecido como banco de dados *snapshot* (BDS). No entanto, dependendo da aplicação, pode-se precisar do histórico dos dados, requerendo o armazenamento de múltiplos pontos no tempo. A partir dessa problemática, foi proposta uma extensão ao banco de dados tradicional, dando origem ao banco de dados temporal (BDT), podendo ser aplicada em várias áreas, a exemplo da *Mineração Temporal* [49], pois através do histórico dos dados é possível a detecção de padrões nas séries de tempo.

2.2.1 BDS X BDT

Dentre as principais diferenças entre um BDS e um BDT, destacam-se duas: *atualizações* e *consultas*. Na Tabela 2.2 são ilustradas operações de atualização de dados em um banco de dados temporal, e esse é o primeiro ponto divergente entre os tipos de bancos de dados:

Em um BDS, a atualização de uma tupla é realizada através da modificação de valores da

Data	Nome	Endereço	Salário
12/01/10	Ulrich Schiel	Rua dos Professores	R\$15.000
12/06/10	Ulrich Schiel	Rua dos Professores	R\$20.000
12/12/10	Ulrich Schiel	Rua dos Coordenadores	R\$20.000

Tabela 2.2: Atualização de informações armazenadas em um BDT

mesma, sem a adição de novas tuplas. No exemplo acima, apenas a última tupla continuaria armazenada, com os valores modificados dos atributos *Data*, *Endereço* e *Salário*.

Em um BDT, as atualizações são feitas dessa forma apenas quando um valor incorreto é inserido. Via de regra, as atualizações determinam em adições de novas tuplas com os valores modificados, como ilustra a Tabela 2.2, na qual as três tuplas são mantidas. Essa abordagem permite o acompanhamento das versões dos dados, e apesar do exemplo não refletir a forma mais eficiente de armazenamento (devido à quantidade de informação redundante), é suficiente para apresentar a lógica empregada nos bancos de dados temporais.

O segundo ponto de diferenciação entre um BDS e um BDT é a complexidade das consultas [34]. Consultas a valores situados em faixas temporais ilustram essa afirmação, a exemplo de “retorne o cpf de todos os funcionários de uma empresa que possuíam salário superior a R\$5.000 entre 01/01/2010 e 01/01/2011”. Esse tipo de consulta só é possível em um BDT, pois em um BDS não haveria um histórico temporal com as movimentações salariais dos funcionários.

Supondo uma consulta temporal q com uma faixa temporal válida $V^q = [t_i^q, t_f^q]$ (sendo t_i^q o tempo inicial e t_f^q o tempo final) e uma tupla “a” com a faixa temporal válida $V^a = [t_i^a, t_f^a]$, várias combinações de consultas podem envolver as duas faixas [34]:

- *Interseção*: a tupla “a” será recuperada se $V^a \cap V^q \neq \emptyset$.
- *Inclusão*: a tupla “a” será recuperada se $t_i^q \leq t_i^a \leq t_f^a \leq t_f^q$, ou seja, $V^a \subseteq V^q$.
- *Está Contido*: a tupla “a” será recuperada se $t_i^a \leq t_i^q \leq t_f^q \leq t_f^a$, ou seja, $V^q \subseteq V^a$.
- *Ponto*: a tupla “a” será recuperada se $t_i^a = t_i^q = t_f^q = t_f^a$.

2.2.2 Tempo válido e tempo transacional

No âmbito dos BDTs, o conceito de *valid time* ou *tempo válido* está associado ao tempo em que a informação é válida no mundo real [72], ou seja, está associado ao **fato**. Na tabela

2.2, a partir da data 12/06/10 o salário do professor Ulrich Schiel efetivamente passou a ser R\$20.000, mesmo que essa informação tenha sido inserida em uma data posterior no BDT. A data em que o fato é inserido no banco de dados é conhecida como *transaction time*, ou *tempo transacional*, associada à **entidade** do BDT, não ao fato em si [45].

O *tempo válido* de uma informação é determinado por um intervalo $[t_i, t_f]$, sendo t_i o tempo inicial e t_f o tempo final. Quando não houver um tempo final determinado, ele será representado por ∞ ou por algum termo que represente essa situação (Jensen [45] sugere “agora”), indicando que a informação continua válida por tempo indeterminado. Na tabela (2.2), o tempo válido do salário de Ulrich Schiel igual a R\$15.000 pode ser representado por $[12/01/10, 11/06/10]$. Se um fato é válido por vários intervalos de tempos disjuntos, esta coleção de intervalos é denominada de *elemento temporal*.

No contexto desta dissertação, o conceito de *tempo válido* está associado à vigência das normas e de seus itens, sendo desejado que todas as suas versões estejam disponíveis e que seja possível realizar consultas de validação temporal, cenário que não poderia ser implementado em um BDS. Por exemplo, “dado um documento jurídico, quais de suas referências não estão mais válidas de acordo com o ordenamento jurídico na data X ?”.

2.3 Extração de Informação

Assim como as fontes do Direito, documentos em geral são escritos em linguagem natural, contendo uma grande quantidade de informação não processável diretamente por computadores. Os sistemas que implementam a Extração de Informação (EI) visam à análise e extração de partes importantes do documento, a fim de tornarem a informação **estruturada**, através do preenchimento de campos (*slots*) pré-definidos.

O conceito de estruturação da linguagem natural contida nos documentos não é novo, pois na década de 50 pesquisadores já haviam percebido o potencial dessa linha de pesquisa, desenvolvendo projetos para estruturar enciclopédias inteiras [43]. Alguns desafios da aplicação de EI foram detectados nesses projetos, a exemplo do processamento da linguagem natural e representação do conhecimento, os quais ainda persistem.

Como elucidam Gaizauskas & Wilks [39], o conceito de *Extração de Informação* (EI) não deve ser confundido com *Recuperação de Informação* (RI). Enquanto a técnica de *RI* se propõe a recuperar documentos de uma coleção, a proposta da *EI* é extrair informações

relevantes dos documentos. São técnicas complementares utilizadas no processamento de textos.

2.3.1 Fases

O processo de EI é composto por três partes [43], levando em consideração o conceito de “fato” como sendo um trecho de informação relevante a ser extraída do documento. Na primeira, há a extração dos fatos individuais de um texto, em seguida esses fatos são integrados, resultando na produção de fatos maiores ou novos fatos. Por fim, os fatos pertinentes são traduzidos para o formato de saída pré-determinado.

Ainda segundo Grishman [43], o processo de extração de informação é composto por seis fases, divididos nas três partes supracitadas:

1. *Análise léxica*: análise morfossintática de palavras e sentenças;
2. *Reconhecimento de nomes*: identificação de nomes e outras estruturas léxicas;
3. *Análise sintática parcial*: identifica grupos de substantivos, verbos e estruturas complementares;
4. *Casamento de padrões*: utilização de padrões para identificar os fatos de interesse;
5. *Análise correferencial*: resolve problemas de *correferência* (quando múltiplos termos ou sentenças diferentes denotam a mesma coisa);
6. *Inferência*: faz inferências através de fatos explícitos do documento.

Na primeira fase o texto é dividido em *tokens* e sentenças, e através de um dicionário é feita a análise morfossintática. Já existem inclusive dicionários com nomes de pessoas e empresas, facilitando sua marcação na fase seguinte. A fase de *reconhecimento de nomes* identifica os nomes e estruturas especiais, a exemplo de nomes pessoais e datas. Essa identificação é feita a partir de um conjunto de padrões, representados por expressões regulares.

A identificação de nomes também tenta resolver o problema de *correferência*. No âmbito jurídico, um exemplo seria “matar alguém” e possivelmente “homicídio”. Outro exemplo é

a detecção de nomes de empresas, quando podem ser utilizados dicionários especiais de identificadores como *Ltda*, *S/A* e abreviações como *IBM*, *UFMG* e *OAB*.

A terceira fase - análise sintática parcial - tenta identificar a correlação sintática dos termos na frase em relação aos substantivos e verbos. Exemplo:

O réu primário João da Silva cometeu um delito, homicídio com arma de fogo, no dia 15 de Fevereiro de 2010. O réu alegou que agiu em legítima defesa.

Através da análise sintática parcial é possível detectar que os termos “réu” e “primário” se referem a João da Silva. Esse conhecimento é armazenado e é aplicado na fase de *análise correferencial*, a exemplo da segunda sentença, onde o sistema deve entender que quem alegou legítima defesa foi João da Silva. Essa tarefa não é trivial, prova disso é que parte dos sistemas de EI não implementam essa fase [43].

As três fases anteriores serviram de preparação para a fase de *casamento de padrões*. Com a identificação e marcação dos elementos do texto, é possível aplicá-los ao padrão pré-estabelecido. O padrão da epígrafe de uma norma pode ser descrito como:

***Tipo da norma**, seguido pelo identificador N^o, seguido pelo **número** da norma, seguido por uma **vírgula**, seguida pela **data** de edição da norma e por fim (opcionalmente) um **ponto final**.*

Exemplo de uma epígrafe, referente ao Código Penal:

DECRETO-LEI N^o 2.848, DE 7 DE DEZEMBRO DE 1940.

A partir da definição de um padrão (*template*) e do pré-processamento do documento nas três fases anteriores, é possível preencher os *campos* desse padrão, tornando a informação estruturada e pronta para o armazenamento na base de dados. No entanto, para tentar melhorar a qualidade desses dados, Grishman [43] descreve mais duas fases.

Na fase de *análise correferencial* são aplicadas no documento as relações entre os termos detectadas na fase de *análise sintática parcial*. Na última fase, são feitas inferências no texto na tentativa de detectar informações implícitas ou explícitas. Exemplo:

FHC era presidente. Ele foi sucedido por Lula.

FHC será o presidente. Ele sucede Lula.

A fase de inferência deve assegurar que a informação de quem era e quem é presidente no momento seja extraída de forma correta. Um outro exemplo é a extração de eventos temporais através de termos como “hoje” ou “na última semana”, a partir da data do documento.

2.3.2 Métricas de Avaliação

Na Extração de Informação são aplicadas três métricas de avaliação: cobertura (*recall*), precisão (*precision*) e *f-measure*. A primeira medida avalia a quantidade de informações relevantes corretamente extraídas de todas as informações relevantes no documento, enquanto a segunda avalia a corretude das informações extraídas. A terceira medida é uma média harmônica que combina as duas medidas anteriores.

Dado N_t como o número total de campos a serem preenchidos do padrão (*template*), N_p como o número de campos que foram preenchidos corretamente após a extração e N_c como o número de campos preenchidos corretamente, ou seja, com os valores esperados, então:

$$\text{cobertura} = \frac{N_p}{N_t}; \text{ precisão} = \frac{N_c}{N_p}.$$

Utilizando novamente o exemplo da epígrafe do Código Penal, espera-se que três campos sejam preenchidos após a aplicação da EI: tipo da norma, número e data de edição. Entretanto, supondo que os campos foram preenchidos da seguinte forma:

Campo	Resultado
Tipo da Norma	DECRETO-LEI
Número da Norma	
Data de Edição	7 DE DEZEMBRO DE

Tabela 2.3: Preenchimento dos campos da epígrafe

Segundo a Tabela 2.3, foram extraídos os campos *tipo da norma* e *data de edição*, porém apenas o primeiro está correto. Nesse contexto, o índice de cobertura é 1/3 (33,33%) e o de precisão é 1/2 (50%).

A terceira métrica (*f-measure*) combina precisão (P) e cobertura (C):

$$\text{f-measure} = \frac{2 \times P \times C}{P + C}.$$

2.4 Forma Estendida de Backus-Naur

A *Forma Estendida de Backus-Naur* (EBNF) está na categoria das *metalinguagens sintáticas*, as quais podem ser definidas como notações para a descrição formal da sintaxe de outras linguagens através de regras [36]. Um âmbito comum de aplicação é a definição de linguagens de programação, a exemplo da linguagem *Algol 60*, descrita pela *Forma de Backus-Naur* (BNF).

2.4.1 Metalinguagem Sintática

Cada regra de uma metalinguagem sintática (MS) é composta por símbolos *não-terminais* e *terminais*. Os símbolos *não-terminais* são conceitos que podem ser particionados em outros elementos, enquanto os *terminais* são atômicos, constituindo a menor parte da MS.

Segundo a norma ISO/IEC 14977:1996 [36], é recomendada a utilização de uma metalinguagem sintática quando houver a necessidade de uma descrição ou definição formal clara. Ainda segundo a norma ISO/IEC 14977, é desejável que uma MS seja:

1. *Concisa*: para que as linguagens sejam descritas de forma breve;
2. *Precisa*: para que as regras não contenham ambiguidade;
3. *Formal*: para que seja possível o processamento das regras por um computador;
4. *Natural*: para que seja simples o suficiente de ser entendida por pessoas que não projetaram a linguagem;
5. *Geral*: para que a notação possa ser aplicada na descrição de várias linguagens;
6. *Simples*: para que o conjunto de caracteres utilizados estejam presentes nos teclados comuns e possam ser entendidos por outros softwares;
7. *Auto-descritiva*: para que a notação seja capaz de se auto-descrever;
8. *Linear*: para que a sintaxe possa ser representada por um único fluxo de caracteres, facilitando o processamento dos computadores.

A *Forma de Backus-Naur* é atualmente uma das metalinguagens sintáticas mais utilizadas na comunidade científica, tanto em sua forma original quanto através de suas variações, a

exemplo de *Extended BNF* (EBNF) e *Augmented BNF* (ABNF). Essas variações surgiram devido a limitações do BNF, como a descrição dos seus símbolos reservados na linguagem a ser definida e a complexidade em descrever algumas construções, a exemplo da repetição [36].

Devido aos problemas supracitados do BNF, a MS escolhida para a descrição formal das fontes do Direito foi a *Forma Estendida de Backus-Naur*, pois esta atende a todos os requisitos enumerados pela ISO/IEC 14977 e as fontes podem ser descritas em uma linguagem linear.

2.4.2 Sintaxe

A especificação BNF é composta por regras, tornando possível especificar completamente a sintaxe válida de uma fonte textual. Essas regras possuem a seguinte forma:

$$\langle \text{símbolo_não-terminal} \rangle ::= \text{expressão}$$

Cada regra representa uma substituição válida de um *símbolo não-terminal* por uma *expressão*, formada por símbolos terminais e/ou não-terminais. Para escrever uma regra são usados meta-símbolos (como “::=” e “[]”) que não devem ser confundidos com os símbolos da metalinguagem representada. Assim como em outras variações do BNF, o EBNF contém algumas modificações para facilitar a representação de uma linguagem [36]:

- Os símbolos reservados da metalinguagem podem ser citados através de aspas, assim não há restrições na utilização deles na linguagem a ser descrita;
- [e] expressa que um símbolo ou expressão é opcional;
- { e } também expressam repetição, além do símbolo * (asterisco) advindo do BNF;
- O fim de cada regra é indicado pelo símbolo ; (ponto-e-vírgula), assim não há ambiguidade quanto ao início e fim das regras;
- (e) agrupam os símbolos, assim como na representação matemática.

Além dessas modificações comuns, o EBNF apresenta algumas extensões próprias [36]:

1. *Especificação da quantidade de itens*, caso o projetista de uma linguagem queira expressar que determinado símbolo possui valor fixo;
2. *Definição através de casos*, a exemplo de um comentário na linguagem Algol, que pode terminar com o primeiro *end*, *else* ou *ponto-e-vírgula*;
3. *Introdução de comentários nas regras*, facilitando o entendimento de regras complexas sem a necessidade de documentação anexa;
4. *Meta-identificadores*, permitindo que os símbolos não-terminais possuam mais de um nome sem deixar a regra ambígua. Isso é possível através da introdução de um símbolo explícito de concatenação: *,* (vírgula);
5. *Possibilidade de extensões*, através do símbolo especial *?* (interrogação).

A Tabela 2.4 exibe a descrição dos símbolos do EBNF, adaptada da norma ISO/IEC 14977 [36]:

Símbolo	Representação Inicial	Representação Final
Concatenação	,	
Definição	=	
Exceção	-	
Repetição	*	
Sequência Especial	?	
Separador-definições		
Fim-regra	;	
Comentário	(*	*)
Agrupamento	()
Opção	[]
Repetição (expressão)	{	}
Citação (aspas)	“	”
Citação (aspas simples)	‘	’

Tabela 2.4: Símbolos do EBNF (adap. de ISO/IEC 14977:1996 [36])

2.4.3 Exemplos

Exemplo 1, com os símbolos de concatenação, definição, separador-definições e fim-regra:

redação = *introdução* , *desenvolvimento* , *conclusão* ;

letra = *cursiva* | *de forma* ;

No primeiro exemplo, foi descrito que uma redação é composta por introdução, desenvolvimento e conclusão, dispostos nessa ordem. A letra a ser utilizada pode ser tanto cursiva quanto de forma.

Exemplo 2, retirado da ISO/IEC 14977 [36], o qual ilustra como são representadas as repetições:

$aa = \textit{“A”};$

$bb = 3 * aa, \textit{“B”};$

$cc = 3 * [aa], \textit{“C”};$

$dd = \{aa\}, \textit{“D”};$

$ee = aa, \{aa\}, \textit{“E”};$

$ff = 3 * aa, 3 * [aa], \textit{“F”};$

Saída:

$aa: A$

$bb: AAAB$

$cc: C AC AAC AAAC$

$dd: D AD AAD AAAD AAAAD \textit{etc.}$

$ee: AE AAE AAAE AAAAE AAAAAE \textit{etc.}$

$ff: AA AF AAA AF AAA AF AAA AF$

Para indicar que uma repetição é realizada pelo menos uma vez, é utilizado o símbolo de exceção após os símbolos de repetição: $\{símbolo\}-$. Caso a regra dd fosse $\{aa\}-$, $\textit{“D”}$, a saída D não seria válida.

O objetivo da descrição formal das fontes do Direito na *Forma Estendida de Backus-Naur* é servir como padrão para a extração automática de partes dos documentos, tornando possível o armazenamento estruturado destes. Essa descrição será utilizada na fase de *casamento de padrões*, supracitada na Seção 2.3.1.

2.5 Sistema de Gerenciamento de Conhecimento Legal

Devido ao constante aumento no volume de informações a serem processadas pelos operadores do Direito, surgiu a necessidade da criação de ferramentas que os auxiliem em suas tarefas. A partir dessa problemática, foram desenvolvidos os SGCLs, cujas principais funções segundo Riveret [64] são: *aquisição, modelagem, armazenamento, recuperação, disseminação e manutenção* do conhecimento legal.

2.5.1 Aquisição

A primeira função é de resolver como será feita a aquisição dos dados. Via de regra as fontes do Direito encontram-se espalhadas, tornando-se o primeiro obstáculo na pesquisa normativo-jurisprudencial do operador. É importante que o SGCL possua um módulo de aquisição de documentos de diversas fontes, transparente ao usuário final, possibilitando-o consultar essas fontes através de um ponto centralizado.

Além de oferecer suporte a uma pesquisa distribuída, um SGCL também precisa selecionar quais fontes são mais importantes para seu público-alvo. Um SGCL destinado apenas a buscas jurisprudenciais, por exemplo, deve agregar os documentos jurídicos de vários tribunais, porém não deve incluir as normas jurídicas nas fontes.

2.5.2 Modelagem e Armazenamento

Para que o conhecimento legal possa ser processado por computadores, é necessário que antes esse conhecimento seja modelado, de acordo com o objetivo do SGCL. Há várias propostas de modelos de fontes do Direito, utilizando desde ontologias até descrições em XML e DTD.

Para SGCLs que visam a resolver problemas na área de Inteligência Artificial, os modelos são geralmente descritos em ontologias, através da *Web Ontology Language* (OWL). O objetivo dessa abordagem é a representação do conhecimento contido não apenas nos documentos, mas também dos operadores. Um exemplo de SGCL com esse intuito é o *Iuriservice II* [28], cujo modelo inclui as questões jurídicas mais comuns que os juízes espanhóis resolviam.

Quando o objetivo do SGCL é integrar diversas fontes, os modelos geralmente são des-

critos através da *Extensible Mark-up Language* (XML). Além de ser o padrão recomendado pela *World Wide Web Consortium* (W3C), o XML permite a representação das informações através de esquemas pré-definidos, via *Document Type Definition* (DTD) ou *XML Schema* [64]. Exemplos dessa categoria de SGCLs são os portais que permitem buscas integradas em normas e jurisprudências: LexML Brasil [50] e AKOMA NTOSO [1].

Existe ainda o projeto europeu Estrella¹ de gerenciamento de conhecimento legal que utiliza o padrão XML aplicado à *Web Semântica*, através da OWL para descrever as ontologias Web e o *Resource Description Framework* (RDF) para descrever as fontes.

Após a modelagem, o SGCL deve prover o armazenamento das fontes de forma estruturada, através da *EI*. Os bancos de dados mais utilizados são os *relacionais* e os *orientados a objeto* [64]. Entretanto, devido ao aumento da utilização do padrão XML para representar as fontes, vários projetos estão estudando a alternativa de adotar bancos de dados XML.

2.5.3 Recuperação e Disseminação

Recuperar informações advindas de diversas fontes de forma integrada não é uma tarefa trivial, pois um SGCL deve estar preparado para trabalhar com milhares de documentos. Segundo Riveret [64], a técnica mais utilizada atualmente na recuperação de conteúdo legal é a técnica padrão: indexação dos documentos para prover buscas mais relevantes.

Nos SGCLs voltados para a Web Semântica, a representação do conhecimento legal é enriquecido com as ontologias, e as buscas semânticas tentam “podar” resultados não relevantes para o usuário. As ontologias também são amplamente utilizadas na recuperação de informações no contexto da Inteligência Artificial.

A questão da disseminação está diretamente ligada à recuperação da informação, pois trata a questão de que conteúdo deve ser direcionado a cada tipo de usuário. A disseminação tem a ver com a definição de **quais** informações são importantes, **quando** elas são importantes e qual o melhor **formato** para elas serem apresentadas para cada usuário.

2.5.4 Manutenção

Por último, mas não menos importante, uma das funções de um SGCL é a manutenção dos dados. Devido ao caráter extremamente mutável das fontes do Direito, o sistema precisa

¹<http://www.estrellaproject.org> (Acesso em Junho/2010)

estar sempre atualizando a sua base com as alterações realizadas no ordenamento jurídico. De acordo com a necessidade dos usuários, pode ser interessante manter um histórico das versões dos dados, utilizando os conceitos de um banco de dados temporal.

2.6 Considerações Finais

Neste capítulo foram apresentados conceitos adotados no presente trabalho. Após a introdução dos conceitos de Direito e suas fontes, foi apresentado o modelo de Banco de Dados Temporal, adotado pelo sistema *JudLaw*. Em seguida, foi explicado o que é Extração de Informação, pois a descrição formal em EBNF na seção seguinte se encaixa em uma das fases da EI. Por fim, foram explicadas as funções de um SGCL. No próximo capítulo serão apresentados os trabalhos relacionados.

Capítulo 3

Trabalhos Relacionados

Neste capítulo será apresentado o estado da arte relativo ao escopo desta dissertação. Na Seção 3.1, serão discutidos trabalhos que tratam da modelagem das fontes do Direito. Como muitos desses trabalhos discorrem sobre modelos aplicados a sistemas de gerenciamento de conhecimento legal, houve a necessidade da agregação dos dois tópicos (modelos e SGCLs) em uma única seção.

Na Seção 3.2 serão apresentados trabalhos que demonstram a relevância da descrição formal das fontes do Direito na extração de informação dos documentos. Por fim, na Seção 3.3 será feita uma análise comparativa entre trabalhos relacionados e esta dissertação, a fim de definir as contribuições desta em relação ao estado da arte, destacando os pontos positivos e negativos dos modelos apresentados e das funcionalidades dos sistemas que os implementam.

Foi realizada uma revisão bibliográfica relativa aos modelos e SGCLs propostos na literatura, a fim de determinar quais funcionalidades o *JudLaw* deveria implementar. O objetivo desse levantamento foi a definição das melhores práticas na modelagem das fontes do Direito e a detecção das incompletudes dos modelos em relação às funcionalidades desejadas. As principais fontes de trabalhos relacionados utilizadas nesta dissertação foram:

- Conferência Internacional de Conhecimento e Sistemas de Informação Legais (*JURIX: International Conference on Legal Knowledge and Information Systems*) e Conferência Internacional de Inteligência Artificial e Direito (*International Conference on Artificial Intelligence and Law*);
- Modelos adotados como padrão na representação de fontes do Direito de países (Bélgica, Suíça, Austrália, Dinamarca, Brasil e Itália) e continentes (Europa e África);

- Pesquisas desenvolvidas no Centro Leibniz de Direito (*Leibniz Center for Law*).

3.1 Modelagem e SGCLs

A modelagem das fontes do Direito está diretamente ligada ao cenário no qual ela está inserida [9]. Os modelos podem ser projetados para as mais diversas aplicações: raciocínio baseado em casos legais, suporte à decisão, recuperação de informação, etc. Devido a essa diversidade de domínios, as tecnologias utilizadas para a modelagem do conhecimento também são variadas, a exemplo de ontologias e XML.

Devido ao foco do sistema *JudLaw* ser o *gerenciamento do conhecimento legal*, serão destacados os modelos e sistemas cujo propósito seja semelhante. A título de informação, a área de Inteligência Artificial e Direito também é amplamente pesquisada e rica em trabalhos científicos, principalmente na área de modelagem do raciocínio legal, a exemplo de [8] [11] [22] [27] e [40]. A fonte do Direito preferida das pesquisas nessa área é a *jurisprudência*.

Foram selecionados na revisão bibliográfica os trabalhos voltados para a modelagem *estrutural e dinâmica* das fontes. As questões semânticas não são objetos de estudo desta dissertação, porém estão cogitadas para serem aprofundadas em trabalhos futuros, visto sua importância na área interdisciplinar de *Computação e Direito*.

O restante da seção está dividida da seguinte forma: na Subseção 3.1.1 serão destacados trabalhos brasileiros mais relevantes na área de modelagem estrutural e dinâmica das fontes do Direito. Em seguida, na Subseção 3.1.2 serão discutidos os trabalhos que tratam da dinâmica das fontes, ou seja, suas interações e temporalidade. Por fim, na Subseção 3.1.3 serão apresentados modelos e SGCLs provenientes de iniciativas de países de todo mundo, inclusive alguns com incentivos governamentais.

3.1.1 Modelagem Normativo-Jurídica

A produção científica nacional na área de gerenciamento de conhecimento legal ainda é escassa, principalmente em relação aos países europeus. Em geral, a estrutura e dinâmica das normas jurídicas de outros países não é muito diferente do ordenamento jurídico brasileiro, fato que não se repete quanto aos *documentos jurídicos*. Por isso, para a modelagem desse tipo de documento, foram pesquisados essencialmente trabalhos nacionais, dos quais destacam-se dois: Bueno [24] e Vanzin *et al.* [74].

Na área de modelagem normativa, destaca-se a dissertação de Mestrado de Carvalho Junior [25], a tese de doutorado de Lima [54] e o *Projeto Ontojuris*¹. Este trata-se de um consórcio entre Brasil, Chile, Espanha e Argentina para facilitar o acesso à legislação referente a Propriedade Intelectual, Direito do Consumidor e Direito Eletrônico. A base do sistema são as ontologias ONTOJURIS e ONTOTRIB [2].

O foco dos dois últimos trabalhos é a semântica das fontes, através da modelagem do conhecimento por ontologias, fugindo do escopo desta dissertação. Por isso, serão detalhados a seguir apenas os três trabalhos brasileiros mais relevantes na modelagem estrutural normativo-jurídica: Bueno [24], Vanzin *et al.* [74] e Carvalho Junior [25]. O projeto LexML Brasil [50] também se enquadra nessa categoria, porém será discutido posteriormente, na Seção 3.1.3.

Bueno [24]

No trabalho de Bueno é proposto um modelo de recuperação de documentos jurídicos com características semelhantes, a fim de auxiliar os operadores do Direito na solução de problemas jurídicos. A abordagem não utiliza apenas técnicas de Inteligência Artificial, mas também um *vocabulário controlado* e um *dicionário de termos normativos*. Além da RI, também é discutido o processo de extração do conhecimento dos documentos.

Segundo a autora, em uma consulta utilizando as ferramentas tradicionais (geralmente sítios), é retornado um grande número de resultados não relevantes. Isso se deve à complexidade semântica do conteúdo dos documentos, o que justifica a utilização de técnicas de IA para recuperar eficientemente a informação. O âmbito do desenvolvimento e validação da pesquisa foi o *Direito Criminal*.

A abordagem definida na dissertação de Bueno utiliza o *Raciocínio Baseado em Casos* (RBC) na recuperação de casos anteriores e semelhantes, sendo composta pelas seguintes partes [24]:

1. *Definição da estrutura de representação de casos legais e identificação de informação relevante para a recuperação de jurisprudência apropriada;*
2. *Extração automática da informação do texto legal na inclusão de novos casos jurídicos na base de casos;*

¹<http://www.i3g.org.br/ontojuris> (Acesso em Junho/2010)

3. *Processo de recuperação baseado na similaridade através do matching parcial entre o ajuste da situação fornecido inicialmente em linguagem natural, e os casos na base de casos, usando uma medida de similaridade.*

Para realizar a recuperação da informação, foram determinados os *índices* relevantes de um acórdão: *número do acórdão, data de publicação, local, relator, tipo de recurso, expressões jurídicas, resultado e tipificação*. De acordo com a análise da autora, esses índices estão distribuídos da seguinte forma no acórdão:

Estrutura	Descrição	Índices
A	Informações do acórdão	Número do acórdão, data de publicação, local, relator, tipo do recurso
B	Ementa do acórdão	Expressões jurídicas
C	Nome das partes e a decisão do recurso	Resultado
D	Principal estrutura do acórdão. Contém os acontecimentos de fato e de direito relatados no processo, além da fundamentação da decisão	Expressões jurídicas

Tabela 3.1: Estrutura de um acórdão (adap. de Bueno [24])

O intuito do tipo de estruturação representado na Tabela 3.1 é a extração do conhecimento dos documentos, através da *localização* desses índices. A partir da observação de acórdãos do Tribunal de Santa Catarina, foram definidos os padrões para a extração de alguns atributos [24]:

- *Data de publicação*: descrita na primeira linha do documento, precedido da palavra-chave “DATA”;
- *Número*: descrito na segunda linha do documento, é a expressão posterior ao identificador “n.”;
- *Local*: também descrito na segunda linha do documento, é a expressão posterior à palavra-chave “de”;
- *Relator*: descrito na terceira linha do documento, é a expressão posterior ao identificador “Relator.”;

- *Resultado*: descrito na estrutura C da Tabela 3.1, são as expressões que indicam conceder ou denegar o pedido e suas variantes, posterior ao identificador “ACÓRDAO,”.

O trabalho de Bueno ainda apresenta duas outras formas de extração do conhecimento: através do uso de um vocabulário controlado e dicionário de termos normativos e por inferência. Tanto essas formas quanto questões da recuperação da informação tratadas no trabalho fogem ao escopo desta dissertação.

A principal contribuição do trabalho é a metodologia de recuperação de informação em documentos jurídicos, utilizando técnicas de Inteligência Artificial, mais especificamente o RBC. São contribuições secundárias a definição dos índices de recuperação e das três formas de extração de conhecimento.

Como primeiro ponto fraco, apenas os documentos jurídicos foram contemplados na modelagem proposta, não sendo representada a integração desses com as normas. Além disso, a descrição dos padrões de posição dos índices de recuperação foi feita em linguagem natural, apenas compreensível para seres humanos. A formalização desses padrões é parte fundamental na extração automática dos atributos das fontes.

Vanzin *et al.* [74]

No trabalho de Vanzin *et al.* é descrita uma metodologia para a extração de estruturas dos acórdãos, a fim de tornar possível o armazenamento automático desses documentos jurídicos, originalmente semi-estruturados, em um banco de dados. Essa pesquisa foi aplicada em um sistema de gerenciamento de documentos, chamado *GlocalDoc*. Os acórdãos são descritos em *XML*, seguindo uma estrutura definida pelos especialistas em *DTD*, e ainda definidos conceitualmente através de ontologias.

Os documentos jurídicos utilizados como base para pesquisa foram coletados do Tribunal Regional do Trabalho da 4ª Região no Rio Grande do Sul. Segundo os autores, esses acórdãos possuem uma estrutura bem definida, pois os relatores seguem regras em sua redação. Como descrito na Seção 2.1.2, foram determinadas as subestruturas dos acórdãos em Vanzin *et al.* [74], além de definidos os atributos contidos em cada uma:

- *Cabeçalho*: número do processo, sigla do recurso, relator e data de publicação;
- *Ementa*: tema;

- *Pessoas Envolvidas*: recurso, procedência, reclamante e reclamado;
- *Relatório*: tema, referências legislativas;
- *Voto*: tema, referências legislativas;
- *Sentença*: órgão julgador, tribunal, quantidade de votos, decisão e referências legislativas;
- *Encerramento*: local, data e participantes.

As subestruturas *Ementa*, *Relatório* e *Voto* possuem o atributo *tema do acórdão* em comum, o qual trata de assuntos abordados no acórdão. O atributo *referências legislativas*, presente nas subestruturas *Relatório*, *Voto* e *Sentença*, representa as normas ou itens da norma citados no documento jurídico.

Em nível conceitual, foi desenvolvido o modelo abstrato da estrutura dos acórdãos, através de uma ontologia de domínio. Foram definidas como *Conceito* as classes abstratas ou concretas, elementares ou compostas, reais ou fictícias. Os atributos de cada conceito são representados entre chaves e as relações entre os conceitos são do tipo *PART-OF*, como ilustra a Figura 3.1.

Os documentos jurídicos também foram descritos em XML, por ser uma linguagem de marcação aceita como padrão atualmente, facilitando a troca dos dados entre diversos tipos de aplicativos, mas não apenas por isso. Através do XML também é possível representar as características descritas na ontologia, além de definir a estrutura dos acórdãos utilizando o DTD associado.

São discutidos ainda no trabalho os métodos utilizados para a extração automática de valores de cada atributo, de acordo com a natureza das informações. O primeiro método é o de *extração baseado na localização da informação*, no qual o conteúdo de cada atributo é identificado pela sua localização no documento e por palavras-chaves. Exemplos da localização dos atributos de um acórdão são apresentados na Tabela 3.2.

No segundo método de extração, é empregado o uso de *dicionário*, que descreve as características dos termos - a exemplo de morfologia e sintaxe -, possibilitando a definição do contexto no qual eles estão inseridos. O valor do atributo *tema*, presente nas subestruturas

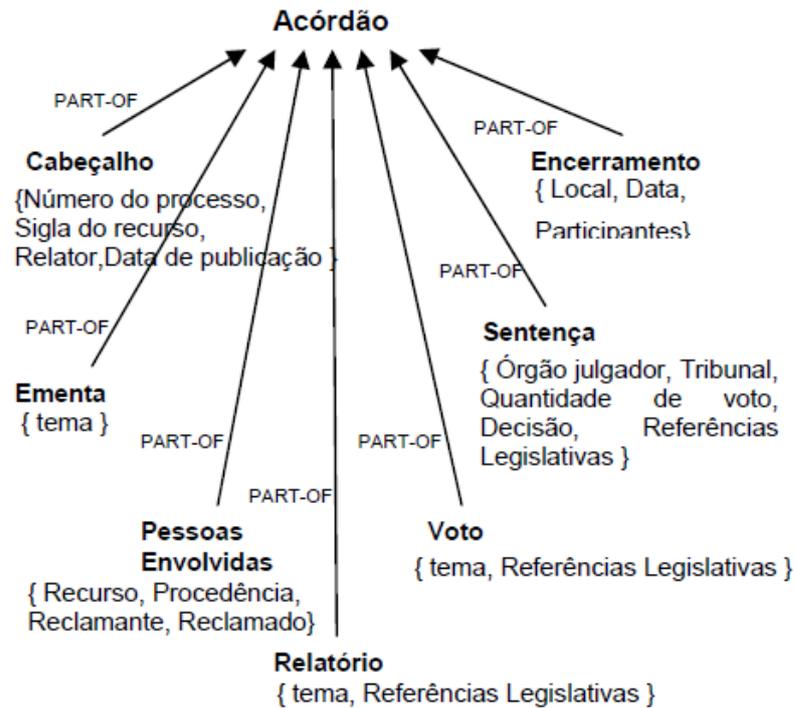


Figura 3.1: Ontologia de domínio, relativa ao acórdão (Vanzin *et al.* [74])

Subestrutura	Atributos	Localização
Cabeçalho	Número do Processo Sigla do Recurso Data de Publicação	Primeira linha, antecedido por “Número do Processo” Entre parênteses depois do número do processo Terceira linha, após “Data de Publicação”
Sentença	Órgão Julgador Tribunal Quantidade de voto	Após a palavra-chave “Acordam” Após a identificação do atributo <i>Órgão Julgador</i> Após o atributo <i>Tribunal</i> . Utiliza-se uma lista de expressões para extrair o valor desse atributo.

Tabela 3.2: Localização dos atributos de um acórdão (adap. de Vanzin *et al.* [74])

Ementa, *Relatório* e *Voto*, é extraído através desse método, mais precisamente com o auxílio de um *Vocabulário Controlado*.

As principais contribuições do trabalho são: a proposta de modelagem usando ontologia, DTD e XML dos documentos jurídicos e a metodologia de extração automática de valores dos atributos. Como contribuição secundária, a definição da localização dos atributos dos acórdãos.

Os pontos fracos deste trabalho são parecidos com os destacados no trabalho anterior (Bueno [24]). Apesar da representação da integração dos acórdãos com as normas através do atributo *referências legislativas*, estas fontes não foram modeladas. Além disso, a

descrição da localização dos atributos foi feita em linguagem natural, incompreensível para uma máquina.

Carvalho Junior [25]

Segundo o autor, o objetivo do trabalho é “definir um DTD para ser utilizado como padrão para construção de documentos XML de qualquer um dos seis (6) atos normativos referidos no Art. 59 da Constituição Federal”. Apesar de Carvalho Junior se referir a seis atos normativos, existem sete atos no artigo 59 da Carta Magna: emendas à Constituição, leis complementares, leis ordinárias, leis delegadas, medidas provisórias, decretos legislativos e resoluções.

O padrão proposto no trabalho utiliza DTD e XML, a fim de não apenas representar a estrutura das normas, mas também os seus relacionamentos e aspectos temporais. Os dados são representados por atributos, definidos como *CDATA*. Caso o atributo não permita valores nulos, ele também deve estar definido como *#REQUIRED*; caso contrário, ele recebe o identificador *#IMPLIED* [25].

O código fonte 3.1 descreve os atributos das partes iniciais e finais de uma norma, através dos respectivos prefixos *pp* e *pf*. No código-fonte 3.2, a LCP95 é representada em XML, seguindo a descrição DTD² anteriormente definida. Os atributos apresentados são: epígrafe, ementa, preâmbulo, local, data, fecho e autoridade(s).

Código Fonte 3.1: Descrição das partes iniciais e finais de uma norma em DTD (Carvalho Junior [25])

```

1 <!ELEMENT norma ( autoridade + )>
2 <!ATTLIST norma
3     pp_epigrafe_tipo      CDATA #REQUIRED
4     pp_epigrafe_numero   CDATA #REQUIRED
5     pp_epigrafe_data     CDATA #REQUIRED
6     pp_ementa            CDATA #REQUIRED
7     pp_preambulo        CDATA #REQUIRED
8     pf_local             CDATA #REQUIRED
9     pf_data              CDATA #REQUIRED
10    pf_fecho             CDATA #REQUIRED>
11
12 <!ELEMENT autoridade EMPTY>
13 <!ATTLIST autoridade
14     pf_autoridade_nome   CDATA #REQUIRED
15     pf_autoridade_cargo  CDATA #REQUIRED>

```

²Percebe-se, no entanto, um erro do autor ao descrever o atributo *autoridade_cargo* como *#REQUIRED*, pois seu valor está como nulo (vazio) no exemplo seguinte.

Código Fonte 3.2: XML resultante da descrição em DTD (Carvalho Junior [25])

```

1 <norma>
2     pp_epigrafe_tipo="Lei Complementar"
3     pp_epigrafe_numero="95"
4     pp_epigrafe_data="26/02/1998"
5     pp_ementa="Dispõe sobre a elaboração, a redação..."
6     pp_preambulo="O PRESIDENTE DA REPÚBLICA Faço saber que..."
7     pf_local="Brasília"
8     pf_data="26/02/1998"
9     pf_fecho="177º da Independência e 110º da República">
10    <autoridade
11        pf_autoridade_nome="Fernando Henrique Cardoso"
12        pf_autoridade_cargo="Presidente da República" />
13    <autoridade
14        pf_autoridade_nome="Íris Rezende"
15        pf_autoridade_cargo="" />
16 </norma>

```

A *parte normativa* (prefixo *pn*) que compreende os itens de uma norma também é descrita, vide o código fonte 3.3:

Código Fonte 3.3: Descrição da parte normativa de uma norma em DTD (Carvalho Junior [25])

```

1 <!ELEMENT artigo (paragrafo*|inciso*)>
2 <!ATTLIST artigo
3     pn_ar_numero CDATA #REQUIRED
4     pn_ar_texto CDATA #IMPLIED>
5 <!ELEMENT paragrafo (inciso*)>
6 <!ATTLIST paragrafo
7     pn_pa_numero CDATA #REQUIRED
8     pn_pa_texto CDATA #IMPLIED>
9 <!ELEMENT inciso (alinea*)>
10 <!ATTLIST inciso
11     pn_in_numero CDATA #REQUIRED
12     pn_in_texto CDATA #IMPLIED>
13 <!ELEMENT alinea (item*)>
14 <!ATTLIST alinea
15     pn_al_numero CDATA #REQUIRED
16     pn_al_texto CDATA #IMPLIED>
17 <!ELEMENT item EMPTY>
18 <!ATTLIST item
19     pn_it_numero CDATA #REQUIRED
20     pn_it_texto CDATA #IMPLIED>

```

Esse exemplo corresponde à representação da estrutura do artigo, ilustrada anteriormente na Figura 2.1. Não foi explicado por que o atributo “texto” dos elementos da norma estão definidos como *#IMPLIED*, o que significa que eles podem conter valores nulos. Dessa forma, a modelagem está incorreta, pois não faz sentido definir um elemento normativo sem um texto associado.

Os relacionamentos externos também foram representados em DTD, porém apenas as *modificações*: alterações e inclusões. Das alterações fazem parte as *revogações* e a *designação de novas redações*, enquanto a *inclusão* adiciona novos elementos (artigos, parágrafos, etc.) à norma. A proposta do trabalho define que as modificações devem ser representadas tanto na norma que as executou quanto na que as recebeu. Os relacionamentos são descritos através do elemento `<refrecord>`:

Código Fonte 3.4: Elemento de referência `<refrecord>` (Carvalho Junior [25])

```

1 <!ELEMENT refrecord (artigo*|paragrafo*|inciso*|alinea*|item*)>
2 <!ATTLIST refrecord
3     pp_epigrafe_tipo      CDATA #REQUIRED
4     pp_epigrafe_numero   CDATA #REQUIRED
5     pp_epigrafe_data     CDATA #REQUIRED
6     acao (altera|inclui) #REQUIRED>

```

O atributo *epígrafe* informa qual a norma relacionada, enquanto o atributo *acao* define qual a modificação executada. O elemento `<refrecord>` tanto está presente na norma alteradora quanto na norma alterada. Seguem dois exemplos, referentes à representação de relacionamentos de alteração e inclusão:

Código Fonte 3.5: LCP107 altera o inciso II da LCP95 (adap. de Carvalho Junior [25])

```

1 LCP107
2 <artigo
3     pn_ar_numero="12">
4     ...
5     <inciso
6         pn_in_numero="II"
7         pn_in_texto="mediante revogação parcial;">
8         <refrecord
9             pp_epigrafe_tipo="Lei Complementar"
10            pp_epigrafe_numero="95"
11            pp_epigrafe_data="26/02/1998"
12            acao=altera>
13         </refrecord>
14     </inciso>
15     ...
16 </artigo>
17
18 LCP95
19 <artigo
20     pn_ar_numero="12"
21     pn_ar_texto="A alteração da lei será feita:">
22     ...
23     <inciso
24         pn_in_numero="II"
25         pn_in_texto="há hipótese de revogação;">
26         <refrecord
27             pp_epigrafe_tipo="Lei Complementar"
28             pp_epigrafe_numero="107"

```

```

29             pp_epigrafe_data="26/04/2001"
30             acao=altera>
31         </refrecord>
32     </inciso>
33     ...
34 </artigo>

```

No primeiro exemplo (3.5), a LCP107 alterou o inciso II da LCP95. Uma crítica a essa representação é a similaridade entre os elementos *<refrecord>* das normas, requerendo uma análise minuciosa dos demais elementos a fim de identificar quem está modificando ou sendo modificado. Outra limitação é que não há registro do que foi modificado, informação que deveria fazer parte do elemento *<refrecord>*.

No exemplo seguinte (3.6) é realizada a inclusão do parágrafo 1º no artigo 12 da LCP95, e as mesmas críticas ao exemplo anterior valem para este. O registro da inclusão é pobre e bastante similar nas duas normas, sendo o único ponto positivo a representação do elemento incluído na norma modificada, mesmo que seja apenas seu número - o recomendado seria número e texto.

Código Fonte 3.6: LCP107 inclui o parágrafo 1º no artigo 12 da LCP95 (Carvalho Junior [25])

```

1 LCP107
2 <artigo
3     pn_ar_numero="12">
4     ...
5         <paragrafo
6             pa_pa_numero="1"
7             pa_pa_texto="O termo 'dispositivo' mencionado
8                 nesta Lei refere-se a artigos, parágrafos,
9                 incisos, alíneas o itens.">
10            <refrecord
11                pp_epigrafe_tipo="Lei Complementar"
12                pp_epigrafe_numero="95"
13                pp_epigrafe_data="26/02/1998"
14                acao=inclui>
15            </refrecord>
16        </paragrafo>
17    ...
18 </artigo>
19
20 LCP95
21 <artigo
22     pn_ar_numero="12"
23     pn_ar_texto="A alteração da lei será feita:">
24     <refrecord
25         pp_epigrafe_tipo="Lei Complementar"
26         pp_epigrafe_numero="107"
27         pp_epigrafe_data="26/04/2001"
28         acao=inclui>

```

```
27         <paragrafo
28             pa_pa_numero="1">
29         </paragrafo>
30     <refrecord>
31     ...
32 </artigo>
```

A principal contribuição do trabalho é a proposta de um padrão independente de plataforma para o armazenamento das normas em formato digital. No entanto, além das falhas supracitadas no modelo, não foram representados outros tipos de relacionamentos, a exemplo das *citações*, tanto relativas a outras normas quanto a documentos jurídicos. Por sinal, estes nem foram modelados, pois o foco do trabalho foi apenas a legislação, ou seja, as fontes normativas.

Outro ponto negativo é a especificação pobre das operações de alteração e inclusão, além de gerar dúvidas em relação ao sentido de cada operação. Também não foi discutido como é armazenado o registro das versões das normas e de seus elementos, um ponto fundamental na modelagem desse tipo de fonte.

3.1.2 Dinâmica das Fontes

As fontes do Direito - especialmente as normativas - possuem características únicas em relação a outros tipos de documentos, das quais destacam-se duas: *temporalidade* e *relacionamentos*. As leis sofrem modificações constantemente, a fim de que se adequem aos valores e costumes da sociedade na qual estão inseridas. Essas modificações refletem diretamente na forma como são julgados os problemas jurídicos, pois as jurisprudências estão intrinsecamente relacionadas com as normas, estas constituindo a base legal para as decisões dos magistrados.

Nesse contexto, serão discutidos nesta seção os principais trabalhos que tratam **exclusivamente** da dinâmica das fontes, ou seja, como elas interagem entre si, como são feitas suas modificações e como são tratadas suas versões ao longo do tempo. A discussão desses aspectos é fundamental, a fim de entender o motivo da especificação de algumas características do modelo proposto nesta dissertação. Serão apresentados três trabalhos: dois relativos à temporalidade das fontes - Grandi *et al.* [42] e Arnold-Moore [5] - e um relativo aos seus relacionamentos - Zhang & Koppaka [76].

Grandi et al. [42]

O primeiro trabalho relativo à temporalidade das fontes tem como foco a modelagem e gerenciamento temporal de documentos normativos em XML. É discutida a representação da evolução de uma norma, tomando como base quatro dimensões temporais: *publicação*, *validade*, *eficácia* e *tempo transacional*.

- *Publicação*: data da publicação da norma;
- *Validade*: intervalo em que a norma está vigente;
- *Eficácia*: intervalo em que a norma pode ser aplicada ao caso concreto;
- *Tempo Transacional*: intervalo em que a norma está armazenada na base de dados.

Na concepção dos autores, as dimensões *validade* e *eficácia* possuem o valor semântico de *tempo válido*, conceito associado aos bancos de dados temporais, apresentado na Seção 2.2.2. A diferença é que o conceito de *eficácia* é aplicado quando a norma sofre *derrogação*, ou seja, quando apenas parte dela é revogada. Nesse caso, o intervalo de validade da norma é fechado, porém o de eficácia continua aberto.

Um conceito introduzido no trabalho é o de *pertinência temporal* de uma versão, representada pelo espaço tridimensional *validade* \times *eficácia* \times *tempo transacional*. Tendo como base o modelo estrutural de uma norma, a pertinência está presente em todos os níveis: norma, artigos, parágrafos, etc. A configuração inicial é definida pelos *nós pais* repassando a pertinência para os *nós filhos*, porém caso haja uma modificação em um destes, sua pertinência será modificada.

O esquema XML proposto na Figura 3.2 para a representação das normas é a extensão de um dos DTDs especificados no projeto *Norme in Rete* [56], financiado pelo governo italiano. As letras *R* e *O* ao lado dos atributos significam respectivamente *required* (requerido) e *optional* (opcional), e as caixas pontilhadas representam elementos opcionais.

O elemento *contents* (conteúdo) possui os atributos referentes às propriedades temporais: *publication* (data de publicação), *vt_Start* e *vt_End* (intervalo de validade), *tt_Start* e *tt_End* (intervalo transacional) e *et_Start* e *et_End* (intervalo de eficácia).

O elemento *ver* corresponde à versão de cada elemento, cujos atributos são *num* (número da versão) e *an_ref* (referência para o responsável pela criação de uma nova versão da

norma). Cada versão possui um elemento *TA*, o qual representa o intervalo tridimensional $[vt_Start, vt_End] \times [tt_Start, tt_End] \times [et_Start, et_End]$.

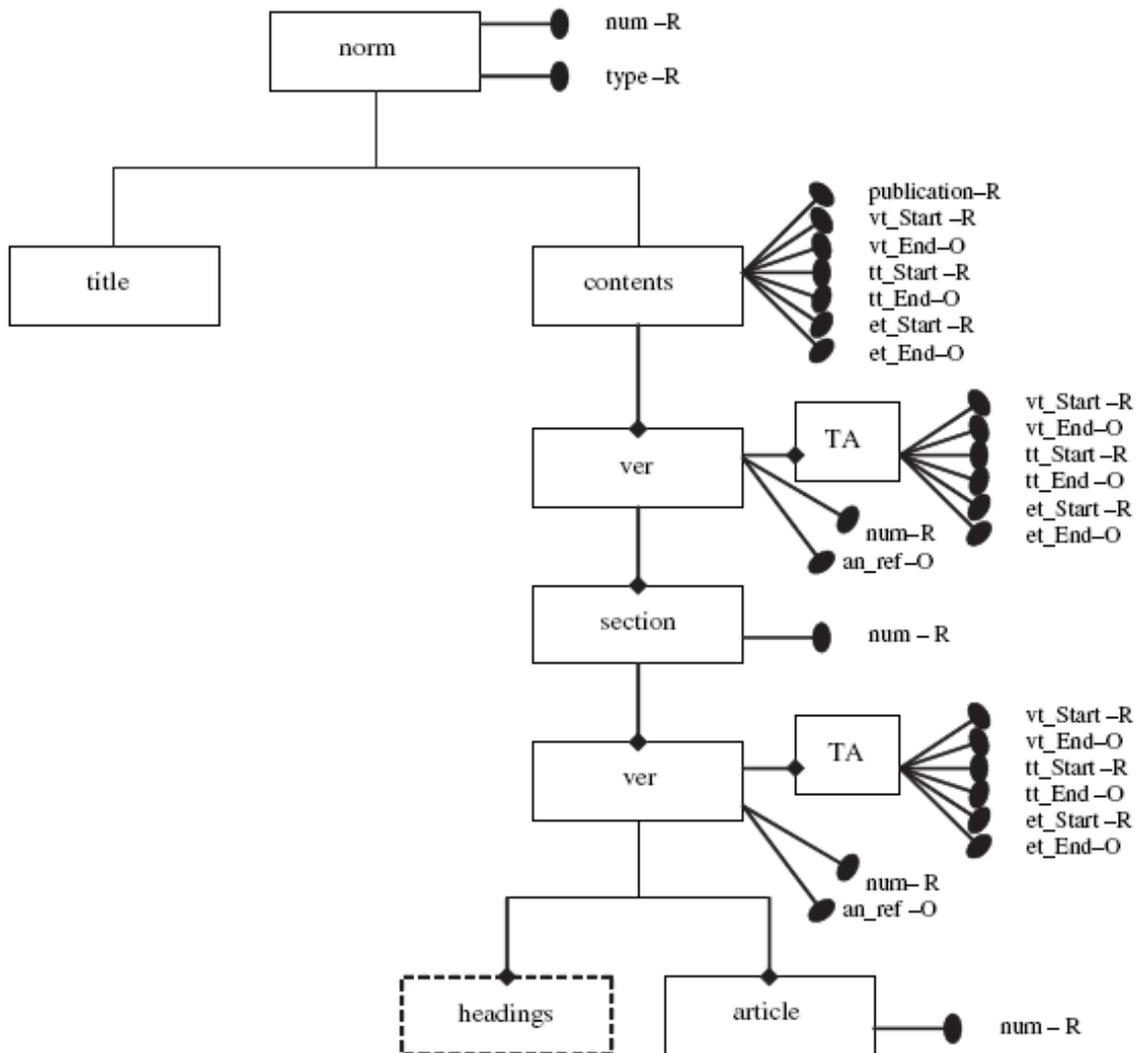


Figura 3.2: Esquema XML de uma norma (adap. de Grandi *et al.* [42])

Além da representação temporal, o trabalho também propõe duas operações para o gerenciamento das modificações das normas, em documentos XML: *changeText* e *changeTime*. A primeira implementa as modificações textuais no documento, enquanto a segunda implementa as modificações na persistência temporal, ambas atuando sobre o elemento alterado.

A operação *changeText* recebe como parâmetros: (i) a norma ou elemento da norma a ser modificado, (ii) validade e eficácia da nova versão, (iii) o novo texto, (iv) número da versão e (v) referência para o elemento modificador. Como saída, o XML da norma é modificado, criando-se uma nova versão do elemento com os atributos temporais definidos.

A operação *changeTime* também recebe como entrada o caminho do elemento passivo (o qual recebe a ação), além de um ponto temporal tridimensional (*vt*, *et*, *tt*) para identificar a versão a ser modificada e um elemento temporal bidimensional contendo a nova validade e eficácia da versão.

Outra contribuição do trabalho é a definição de consultas temporais, visando à reconstrução das normas em um dado ponto no tempo. Essa operação vai auxiliar o operador do Direito a não escolher fontes que já estejam defasadas, a fim de garantir a correteza de sua fundamentação em um caso concreto. A problemática envolvendo a temporalidade das fontes foi discutida na Seção 1.1. A sintaxe das consultas temporais possuem o seguinte formato:

Código Fonte 3.7: Reconstrução temporal em XQuery (Grandi *et al.* [42])

```
1 FOR $a IN path
2 WHERE constraints on $a
3 RETURN const-tree(document($a), temporal specs)
```

A linguagem utilizada no código 3.7 foi o *XQuery*³, o padrão SQL para consultas em XML, adotado pela *W3C*⁴ (World Wide Web Consortium). O identificador *\$a* representa o nó a ser pesquisado e o operador *const-tree()* é o retorno da função, ou seja, a norma reconstruída. Ele recebe como parâmetros os documentos que possuem o nó de interesse - *document(\$a)* - e as especificações temporais - *temporal specs*.

Essas especificações utilizam as quatro dimensões temporais propostas no trabalho, na forma [42]: *dimension [NOT] op VALUE*. O identificador *dimension* pode assumir os valores *PUBLICATION*, *VALIDITY*, *EFFICACY* e *TRANSACTION*, enquanto o operador *op* pode ser: *PRECEDES*, *FOLLOWS*, *=*, *OVERLAPS*, *MEETS*, *MET-BY*, *CONTAINS* e *CONTAINED-IN*. Esses operadores tanto comparam intervalos entre si quanto datas e intervalos. Exemplo de uma consulta temporal:

Código Fonte 3.8: Consulta temporal em XQuery (adap. de Grandi *et al.* [42])

```
1 FOR $a IN // article / paragraph
2 WHERE contains ($a, "crimes cibernéticos")
3 RETURN const-tree(document($a),
4     VALIDITY CONTAINS "1999-01-01" and
5     PUBLICATION PRECEDES "2001-01-01")
```

³<http://www.w3.org/TR/xquery> (Acesso em Junho/2010)

⁴<http://www.w3.org> (Acesso em Junho/2010)

No exemplo 3.8 é ilustrada uma busca por versões de normas cujos parágrafos contêm a expressão *crimes cibernéticos*; as normas devem ser válidas na data 01/01/1999 e sua publicação anterior a 01/01/2001. Por não terem seus valores definidos na consulta, as dimensões *tempo transacional* e *eficácia* recebem os valores padrão: a primeira é atribuída como *NOW* (agora) e à segunda é atribuída a expressão relativa à dimensão temporal *validade* (*EFFICACY CONTAINS "1999-01-01"*).

São várias as contribuições do trabalho: a modelagem temporal das fontes normativas, as operações *changeText* e *changeTime* e a definição de consultas de reconstrução temporal; no entanto há de se fazer uma ressalva em relação às dimensões temporais definidas. Como discutido anteriormente, no ordenamento jurídico brasileiro os conceitos de *vigência* e *eficácia* se confundem, enquanto há uma diferenciação com o conceito de *validade*.

Sendo assim, fazendo um paralelo entre o trabalho de Grandi *et al.* e esta dissertação, a modelagem proposta nesta contempla três dimensões temporais: *data de publicação*, *validade* e *vigência*. O *tempo transacional* foi considerado dispensável para fins deste trabalho, enquanto a validade corresponde ao período que vai desde a publicação da norma até o início de sua vigência. A discussão dos aspectos temporais das fontes foi descrita na Seção 2.1.1 do Capítulo 2.

Arnold-Moore [5]

O trabalho em pauta discute aspectos de armazenamento, manutenção, apresentação e citação de fontes normativas em repositórios *point-in-time* (ponto-no-tempo). Nesse tipo de repositório são armazenadas as diferentes versões das normas, provenientes das alterações feitas por outras normas. A cada versão são associados o tempo de início (*ValidStart*) e fim (*ValidEnd*) de sua validade.

Há várias formas de armazenar as alterações das normas, e para lidar com grandes volumes de texto de forma eficiente, é necessário evitar o armazenamento redundante dos dados. Sendo assim, a sugestão do autor é que se divida a lei em *fragmentos*, armazenando apenas as versões desses fragmentos modificados, em vez de salvar toda lei novamente com apenas uma parte alterada. Para atualizar a base de dados, basta adicionar as novas partes e definir o fim da validade das partes substituídas e o intervalo das partes substitutas.

Outro ponto discutido no artigo é a forma como são apresentadas as modificações para

os usuários. É citado o exemplo de como eram publicadas as alterações no sítio *Tasmanian Legislation Online*⁵, no qual são publicadas as normas da Tasmânia. A interface era simples, constituída por uma grande tabela com três colunas, do tipo:

Atos que modificaram a Lei Criminal 4781		
Ato modificador	Número e ano	Data da alteração
Ato de Revisão Legal 2010	No. 25 de 2010	30.03.2010

Tabela 3.3: Modificações nas normas da Tasmânia (adap. de Arnold-Moore [5])

Através dos valores dos atributos *Ato modificador* e *Data de alteração*, era possível visualizar os textos dos atos modificadores e a versão principal da norma modificada, respectivamente. Em um outro modo de visualização, era possível navegar pela versão dos fragmentos através de setas orientadas para a direita e esquerda, com um relógio desenhado em seu interior. Esse conceito foi adotado devido aos símbolos universais de *Play* e *Rewind*, utilizados em tocadores de música e VCRs (*Video Cassette Recorder*) [5].

São discutidos ainda no trabalho os conceitos de bancos de dados temporais aplicados ao âmbito legislativo, mais especificamente os conceitos de *tempo válido* e *tempo transacional*. Discussão semelhante foi apresentada na Seção 2.2.2 e no trabalho anteriormente citado (Grandi *et al.* [42]). A última seção do trabalho trata de aspectos temporais das citações em documentos legislativos, pois segundo a Associação dos Bibliotecários Legais Americanos (*American Association of Law Librarians*, AALL), toda citação que não possui informações temporais associadas é **incompleta** [5].

Exemplo dessa afirmação é a citação a uma norma feita por um documento jurídico: um documento jurídico *DJ1* tem sua fundamentação a partir de uma citação à versão *VI* de uma norma *N*, a qual foi posteriormente atualizada para a versão *V2*; caso um outro documento jurídico *DJ2* utilize a mesma fundamentação de *DJ1*, *DJ2* estará inconsistente temporalmente. Por isso, é essencial saber quando foi feita a citação por *DJ1*, sabendo assim à qual versão da norma *N* ele está se referindo.

Apesar de destacar a importância da temporalidade para as fontes normativas, o autor abordou apenas superficialmente o tema. Merecem destaque as idéias de armazenar apenas as novas versões dos fragmentos das normas e de associar o tempo às citações feitas entre documentos. Em um artigo mais teórico, Arnold-Moore [6] detalha o processo de alteração

⁵<http://www.thelaw.tas.gov.au> (Acesso em Junho/2010)

legislativa da Tasmânia, comparando-o ao de outros países anglofônicos.

Zhang & Koppaka [76]

Uma característica já mencionada das fontes do Direito é a grande quantidade de referências feitas e recebidas por elas. O trabalho de Zhang & Koppaka destaca esse ponto e propõe a modelagem desses relacionamentos através de uma *Rede Semântica de Citações*, incorporada a uma ferramenta de auxílio a pesquisas em documentos jurídicos.

A proposta é que, dado um caso concreto *raiz*, o usuário possa navegar por todos os outros casos citados por ele, e que essa operação possa ser realizada também nesses casos citados, recursivamente. Essa abordagem é baseada no modelo mental do operador do Direito descrito por Sutton [70], ilustrado na Figura 1.1.

As citações feitas em documentos jurídicos são *multidimensionais*, pois tanto um caso pode citar outros por diferentes propósitos quanto pode receber várias citações por razões diversas [76]. Em um exemplo citado em Zhang & Koppaka [76], um caso nos Estados Unidos recebeu centenas de citações, agrupadas em pelo menos dez categorias. Isso demonstra que nem todas essas citações eram semanticamente próximas, apesar de apontarem para o mesmo documento.

Para lidar com essa situação, a proposta de Zhang & Koppaka é dividir a rede semântica de citações em sub-redes, de acordo com o assunto legal tratado, constituindo um dos desafios para os pesquisadores. Para resolver esse problema, foi utilizada a técnica de *Reason for Citing* (RFC), a qual se baseia na análise do texto ao redor da citação para detectar o seu ramo semântico na árvore de citações.

Através dessa técnica, o usuário pode navegar pelas citações de forma refinada, escolhendo quais os casos relevantes de acordo com os assuntos de interesse. Na Figura 3.3 são ilustrados três casos, com as setas vermelhas indicando os assuntos abordados por eles e as verdes representando as citações feitas. No exemplo em questão, um *Caso Inicial* cita o *Caso1*, mais especificamente o *Assunto1* tratado por ele. O próprio *Caso Inicial* é citado pelo *Caso2*, o qual referencia o *Assunto5*.

Ainda no trabalho de Zhang & Koppaka, é apresentado um protótipo desenvolvido para testar a eficácia do método. Além disso, um módulo de visualização da rede semântica de citações faz parte desse protótipo, permitindo que o usuário consulte os casos de forma

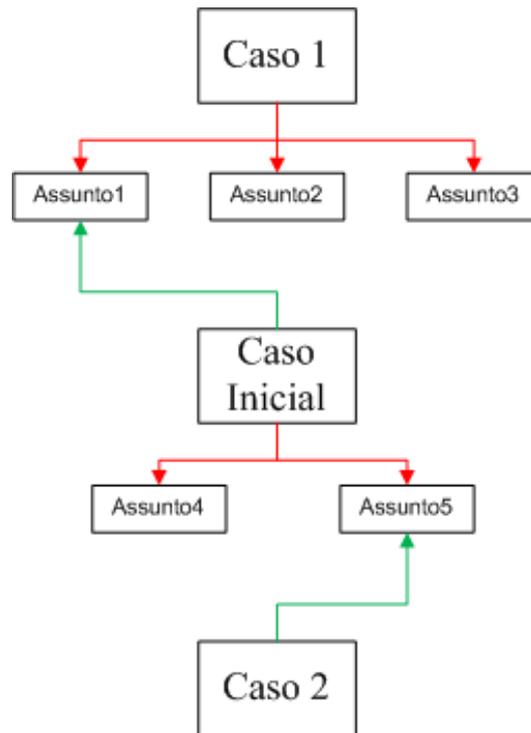


Figura 3.3: Rede semântica simplificada de citações (adap. de Zhang & Koppaka [76])

bidirecional, ou seja, a partir de um caso é possível acessar os casos citados por ele e os casos que o citam. A avaliação da técnica de RFC aplicada no protótipo foi feita por especialistas (advogados), os quais avaliaram se os assuntos foram extraídos de forma correta dos casos. O protótipo obteve avaliação geral de 91,05% de precisão nesse quesito [76].

Como ponto forte do trabalho, pode-se citar a representação do modelo mental dos operadores do Direito, através da modelagem de uma rede semântica de casos semelhantes. Para isso, foi empregada uma análise da semântica das informações para a extração de citações, a fim de identificar os casos semelhantes a partir dos assuntos tratados. Essa abordagem não foi encontrada em nenhum outro trabalho no levantamento bibliográfico.

Um ponto fraco detectado foi a ausência da representação das citações de documentos jurídicos a fontes normativas, pois esse tipo de referência é extremamente utilizado, muitas vezes até mais que a referência a outros casos. Também não foi modelada a temporalidade das citações, condição já mencionada como fundamental pela Associação dos Bibliotecários Legais Americanos. A consistência temporal da fundamentação de um caso é um fator determinante para definir sua relevância para outros casos.

3.1.3 Modelos e SGCLs Pátrios

Serão apresentados nesta seção os principais modelos e SGCLs propostos em diferentes países, inclusive alguns provenientes de investimentos governamentais. Em busca de uma padronização na representação das fontes do Direito, existem tanto propostas em nível de países quanto continentes, dos quais serão detalhados: *Agora-Lex* (Bélgica), *AKOMA NTOSO* (África), *EnAct* (Tasmânia, Austrália), *Estrella* (Europa) e *LexML Brasil*.

A escolha desses trabalhos foi baseada em dois critérios: qualidade das idéias apresentadas e heterogeneidade entre os modelos. O primeiro visa à seleção das melhores práticas de modelagem propostas na literatura, enquanto o segundo procura eliminar a redundância em modelos semelhantes. Quando encontrados modelos nessa situação, foi escolhido o mais completo entre eles para ser discutido.

Agora-Lex

O projeto *Agora-Lex* foi fundado no ano de 2000, pelo Escritório Federal Belga de Assuntos Culturais, Técnicos e Científicos, e desenvolvido pelo Centro Interdisciplinar de Direito e Tecnologia da Informação da *Katholieke Universtiteit Leuven* em parceria com a *Université Libre de Bruxelles* [61], ambas situadas na Bélgica.

Os objetivos do projeto são descritos em Logghe *et al.* [55], sendo o objetivo principal desenvolver um SGCL para o governo belga. Como objetivos secundários, a modelagem com foco no gerenciamento de versões e a definição de uma metodologia para a troca de dados entre diferentes bases de dados federais. Ainda segundo Logghe *et al.* [55], são requisitos do sistema:

1. *Versionamento das fontes*: além de armazenar as diferentes versões das normas e de seus elementos, o sistema deve ser capaz de oferecer ao usuário a opção de visualizá-las;
2. *Interface Web*: visando à facilitação do acesso ao sistema, a interface desse deve ser desenvolvida com tecnologias Web;
3. *Manutenção automática das informações*: pelo fato de as leis serem fundamentalmente mutáveis, é necessário que o sistema se mantenha sempre atualizado. No entanto, essa é uma tarefa dispendiosa de ser realizada manualmente;

4. *Comunicação entre bases de dados*: devido à estrutura legislativa da Bélgica, a qual possui diferentes governos regionais, há várias bases de fontes legislativos. Sendo assim, o sistema precisa oferecer acesso unificado a todas elas, provendo uma interface de comunicação.

Para representar a temporalidade dos elementos de uma norma, foram modelados dois atributos: *Start date* e *End date*. Também foram determinados os atributos *Previous version link* e *Next version link* para a navegação entre as versões antigas e atuais dos elementos, respectivamente. A primeira falha detectada no modelo é que a temporalidade das normas não foi representada, apenas de seus elementos. Além disso, não foram representados todos os componentes da norma, a fim de simplificar o sistema [55].

A interface do sistema foi projetada para dar suporte a consultas usando atributos textuais e temporais, sendo utilizadas para a reconstrução de uma norma em um dado ponto no tempo. Como última funcionalidade, o *Agora-Lex* visa à integração e troca de informações entre diferentes fontes de dados legais, devido ao caráter descentralizado das normas na Bélgica.

Na modelagem EER (*Extended Entity-Relationship*) apresentada em Logghe *et al.* [55], as principais entidades são: *Estatuto*, *Componente do Estatuto*, *Dispositif*, *Rubric*, *Artigo* e *Modificação*. As duas primeiras representam a estrutura do estatuto, enquanto *Dispositif* está relacionado com o corpo do documento. Os *Artigos* são os elementos de menor granularidade manipulados pelo sistema, podendo ser agrupados em entidades denominadas *Rubric*. Por fim, a entidade *Modificação* representa as versões apenas dos estatutos, o que é uma falha no modelo, por essa entidade não se relacionar também com os elementos do estatuto.

Ainda sobre o projeto *Agora-Lex*, é discutido em Moens & Logghe [61] a importância e complexidade da atualização automática das fontes em um sistema gestor de conhecimento legal, problemática relacionada à manutenção dos dados. Segundo os autores, essa seria uma tarefa tediosa caso fosse efetuada manualmente por especialistas, porém a sua automação não é simples. Precisam ser levados em consideração diversos fatores, desde a aquisição e processamento das fontes, até a garantia de qualidade das novas informações.

Dentre os pontos positivos do trabalho, pode-se destacar a definição de requisitos importantes na implementação de um SGCL, bem como uma modelagem de fontes do Direito com suporte ao versionamento dos estatutos. Como falhas, alguns elementos dos estatutos não foram modelados (a exemplo dos parágrafos) e a esses elementos não foram associa-

das versões, apesar de eles se comportarem de forma semelhante aos estatutos em relação à temporalidade das modificações.

AKOMA NTOSO

A denominação *AKOMA NTOSO* é o acrônimo de *Architecture for Knowledge-Oriented Management of African Normative Texts using Open Standards and Ontologies*, e também significa “corações ligados” na linguagem de um povo do oeste da África. O projeto é parte do *Africa i-Parliament Action Plan*⁶ em parceria com o Departamento de Assuntos Econômicos e Sociais das Nações Unidas⁷. O objetivo dessa iniciativa é representar as fontes do Direito de forma simples, independente de plataforma e que possa ser compreendida por computadores [1].

Tendo como supervisores dois pesquisadores - Fabio Vitali e Monica Palmirani - da Universidade de Bologna, o *AKOMA NTOSO* (AN) foi concebido para ser o padrão de representação de fontes legais e jurídicas dos países africanos, tendo sua primeira versão lançada em 2009. Atualmente é um dos maiores projetos atualmente na área de gerenciamento do conhecimento legal, só podendo ser comparado em questões de abrangência com o projeto *Estrella*, o qual se propõe a padronizar a forma de representar as fontes do Direito em toda a Europa.

O formato adotado para a modelagem das fontes do Direito foi o XML, em conjunto com padrões desenvolvidos visando a uma estruturação documental genérica, porém sendo aplicados ao âmbito legal e jurídico. São descritos no manual de referência do AN [1] cinco categorias de padrões : marcadores, blocos, contêineres, hierarquia e *inlines*. Vale ressaltar o primeiro desses, pois nessa categoria estão incluídos os *metadados*.

Definidos pelo identificador *<meta>*, os metadados representados pelo AN são reflexos da importância de se modelar a temporalidade e alto grau referencial das normas. Essa riqueza de informações é o fator diferencial do modelo do AKOMA NTOSO, o qual o torna importante como objeto de estudo. Nos metadados estão contidas informações referentes a *identificação, publicação, ciclo de vida, modificações e citações* das normas:

- *Identificação (<identification>)*: detalhes sobre a identificação da norma;

⁶<http://www.akomantoso.org> (Acesso em Junho/2010)

⁷<http://www.un.org/en/development/desa> (Acesso em Junho/2010)

- *Publicação* (<publication>): detalhes sobre a publicação original;
- *Ciclo de vida* (<lifecycle>): dados sobre a evolução e versionamento da norma, incluindo informações como datas e identificadores dos elementos modificadores.
- *Modificações* (<analysis>): divididas em duas seções, modificações ativas (<activeModifications>) e passivas (<passiveModifications>). São representadas tanto as origens quanto os destinos das modificações, além das informações temporais;
- *Citações* (<references>): dados sobre as citações (referências) feitas pela norma, identificando detalhadamente o elemento destino da citação. No entanto, uma falha na modelagem é que não foram representadas as citações recebidas pela norma.

Na Figura 3.4 é apresentado um exemplo que ilustra os metadados do ato queniano nº 16 de 6 de Dezembro de 1989, de acordo com o modelo AKOMA NTOSO, em XML. Os subníveis de identificação ontológica FRBRWork, FRBRExpression e FRBRManifestation são especificações da ontologia FRBRoo (*Functional Requirements for Bibliographic Records*) [10], distinção feita de acordo com a semântica dos documentos, a qual não faz parte do escopo desta pesquisa.

Além dos metadados, são definidas três seções na descrição XML da norma, sendo duas preliminares (prefácio <preface> e preâmbulo <preamble>) e uma que representa o corpo da norma, identificada por <body>. No corpo da norma, são utilizados identificadores para cada elemento, a exemplo de artigo (<article>), seção (<section>) e cláusula (<clause>). Nessas seções são descritas apenas o texto e o identificador de cada elemento, pois as demais informações estão contidas nos metadados. Uma seção de exemplos⁸ foi disponibilizada no sítio do AKOMA NTOSO, a fim de ilustrar a representação de algumas fontes do Direito no formato proposto.

O modelo desenvolvido para o AN também prevê a padronização de nomes para os itens. Como o meio de disseminação das fontes é a Internet, foi determinado que as URLs das normas seriam todas relativas, relacionadas com seus elementos de identificação. A adoção de um identificador único (*Uniform Resource Identifier*) para representação de fontes do Direito também foi adotada pelo LexML Brasil [50] e pelo presente trabalho.

⁸<http://examples.ekomantoso.org> (Acesso em Junho/2010)

```

[-] <akomaNtoso>
  [-] <act contains="originalVersion">
    [-] <meta>
      [-] <identification source="#bungeni">
        [-] <FRBRWork>
          <this value="/ke/act/1989-12-06/16/main"/>
          <uri value="/ke/act/1989-12-06/16"/>
          <date date="1989-12-06" name="Generation"/>
          <author href="#parliament" as="#author"/>
        [+ ] <components>
          </FRBRWork>
        [+ ] <FRBRExpression>
        [+ ] <FRBRManifestation>
        </identification>
        <publication name="" date="1989-12-15" showAs=""/>
      [-] <lifecycle source="#bungeni">
        <event id="e1" date="1989-12-15" source="#ro1" type="generation"/>
        </lifecycle>
      [-] <analysis source="#bungeni">
        [-] <activeModifications>
          [+ ] <textualMod type="repeal" id="am1">
          [+ ] <textualMod type="repeal" id="am2">
          [+ ] <textualMod type="substitution" id="am3">
          [+ ] <textualMod type="insertion" id="am4">
        </activeModifications>
        </analysis>
      [-] <references source="#bungeni">
        <original id="ro1" href="/ke/act/1989-12-06/16/eng@/main" showAs="Original"/>
        <activeRef id="ra1" href="/ke/act/1980-01-01/1/main" showAs=""/>
        <activeRef id="ra2" href="/ke/act/1982-12-12/2/main" showAs=""/>
        <TLCRole id="editor" href="/ontology/roles/editor" showAs="Editor"/>
        <TLCRole id="jurist" href="/ontology/roles/jurist" showAs="Jurist"/>
        <TLCRole id="author" href="/ontology/roles/author" showAs="Author"/>
        </references>
      </meta>

```

Figura 3.4: Metadados do AKOMA NTOSO (<http://examples.akomantoso.org>)

Os pontos positivos do trabalho são inúmeros, porém o principal é a proposição de esquemas e formatos de fontes do Direito, com foco nos dispositivos legais, porém extensíveis para outras fontes. Como pontos negativos, alguns detalhes na forma de modelagem, a exemplo da falta de referências cruzadas entre dispositivos legais e jurídicos, além da centralização de todos os metadados dos elementos em uma única seção, o que pode não ser conveniente para algumas operações de navegação pelos elementos da norma, de acordo com a necessidade do operador do Direito.

EnAct

A partir da necessidade de prover informações legais atualizadas para a sociedade, o governo da Tasmânia (ilha e estado australiano) investiu no projeto *EnAct*, o qual visava ao suporte a modificações, gerenciamento e publicação das leis [7]. O projeto foi criado no ano de 2000 e sua principal motivação é a disseminação das fontes legais do Direito para todos, através da Internet.

Boas idéias foram sugeridas em Arnold-Moore & Cledes [7], relacionadas com a discussão de aspectos dos textos legais e de funcionalidades de um SGCL. Segundo os autores, as normas são elementos muito extensos, os quais contém uma grande quantidade de informações. A sugestão é que as leis sejam decompostas em unidades lógicas menores, com atributos próprios e associadas ao contexto que as cercam.

Um outro ponto abordado no trabalho é o caráter referencial das normas, envolvendo tanto as normas quanto seus elementos; foram definidas tanto as referências *internas* (entre elementos de uma norma) quanto *externas* (entre normas ou elementos de normas distintas). Para representar a estrutura lógica das leis ao armazená-las em repositórios digitais, foi sugerida a metalinguagem de marcação *Standard Generalized Markup Language* (SGML).

As modificações dos atos (normas) são apresentadas no trabalho: *revogação*, *alteração* e *inclusão*; em conjunto com essas, são discutidas as características temporais das normas, sendo defendido o armazenamento das diferentes versões das leis, posição adotada por outros autores supracitados nesta dissertação.

Na Figura 3.5 os autores ilustram ações típicas sofridas pelos atos principais (*original act*), realizadas pelos atos alteradores (*amending act*). O ato principal possui cinco seções (*s1*, *s2*, *s3*, *s4* e *s5*), adicionadas em pontos diversos no tempo. O segundo ato alterador

(*amending act 2*) modifica a seção *s4* e adiciona a seção *s6*. Já o primeiro ato alterador (*amending act 1*), altera os atos *s3* e *s5*.

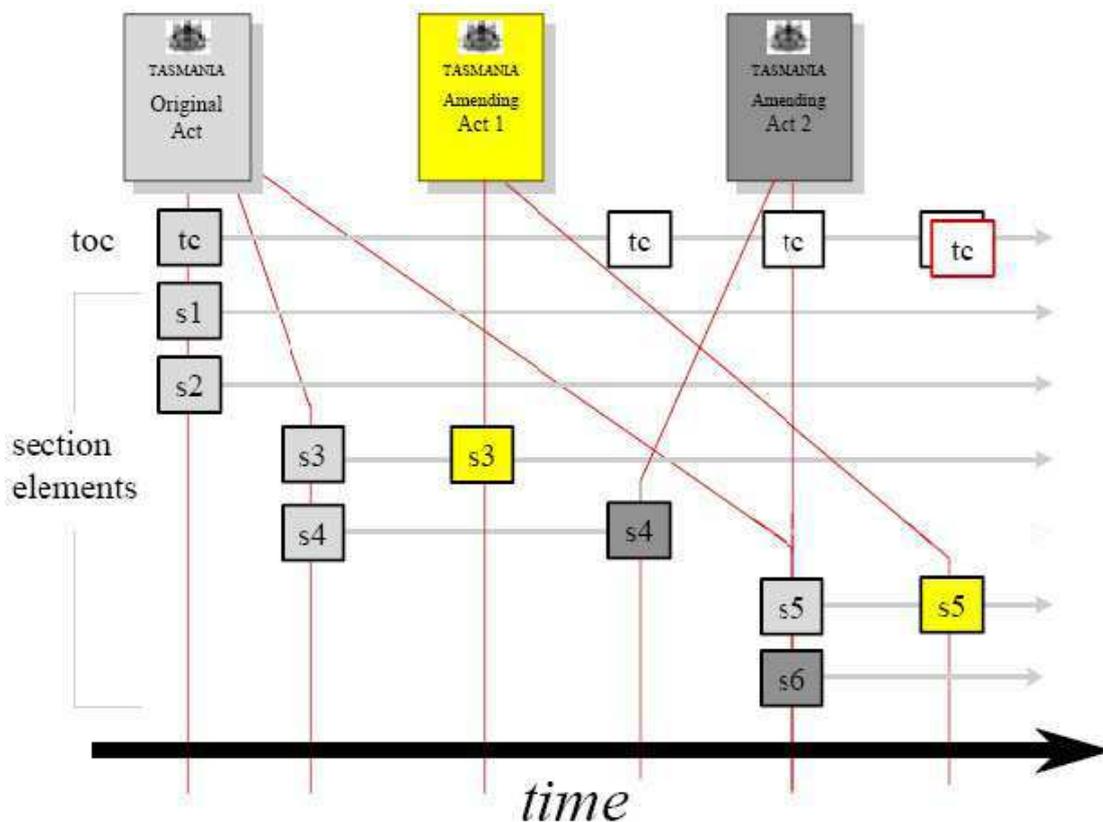


Figura 3.5: Linha de tempo de uma norma (Arnold-Moore & Cledes [7])

De acordo com Arnold-Moore & Cledes [7], o usuário pode estar interessado não apenas em recuperar a versão original do ato, mas também às suas versões antigas. A motivação para essa prática é o melhor entendimento da dinâmica das modificações legislativas por parte dos operadores do Direito, auxiliando também a compreensão das decisões advindas dos documentos jurídicos, pois esses utilizam diretamente as fontes legais.

Os autores corroboram com a idéia de Moens & Logghe [61] de que um ponto fundamental em um sistema gestor de conhecimento legal é a atualização automática das fontes do Direito. Tendo como base o esquema representado em *SGML*, foi desenvolvido um módulo para capturar as modificações em textos legais, denominado *Change Description Document* (CDD).

Através do *CDD* são gerados os fragmentos atualizados da norma, incorporados posteriormente à base de dados, operação representada pela figura 3.6. Os detalhes de como são

detectados e manipulados os textos modificadores são discutidos em [3] e [4]. Atualmente não existem propostas semelhantes na literatura para as fontes do Direito brasileiro, por isso foi projetado para o sistema *JudLaw* um módulo com essa finalidade.

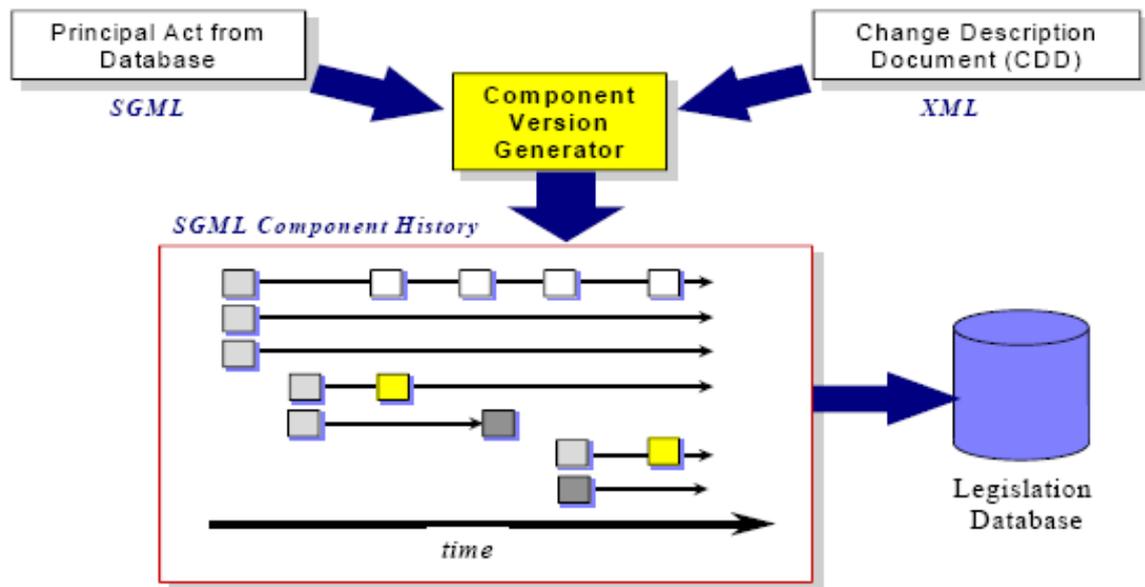


Figura 3.6: Geração de fragmentos através do CDD (Arnold-Moore & Cledes [7])

As principais contribuições do trabalho de Arnold-Moore & Cledes ao descreverem o *EnAct* [7] foram as definições de funcionalidades importantes para um SGCL (a exemplo do módulo CCD) e a exposição de características das normas que devem ser modeladas (a exemplo das alterações e temporalidade). Como deficiência na modelagem, pode ser apontada a ausência da representação de documentos jurídicos.

Estrella

O projeto *Estrella* (acrônimo de *European project for Standardized Transparent Representations in order to Extend Legal Accessibility*) foi iniciado em 2006 e tem como proposta o desenvolvimento e validação de uma plataforma aberta e baseada em padrões aceitos na comunidade, visando dar suporte ao desenvolvimento de aplicações no âmbito legislativo. Além disso, espera-se que as soluções propostas se tornem o padrão de representação de dispositivos legais em todo o continente europeu, o que está próximo de acontecer.

Existem duas grandes linhas de pesquisa no *Estrella*: a primeira, envolve o gerenciamento do conhecimento legal (GCL) e gerou como resultado o *framework CEN MetaLex*; a

segunda propôs a ontologia LKIF [17] (*Legal Knowledge Interchange Format*), na área de Inteligência Artificial e Direito, a qual foi especificada em *OWL*. Pelo fato do trabalho desta dissertação ter como foco a primeira área de pesquisa, será discutido apenas o *CEN MetaLex* (*CML*).

Segundo Boer *et al.* [16], o *CEN MetaLex* foi resultado de um estudo de outros modelos de representação de dispositivos legais, a exemplo de *AKOMA NTOSO* [1], *Norme in Rete* [56], *LexDania* [47] e *CHLexML* [63]. Originalmente, a denominação do *framework* era simplesmente *MetaLex*, porém como foi adotado como padrão pelo Comitê Europeu de Normalização⁹, recebeu o prefixo *CEN*.

O principal objetivo do *CML* é prover um arcabouço para a representação de fontes do Direito, a fim de facilitar o desenvolvimento de aplicações que lidem com esse tipo de informação. Como objetivos secundários, o *framework* foi projetado para permitir o intercâmbio e a interoperabilidade entre diferentes fontes, descrever os relacionamentos entre documentos legislativos, permitir buscas sobre as informações e dar suporte ao gerenciamento de versões [19].

O *CEN MetaLex* descreve os dispositivos legais em XML, enquanto suas propriedades são descritas em RDF. Através de modelos *XSL* (*Extensible Stylesheet Language*) também é possível representar as informações em outros formatos, a exemplo do *XHTML*. Como o *framework* foi projetado para servir às legislações de diversos países, os identificadores utilizados possuem denominações genéricas; o identificador *element*, por exemplo, pode representar capítulos, artigos, seções, etc. Para representar os aspectos temporais das normas e de seus elementos, são descritos seis atributos [20][21]:

- *date-publication*: data em que foi publicada a norma;
- *date-enacted*: data em que a norma se torna vigente, obrigatoriamente igual ou maior que a data de publicação;
- *date-repealed*: data em que a norma foi revogada;
- *date-effective*: data em que a norma - ou parte dela - se torna eficaz para determinados casos;

⁹Comité Européenne de Normalisation

- *date-version*: data que indica quando a norma foi considerada *temporalmente correta*. É verificada a corretude dos seus atributos temporais e de seus filhos, de acordo com as referências feitas por outros elementos (a exemplo de alterações e revogações);
- *ref-date*: atributo opcional, caso seja desejado acrescentar uma marca de tempo para identificar a qual versão da norma a referência foi feita.

Após uma análise crítica dos atributos temporais do *CEN MetaLex*, percebe-se que alguns são desnecessários para o âmbito desta pesquisa. Ao modelar um intervalo de vigência, a data de revogação torna-se parte dele, não sendo necessário um atributo separado. Como discutido em seções anteriores, não são feitas distinções entre o intervalo de vigência e o de eficácia de uma norma. Os próprios autores do *MetaLex* não aconselham a utilização desse atributo em Boer *et al.* [21], porém a pedidos eles o adicionaram no *framework*.

Em relação ao atributo *date-version*, a operação de atualizá-lo automaticamente tem um alto custo computacional. Além disso, não foi identificada sua utilidade, visto que a verificação da consistência temporal quase sempre está relacionada com os elementos específicos da norma, e não com ela em si. Assim como o *AKOMA NTOSO*, todas as informações temporais são armazenadas à parte nos metadados das normas, o que pode dificultar operações avançadas de navegação estrutural e temporal em uma norma.

A proposta de práticas adotadas pelo *CEN MetaLex* para o ano de 2010 foi aprovada em 2009 [18], no *CEN Workshop on an Open XML Interchange Format for Legal and Legislative Resources*. No documento estão contidas todas as especificações necessárias para a implementação do *framework*, as quais foram fortemente influenciadas pelo modelo do *AKOMA NTOSO*. Assim como no padrão africano, o *CML* descreve uma convenção de nomes relativos para representar os documentos.

Apesar de não ser um projeto totalmente inovador por se basear em outros modelos, como principal ponto positivo do *CEN MetaLex* pode-se apontar a abordagem de extrair as melhores práticas desses modelos e a partir disso definir um *framework*. Os demais pontos positivos e negativos do *AKOMA NTOSO* podem ser aplicados ao *CML*, dada a semelhança entre ambos. No entanto, em alguns pontos o padrão africano é melhor detalhado que o padrão europeu, a exemplo das informações associadas a cada referência e da clara especificação das citações passivas.

LexML Brasil

O portal LexML Brasil¹⁰ (LexMLBR), cuja primeira versão foi lançada em 2008, é uma iniciativa liderada pelo Senado Federal, em parceria com órgãos do *GT LexML*¹¹ da comunidade *TIControl*¹². Nele estão reunidas e organizadas informações legislativas e jurídicas, provenientes de órgãos dos três poderes (Executivo, Legislativo e Judiciário), nas três esferas (Federal, Estadual e Municipal). São objetivos do *LexMLBR*: (i) integrar e publicar essas fontes; (ii) prover a infraestrutura para lidar com grandes quantidades de dados, representando-os em formato *XML*.

Assim como o *AKOMA NTOSO* e o *CEN MetaLex*, o *LexMLBR* também utiliza identificadores unívocos para representar os documentos. Uma das justificativas é que mesmo que haja modificações nos endereços virtuais das normas (URLs), pode-se evitar o erro 404 (recurso não encontrado) caso sejam utilizados esses identificadores relativos [50]. A descrição detalhada de como são compostos os nomes uniformes de recursos pode ser encontrada no documento LexML URN [52]. Um exemplo simplificado retirado do portal LexMLBR:

Passos para a criação do identificador unívoco referente à LEI Nº 11.705, DE 19 DE JUNHO DE 2008.

1. *Simplificação de letras e números*: “LEI Nº 11.705, DE 19 DE JUNHO DE 2008” = “lei 11705, de 19/06/2008”;
2. *Conversão do formato da data*: “19/06/2008” = “2008-06-19”;
3. *Colocação dos elementos na ordem padrão LexML*: “lei 11705, de 2008-06-19” = “lei;2008-06-19;11705”;
4. *Inclusão do contexto (esfera Federal)*: “br:federal:lei;2008-06-19;11705”.

A linguagem utilizada pelo *LexMLBR* na modelagem das normas brasileiras seguiu a tendência dos padrões internacionais *AKOMA NTOSO* e *CEN MetaLex*, sendo escolhida a linguagem de marcação XML. As bases do modelo foram as Leis Complementares nº 95/1998 e 107/2001, vide a Figura 3.7, a qual ilustra o artigo 1º da Constituição Federal de 1988.

¹⁰<http://www.lexml.gov.br> (Acesso em Junho/2010)

¹¹<http://projeto.lexml.gov.br/gt-lexml/composicao> (Acesso em Junho/2010)

¹²<http://www.ticontrol.gov.br> (Acesso em Junho/2010)

```

<Artigo id="art1">
  <Rotulo>Art. 1º</Rotulo>
  <Caput id="art1_cpt"><p>A República Federativa do Brasil, formada pela união indissolúvel dos Estados e
Municípios e do Distrito Federal, constitui-se em Estado democrático de direito e tem como fundamentos:</p>
  <Inciso id="art1_cpt_inc1"><Rotulo>I - </Rotulo><p>a soberania;</p></Inciso>
  <Inciso id="art1_cpt_inc2"><Rotulo>II - </Rotulo><p>a cidadania;</p></Inciso>
  <Inciso id="art1_cpt_inc3"><Rotulo>III - </Rotulo><p>a dignidade da pessoa humana;</p></Inciso>
  <Inciso id="art1_cpt_inc4"><Rotulo>IV - </Rotulo><p>os valores sociais do trabalho e da livre
iniciativa;</p></Inciso>
  <Inciso id="art1_cpt_inc5"><Rotulo>V - </Rotulo><p>o pluralismo político.</p></Inciso>
</Caput>
  <Paragrafo id="art1_par1"><Rotulo>Parágrafo único.</Rotulo><p>Todo o poder emana do povo, que o exerce
por meio de representantes eleitos ou diretamente, nos termos desta Constituição.</p></Paragrafo>
</Artigo>

```

Figura 3.7: Art 1º da CF/88 em XML (adap. de Projeto LexML Brasil [50])

Outro ponto em comum entre o *LexMLBR*, *AKOMA NTOSO* e o *CEN MetaLex* é a utilização da ontologia *FRBRoo* [10]. O principal argumento defendido pelos autores, descrito no modelo de referência do *LexMLBR* [51], é que o acesso a diferentes versões da norma só é possível através de modelos que se baseiem nessa ontologia. No entanto, quando a modelagem é puramente estrutural (excetuando-se a semântica dos documentos), ao serem empregados os conceitos de um banco de dados temporal, essa operação é perfeitamente possível, mesmo sem utilizar a ontologia *FRBRoo*.

Assim como no *CEN MetaLex*, os esquemas dos projetos *AKOMA NTOSO* e *Norme in Rete* serviram de base para o desenvolvimento do esquema do *LexMLBR* [53]. Foram definidos dois esquemas para a representação das normas: *rígido* e *flexível*. O primeiro é voltado para as normas que seguem rigorosamente as regras descritas nas LCPs 95/1998 e 107/2001, enquanto o segundo é utilizado para as leis que não seguem essas regras, a exemplo das leis estaduais e municipais.

Segundo a descrição do esquema *LexMLBR* [53], os seus elementos utilizam a notação *UpperCamelCase*, enquanto os atributos utilizam *lowerCamelCase*. O *encoding* adotado para as instâncias dos documentos foi o *UTF-8*, visto que ele “oferece o armazenamento em um único *byte* para os caracteres mais comuns e permite a utilização dos caracteres definidos no padrão *Unicode*”. A parte inicial do esquema foi derivado do *AKOMA NTOSO*, o qual define tipos simples (*strings*) e *complexos* (elementos com estrutura hierárquica, de conteúdo misto ou vazio e sequências de elementos de bloco).

Os tipos definidos pelo *LexMLBR* são: *Norma* (Constituição, leis, etc.), *ProjetoNorma*

(projetos de lei), *Jurisprudencia* (súmulas e acórdãos), *DocumentoGenerico* (outros tipos) e *Anexo* (anexos aos tipos supracitados). O elemento *Norma* é composto de *ParteInicial*, *Articulacao* e *ParteFinal*, divisão baseada no artigo 3º da LCP95, apresentado na Seção 2.1.1. Além do elemento *Artigo*, seus agrupadores e desdobramentos também são descritos. Os identificadores desses elementos são criados a partir da “concatenação do identificador de cada nível hierárquico desde o nível mais alto até o nível que se está identificando” [53]. Exemplo: o identificador da Seção IV do Capítulo I é escrito como *id= “cap1_sec4”*.

A estrutura dos acórdãos é composta por seis elementos, um complexo e cinco textuais: *CabecalhoAcordao* (epígrafe, data de julgamento, órgão julgador e partes), *EmentaTexto*, *AcordaoTexto*, *RelatorioTexto*, *VotoTexto*, *DebateTexto* e *ExtratoAtaTexto*. As referências de qualquer tipo são armazenadas no elemento *<Remissao>*, o qual se refere ao identificador do dispositivo apontado, de forma relativa (a exemplo de “*art1_par1*”). Quando a referência é uma alteração, o elemento *<Alteracao>* armazena a referência absoluta da norma (a exemplo de *br;sao.paulo:lei:2004-02-11;123*), agrupando o elemento *<Remissao>*. Ainda há a possibilidade de múltiplas referências através do elemento *<RemissaoMultipla>*.

Os metadados do esquema possuem o mesmo conteúdo do esquema do *AKOMA NTOSO*, ilustrados na Figura 3.4. A única diferença é a tradução dos elementos para o português, porém o conteúdo dos elementos é o mesmo. Percebe-se que as diferenças latentes entre o *LexMLBR* e o *AKOMA NTOSO* estão relacionadas com a modelagem dos elementos da parte normativa dos dispositivos legais, com os elementos referenciais *<Alteracao>*, *<Remissao>* e *<RemissaoMultipla>* (inexistentes no *AN*) e com a integração dos documentos jurídicos e legais.

Como pontos positivos, a proposta brasileira possui uma sólida modelagem das fontes do Direito brasileiro, baseada em práticas adotadas internacionalmente. Aspectos como a temporalidade e as referências também foram contempladas. O módulo de busca a documentos é satisfatório (funciona, porém possui falhas no retorno dos resultados), inclusive pretende-se utilizá-lo em um dos módulos do *JudLaw*.

Como um dos pontos negativos, pode-se apontar a centralização de todas as informações temporais em um único lugar (limitação detectada também no *AN*). Além disso, os elementos referenciais *<Alteracao>*, *<Remissao>* e *<RemissaoMultipla>* são utilizados apenas para referências ativas. Não foi especificado como as fontes são processadas no sistema, o que

sugere que não foi utilizado nenhum modelo formal para a extração dos atributos. Sendo assim, apesar do modelo descrever os atributos e referências dos documentos, ambos teriam que ser armazenados de forma manual por um especialista, o que não é desejável.

3.2 Descrição Formal das Fontes

A digitalização de documentos legais e jurídicos é uma realidade em quase todos os países. No entanto, quando essas fontes são publicadas na Internet pelas entidades governamentais, não há a preocupação em fazê-lo de forma que os documentos estejam estruturados. Diante disso, para que um SGCL seja capaz de processar essas fontes, é necessário que haja uma forma de extrair os atributos dos documentos.

Extrair manualmente essa informação é uma tarefa custosa e tediosa, além da necessidade de um especialista para realizá-la. Para que a extração seja automática, os autores apontam a *formalização das fontes* como passo fundamental do processo. Esta seção apresentará apenas os principais trabalhos científicos internacionais que tratam dessa temática, pois não foi encontrado nenhum trabalho nacional que propusesse uma gramática formal para fontes do Direito.

A idéia de formalizar os documentos legais e jurídicos não é nova - um trabalho a respeito - Palmirani *et al.* [62] - foi publicado em 2003. No entanto, ainda é atual e muito utilizada, vide o trabalho de Wyner [75] de 2010, que trata da mineração textual em documentos jurídicos. Os campos de pesquisa relacionados com a formalização das fontes estão relacionados com a divisão da área interdisciplinar da *Ciência da Computação e Direito*. Os trabalhos se encaixam nas subáreas da *Inteligência Artificial e Direito* e do *Gerenciamento de Conhecimento Legal*, dependendo do foco da pesquisa.

No primeiro grupo, estão as pesquisas voltadas para a formalização da semântica dos documentos, relacionadas com o raciocínio baseado em casos. Podem ser citados como exemplo: Biagioli *et al.* [13] e [14], de Maat *et al.* [30], [31] e [32], Hachey & Grover [44], van Engers [73] e Wyner [75]. As técnicas mais utilizadas estão relacionadas com a Inteligência Artificial, dentre as quais está o Processamento de Linguagem Natural.

Como a temática desta dissertação está relacionada com o gerenciamento do conhecimento legal, a linha de pesquisa do segundo grupo se torna mais interessante de ser detalhada. O foco é a formalização da sintaxe dos documentos, a fim de extrair seus atributos e

armazená-los de forma estruturada na base de dados. Exemplos desse grupo são: de Maat *et al.* [33], Farah & Rousselot [35], Vanzin *et al.* [74], Bueno [24] e Schilder & McCulloh [66].

Alguns trabalhos definem apenas índices para a extração de informação, e não uma gramática formal, a exemplo dos três últimos supracitados. Em Vanzin *et al.* [74] e Bueno [24], são descritos índices dos documentos jurídicos brasileiros. Em Schilder & McCulloh [66], a ênfase é na descrição de índices para a extração de expressões temporais em textos legais. No trabalho em pauta, são discutidas as diferentes formas em que as datas são apresentadas, abreviações de atributos (dia, mês e ano) e expressões literais, por exemplo:

- O roubo aconteceu em 01/06/2010;
- O roubo aconteceu dia 1 de Junho de 2010;
- O roubo aconteceu no primeiro dia de Junho de 2010;
- O roubo aconteceu em 2010, no primeiro dia do mês de Junho.

Em de Maat *et al.* [33], são formalizados os aspectos referenciais das leis holandesas, cujo estudo de caso é a legislação de impostos e importações. Inicialmente, são descritos quatro tipos de referências simples: *por nome* (“Lei das Importações”), *por nome e número* (“artigo 1”), *variação de nome e número* (“a lei de 13 de Abril de 1995”) e *indireta* (“o próximo artigo”). Dentre as referências complexas, são detalhadas dois tipos: *multivalorada* (“artigos 12, 13 e 20-25”) e *multicamada* (“parágrafo I, artigo 15, Lei nº 3418/2010”).

Assim como nas fontes brasileiras, as referências não são totalmente uniformes, gerando variações. Há também casos especiais, por exemplo, “*caput* do artigo 20” (a referência está apontando para o enunciado do artigo 20). Para uma extração automática e completa, é preciso que o maior número dessas mutações seja formalizado.

No trabalho é apresentada uma formalização para referências, através de uma linguagem que contém elementos de *BNF* e *EBNF* (Seção 2.4) [33]. Apesar do exemplo ser simples, através dessa gramática é possível representar diversos níveis referenciais (*lower_level* e *higher_level*). Foi detectado um padrão semelhante nas fontes do Direito brasileiro, sendo assim uma formalização semelhante foi desenvolvida.

```

article → “article” <designation> [[“,”] <lower_level>]
[ “-” <designation> [[“,”] <lower_level>]
[
([“,”]<designation> [[“,”] <lower_level>])*
“and” <designation> [[“,”] <lower_level>]
]
[[“,”] [“of”] <higher_level>]

```

O símbolo *designation* define o identificador do artigo, enquanto *low_level* e *higher_level* são referentes, respectivamente, aos seus desdobramentos e agrupadores. A gramática visou à formalização tanto de referências simples, quanto complexas. O caractere * (asterisco) representa uma possível multiplicidade da expressão entre parênteses, de zero ou mais repetições.

Um dos desafios da extração de informação apontado por de Maat *et al.* [33] é o mapeamento das referências para os atributos da norma. Para a resolução de referências indiretas, a exemplo de “o artigo anteriormente mencionado”, os autores sugerem que seja armazenado - temporariamente - um histórico de todos os elementos já referenciados.

O principal ponto positivo do trabalho é a proposta de uma formalização para as referências nas leis holandesas. Os resultados foram bastante satisfatórios, chegando aos índices de 99% de detecção de referências simples e 95% de referências compostas [33]. No entanto, o trabalho possui algumas limitações.

Apenas os dispositivos legais foram contemplados, e as únicas informações formalizadas foram as referências. Além disso, a formalização sugerida foi bastante superficial. Quanto aos desafios, não foi mencionado como são diferenciadas as referências a diferentes versões das normas ou de seus elementos.

Em Farah & Rousselot [35] é apresentado o sistema *DARES (Documents annotation and recombining system)*. Através da definição de um modelo de anotação para o domínio desejado, o *DARES* é capaz de extrair os atributos de novos documentos, contanto que eles se encaixem nessa descrição formal. Assim, é possível armazenar, indexar e acessar as informações, as quais estarão estruturadas. Os estudos de caso do trabalho em pauta foram leis europeias.

Para a formalização dos documentos legais, foi proposta uma solução mista com ex-

pressões regulares e a gramática EBNF [35]. Existe um diferencial do trabalho em pauta em relação aos supracitados na seção, pois ele trata tanto de aspectos sintáticos quanto semânticos. São apresentados índices para a extração de atributos (título, preâmbulo e referências) e descrita formalmente - mesmo que superficial - a semântica das alterações, vide a Figura 3.8.

Type of update	Grammar rule
Deletion	Deletion := Part ('shall' 'be' 'is') 'deleted'
Adding	Adding := Part ('shall' 'be' 'is') 'added'
Insertion	Insertion := ('in' Part (',')? Part ('shall' 'be' 'is') 'inserted' (('before' 'after') Part)?) (Part ('shall' 'be' 'is') 'inserted' ('in' Part) (('before' 'after') Part)?)
Replacement	Replacement := Part ('shall' 'be' 'is') 'replaced' 'by' Part
Part	Corresponds to the structures identification. A regular expression is generated by the system according to the DTD given. This expression has been described in the semantic annotation section.

Figura 3.8: Gramática das expressões de alterações em EBNF (Farah & Rousselot [35])

O principal ponto positivo do trabalho é a proposta de um sistema de gerenciamento de conhecimento genérico, o qual pode ser aplicado a diferentes domínios. Foram projetados tanto os aspectos estruturais quanto semânticos dos documentos, o que também é uma contribuição. A maioria dos SGCLs discutidos neste capítulo não trataram dessa temática, porém ao analisar a literatura a respeito do assunto, percebe-se a sua importância.

Como o âmbito das leis europeias foi apenas um estudo de caso para demonstrar as ideias dos autores, as descrições foram breves e incompletas. Apenas algumas partes das normas foram formalizadas e os documentos jurídicos - fontes mais utilizadas no exterior - não foram discutidos no trabalho.

3.3 Análise Comparativa

Esta seção tratará de uma análise crítica e comparativa dos trabalhos apresentados, relacionando-os entre si e com as proposições desta dissertação: modelo *ITMJudLaw*, descrição formal das fontes e SGCL *JudLaw*. Serão discutidas as melhores práticas e limitações encontradas na literatura sobre esses pontos. O objetivo dessa análise é deter-

minar o diferencial relevante desta pesquisa em relação à literatura na área.

Apesar de não fazer parte diretamente do escopo desta dissertação, existe uma linha de pesquisa que visa à formalização semântica dos relacionamentos das normas. Nesse contexto, podem ser citados os dois trabalhos italianos Brighi & Palmirani [23] e Governatori *et al.* [41], cujas abordagens são diferentes. O primeiro emprega técnicas de *Processamento de Linguagem Natural*, enquanto o segundo usa extensão temporal da *Lógica Anulável* (Defeasible Logic).

Existem ainda na literatura outros trabalhos que estão de alguma forma relacionados com a presente pesquisa, porém devido à sua pouca contribuição em relação aos demais, esses trabalhos não foram detalhados. Um exemplo disso é o artigo de Martinek & Cybulka [57], o qual trata apenas da definição de assinaturas de operações de alteração, além de ser focado na legislação polonesa.

Após o estudo dos trabalhos na área de gerenciamento do conhecimento legal, foi possível determinar pontos importantes na modelagem das fontes do Direito. Dentre eles, destacam-se: (i) modelagem integrada de dispositivos legais e documentos jurídicos, (ii) temporalidade das fontes e (iii) operações sobre os documentos.

A importância da modelagem integrada entre fontes do Direito foi detectada a partir de entrevistas informais a operadores do Direito. Advogados, juízes e promotores foram consultados e o resultado foi unânime: ao receberem novos casos, as consultas a processos, acórdãos e leis raramente é feita de forma separada. Dos modelos apresentados, apenas o *AKOMA NTOSO*, *CEN MetaLex* e *LexMLBR* previram esse comportamento.

Essa integração é descrita através das *referências*, um quesito fundamental na modelagem. O *ITMJudLaw* descreve não apenas as referências feitas entre dispositivos legais, mas também as referências feitas por documentos jurídicos a outros documentos jurídicos e a dispositivos legais. Esta pesquisa e o *LexMLBR* foram os únicos projetos brasileiros a modelarem esse comportamento.

No entanto, é importante analisar como as referências são modeladas. Como já discutido, em Carvalho Junior [25] a semântica das referências é pobre, podendo causar até ambiguidades. As referências devem ser claras, a ponto de facilmente serem identificados o tipo (citação ou alteração), origem, destino e a *data*. Quanto a esta, novamente, apenas os três trabalhos (*AN*, *CML* e *LexMLBR*) de todos pesquisados, modelaram a temporalidade das

referências.

Saber *quando* uma referência foi feita é fundamental, para saber à qual versão da norma ou elemento ela está se referindo. Sendo assim, foi um ponto contemplado no *ITWMLaw*. Não apenas os aspectos temporais das referências, mas também das *fontes* são essenciais. Nem todos os trabalhos trataram de questões como *versionamento* e *histórico*, e menos ainda abordaram com base nos conceitos de Bancos de Dados Temporais, como é o caso desta pesquisa.

Apesar do *ITWMLaw* ter sido influenciado pelos trabalhos relacionados, definiu-se que um dos requisitos seria a simplicidade (apenas o mínimo necessário), contanto que não houvesse perda de poder operacional em relação aos demais. O objetivo de alguns modelos (*AN*, *CML* e *LexMLBR*, por exemplo) é ser o padrão de representação para fontes do Direito. Sendo assim, eles definem inúmeras convenções e regras para descreverem os documentos.

No entanto, o foco desta pesquisa é modelar de forma *enxuta* - porém eficaz - os documentos e seus comportamentos referenciais e temporais. Além disso, dar suporte a diversos tipos de operações simples e avançadas, a fim de automatizar algumas tarefas realizadas pelos especialistas. Após visitas a escritórios de advocacia e de magistrados, fóruns e tribunais, percebeu-se que há uma enorme carência por ferramentas gestoras do conhecimento legal.

Apesar de muitos trabalhos relacionados a SGCLs sugerirem formas de representar as fontes do Direito, poucos definem operações sobre esses documentos. Desses poucos, a única operação que geralmente é oferecida é a de busca às fontes. Os trabalhos que definem essas operações são isolados, a exemplo de Grandi *et al.* [42]. Uma das contribuições desta dissertação é a definição, implementação e avaliação de algumas operações estruturais e temporais avançadas, especificadas a partir da necessidade de especialistas na área do Direito.

Como já citado, não foram encontrados trabalhos brasileiros que descrevam gramáticas formais para a representação de fontes do Direito. Dos trabalhos apresentados anteriormente, alguns são incompletos, pois apenas sugerem índices para a extração de atributos. Uma das limitações da descrição proposta é o foco apenas na sintática, sendo a formalização semântica uma sugestão para trabalhos futuros.

Embora em Riveret [64] tenham sido descritos os módulos fundamentais de um SGCL, não foram encontrados na literatura sistemas que implementassem todos. A última

contribuição desta pesquisa é o projeto de um sistema gestor de conhecimento legal (*JudLaw*), no qual serão desenvolvidas todas as especificações de Riveret. Além disso, idéias de outros SGCLs da literatura também serão utilizadas. Atualmente já existe um protótipo funcional, com o modelo e operações implementadas. A partir da revisão bibliográfica, foram determinados seis critérios para avaliar a qualidade dos modelos de fontes do Direito:

1. Modelagem integrada das fontes;
2. Modelagem referencial;
3. Modelagem temporal (versionamento);
4. Definição de operações estruturais e temporais;
5. Descrição formal das fontes;
6. Acesso distribuído a diferentes repositórios.

Após uma análise dos trabalhos apresentados neste capítulo, foram geradas as Tabelas 3.9 e 3.10, as quais ilustram a classificação de como os modelos e SGCLs implementaram os critérios de avaliação supracitados. Por serem bastante específicos e não se encaixarem nos critérios adotados, os trabalhos que tratam apenas da dinâmica e descrição formal das fontes não foram representados nas tabelas.

Foram adotados três símbolos para a avaliação: *positivo* (+), *negativo* (-) e *intermediário* (+-). O primeiro indica que o trabalho contemplou o critério, enquanto o segundo indica o contrário. O último significa que o trabalho não contemplou de forma satisfatória. Todos os trabalhos que estão nas tabelas foram detalhados neste capítulo, a fim de que não haja dúvidas quanto à avaliação realizada.

Apesar de não constar nas ilustrações, o sistema *JudLaw* (o qual possui como base o modelo *ITMJudLaw*) contempla todos os critérios, de forma positiva, como será demonstrado nos capítulos seguintes. O intuito desta forma de representação é que fique - ainda mais - clara a real contribuição desta dissertação, em relação à literatura na área de modelagem e gerenciamento do conhecimento legal.

	Bueno (1999)	Vanzin <i>et al.</i> (2002)	Carvalho Jr. (2004)
Mod. integrada	-	-	-
Mod. referencial	-	+ -	+ -
Mod. temporal	-	-	-
Operações	+ -	+ -	+ -
Descrição formal	+ -	+ -	-
Acesso distribuído	-	-	-

Figura 3.9: Tabela Comparativa 1 - Trabalhos Nacionais

	Agora- Lex	EnAct	AKOMA NTOSO	CEN MetaLex	LexML Brasil
Mod. integrada	-	-	+	+	+
Mod. referencial	-	+	+	+	+
Mod. temporal	+ -	+	+	+	+
Operações	+	+	+ -	+ -	+ -
Descrição formal	-	-	-	-	-
Acesso distribuído	+	-	-	-	+

Figura 3.10: Tabela Comparativa 2 - SGCLs Pátrios

3.4 Considerações Finais

Neste capítulo foram apresentados os trabalhos relacionados a esta dissertação. Inicialmente, foram apresentadas pesquisas que tratam de aspectos da modelagem e dinâmica das fontes do Direito, além de sistemas que os implementam (SGCLs). Em seguida, foram contemplados os trabalhos sobre descrição formal de fontes do Direito. Por fim, foi feita uma análise sobre os trabalhos anteriormente descritos, comparando-os entre si e com a presente pesquisa.

O propósito de apresentar os trabalhos relacionados a esta pesquisa, destacando seus pontos positivos e negativos, é deixar evidente a relevância deste trabalho. No próximo capítulo, será detalhado o modelo *ITMJudLaw*. Além disso, serão apresentados (i) o modelo *TTMDoc*, base para o *ITMJudLaw*, (ii) as operações simples e avançadas definidas para o modelo e (iii) como foi feita a avaliação do modelo. Também será introduzido o protótipo do *JudLaw*.

Capítulo 4

Modelo ITMJudLaw

Neste capítulo será introduzido o modelo *ITMJudLaw* (*Integrated Temporal Model JudLaw*), bem como os aspectos relacionados a ele. Inicialmente, será apresentado na Seção 4.1 o *Temporal Text Model of Documents* (TTMDoc), um modelo genérico de documentos textuais-temporais, o qual foi criado como base para o *ITMJudLaw*. Em seguida, na Seção 4.2, serão introduzidas as entidades e atributos do modelo *ITMJudLaw*, como caso especial do *TTMDoc* no domínio de documentos legais e jurídicos. Na Seção 4.3, as operações previstas para o modelo serão descritas, visando a dar suporte às tarefas realizadas pelos operadores do Direito na pesquisa normativo-jurisprudencial. Por fim, a avaliação do modelo é discutida na Seção 4.4.

4.1 TTMDoc

O *Temporal Text Model of Documents* é um modelo de dados para Bancos de Dados Textuais-Temporais, nos quais os textos armazenados têm uma variabilidade temporal expressiva. Nesse cenário, um documento não é algo estático e imutável, mas sofre constantes modificações em seu conteúdo, e o histórico dessas modificações deve ser mantido no sistema. Sendo assim, o *TTMDoc* também pode ser considerado um modelo para documentos versionados.

O restante desta seção está dividida em duas partes. Na Subseção 4.1.1 serão apresentados os conceitos estruturais dos documentos. Na Seção 4.1.2 será tratada a dinâmica desses, ou seja, a manutenção/atualização de seus itens.

4.1.1 Estrutura

No modelo, um *documento* é considerado um texto contínuo, cujo início e fim estão bem definidos, além de apresentar um determinado conteúdo específico. Os exemplos são os mais variados, desde livros até tratados, normas e artigos científicos. Apesar de serem considerados não-estruturados, documentos raramente são simples textos corridos. Geralmente eles possuem metadados, como *criador*, *tamanho* e *data de criação*, além de serem compostos por itens.

Tanto documentos quanto seus itens podem ter uma validade temporal bem definida. A um parágrafo de uma norma, por exemplo, estão associadas informações sobre aspectos temporais, a exemplo de sua data de publicação e suas versões anteriores e posteriores. Ou ainda, sobre aspectos referenciais, a exemplo de quais as citações feitas e recebidas por ele. Sendo assim, um documento pode ser definido da seguinte forma:

$$(1) \textit{Documento} = \langle \textit{Metadados}_D, \textit{Corpo}_D \rangle$$

Em $\langle \textit{Metadados}_D \rangle$ estão as informações associadas ao documento, relacionadas com o *arquivo digital*: (i) *identificador*, (ii) *data de criação*, (iii) *nome do arquivo*, (iv) *autor* e eventualmente (v) *extensão do arquivo*, o qual pode ser importante para dar indícios sobre informações estruturais. Quando um documento se relaciona com outro através da relação *parte-de* (por exemplo, uma notícia é parte de uma edição de um jornal), um metadado comum associado a ele é o identificador do documento no qual ele está contido.

O $\langle \textit{Corpo}_D \rangle$ representa o conteúdo propriamente dito de um documento. Nessa estrutura, também poderão ser distinguidas informações que podem ser *metadados* ou *texto corrido*. É ainda essencial distinguir as unidades de texto que podem sofrer inclusões, remoções e modificações. No modelo *TTMDoc*, essas unidades são denominadas *itens*. Sendo assim:

$$(2) \langle \textit{Corpo}_D \rangle = \langle \textit{Metadados}_C, \textit{Itens}_C \rangle$$

$$(3) \langle \textit{Itens}_C \rangle = \{ \langle \textit{Item} \rangle \}$$

Os $\langle \textit{Metadados}_C \rangle$ representam as informações globais referentes ao *conteúdo* do documento. Podem ser citados como exemplo: *título*, *ano de publicação*, *autor*, *editor*, *idioma*, *classificação do conteúdo*, *índice* e *referências*. Uma mudança em um item de um

documento pode ter diversas consequências nos seus metadados. A diferença básica entre $\langle \text{Metadados}_C \rangle$ e $\langle \text{Metadados}_D \rangle$ é que os últimos não são visíveis no documento, e estão associados ao sistema de arquivos no qual ele está armazenado.

Na modelagem de documentos com características temporais, os metadados devem prover um meio de tornar acessível o histórico de versões. O atributo mais importante dessa categoria é o *intervalo de validade*, e opcionalmente, um número que indique a *versão* do documento. A depender do tipo de documento, sua validade pode não estar associada diretamente à de seus elementos. É o caso das *normas*, cujos artigos podem sofrer revogações, porém o intervalo de validade da norma, como um todo, permanece intacto.

O conteúdo textual do documento está contido nos seus itens, representados por $\{\langle \text{Item} \rangle\}$. Cada item, por sua vez, também pode conter metadados associados a ele. Dentre estes metadados, podem ser destacados o *identificador* do item, *posição* no documento, *intervalo de validade* e uma *referência* ao elemento - ou ao documento - no qual ele está contido. Um item ainda poder conter outros sub-itens, a exemplo dos artigos de uma norma, os quais podem ser desdobrados em parágrafos ou incisos. Sendo assim, um item pode ser descrito como:

$$(4) \langle \text{Item} \rangle = \langle \text{Metadados}_I, \text{texto}, \{\langle \text{Item} \rangle\} \rangle.$$

4.1.2 Dinâmica

Como modelo comportamental ou dinâmico dos documentos, serão consideradas as principais operações de manutenção neles e em seus itens. O objetivo é descrever como os itens são alterados quando o documento sofre operações de *modificação*, *inclusão* e *remoção*. Dado um par $\langle D, I \rangle$, em que I é um item de um documento D :

1. *Modificação*: representada por $\langle D, I, t \rangle \rightarrow \langle D', I' \rangle$, onde I' é a versão modificada do item I e t é o tempo quando ocorreu a ação. Após a execução da operação $\text{modifica}(id_D, id_I, I', t)$, é criada uma nova versão D' do documento D , tal que (i) $\text{Metadados}_{D'} = \text{Metadados}_D$ e (ii) $\text{Corpo}_{D'} = \text{Corpo}_D$, exceto pela troca de I por I' ¹. O intervalo de validade de I' passa a ser $[t, +\infty[$ e o de I é fechado em $t-I$.

¹A rigor, uma modificação em um item pode provocar uma mudança em Corpo_D , por exemplo, a atualização do índice remissivo. Por simplicidade, esses casos não serão levados em consideração em nenhuma das operações.

2. *Inclusão*: representada por $\langle D, I, E, t \rangle \rightarrow \langle D', E' \rangle$, onde E tanto pode ser o documento ($E = D$) quanto um item ($E = I_0$), ao qual será adicionado o novo item I . Caso aconteça a primeira situação, a função $incluir(id_E, id_I, I, t)$ cria um novo documento D' tal que (i) $Metadados_{D'} = Metadados_D$ e (ii) $Corpo_{D'} = Corpo_D$, exceto pelo acréscimo de I . Sendo assim, $Itens_{D'} = Itens_D \cup I$, $pai(I) = D$ e $validade(I) = [t, +\infty[$.
3. *Remoção*: representada por $\langle D, I, t \rangle \rightarrow \langle D', \emptyset \rangle$. Um item I é removido do documento D , através da função $remove(id_D, id_I, t)$. No documento modificado D' , $Metadados_{D'} = Metadados_D$ e (ii) $Corpo_{D'} = Corpo_D$, exceto pela exclusão de I . O intervalo de validade de I e de todos os elementos contidos nele passa é fechado em $t-I$.

4.2 Visão geral

Para o desenvolvimento do modelo *ITMJudlaw*, foram estudados detalhadamente os aspectos dos modelos existentes, incluindo seus pontos positivos e negativos. A partir dessa análise, foi possível criar um modelo que proovesse uma fundação sólida para representar as fontes do Direito, aplicar conceitos dos bancos de dados temporais e automatizar operações realizadas pelos especialistas.

Alguns modelos apresentados no capítulo anterior estavam associados a uma ontologia, a exemplo do AKOMA NTOSO, CEN MetaLeX e LexMLBR. No entanto, para os objetivos deste trabalho, essa abordagem se mostrou dispensável. A *semântica* dos documentos não é objeto de estudo desta pesquisa, por isso optou-se por não incluir conceitos ontológicos na modelagem. O foco do estudo foi a modelagem da estrutura, temporalidade e referencialidade das fontes do Direito.

A visão geral do modelo é apresentada na figura 4.1. Uma das características das fontes é um alto nível de relacionamento entre elas, mas apenas alguns dos modelos estudados consideram esse comportamento. O restante da seção está dividida da seguinte forma: na Subseção 4.2.1 serão detalhados as classes e os atributos referentes às fontes do Direito, enquanto na Subseção 4.2.2 serão abordados os tipos de referências e suas características.

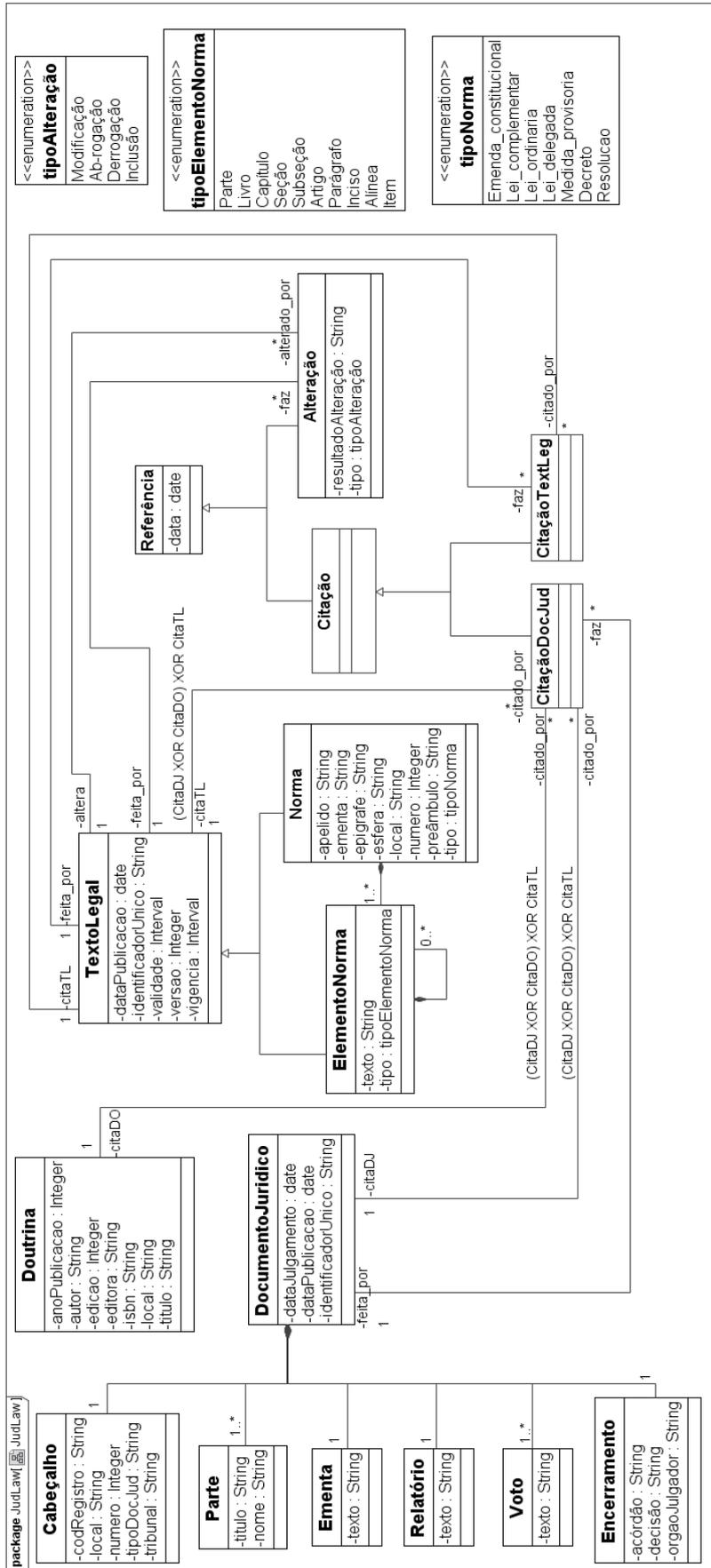


Figura 4.1: Modelo ITMJudLaw.

4.2.1 Fontes

Apesar de não ter sido considerada em *nenhum* dos modelos estudados, a *Doutrina* também é uma fonte importante. Isso se deve ao fato de que, contrariando a maioria, alguns operadores utilizam a doutrina como a principal fonte de pesquisa. Os seus atributos foram determinados a partir da observação de como são feitas as referências a esse tipo de fonte, realizadas pelos documentos jurídicos.

Os atributos da classe *Doutrina* são: *título* (título da obra), *autor* (autor da obra), *edição* (número da edição), *editora* (editora da obra), *local* (local de publicação da obra), *anoPublicação* (ano de publicação da obra) e *isbn* (numeração padrão internacional que identifica a obra). Como raramente esse tipo de fonte é publicada na Internet, geralmente as informações armazenadas no *JudLaw* acerca de uma doutrina estão relacionadas com os *metadados* da obra, caso ela tenha sido referenciada por outra fonte.

DocumentoJuridico

A entidade *DocumentoJuridico* representa os documentos jurídicos, com ênfase nos acórdãos. A análise de acórdãos emanados de diversos colegiados - STJ, STF, TJRS e TJSC - constituiu a base para a modelagem dessa classe. Além desses, os trabalhos de Vanzin *et al.* [74] e Bueno [24] também foram utilizados para o mesmo fim.

O *DocumentoJuridico* foi dividido em seis partes: *Cabeçalho*, *Ementa*, *Parte*, *Relatório*, *Voto* e *Encerramento*. Apesar de uma parte dos atributos estarem espalhados nessas partes, a entidade possui três atributos: *identificadorUnico*, *dataJulgamento* e *dataPublicação*. O primeiro está relacionado com a idéia de utilizar identificadores unívocos para representar os documentos.

Essa prática está presente em três dos mais importantes trabalhos apresentados: AKOMANTOSO, CEN MetaLeX e LexMLBR. Do próprio conteúdo do documento jurídico é possível extrair esse identificador único, pois há certa padronização nesse quesito. Geralmente é estruturado pela união de *abreviação_tipo_documento + número + local*. Exemplo: RECURSO ESPECIAL Nº 698.900 - SP = *REsp698900SP*.

Os demais atributos representam os aspectos temporais dos documentos jurídicos, os quais não são tão complexos quanto os das normas. Não existe versionamento nos documentos jurídicos, visto que uma vez que eles são publicados, não podem mais serem mo-

dificados. Por isso, os únicos metadados temporais associados a eles são a data de julgamento do processo (*dataJulgamento*) e a data de publicação da decisão do(s) magistrado(s) (*dataPublicação*).

A semântica das partes de um documento jurídico foi apresentada na Seção 2.1.2. Essas partes são agregadas ao *DocumentoJuridico* através da *composição*, pois ela só possuem sentido quando inseridas no contexto do acórdão. O *Cabeçalho* possui diversas informações sobre o documento jurídico, distribuídas nos atributos:

- *tribunal*: entidade à qual o colegiado que emanou a decisão pertence;
- *tipoDocJud*: tipo do documento jurídico;
- *numero*: identificação numérica do documento jurídico;
- *local*: local de onde partiu a ação do processo julgado;
- *codRegistro (opcional)*: código de registro do documento jurídico.

Como não existe um padrão estrutural para os documentos nacionais, alguns não possuem certos atributos (a exemplo de *codRegistro*) ou eles estão em posições diferentes no texto. A fim de descrever genericamente todos os documentos, os atributos foram incluídos na entidade *Cabeçalho* de acordo com seu *conteúdo*. Após o cabeçalho, são descritas as *partes* do processo, incluindo o relator do processo, os membros de acusação e de defesa. A classe *Parte* é composta pelos atributos *título* e *nome*, as quais representam o papel da pessoa no processo e o seu nome, respectivamente.

Em seguida, são modeladas três entidades cujo único atributo é o seu *texto*: *Ementa*, *Relatório* e *Voto*. Geralmente, o documento jurídico contém apenas um voto, referente ao do relator, enquanto os demais magistrados apenas concordam com o que foi dito. No entanto, cada um deles é livre para escrever seu próprio voto, caso sinta a necessidade de definir seu ponto de vista. Essas três entidades possuem outro ponto em comum: nelas - e apenas nelas - podem ser feitas referências às demais fontes do Direito.

Por fim, a classe *Encerramento* representa as considerações finais do documento jurídico, além de conter um resumo das informações do documento. Cada tribunal possui seus próprios atributos de encerramento, inclusive alguns já apresentados ao longo do documento

são repetidos (a exemplo de número, local e partes). Sendo assim, ao modelar essa entidade, os atributos já apresentados não foram descritos, mesmo que eles estejam no encerramento. Os atributos da classe são:

- *decisão*: referente à sentença emanada pelo colegiado;
- *acórdão (opcional)*: presente apenas em acórdãos, o seu conteúdo é semelhante ao da *decisão*;
- *orgaoJulgador*: identifica o colegiado que julgou o processo.

No apêndice B é apresentado um exemplo prático da aplicação do modelo ITMJudLaw a um documento jurídico.

Texto Legal

A entidade *TextoLegal* representa tanto as *Normas* quanto os seus elementos, denominados *ElementoNorma*. Como atributos presentes nas duas subclasses, a superclasse define: *identificadorUnico*, *dataPublicação*, *validade*, *vigência* e *versão*. Como discutido anteriormente, a utilização de um identificador unívoco foi uma proposição sugerida por alguns modelos da literatura, adotada nesta dissertação. Em relação ao *TextoLegal*, o padrão adotado para representação é o do *LexML*, detalhado em LexML URN [52].

Os aspectos temporais dos textos legais são definidos pelos outros quatro atributos, os quais são fundamentais para operações temporais, tais como *versionamento* e *verificação da consistência temporal*. A data de publicação da norma ou de seus elementos é representada pelo atributo *dataPublicacao*. Os dois intervalos temporais de *validade* e *vigência* - cuja semântica foi discutida na Seção 2.1.1 - representam o tempo em que o texto legal está válido e/ou vigente. Por fim, o atributo numérico *versão* indica qual a versão do texto legal, caso sua versão original tenha sofrido modificações.

É importante ressaltar que os atributos temporais da norma e de seus elementos são distintos e independentes. Quando uma nova norma é publicada, seus elementos passam a ter a mesma data de publicação e os mesmos intervalos temporais, a menos que haja uma disposição em contrário a isso. No entanto, ao longo do tempo, pode ser que alguns elementos da norma sofram modificações ou revogações, gerando mudanças em seus intervalos. Nessas situações, os intervalos temporais da norma permanecem inalterados.

A classe *Norma*, a qual é uma especialização do *TextoLegal*, possui atributos próprios. Estes são exclusivos das normas, não sendo encontrados nos seus itens (artigos, parágrafos, etc.). São eles: *epígrafe*, *ementa*, *preâmbulo*, *tipo*, *numero*, *esfera*, *local* e *apelido*. Os três primeiros formam a *parte preliminar* das normas, apresentados no artigo 3º da LCP95. Além desses, são modelados o *tipo* da norma, seu *número*, a *esfera* na qual ela está inserida (federal, municipal, estadual ou distrital) e o *apelido* (atributo opcional, a exemplo de Lei Maria da Penha). A norma ainda possui apontadores para todas as versões dos seus filhos, denominados *elementos da norma*.

A entidade *ElementoNorma* representa os itens da norma. Como atributos, além dos herdados via especialização, são descritos o *texto* (que pode ser denominado *caput*, no caso de um artigo) e o *tipo* do item. Como explicado na descrição do esquema XML do LexML [53], estados e municípios não utilizam as mesmas regras de estruturação das normas federais. Por isso, o esquema do ITMJudLaw foi definido como *flexível*, não sendo feitas validações estruturais em relação ao agrupamento dos elementos da norma. Os tipos da norma e de seus elementos foram descritos nas enumerações *tipoNorma* e *tipoElementoNorma*, baseadas nas normas CF/88 e LCP95.

Outro diferencial do ITMJudLaw em relação aos modelos do AKOMA NTOSO, CEN MetaLex e LexML é a *posição dos metadados*. Enquanto nos três modelos pátrios citados os metadados estão todos agrupados em um único elemento (*<meta>*), neste trabalho optou-se por atribuir aos itens os seus metadados, de forma *descentralizada*. Em cada elemento estão as suas informações estruturais, referenciais e temporais, além de apontadores para todas as versões do seu pai e dos seus filhos.

Ao analisar o método de pesquisa normativo-jurisprudencial dos operadores do Direito, percebeu-se que em certos momentos ele está interessado apenas em um artigo ou inciso de uma lei. Após acessar o item desejado, todas as informações que ele precisa sobre o elemento já estão carregadas na memória, não precisando acessar novamente a base de dados para obtê-las. Espera-se que haja um ganho no desempenho geral do sistema com essa abordagem, a ser comprovada em testes de desempenho com o *JudLaw*, em trabalhos futuros.

No apêndice B é apresentado um exemplo prático da aplicação do modelo ITMJudLaw a um texto legal, logo após o exemplo relativo ao documento jurídico.

4.2.2 Referências

O comportamento referencial das fontes do Direito foi modelado a partir da classe *Referência*. Ela possui duas especializações: *Citação* e *Alteração*. O atributos que uma referência deve conter são basicamente a *data* quando ela ocorreu, além da *origem* e *destino*. O primeiro foi descrito de forma explícita, enquanto os demais foram modelados através dos relacionamentos de suas subclasses.

Uma *Citação* ocorre quando uma fonte é mencionada por outra, sem sofrer nenhuma alteração. A origem é determinada pelo relacionamento *faz* e o destino por *citado_por*. A citação possui duas especializações: *CitaçãoDocJud* e *CitaçãoTextLeg*. A primeira representa as citações feitas por documentos jurídicos, os quais citam todos os tipos de fontes. Apesar de poder ter como destino os três tipos de fontes, cada *CitaçãoDocJud* se relaciona com um por vez, restrição representada pelo operador *XOR*. As citações feitas por textos legais - representadas por *CitaçãoTextLeg* - têm como destino apenas outros textos legais.

Os textos legais recebem tanto citações de documentos jurídicos quando de outros textos legais. No entanto, os documentos jurídicos e doutrinas apenas recebem citações de outros documentos jurídicos. Apesar de uma fonte poder ser citada diversas vezes, cada instância de uma *Citação* possui apenas uma origem e um destino. Para citações múltiplas, são criados diferentes objetos para cada uma delas.

A segunda forma de referência é a *Alteração*, cujos tipos foram descritos na enumeração *tipoAlteração*. São eles: *inclusão*, *modificação*, *ab-rogação* e *derrogação*. Os dois últimos são tipos de revogação, apresentados na Seção 2.1.1. A *inclusão* ocorre quando um novo item é adicionado à norma ou a um elemento da norma e a *modificação* quando um item tem seu texto substituído por outro.

O atributo *resultadoAlteração* armazena os resultados da inclusão e da modificação, a exemplo de um novo elemento ou um elemento modificado. As únicas fontes que fazem e sofrem o tipo *Alteração* são os textos legais. O ordenamento jurídico brasileiro define que apenas uma norma de igual ou superior hierarquia é capaz de alterar outra norma ou algum de seus elementos.

Um aspecto muito importante é o *tempo*, pois através da data em que foi feita a referência, é possível determinar a consistência temporal dos documentos. Esse atributo está associado à data de publicação do elemento origem da citação ou alteração. Apesar da importância da

temporalidade das referências, nem todos os modelos estudados contemplaram esse aspecto.

4.3 Operações

Através de conversas com operadores do Direito, foram determinadas algumas operações realizadas manualmente por eles, as quais poderiam ser automatizadas por um sistema de gerenciamento de conhecimento legal. Aliando conceitos de Bancos de Dados Temporais ao modelo *ITMJudLaw*, algumas dessas operações foram implementadas e testadas através de um *protótipo*.

As principais operações definidas neste trabalho estão relacionadas com as referências e temporalidade dos elementos. Algumas são mais simples, a fim de facilitar a navegação entre as fontes do Direito, porém outras são mais complexas, a exemplo da reconstrução temporal de uma norma e a verificação da consistência temporal das referências.

O restante desta seção está dividida da seguinte forma: inicialmente, serão apresentadas as operações de manutenção das fontes, na Subseção 4.3.1. Na Subseção 4.3.2, serão descritas as operações voltadas para o usuário final, a fim de facilitar e automatizar o trabalho dos especialistas.

4.3.1 Operações de Manutenção

Devido à estrutura hierárquica da norma e de seus elementos, quando um item sofre uma alteração, seus elementos filhos também são alterados. A operação mais simples é a de inclusão, pois apenas o elemento pai é modificado, apenas sofrendo um acréscimo de um novo item. No entanto, as operações de modificação e revogação são mais complexas, pois requerem a atualização das informações de vários elementos.

Quando um item é revogado, os intervalos de validade e vigência de todos os elementos inferiores hierarquicamente precisam ser modificados. Foi especificada uma operação recursiva para refletir essa realidade, ilustrada no código 4.1. Ambos os intervalos temporais são fechados na data em que foi publicada a revogação.

A operação de *modificação* de um item atualiza tanto seu elemento pai quanto filhos. Seguindo o que foi definido no *TTMDoc* e os conceitos de bancos de dados temporais, quando há uma modificação, é preciso que:

1. Os intervalos temporais da antiga versão do elemento sejam fechados, em uma data correspondente ao dia anterior à publicação da modificação, salvo disposição em contrário;
2. A hierarquia à qual o elemento modificado pertencia deve ser configurada para o novo elemento;
3. Os elementos pai e filhos devem ser atualizados com a nova versão do elemento.

Código Fonte 4.1: Revogação recursiva de elementos da norma

```

1  revogaElementosNormaRecursivo(elemento_norma , data_expirada) {
2      elemento_norma.vigencia = data_expirada;
3      elemento_norma.validade = data_expirada;
4      FOR each filho of elemento.filhos {
5          filho.vigencia = data_expirada;
6          filho.validade = data_expirada;
7          salva(filho);
8          revogaElementosNormaRecursivo(filho , data_expirada);
9      }
10 }
```

No código-fonte [4.2](#) é especificado o algoritmo para a atualização do sistema, ao ocorrer uma modificação a um elemento da norma, feita por uma norma.

Código Fonte 4.2: Modificação de um elemento da norma

```

1  criaAlteracaoModificacao(norma_origem , elemento_norma_destino ,
2      elemento_norma_modificado , data_modificacao) {
3      fechaIntervaloTemporal(elemento_norma_destino.vigencia , diaAnterior(
4          data_modificacao));
5      fechaIntervaloTemporal(elemento_norma_destino.validade , diaAnterior(
6          data_modificacao));
7      salva(elemento_norma_destino);
8      configuraAtributos(elemento_norma_destino , elemento_norma_modificado
9          , data_modificacao);
10     IF elemento_norma_modificado.normasPais.size > 0 {
11         FOR each norma_pai of elemento_norma_modificado.normasPais {
12             norma_pai.adicionaFilho(elemento_norma_modificado);
13             salva(norma_pai);
14         }
15     } ELSE {
16         FOR each elemento_norma_pai of elemento_norma_modificado.
17             elementosNormaPais {
18                 elemento_norma_pai.adicionaFilho(elemento_norma_modificado
19                     );
20                 salva(elemento_norma_pai);
21             }
22     }
23     salva(elemento_norma_modificado);
24 }
```

O item *elemento_norma_modificado* representa a nova versão do elemento da norma. Seus atributos e estrutura hierárquica são determinados através do método *configuraAtributos* (linha 5). Em seguida, as versões dos seus pais são atualizadas com o novo elemento.

4.3.2 Operações para o Usuário Final

Os elementos *DocumentoJuridico*, *Norma* e *ElementoNorma* possuem atributos que armazenam todas as referências feitas e sofridas por eles. Caso o operador queira acessar apenas tipos específicos de referências, também é possível, a fim de otimizar a busca por uma determinada informação. As formas de acesso às referências são:

- *getCitacoesFeitas*: retorna as citações feitas pela fonte;
- *getCitacoesRecebidasDocJud*: retorna as citações recebidas, feitas por documentos jurídicos;
- *getCitacoesRecebidasTextLeg*: retorna as citações recebidas, feitas por textos legais;
- *getAlteracoesFeitas*: exclusiva dos textos legais, retorna as alterações feitas a outros textos legais;
- *getAlteracoesRecebidas*: também exclusiva dos textos legais, retorna as alterações recebidas, feitas por textos legais.

O sistema *JudLaw* foi projetado para permitir que os usuários possam navegar entre os elementos de forma *referencial* e *temporal*. Um exemplo prático: ao analisar um acórdão, o especialista se interessou por uma referência ao parágrafo § 2º do artigo 32 da lei nº 1739. Ao ler o parágrafo, o operador tem interesse em conhecer versões antigas e atuais dos elementos que o cercam, para entender como o elemento foi modificado ao longo do tempo.

Através do modelo proposto e implementado nesta pesquisa, estão disponíveis todas as versões não apenas do parágrafo, mas também do elemento no qual ele está inserido (artigo) e dos seus filhos (incisos). A operação de navegação é livre em qualquer sentido, tanto em relação aos elementos quanto em relação às suas versões, vide a figura 4.2.

Além da implementação das operações referenciais do modelo, duas operações temporais foram desenvolvidas, a fim de automatizar tarefas realizadas pelos operadores. Uma delas,

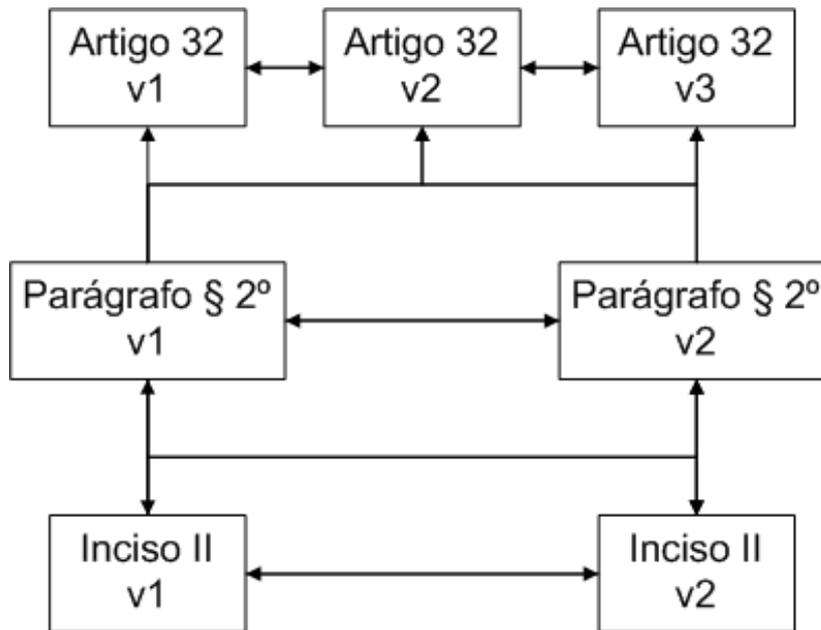


Figura 4.2: Navegação estrutural e temporal nos elementos da norma

a *reconstrução temporal* das normas, foi proposta em Grandi *et al.* [42]. A outra, que visa à *verificação da consistência temporal* dos elementos, não foi encontrada em nenhum dos trabalhos estudados.

Ao longo do tempo, é comum os elementos das normas sofrerem modificações, gerando diversas versões desses. A operação de reconstrução temporal oferece ao usuário uma determinada visão da norma, em um ponto específico do tempo que se adeque ao caso que o especialista deseja resolver.

Ilustrando esse cenário com um exemplo prático: o cliente de um advogado cometeu um crime em Junho de 2010, porém antes do julgamento, o código penal sofreu diversas modificações. Uma delas foi o aumento da punição para o crime cometido pelo réu, além da condição de agora ser inafiançável. Para o advogado, é interessante conhecer como estava descrita a lei à época do crime, para poder fundamentar a defesa de seu cliente.

Com a reconstrução temporal, uma *data* é passada como parâmetro, e tanto a norma quanto seus elementos são apresentados de acordo com a versão vigente à época desejada. Além dos aspectos estruturais, as referências (citações e alterações) também são levadas em consideração na visão final apresentada, como descreve o código-fonte 4.3. Dessa forma, o operador tem à sua disposição uma versão completa da norma em um determinado ponto no tempo.

Código Fonte 4.3: Reconstrução temporal da norma

```

1 reconstroiNormaTemporalVigencia(norma, data) {
2     configuraAtributos(norma, visao_temporal_norma);
3     validaElementosNormaVigencia( visao_temporal_norma, data );
4     validaReferencias( visao_temporal_norma, data);
5     return visao_temporal_norma;
6 }

```

As partes mais importantes da rotina são as operações *validaElementosNormaVigencia* e *validaReferencias*, as quais têm a função de excluir da visão os elementos que não são válidos em relação à data desejada. No código 4.4 é apresentado o método recursivo que valida temporalmente os filhos de um elemento raiz:

Código Fonte 4.4: Validação recursiva de elementos a partir de uma raiz

```

1 filhosValidosRecursivo(elemento_norma, data) {
2     elemento_norma.setElementosNorma( elementosNormaValidosData(
3         elemento_norma.filhos, data) );
4     FOR each filho of elemento.filhos {
5         filhosValidosRecursivo(filho, data);
6     }
7 }

```

Após a definição de um elemento raiz, seus filhos inválidos são podados pelo método *elementosNormaValidosData*. Retornada a lista de filhos válidos da raiz, cada filho também passará a ser a raiz na chamada recursiva, passando pelo mesmo processo do primeiro elemento raiz. Com isso, todos os elementos da norma são verificados.

O método que valida as referências é mais simples, pois as rotinas de verificações de validade das referências não precisam ser recursivas. No código 4.5 é ilustrada a validação das referências.

Código Fonte 4.5: Validação das referências

```

1 validaReferencias(norma, data) {
2     norma.setCitacoesFeitas( citacoesTextLegValidas(norma.citacoesFeitas
3         , data) );
4     norma.setCitacoesRecebidasTextLeg( citacoesTextLegValidas(norma.
5         citacoesRecebidasTextLeg, data) );
6     norma.setCitacoesRecebidasDocJud( citacoesDocJudValidas(norma.
7         citacoesRecebidasDocJud, data) );
8     norma.setAlteracoesFeitas( alteracoesValidas(norma.alteracoesFeitas,
9         data) );
10    norma.setAlteracoesRecebidas( alteracoesValidas(norma.
11        alteracoesRecebidas, data) );
12 }

```

As fontes do Direito são documentos que possuem um alto índice de referências entre si. Quando um operador do Direito analisa uma fonte, ele precisa se certificar de que todas

as referências utilizadas como base para a sua construção estejam de acordo com o atual ordenamento jurídico nacional.

Essa etapa demanda tempo e esforço do operador, podendo até ser dispensada caso o especialista esteja sem tempo para analisar o documento. Essa atitude pode gerar falhas grosseiras na defesa do cliente, a exemplo de se basear em uma jurisprudência ultrapassada, dado o caráter dinâmico das leis.

A operação de *verificação da inconsistência temporal* das fontes do Direito, inédita na literatura, foi desenvolvida para três situações: (i) citações feitas por textos legais, (ii) alterações feitas por textos legais e (iii) citações feitas por documentos jurídicos. Por elas serem bastantes parecidas, apenas o terceiro caso será detalhado. A operação possui duas versões: *original* e *expandida*. Na verificação temporal *original*, são verificadas apenas as inconsistências temporais das citações ou alterações, vide o código 4.6.

Código Fonte 4.6: Verificação da consistência temporal original de citações feitas por documentos jurídicos

```

1 List inconsistenciaTemporalOriginal(documento_juridico) {
2     List citacoes_inconsistentes;
3     FOR each citacao of documento_juridico.citacoes {
4         IF citacao.destino.norma != null {
5             IF citacao.data > norma_destino.vigencia.fim {
6                 citacoes_inconsistentes.add(citacao);
7             }
8         } ELSE {
9             IF citacao.data > elemento_norma_destino.vigencia.fim {
10                citacoes_inconsistentes.add(citacao);
11            }
12        }
13    }
14    return citacoes_inconsistentes;
15 }

```

A inconsistência temporal ocorre quando a *data da citação*, feita pelo documento jurídico, é *mais atual* que a *data do fim da vigência* de uma norma ou elemento. Sendo assim, apenas os textos legais são verificados, pois os documentos jurídicos não possuem vigência.

Além da verificação da norma ou elemento desejado pelo usuário, também há a opção de serem verificados os itens ao redor, através da verificação da inconsistência temporal *expandida*. Essa operação foi implementada a partir da sugestão de um promotor de justiça, cuja metodologia de pesquisa incluía não apenas os elementos citados, mas os demais que estivessem próximos.

Na operação expandida, são armazenados os itens vizinhos cuja *data de publicação é mais recente* que a do elemento citado. Caso a citação seja feita a uma norma, são verificadas todas as versões dos filhos. Caso o destino da referência seja um elemento da norma, são verificadas as versões do pai e dos filhos. Isso é interessante para o operador do Direito, pois há a possibilidade de o elemento citado estar diretamente ligado aos elementos próximos hierarquicamente.

Código Fonte 4.7: Verificação da consistência temporal expandida de citações feitas por documentos jurídicos

```

1 List inconsistenciaTemporalExpandida ( documento_juridico ) {
2     List citacoes_inconsistentes ;
3     FOR each citacao of documento_juridico.citacoes {
4         IF citacao.destino.norma != null {
5             IF citacao.data > norma_destino.vigencia.fim {
6                 citacoes.inconsistentes.add(citacao);
7             }
8             List filhos_norma_atualizados ;
9             FOR each filho of norma_destino.filhos {
10                IF filho.dataPublicacao > norma_destino.
11                    dataPublicacao {
12                    filhos_norma_atualizados.add(filho);
13                }
14            }
15        } ELSE {
16            IF citacao.data > elemento_norma_destino.vigencia.fim {
17                citacoes.inconsistentes.add(citacao);
18            }
19            List normas_pai_atualizados ;
20            FOR each norma_pai of elemento_norma_destino.normasPais {
21                IF norma_pai.dataPublicacao > elemento_norma_destino.
22                    dataPublicacao {
23                    normas_pai_atualizados.add(norma_pai);
24                }
25            }
26            List filhos_elemento_norma_atualizados ;
27            FOR each filho of elemento_norma_destino.filhos {
28                IF filho.dataPublicacao > elemento_norma_destino.
29                    dataPublicacao {
30                    filhos_elemento_norma_atualizados.add(filho);
31                }
32            }
33        }
34    }
35    return citacoes_inconsistentes ;
36 }

```

4.4 Avaliação

A avaliação do modelo ITMJudLaw e das operações foi feita por meio de um protótipo, no qual foram utilizadas como estudo de caso as fontes do Direito brasileiras descritas no apêndice A. Foram implementadas e testadas as operações descritas na seção anterior, além das operações básicas de armazenamento, acesso, atualização e remoção, referentes a todas as classes modeladas. Os requisitos para o modelo foram levantados a partir da revisão bibliográfica dos modelos e sistemas mais influentes atualmente, além de conversas com especialistas do Direito.

Através do povoamento da base de dados de forma manual, foi comprovado que o modelo é capaz de representar (i) a estrutura dos documentos jurídicos e legais brasileiros, (ii) as suas interrelações referenciais, (iii) seus aspectos temporais, a exemplo do versionamento das normas, (iv) realizar as operações de atualização da base de fontes do Direito, garantindo a manutenção do histórico de textos alterados e (v) realizar as diversas formas de consultas de interesse dos usuários do sistema.

As melhores práticas e sugestões dos modelos estudados foram contempladas neste trabalho. O próximo passo da avaliação é criar uma interface gráfica para que operadores do Direito possam interagir com o sistema, descrever suas impressões e sugerir modificações.

4.5 Considerações Finais

Neste capítulo, foram apresentados os principais aspectos relacionados ao modelo proposto nesta dissertação. Inicialmente, foi introduzido o modelo genérico *TTMDoc* para documentos textuais-temporais, o qual serviu de base para o ITMJudLaw. Em seguida, foi contemplada a descrição da visão geral do ITMJudLaw, incluindo aspectos estruturais, referenciais e temporais da modelagem.

Posteriormente, foram descritas as principais operações do sistema, tanto para manter a base atualizada, sem desprezar as características históricas dos textos, como para automatizar verificações realizadas manualmente pelos especialistas. Por fim, foi apresentada a forma de avaliação do modelo e de suas operações. No próximo capítulo, será detalhada a gramática que descreve formalmente as fontes do Direito brasileiras.

Capítulo 5

Descrição Formal das Fontes do Direito

Neste capítulo será apresentada a descrição formal das normas brasileiras e acórdãos do Superior Tribunal Federal (STF) e Superior Tribunal de Justiça (STJ). A linguagem escolhida para a definição da gramática foi a *Forma Estendida de Backus-Naur*, devido à sua simplicidade, aliada a um grande poder expressivo. Esta é uma proposta inédita na literatura nacional, pois não foram encontrados trabalhos que propusessem formalizações para as fontes brasileiras. Serão descritas formalmente as principais partes das fontes do Direito, e para maiores detalhes, a gramática completa pode ser encontrada no apêndice C.

A partir da descrição formal dos dispositivos legais e jurídicos, é possível fazer a extração automática de informações importantes desses, como apontam diversos autores citados anteriormente. Atualmente, essas fontes estão publicadas na Internet como textos semi-estruturados, ou seja, apesar da divisão de seus elementos ser compreendida pelos seres humanos, isso não acontece com as máquinas. Além disso, o armazenamento estruturado feito manualmente é uma tarefa que requer tempo e esforço de um especialista.

O restante do capítulo está dividido da seguinte forma: na Seção 5.1 será apresentada a gramática formal referente à estrutura dos dispositivos legais. Na Seção 5.2 serão descritos formalmente os documentos jurídicos, dentre eles, acórdãos do STF (5.2.1) e do STJ (5.2.2), além de como são feitas as referências nesses documentos (5.2.3). Por fim, na Seção 5.3 será mostrado como a gramática está sendo validada. Os documentos que serviram de base para o desenvolvimento da gramática estão descritos no apêndice A.

5.1 Dispositivos Legais

Após o estudo das fontes do estudo de caso, foi determinada uma gramática formal genérica, aplicável a todos os tipos de normas descritos no artigo 59 da Constituição Federal de 1998. Inicialmente, a norma está dividida em *cabeçalho*, *epígrafe*, *ementa*, *preâmbulo*, *parte normativa* (elementos da norma), *local* e *data de publicação*:

```
1 norma = cabeçalho , epígrafe , ementa , preâmbulo , {elemento_norma}- ,
    local_data ;
```

O cabeçalho é sempre o mesmo em todas as normas, descrito em três sentenças:

```
1 cabeçalho = ‘‘Presidência da República’’ , ‘‘Casa Civil’’ , ‘‘Subchefia
    para Assuntos Jurídicos’’ ;
```

Em seguida, é descrita a epígrafe da norma, a qual contém o seu tipo, número e data de publicação. Os tipos podem ser: lei, lei complementar, lei delegada, decreto, emenda constitucional ou medida provisória, escritos com letras maiúsculas. O ponto final na epígrafe é opcional, presente em apenas algumas normas:

```
1 epígrafe = tipo_dispositivo_legal , ‘‘Nº’’ , número_norma , ‘‘,’’ ,
    data_norma_epígrafe , [‘‘.’’] ;
2
3 tipo_dispositivo_legal = tipo_lei | ‘‘DECRETO’’ | ‘‘EMENDA CONSTITUCIONAL
    ’’ | ‘‘MEDIDA PROVISÓRIA’’ ;
4
5 tipo_lei = ‘‘LEI’’ | ‘‘LEI’’ , tipo_lei_específico ;
6
7 tipo_lei_específico = ‘‘COMPLEMENTAR’’ | ‘‘DELEGADA’’ ;
8
9 data_norma_epígrafe = ‘‘DE’’ , dia_data , ‘‘DE’’ ,
    mês_data_extenso_maiúsculo , ‘‘DE’’ , ano_data ;
```

Da epígrafe é possível retirar os atributos *tipo*, *número* e *dataPublicação* da norma. Em seguida, aparecem na estrutura da norma dois atributos que podem conter referências a outros textos legais, contidas em seu texto:

```
1 ementa = texto_ref ;
2
3 preâmbulo = autoridade , texto_ref ;
```

Alguns verbos recorrentes da ementa que caracterizam o teor da norma: *altera*, *acrescenta*, *dá nova redação*, *dispõe*, *institui*, *cria*, *autoriza*, *abre* (créditos), etc. A autoria de uma norma define quem é a autoridade responsável por sua edição, e cada norma tem seu jeito peculiar de expressar sua autoria. O símbolo *texto_ref* (o qual será apresentado a seguir)

significa um texto que contém referências a outros elementos, enquanto *autoridade* indica qual autoridade foi responsável pela edição da norma. Posteriormente, é definida a parte normativa da lei, composta por seus elementos:

```

1 elemento_norma = id_elemento_norma , texto_ref , {referência_inversa_TL}
  , {elemento_norma} ;
2
3 id_elemento_norma = id_parte | id_livro | id_título | id_capítulo |
  id_seção | id_subseção | id_artigo | id_parágrafo | id_inciso |
  id_alínea | id_item ;
4
5 id_parágrafo = (id_parágrafo_único | id_parágrafo_comum) , ‘-’ ;
6
7 id_parágrafo_único = ‘Parágrafo único’ ;
8
9 id_parágrafo_comum = ‘§’ , numeração_art_par ;

```

Cada *elementoNorma* possui um identificador, dependendo do seu tipo. A forma de identificação de cada *elementoNorma* foi descrita na Seção 2.1.1. Como exemplo, foi demonstrado acima o identificador de um parágrafo. O símbolo *numeração_art_par* sugere a numeração de artigos e parágrafos, cujos 10 primeiros números são escritos na forma ordinal, enquanto os demais na forma cardinal.

Após o identificador, o elemento é composto por (i) seu texto (o qual pode conter referências), (ii) uma referência inversa que indica que o elemento foi alterado por outro dispositivo (opcional) e (iii) um conjunto de elementos filhos (opcional). Exemplo de uma referência inversa feita a um elemento, retirada do Código Penal: *Art. 7º - Ficam sujeitos à lei brasileira, embora cometidos no estrangeiro: (Redação dada pela Lei nº 7.209, de 1984)*

A *referência inversa* foi um dos tipos formalizados pela gramática proposta, presente apenas em dispositivos legais. A denominação *TL* significa *Texto Legal*. Operações de inclusão, modificação e revogação podem ser representadas por ela:

```

1 referência_inversa_TL = ‘(’ , termo_ref_inversa , (‘pelo’ | ‘pela’)
  , tipo_dispositivo_legal , ‘no’ , número_norma , ‘,’ , [‘de’] ,
  data_referência_numeral , ‘)’’ ;
2
3 termo_ref_inversa = termo_inclusão | termo_modificação | termo_revogação
  ;
4
5 termo_inclusão = ‘Incluído’ | ‘Incluídos’ ;
6
7 termo_modificação = ‘Vide’ | ‘Redação dada’ ;
8
9 termo_revogação = ‘Revogado’ | ‘Revogados’ ;

```

O símbolo *texto_ref* representa um texto que pode conter *referências diretas* a um ou mais *dispositivos legais*. A *referência direta* é um dos tipos de referências descritas na gramática, junto com a referência inversa:

```

1 texto_ref = texto , {referência_direta_TL} ;
2
3 referência_TL = referência_direta_TL | referência_inversa_TL ;
4
5 referência_direta_TL = referência_direta_TL_interna |
   referência_direta_TL_externa ;

```

A referência direta se divide em dois tipos: *interna* e *externa*. A primeira acontece quando um elemento da norma se refere a outro(s) elemento(s) contido(s) na mesma norma. A segunda acontece quando o destino da referência não faz parte da norma na qual ele está contido. A referência direta interna pode ser *simples* ou *composta*. Segue a descrição da primeira:

```

1 referência_direta_TL_interna = referência_direta_TL_interna_simples |
   referência_direta_TL_interna_composta ;
2
3 referência_direta_TL_interna_simples = { id_elemento_norma_ref_simples ,
   {separador_referência_simples} }- ;
4
5 id_elemento_norma_ref_simples = id_parte | id_livro | id_título |
   id_capítulo | id_seção | id_subseção | id_artigo_ref_simples |
   id_parágrafo_ref_simples | id_inciso_ref_simples |
   id_alínea_ref_simples | id_item_ref_simples ;
6
7 id_artigo_ref_simples = (‘art.’ | ‘artigo’) , numeração_art_par ,
   [‘,’ , ‘caput’] ;
8
9 separador_referência_simples = ‘de’ | ‘da’ | ‘do’ | ‘,’ ;

```

A descrição supracitada ilustra o poder da gramática proposta, a qual permite representar referências a todos os tipos de elementos previstos na CF/88. Apenas a título de exemplo, foram demonstrados os símbolos que identificam um artigo referenciado. Também são previstas referências em diversos níveis, a exemplo de: “segundo o inciso III do parágrafo § 2º do artigo 32”.

As estruturas das referências internas simples e compostas são bastante parecidas, variando apenas os identificadores dos elementos (que passam a serem descritos no plural) e a presença de elementos de conjunção. Há sutis diferenças entre os identificadores dos elementos quando estes fazem parte da norma e quando são referenciados no texto. Por exemplo, um artigo possui três identificadores associados a ele: *id_artigo*, *id_artigo_ref_simples* e

id_artigo_ref_composta. Segue a descrição da referência direta interna composta:

```

1 referência_direta_TL_interna_composta = {id_elemento_norma_ref_composta
    , {separador_referência_simples , referência_direta_TL_interna_simples
    } }- ;
2
3 id_elemento_norma_ref_composta = id_parte_ref_composta |
    id_livro_ref_composta | id_título_ref_composta |
    id_capítulo_ref_composta | id_seção_ref_composta |
    id_subseção_ref_composta | id_artigo_ref_composta |
    id_parágrafo_ref_composta | id_inciso_ref_composta |
    id_alínea_ref_composta | id_item_ref_composta ;
4
5 id_artigo_ref_composta = (‘arts.’ | ‘artigos’) , numeração_art_par ,
    {separador_referência_composta , numeração_art_par}- ;
6
7 separador_referência_composta = ‘e’ , ‘,’ ;

```

Através da especificação da referência direta interna simples, é possível descrever referências em diversos níveis, a exemplo de “incisos III e IV do artigo 32”. A referência direta *externa* também utiliza a estrutura da referência direta *interna*, com a adição da norma que contém o(s) elemento(s) referenciado(s) ao final da expressão:

```

1 referência_direta_TL_externa = referência_direta_TL_interna , {
    separador_referência}- , tipo_dispositivo_legal_ref , [‘nº’] ,
    número_norma , separador_referência_simples , data_referência ;
2
3 tipo_dispositivo_legal_ref = tipo_lei_ref | ‘Decreto’ | ‘Decreto-lei’
    | ‘Emenda Constitucional’ | (‘Medida Provisória’ | ‘MP’) |
    códigos ;
4
5 data_referência = data_referência_completa | ano_data ;
6
7 data_referência_completa = ‘de’ , dia_data , ‘de’ , mês_data_extenso
    , ‘de’ , ano_data ;

```

Tanto as versões *simples* quanto *composta* da referência direta interna podem compor a referência direta externa. Exemplos: *incisos I e II do artigo 35 do Decreto nº 58321 de 12 de Junho de 2010*. Caso a referência seja feita a um código, não são especificados o número da norma nem a data, por isso essa parte da expressão está como opcional na regra.

Assim como a identificação dos elementos da norma depende do contexto no qual eles estão inseridos (se fazem parte da norma ou se foram referenciados), a mesma lógica serve para os identificadores dos dispositivos legais referenciados. Apesar de alguns estarem explícitos no símbolo *tipo_dispositivo_legal_ref*, outros são descritos em outras regras, a exemplo de *tipo_lei_ref* e *códigos*. Por fim, é descrito o local e data em que a norma foi publicada:

```

1 local_data = “Brasília” , “,” , [“em”] , dia_data , “de” ,
  mês_data_extenso , “de” , ano_data , “;” ;
2
3 mês_data_extenso = “Janeiro” | “Fevereiro” | “Março” | “Abril” |
  “Maio” | “Junho” | “Julho” | “Agosto” | “Setembro” | “
  Outubro” | “Novembro” | “Dezembro” ;

```

5.2 Documentos Jurídicos

Os documentos jurídicos são bastante parecidos estruturalmente. Nesta seção será descrita a parte da gramática referente à estrutura dos acórdãos do Superior Tribunal Federal (5.2.1) e Superior Tribunal de Justiça (5.2.2). Um fato interessante é que eles referenciam outras fontes da mesma forma, por isso foi separada em outra seção (5.2.3) a parte da gramática que descreve essa característica dos documentos.

5.2.1 STF

Os documentos jurídicos provenientes do STF podem ser descritos formalmente da seguinte forma:

```

1
2 documento_jurídico_STF = cabeçalho1 , cabeçalho2 , {parte}- , ementa ,
  acórdão , relatório , {voto}- , extrato_da_ata ;

```

O primeiro cabeçalho dispõe sobre informações gerais, como por exemplo, o *número do diário de justiça eletrônico* no qual foi publicado o acórdão, a *data de divulgação*, *data de publicação* e o *número do ementário*. No segundo cabeçalho, alguns metadados do documento são informados, tais como *data do julgamento*, *órgão julgador*, *tipo*, *número* e *local* de onde partiu a ação do processo:

```

1
2 cabeçalho1 = “Superior Tribunal Federal” , “Coordenadoria de Análise
  de Jurisprudência” , “DJe nº” , número_dje , “Divulgação” ,
  data_divulgação , “Publicação” , data_publicação , “Ementário nº”
  , número_ementário ;
3
4 cabeçalho2 = data_julgamento , órgão_julgador , cabeçalho2_docjudinfo ;
5
6 cabeçalho2_docjudinfo = tipo_dj , número_dj , local ;

```

Em seguida, são apresentadas as *partes* do processo:

```

1
2 parte = título , “:” , nome ;

```

A parte do conteúdo propriamente dito do acórdão é iniciado com a *ementa*. São descritos em sequência o *acórdão*, *relatório* e *voto(s)*. Cada uma dessas seções é iniciada por um termo específico, que também serve para indicar o fim da seção anterior. No entanto, algumas possuem termos que indicam explicitamente o fim da seção, como é o caso de *relatório* e *voto(s)*. Apenas *ementa*, *relatório* e *voto(s)* podem conter referências a outras fontes, por isso, o tipo dos seus textos é diferenciado (*texto_ref_dj*):

```

1
2 ementa = ‘EMENTA:’ , texto_ref_dj ;
3
4 acórdão = ‘A C Ó R D Ã O’ , texto ;
5
6 relatório = ‘R E L A T Ó R I O’ , texto_ref_dj , [‘É o relatório.’] ;
7
8 voto = ‘V O T O’ , texto_ref_dj , [‘É como voto.’] ;
9
10 texto_ref_dj = texto , {referência_DJ} ;

```

A regra *referência_DJ* representa as referências feitas pelos documentos jurídicos. Ela será explicada em detalhes na Seção 5.2.3. O final de um acórdão do STF é marcado pelo *Extrato da Ata*, no qual é informada a decisão do colegiado acerca do caso, além de algumas informações sobre o documento jurídico:

```

1
2 extrato_da_ata = ‘EXTRATO DA ATA’ , tipo_dj , número_dj , ‘PROCED.:’ ,
   local , {parte}- , ‘Decisão:’ , decisão ;

```

5.2.2 STJ

Os documentos jurídicos provenientes do STJ possuem estrutura semelhante aos do STF, e podem ser descritos formalmente da seguinte forma:

```

1
2 documento_jurídico_STJ = ‘Superior Tribunal de Justiça’ , cabeçalho , {
   parte}- , ementa , acórdão , relatório , {voto}- ,
   certidão_de_julgamento ;

```

No cabeçalho dos documentos, são informados o *tipo* do documento jurídico, seu *número*, a *sigla do local* de onde partiu o processo e o *código de registro* do acórdão no tribunal. Este atributo não está presente em documentos do STF.

```

1
2 cabeçalho = tipo_dj , ‘Nº’ , número_dj , ‘-’ , sigla_local , ‘(’ ,
   cod_registro_dj , ‘)’ ;
3
4 cod_registro_dj = ano_data , ‘/’ , código ;

```

Em seguida, são apresentadas as *partes* do processo, cuja estrutura é idêntica a dos documentos do STF:

```
1
2 parte = título , ‘:’ , nome ;
```

As seções referentes ao conteúdo dos documentos do STJ são as mesmas dos documentos do STF, mudando apenas alguns detalhes estruturais, a exemplo dos elementos do cabeçalho. Além disso, a data do julgamento é apresentada no final da Seção *acórdão*.

```
1
2 ementa = ‘EMENTA’ , texto_ref_dj ;
3
4 acórdão = ‘ACÓRDÃO’ , texto , local_acórdão_stj ,
   data_julgamento_extenso ;
5
6 local_acórdão_stj = local | local , ‘(’ , sigla_local , ‘)’ | local ,
   ‘/’ , sigla_local ;
7
8 data_julgamento_extenso = dia_data , ‘,’ , ‘de’ , mês_data_extenso ,
   ‘de’ , ano_data ;
9
10 relatório = ‘RELATÓRIO’ , texto_ref_dj , [‘É o relatório.’] ;
11
12 voto = ‘VOTO’ , texto_ref_dj , [‘É como voto.’] ;
13
14 texto_ref_dj = texto , {referência_DJ} ;
```

A última parte de um acórdão do STJ é a *certidão de julgamento*. Nela estão algumas informações sobre o documento, já citadas anteriormente, além da decisão à qual chegou o colegiado sobre o processo:

```
1
2 certidão_de_julgamento = ‘CERTIDÃO DE JULGAMENTO’ , órgão_julgador , ‘
   Número Registro:’ , cod_registro_dj , sigla_tipo_dj , número_dj ,
   ‘/’ , sigla_local , ‘JULGADO:’ , data_julgamento_numeral , ‘
   CERTIDÃO’ , decisão ;
3
4 data_julgamento_numeral = dia_data , ‘/’ , mês_data_numeral , ‘/’ ,
   ano_data ;
```

5.2.3 Referências

Dada a natureza dos documentos jurídicos, as referências não têm um formato padrão, ficando ao encargo do operador do Direito escolher a melhor forma de fazê-lo. No entanto, ao analisar os 60 documentos jurídicos publicados pelo STF e STJ (30 de cada) mais recentes à época da pesquisa (Janeiro de 2010), constatou-se diversas semelhanças nas citações.

As referências feitas por documentos do STF e STJ são estruturalmente parecidas, por isso foi descrita uma gramática genérica que pode ser utilizada para documentos de ambos tribunais. Existem três tipos de referências, relativos aos tipos de fontes aos quais os documentos jurídicos se referem: documentos jurídicos (*DJ_DJ*), doutrinas (*DJ_DO*) e textos legais (*DJ_TL*). Os dois primeiros tipos são os mais simples:

```

1
2 referência_DJ = referência_DJ_DJ | referência_DJ_TL | referência_DJ_DO ;
3
4 referência_DJ_DJ = [tribunal_abrev] , tipo_DJ_abrev , [exp_num] ,
   número_dj , [-AgR] , '/' , sigla_estado ;
5
6 tribunal_abrev = 'STF' | 'STJ' ;
7
8 sigla_estado = 'AC' | 'AL' | 'AP' | 'AM' | 'BA' | 'CE' | 'DF'
   'ES' | 'GO' | 'MA' | 'MT' | 'MS' | 'MG' | 'PA' | 'PB' |
   'PR' | 'PE' | 'PI' | 'RJ' | 'RN' | 'RS' | 'RO' |
   'RR' | 'SP' | 'SC' | 'SE' | 'TO' | sigla_país ;
9
10 exp_num = 'no' | 'n.' | 'n.º' ;
11
12 referência_DJ_DO = '(' , nome_obra , ',' , edição , editora , local ,
   ano , página , ')' ;

```

A *parte final* de uma referência feita por um documento jurídico pode variar de duas formas: quando a referência é feita a elementos de (i) uma lei ou de (ii) um código. Exemplos:

- Artigo 32 da Lei n.º 64234 de 12 de Junho de 2010;
- Artigo 32 do Código Penal;
- Artigos 32 e 33, todos do Código Penal.

```

1
2 referência_DJ_TL_final_norma_data = {separador_referência_simples}- ,
   tipo_dispositivo_legal_ref , exp_num , número_norma , {
   separador_referência_simples_DJ}- , data_referência_DJ ;
3
4 referência_DJ_TL_final_códigos = ',' , ['todos'] , 'do' , códigos ;
5
6 separador_referência_simples_DJ = separador_referência_simples | '/' ;
7
8 data_referência_DJ = data_referência_completa_DJ | ano_data ;
9
10 data_referência_completa_DJ = 'de' , dia_data , separador_data ,
   mês_data_numeral , separador_data , ano_data ;
11
12 códigos = ('Código Penal' | 'CP') | ('Código de Processo Penal' |
   'CPP') | ('Constituição' | 'Constituição Federal' | 'CF') ;

```

Aliadas a uma das partes finais supracitadas, as referências feitas em um documentos jurídicos são semelhantes às referências feitas em dispositivos legais. Caso seja uma referência simples (a um elemento), é idêntica à *referência direta interna simples* dos textos legais. Caso seja composta, tanto pode ser na forma de uma *referência direta interna composta* (também proveniente dos textos legais) quanto em uma forma exclusiva dos documentos jurídicos, denominada *referência_DJ_TL_composta_especializada*:

```

1
2 referência_DJ_TL = ( referência_direta_TL_interna_simples |
   referência_DJ_TL_composta ) , referência_DJ_TL_final_norma_data |
   referência_DJ_TL_final_códigos ;
3
4 referência_DJ_TL_composta = referência_direta_TL_interna_composta |
   referência_DJ_TL_composta_especializada ;
5
6 referência_DJ_TL_composta_especializada = { id_elemento_norma_ref_simples
   | id_elemento_norma_ref_composta , [ referência_DJ_TL_final_norma_data
   ] , { separador_referência_composta_DJ }- }- ,
   id_elemento_norma_ref_simples | id_elemento_norma_ref_composta ,
   referência_DJ_TL_final_norma_data | referência_DJ_TL_final_códigos ;
7
8 separador_referência_composta_DJ = separador_referência_simples | ‘e’ |
   ‘c/c’ | ‘no’ | ‘na’ ;

```

Um exemplo de referência composta especializada: artigo 32 do Código Penal *c/c* inciso IV da Lei n.º 45312. Esse tipo de referência apenas é encontrada em documentos jurídicos.

5.3 Avaliação

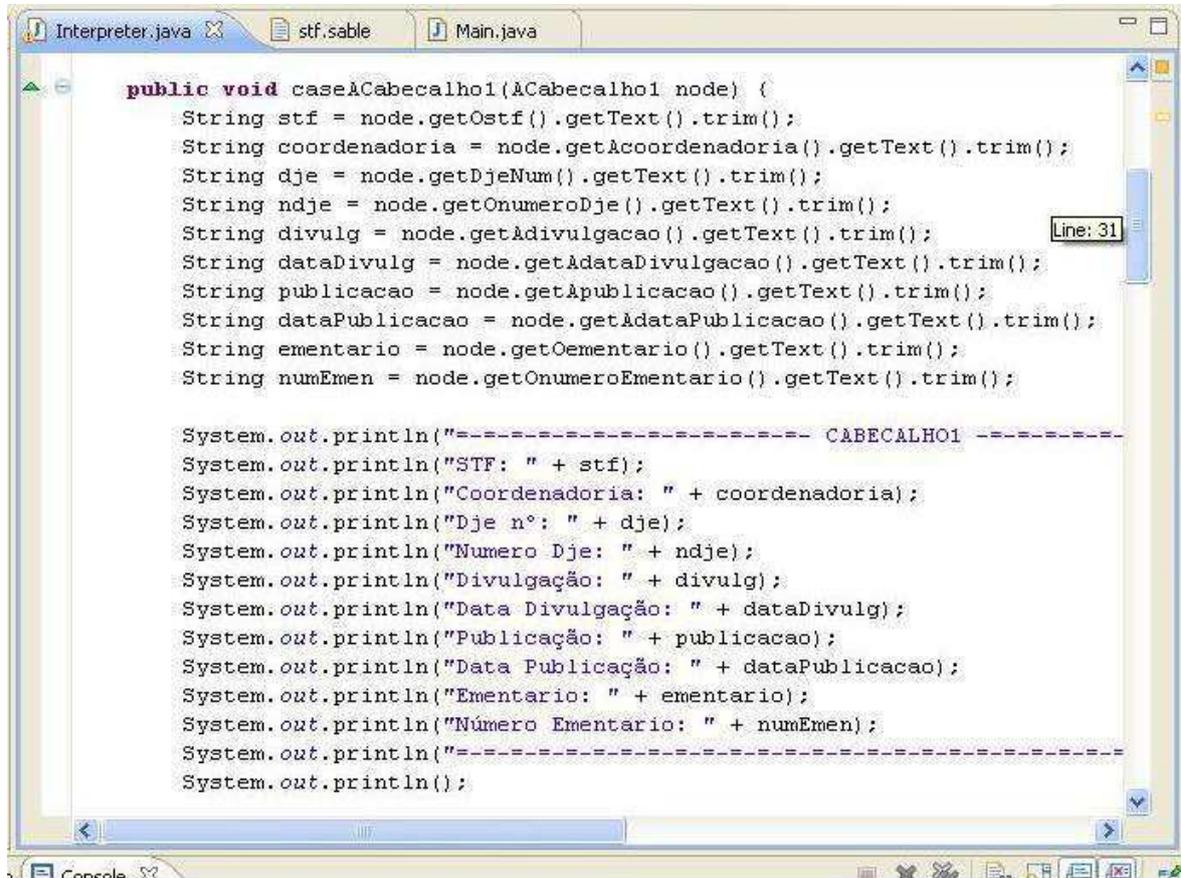
A avaliação da gramática está sendo realizada com o auxílio do *framework SableCC*¹, o qual gera compiladores a partir de gramáticas formais. Das ferramentas pesquisadas, ela foi a mais adequada para a validação da descrição formal em EBNF. Para adequar as gramáticas ao *framework*, foram geradas novas versões da gramática, aplicáveis ao *SableCC*.

Após as diversas mudanças, as três gramáticas foram compiladas sem erros pela ferramenta. Em seguida, foi gerado um *parser* (analisador) para cada uma, possibilitando a criação dos *interpretadores* que irão extrair as informações dos documentos. O objetivo é realizar a extração automática de atributos, a partir da descrição formal proposta.

Na Figura 5.1 é ilustrada uma parte do interpretador para documentos jurídicos do STF, mais especificamente relativo ao cabeçalho. Ao receber como parâmetro um elemento com-

¹<http://sablecc.org>. Acesso em Junho/2010

plexo (com diversos elementos internos), cada uma de suas partes é extraída e apresentada ao operador. Assim como nos demais interpretadores, a base é a descrição formal desenvolvida nesta dissertação.



```
public void caseACabecalho1(ACabecalho1 node) {
    String stf = node.getOstf().getText().trim();
    String coordenadoria = node.getAcoordenadoria().getText().trim();
    String dje = node.getDjeNum().getText().trim();
    String ndje = node.getOnumeroDje().getText().trim();
    String divulg = node.getAdivulgacao().getText().trim();
    String dataDivulg = node.getAdataDivulgacao().getText().trim();
    String publicacao = node.getApublicacao().getText().trim();
    String dataPublicacao = node.getAdataPublicacao().getText().trim();
    String ementario = node.getOementario().getText().trim();
    String numEmen = node.getOnumeroEmentario().getText().trim();

    System.out.println("----- CABECALHO1 -----");
    System.out.println("STF: " + stf);
    System.out.println("Coordenadoria: " + coordenadoria);
    System.out.println("Dje nº: " + dje);
    System.out.println("Numero Dje: " + ndje);
    System.out.println("Divulgação: " + divulg);
    System.out.println("Data Divulgação: " + dataDivulg);
    System.out.println("Publicação: " + publicacao);
    System.out.println("Data Publicação: " + dataPublicacao);
    System.out.println("Ementario: " + ementario);
    System.out.println("Número Ementario: " + numEmen);
    System.out.println("-----");
    System.out.println();
}
```

Figura 5.1: Interpretador para documentos do STF

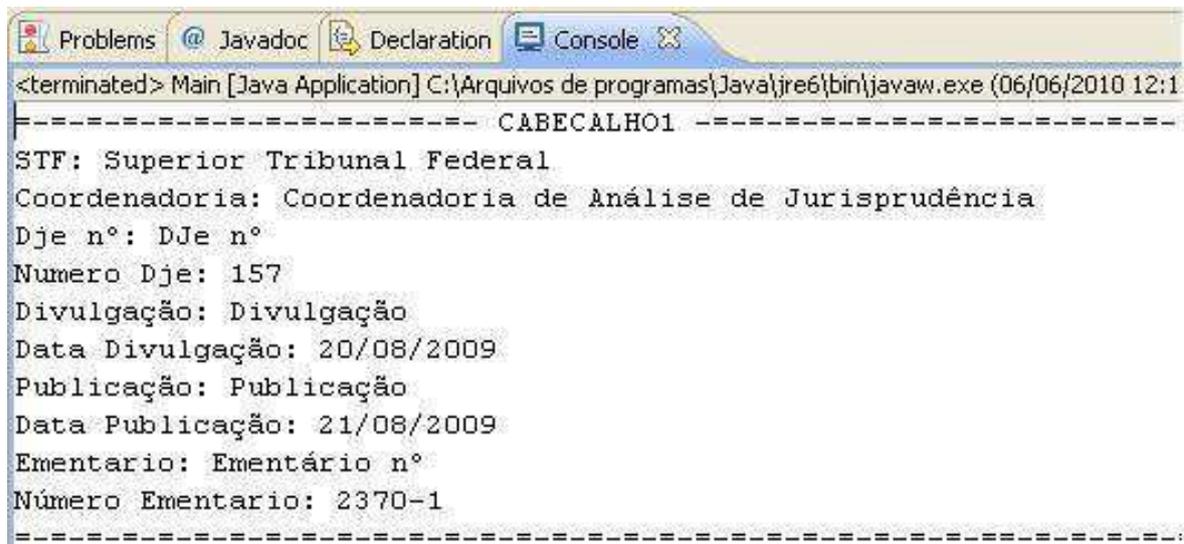
Para avaliar as gramáticas, estão sendo utilizadas as fontes descritas no apêndice A. A fase de validação das gramáticas ainda está em desenvolvimento, e até o presente momento, foram obtidos os seguintes resultados:

1. Compilação sem erros das gramáticas dos dispositivos legais e documentos jurídicos;
2. Extração de atributos dos documentos jurídicos do STF.

Os primeiros testes foram realizados com acórdãos do STF, e os resultados da aplicação da gramática foram satisfatórios. Foram extraídos quase todos os atributos, faltando apenas as *partes* do processo e as *referências*. Provavelmente, a dificuldade de extração destas se

deve ao fato da complexidade de sua descrição. No entanto, acredita-se que com os devidos ajustes, a extração de todas as informações será possível.

Para exemplificar os resultados da extração, nas Figuras 5.2, 5.3 e 5.4 foram apresentados os atributos extraídos do Agravo Regimental na Ação Cível Originária, nº1.1191-2 de Roraima, acórdão proveniente do STF.



```
<terminated> Main [Java Application] C:\Arquivos de programas\Java\jre6\bin\javaw.exe (06/06/2010 12:1
----- CABECALHO1 -----
STF: Superior Tribunal Federal
Coordenadoria: Coordenadoria de Análise de Jurisprudência
Dje nº: DJe nº
Numero Dje: 157
Divulgação: Divulgação
Data Divulgação: 20/08/2009
Publicação: Publicação
Data Publicação: 21/08/2009
Ementario: Ementário nº
Número Ementario: 2370-1
-----
```

Figura 5.2: Extração de atributos de um acórdão do STF - Parte 1

Para a realização dos testes, o acórdão foi replicado em um arquivo do tipo *texto*, pois esse é o formato que o *SableCC* aceita como entrada. Devido ao tamanho de alguns atributos (a exemplo de ementa, relatório e voto), apenas uma parte deles foi descrita no arquivo, o que não diminui a qualidade dos testes.

Os próximos passos para a validação da gramática do STF são: trabalhar na extração (i) das *partes* do processo e (ii) das referências a outros documentos. Em seguida, serão testadas as gramáticas dos documentos do STJ e dos dispositivos legais. Como estruturalmente os documentos do STF e STJ são bastante parecidos, acredita-se que os resultados dos testes com os documentos deste tribunal também serão positivos.

O módulo de Extração de Informação, responsável pela validação da descrição formal das fontes, está sendo desenvolvido com a colaboração de três graduandos do curso de Ciência da Computação da Universidade Federal de Campina Grande, como projeto da disciplina Laboratório de Engenharia de Software.

```

Problems @ Javadoc Declaration Console
<terminated> Main [Java Application] C:\Arquivos de programas\Java\jre6\bin\javaw.exe (06/06/2010 12:10:17)
----- CABECALHO2 -----
Data do Julgamento: 01/07/2009
Orgão Julgador: tribunal pleno
-----

----- PARTE -----
Titulo: TÍTULO
Dois Pontos:
Nome: carmen lucia
-----

----- EMENTA -----
Ementa: EMENTA
Texto Ementa:
agravo regimental na acao civil originaria. concurso publico para provimento de cargo de professor pa
ra magisterio indigena. incompetencia do supremo tribunal federal. art. 102, inc. i, alinea f, da con
stituicao da republica. retorno dos autos a origem. agravo regimental ao qual se nega provimento. ...
-----

----- ACORDÃO -----
Acordão: ACÓRDÃO
Texto Acordão:
visto, relatos e discutidos estes autos, acordam os ministros do supremo tribunal federal, em tribunal
pleno, sob a presidencia do ministro gilmar mendes, na conformidade da ata de julgamento e das notas taqu
igraficas, a unanimidade, em negar provimento ao recurso de agravo, nos termos do voto da relatora. ...
-----

```

Figura 5.3: Extração de atributos de um acórdão do STF - Parte 2

```

Problems @ Javadoc Declaration Console
<terminated> Main [Java Application] C:\Arquivos de programas\Java\jre6\bin\javaw.exe (06/06/2010 12:10:17)
----- RELATÓRIO -----
Relatorio: RELATÓRIO
Texto Relatório:
a senhora ministra carmem lucia - (relatora) : 1. agravo regimental na acao civil publica interposto pelo estado
de roraima, em 27.11.2008, contra decisao pela qual declarei "a incompetencia deste supremo tribunal federal par
a processar e julgar a acao civil publica n. 2007.42.00.002.845 - 2 e determinei o retorno dos autos ao juizo da
2a vara federal da secao judiciaria de roraima (art. 21, 2o, do regimento interno do supremo tribunal federal(
2a vara federal da secao judiciaria de roraima (art. 21, 2o, do regimento interno do supremo tribunal federal(
ados pelos agravados para que se encaminhasse a assembleia legislativa do estado de roraima projeto de lei de cr
iacao de carreira especifica do magisterio indigena "ferem frontalmente o pacto federativo previsto nos arts. 1o
, c(barra)c art. 18, da constituicao federal, pois requerem que o estado - membro legisle sobre materia de compet
encia privativa da uniao (art. 22, inc. xiv, da constituicao da republica)" (fl. 453, grifos no original). (...).
Fim Relatório: é o relatório.
-----

----- VOTO -----
Voto: VOTO
Texto Voto:
a senhora ministra carmem lucia - (relatora) : 1. em 30.6.2008, com respaldo em assentada jurisprudencia, de
clarei a incompetencia do supremo tribunal federal para processar e julgar a acao civil publica n. 2007.42.00
.002.845 - 2 e determinei o retorno dos autos ao juizo da 2a vara federal da secao judiciaria de roraima....
Fim Voto: é como voto.
-----

----- EXTRATO DA ATA -----
Extrato: EXTRATO DA ATA
Tipo Dj: AG. REG.NA AÇÃO CÍVEL ORIGINÁRIA
Numero Dj: 1191-2
Proced.: PROCED.:
Local: roraima
Decisão: Decisão
Decisão:
o tribunal, por unanimidade e nos termos do voto da relatora, negou provimento ao recurso de agravo. ause
ntes, justificadamente, os senhores ministros celso de mello, ellen gracia, eros grau e, licenciado, o sen
hor ministro menezes direito. presidiu o julgamento o senhor ministro gilmar mendes. plenaro, 01.07.2009.
-----

```

Figura 5.4: Extração de atributos de um acórdão do STF - Parte 3

5.4 Considerações Finais

Neste capítulo, foram apresentadas as principais regras da gramática que descrevem formalmente as fontes do Direito. Inicialmente, foram apresentadas as regras referentes aos dispositivos legais e seus elementos. Em seguida, foram descritos formalmente os documentos jurídicos do STF e STJ, além dos seus aspectos referenciais. Por fim, foi apresentada a forma como está sendo feita a avaliação da gramática.

No próximo capítulo, será apresentado o projeto do sistema de gerenciamento de conhecimento legal *JudLaw*. Serão descritos seus módulos e funcionalidades, além do protótipo já desenvolvido, o qual implementa e testa a infraestrutura do sistema.

Capítulo 6

JudLaw

Neste capítulo será apresentado o projeto do *JudLaw*, um sistema de gerenciamento de conhecimento legal. Na Seção 6.1 a arquitetura do sistema será introduzida, além da descrição dos principais módulos. Devido ao escopo definido para este trabalho, apenas a infraestrutura do sistema foi implementada integralmente. Na Seção 6.2 será descrito o protótipo do sistema, o qual foi utilizado para validar o modelo *ITMJudLaw*, e os resultados obtidos dos módulos em desenvolvimento.

6.1 Arquitetura

Em Riveret [64] são descritas as principais funcionalidades que os SGCLs precisam contemplar: *aquisição, modelagem, armazenamento, recuperação, disseminação e manutenção* do conhecimento legal. No entanto, dos sistemas encontrados na literatura, a maioria apenas se preocupou com algumas delas. Essas funcionalidades foram detalhadas na Seção 2.5.

A principal preocupação ao se projetar o *JudLaw* foi minimizar - ou dirimir, quando possível - os problemas enfrentados pelos operadores do Direito na pesquisa normativo-jurisprudencial. O primeiro deles é a busca por documentos em fontes *descentralizadas*. Atualmente, existem profissionais contratados apenas para procurar as fontes do Direito mais atuais para povoar as bases de dados de instituições governamentais. Para centralizar essa função, foi projetado o módulo de *Coleta de Dados*, o qual acessa diversas fontes a partir de única busca feita pela usuário.

No entanto, as fontes estão publicadas em formato semi-estruturado, o qual não é adequado para a composição da base de dados de um SGCL. Como uma das funcionalidades de um sistema dessa natureza é permitir que o usuário faça diversas operações sobre os docu-

mentos, é interessante que essas fontes já estejam disponíveis de forma estruturada.

Pensando nesse cenário, foi projetado o módulo de *Extração de Informação* (EI). Ao receber como entrada os documentos no formato semi-estruturado, o módulo utiliza a descrição formal descrita neste trabalho em conjunto com um interpretador de gramáticas, a fim de extrair de forma automática todas as informações importantes do documento. Estando estruturados, os documentos são armazenados em um banco de dados.

Com a base de dados povoada, o usuário pode fazer realizar buscas e operações sobre os documentos, através do módulo de *Recuperação de Informação* (RI). Com o *JudLaw*, o especialista será capaz de fazer buscas distribuídas, acessando simultaneamente tanto a base local quanto as fontes na Internet. Por fim, a pesquisa feita pelo usuário pode ser visualizada na *Interface* do *JudLaw*, a qual é o meio de interação do operador com o sistema. A arquitetura do *JudLaw* está ilustrada na Figura 6.1.

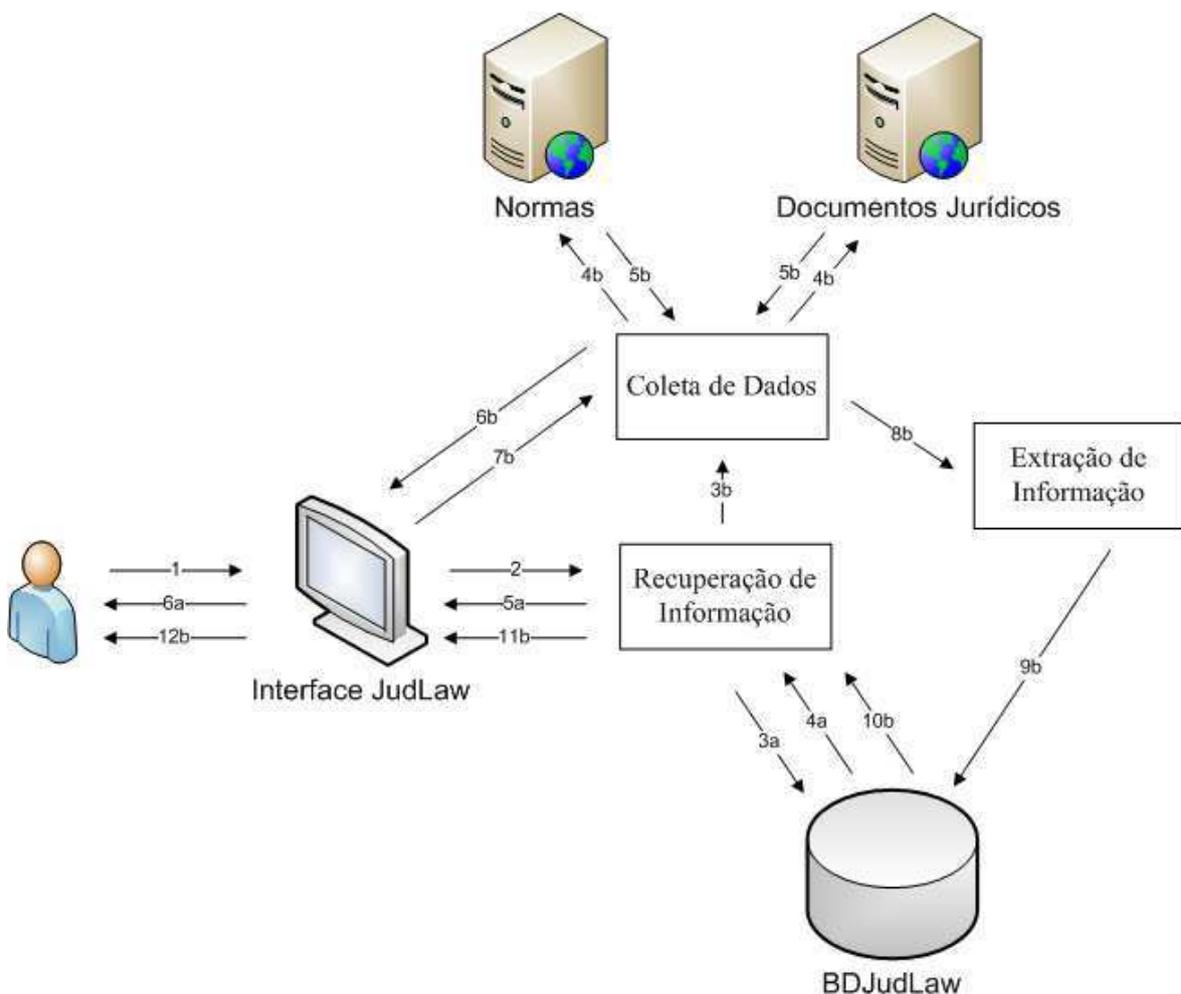


Figura 6.1: Arquitetura do JudLaw

Em resumo, os principais componentes do JudLaw são os módulos de: *coleta de dados* (Seção 6.1.1), *extração de informação* (Seção 6.1.2), *recuperação de informação* (Seção 6.1.3) e interação com o usuário, através de uma *interface* (Seção 6.1.4). O fluxo de execução apresentado na arquitetura pode ser descrito da seguinte forma:

1. O usuário faz uma busca sobre uma fonte específica (uma lei, por exemplo) ou sobre um assunto desejado, interagindo com a interface do JudLaw (1);
2. A busca é passada para o módulo de RI (2), o qual distribui a consulta para as fontes da Internet e para a base local (3a e 3b);
3. Simultaneamente, os resultados locais (5a) e remotos (6b) são mostrados ao usuário através da interface do *JudLaw*;
4. Caso o usuário queira finalizar a consulta apenas com os resultados locais (*BDJudLaw*), a busca está concluída (6a);
5. Caso ele queira utilizar fontes remotas, o usuário informa ao sistema quais fontes (6b e 7b) deverão ser passadas para o módulo de Extração de Informação (8b), onde serão processadas e armazenadas na base de forma estruturada (9b);
6. Um novo resultado é gerado e retornado ao usuário (11b e 12b), contendo as novas fontes obtidas remotamente.

6.1.1 Coleta de Dados

Quando um usuário realiza uma busca no *JudLaw*, ela é distribuída para a base local e para fontes remotas. Para esta segunda situação, foi projetado o módulo de *Coleta de Dados* (CD), o qual é responsável pela visualização e *download* dessas fontes, caso seja de interesse do usuário. O módulo de *CD* se comunica tanto com a *interface* do sistema quanto com o módulo de *Extração de Informação*.

Ao exibir os resultados para o usuário, este pode escolher quaisquer documentos para fazerem parte da base de dados do *JudLaw*. Após escolhidos, o módulo de *CD* baixa os documentos e os converte para formato *texto*, para que eles possam ser processados pelo módulo de *EI*.

Motor de Busca Remota

O motor de busca remota do *JudLaw* é responsável por receber os parâmetros da busca do usuário, compor a *URL* de acordo com o padrão da fonte remota a ser utilizada e distribuir a busca entre os sítios. Naturalmente, é preciso que sejam conhecidas as gramáticas de composição de *URLs* adotadas pelas fontes, por isso foram projetados arquivos de configuração para esse fim.

Na Figura 6.2 é apresentado o esquema do motor de busca remota do *JudLaw*. A classe *LeitorDeEngines* lê o arquivo *sites.list*, o qual possui os nomes dos arquivos de configurações de todas as fontes remotas onde serão feitas as buscas. Cada arquivo de configuração possui as descrições dos mecanismos utilizados por um sítio, a exemplo do portal LexML.

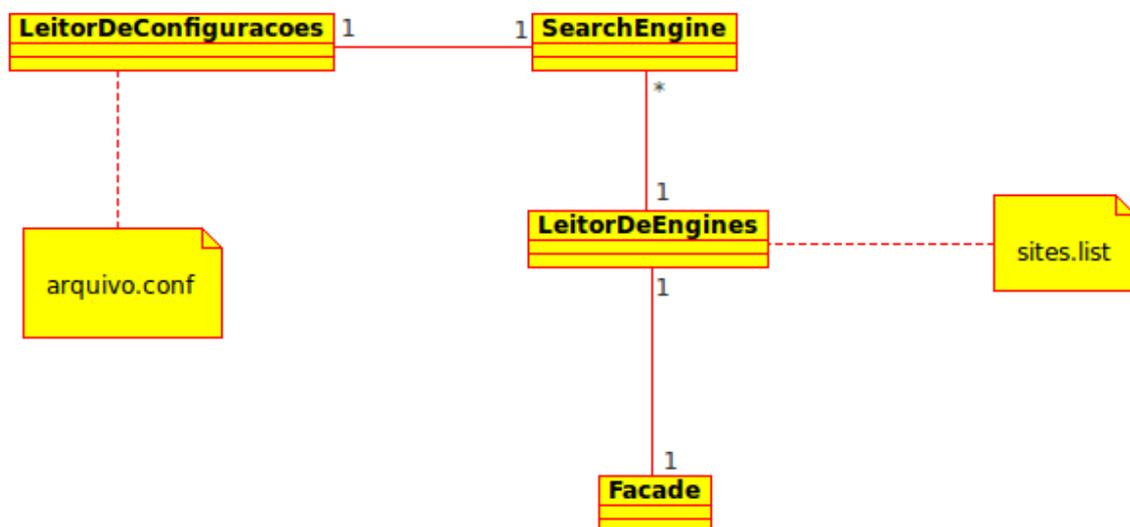
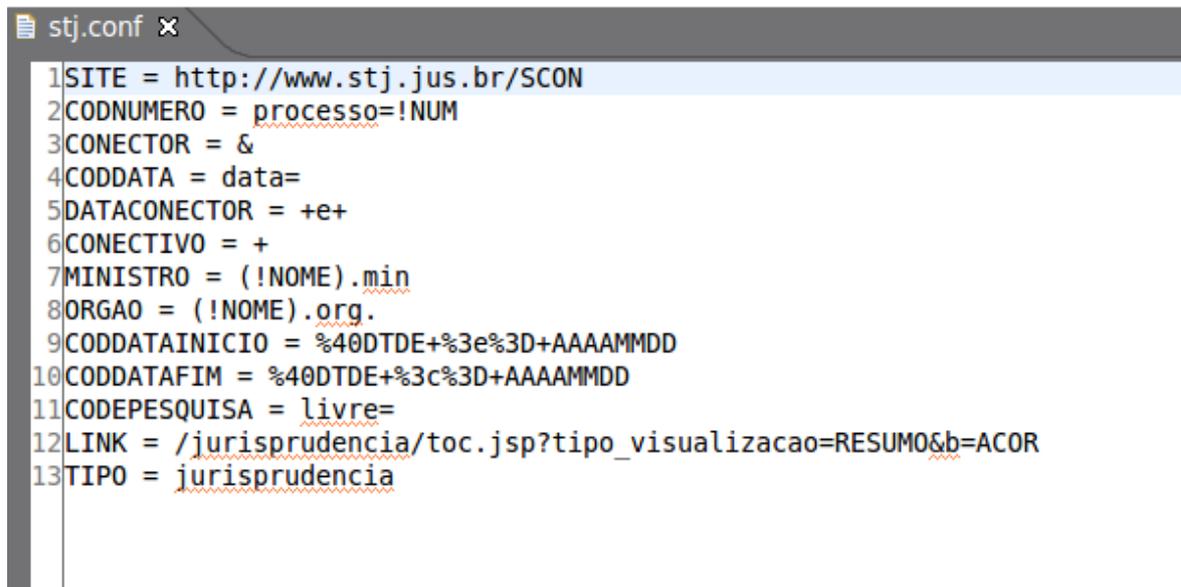


Figura 6.2: Esquema do motor de busca remota

Na Figura 6.3, é ilustrado o arquivo de configuração para a interação com o motor de busca do sítio do STJ. Os valores escritos em maiúsculo correspondem aos atributos do sistema, os quais irão compor a URL. A exceção é o valor *TIPO*, que apenas indica qual o tipo de fonte a ser pesquisada, a exemplo de *jurisprudência* ou *legislação*. Devido à pesquisa ser feita em ambiente HTML, alguns códigos de composição de URLs também são

contemplados (a exemplo de %40 e %3).

A screenshot of a text editor window titled 'stj.conf'. The window contains 13 lines of configuration code. The code is as follows:

```
1SITE = http://www.stj.jus.br/SCON
2CODNUMERO = processo=!NUM
3CONECTOR = &
4CODDATA = data=
5DATACONECTOR = ++
6CONECTIVO = +
7MINISTRO = (!NOME).min
8ORGAO = (!NOME).org.
9CODDATAINICIO = %40DTDE+%3e%3D+AAAAMMDD
10CODDATAFIM = %40DTDE+%3c%3D+AAAAMMDD
11CODEPESQUISA = livre=
12LINK = /jurisprudencia/toc.jsp?tipo_visualizacao=RESUMO&b=ACOR
13TIPO = jurisprudencia
```

Figura 6.3: Arquivo de configuração do sítio do STJ

A classe *Facade* serve de fachada para a comunicação entre a interface gráfica e a lógica do motor. O fluxo de execução do motor de busca do JudLaw pode ser descrito da seguinte forma:

1. A classe *LeitorDeEngines* lê todas as fontes remotas a serem pesquisadas no arquivo *sites.list* e cria uma instância da classe *SearchEngine* para cada fonte;
2. Cada instância de *SearchEngine*, referente a uma fonte remota, recebe as informações do arquivo de configuração correspondente à fonte, através da classe *LeitorDeConfiguracoes*;
3. Cada instância de *SearchEngine* compõe a *URL* de acordo com as configurações recebidas e distribui para os fontes remotas, interagindo com seus motores de busca.

O objetivo é que o *JudLaw* faça essa captura de forma automática, dispondo de uma interface amigável com botões de *download*, para que o usuário apenas precise clicar neles para baixar os documentos.

6.1.2 Extração de Informação

Assim que uma nova fonte é baixada para o sistema, ela precisa ser processada para que seja armazenada de forma estruturada no banco de dados. Essa operação é realizada pelo módulo de Extração de Informação (EI), o qual trabalha em conjunto com o módulo de Coleta de Dados. Estes dois módulos estão atualmente em desenvolvimento.

Apesar do interesse em se extrair automaticamente as informações das fontes, o *JudLaw* vai prover uma interface para que os usuários verifiquem se os atributos foram extraídos corretamente e, caso não tenham sido, poderão ser feitas alterações antes do armazenamento da fonte na base de dados.

No entanto, um dos desafios da aplicação real do módulo de extração é fazer a limpeza dos dados que vêm do módulo de CD. Após o término da fase de validação da descrição formal, o *tratamento dos dados* advindos do módulo de coleta de dados será o próximo ponto a ser trabalhado. Esse tratamento é uma das funções do módulo de Extração da Informação.

6.1.3 Recuperação de Informação

Após o estudo dos módulos de busca dos *SGCLs* encontrados na literatura, foi constatado que a maioria disponibiliza apenas buscas por palavras-chave ou por atributos, além de não agregar nenhum valor semântico aos resultados. Um dos objetivos do *JudLaw* é oferecer um módulo de *Recuperação de Informação* (RI), através do qual os documentos serão *indexados* e as buscas possam se tornar mais *eficientes*.

Quando um sistema não implementa os conceitos da RI, as buscas geralmente apresentam resultados insatisfatórios para os usuários. Como exemplo prático, caso um usuário queira pesquisar sobre “lesão corporal”, serão retornados apenas os documentos que contenham essas duas palavras ou qualquer uma delas, dependendo do conector utilizado pelo usuário (*AND* ou *OR*).

No entanto, poderia ser interessante para o usuário que também fossem retornados os documentos que contivessem o termo “agressão física”. Em contrapartida, será retornado um documento no qual aparece somente uma vez a expressão “lesão corporal”, mas que não tem a ver com o tema central. Sendo assim, esse documento potencialmente não é de interesse do usuário.

Uma forma de resolver essas limitações é a aplicação de técnicas de Recuperação de

Informação, a exemplo de um *ranking* de documentos de acordo com a quantidade de vezes que o termo ou expressão aparece no texto. Ou então, a criação de um *thesaurus* de termos que torne possível o mapeamento de palavras ou expressões semanticamente parecidas, a exemplo de “assassinato” e “homicídio”.

Dessa forma, as pesquisas podem se tornar mais eficientes, retornando apenas os documentos que realmente interessam ao operador do Direito. Essa prática é útil para poupar tempo do especialista, pois este não terá mais que perder horas ou até dias pesquisando fontes que não vão contribuir de forma efetiva para sua pesquisa normativo-jurisprudencial.

Como a ênfase do modelo *ITMJudLaw* são as referências e aspectos temporais dos documentos, também serão disponibilizadas ao usuário consultas avançadas de navegação. A partir de um documento, o especialista poderá acessar todos os tipos de referências feitas e sofridas não apenas por ele, mas também por seus elementos, separadamente.

O mesmo conceito de navegação será aplicado a consultas temporais dos documentos, ou seja, todas as versões antigas e atuais de uma fonte poderão ser acessadas. A partir de um artigo, por exemplo, o operador poderá consultar:

1. Todas as versões do artigo;
2. Todas as versões dos elementos filhos do artigo (parágrafos ou incisos);
3. Todas as versões do elemento pai do artigo (a norma).

O usuário também poderá *reconstruir temporalmente* uma norma, a fim de saber exatamente o seu conteúdo em um determinado ponto do tempo. Além disso, caso ele queira saber se todas as referências de um documento estão de acordo com o ordenamento jurídico, a operação de *verificação da consistência temporal* poderá ajudá-lo nessa tarefa.

6.1.4 Interface

O *JudLaw* ainda não possui uma interface bem definida, embora alguns dos módulos (CD e EI) já ofereçam a possibilidade de o usuário interagir com o protótipo. Para o desenvolvimento da interface do sistema, serão consultados operadores do Direito a fim de definir a melhor forma de interação com as informações, sob a ótica de um especialista.

Através de entrevistas informais com promotores e advogados e do estudo das interfaces de outros sistemas, algumas funcionalidades foram levantadas, a serem implementadas na interface do *JudLaw*:

1. *Pesquisas*: buscas por fontes, tanto locais quanto remotas. Ainda deverão ser disponibilizadas as opções de busca *integrada* ou *separada* de acordo com o tipo do documento;
2. *Temporalidade*: as versões das normas e de seus elementos devem estar disponíveis que o operador possa navegar entre elas, em qualquer sentido na hierarquia dos textos legais. O especialista ainda terá as opções de reconstruir temporalmente uma norma e verificar a consistência temporal de qualquer fonte do Direito;
3. *Referencialidade*: ao acessar um documento, as referências a outras fontes devem estar destacadas, além de acessíveis. Em relação aos dispositivos legais, os operadores terão a opção de navegar em qualquer direção tanto na própria hierarquia dos textos quanto em suas referências;
4. *Edição*: estará disponível para os usuários as operações de manutenção da base de dados.

Essas funcionalidades farão parte da primeira versão da interface do *JudLaw*, porém uma parte muito importante no seu desenvolvimento é o *feedback* de operadores do Direito. Para isso, serão feitos testes de usabilidade da interface do *JudLaw* com diversos tipos de especialistas, dentre eles, advogados, promotores e juízes.

6.2 Protótipo

Devido ao escopo desta dissertação, não foi possível implementar todos os módulos projetados para o *JudLaw*. No entanto, a *infraestrutura* do sistema encontra-se desenvolvida e implementada. Inicialmente, o objetivo da implementação da infraestrutura foi validar o modelo proposto neste trabalho, porém novos módulos foram e estão sendo desenvolvidos.

Atualmente, o protótipo conta com mais de 5000 linhas de código, sendo mais de 2000 só de testes de unidade, a fim de comprovar a corretude das operações implementadas. A

primeira fase foi completada, relativa à implementação do banco de dados utilizando o modelo *ITMJudLaw*, e estão em fase de desenvolvimento os módulos de coleta de dados e de extração da informação. Após o término desses, o módulo de *Recuperação da Informação* e a *interface gráfica* serão contemplados.

O módulo de Coleta de Dados está sendo desenvolvido com a colaboração de dois graduandos do curso de Ciência da Computação da Universidade Federal de Campina Grande, como projeto da disciplina Laboratório de Engenharia de Software. O módulo está habilitado para lidar com os motores de buscas de três fontes: portal *LexML* e sítios do *STF* e *STJ*. Os três possuem uma característica em comum: a busca é feita por *composição de URL*, facilitando a interação de outros sistemas com seus módulos de busca.

Foi desenvolvido um protótipo da interface gráfica de pesquisa remota, ilustrado na Figura 6.4, o qual possui elementos de pesquisa tanto para dispositivos legais quanto para documentos jurídicos. Como exemplo prático, caso o usuário deseje fazer uma busca no campo livre por “assassinato” e o tipo da busca esteja marcado como *Tudo*, serão retornados resultados das três fontes remotas: STJ, STF e LexML.

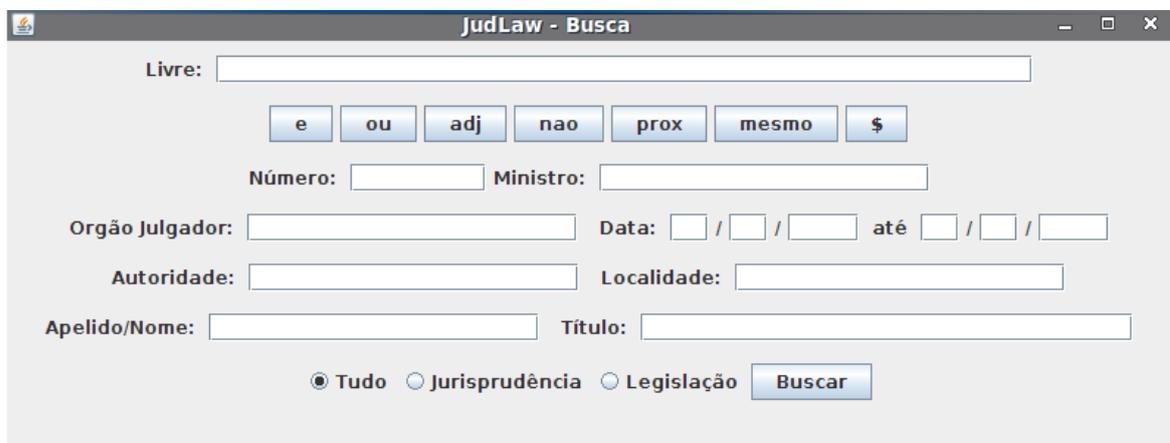


Figura 6.4: Protótipo da interface gráfica do módulo CD. A interface, intitulada "JudLaw - Busca", apresenta um formulário com os seguintes elementos:

- Um campo de texto rotulado "Livre:" para a entrada de termos de busca.
- Botões de filtro: "e", "ou", "adj", "nao", "prox", "mesmo" e "\$".
- Campos para "Número:" e "Ministro:".
- Campos para "Orgão Julgador:" e "Data:" (formato [] / [] / [] até [] / [] / []).
- Campos para "Autoridade:" e "Localidade:".
- Campos para "Apelido/Nome:" e "Título:".
- Opções de busca: radio buttons para "Tudo" (selecionado), "Jurisprudência" e "Legislação".
- Um botão "Buscar" para executar a consulta.

Figura 6.4: Protótipo da interface gráfica do módulo CD

Embora o motor de busca esteja plenamente desenvolvido, o módulo de coleta de dados ainda não está completo. Após escolher quais fontes o usuário deseja baixar para o *JudLaw*, ele precisa captar manualmente os *links* e informá-los ao sistema, para que essas possam ser baixadas. Quanto ao módulo de *Recuperação da Informação*, o protótipo do *JudLaw* já possibilita uma série pesquisas ao usuário:

1. *Sobre os atributos*: o documento não é considerado uma unidade indivisível, e cada

- componente pode ser acessado individualmente;
2. *Sobre as referências:* todas as referências ativas e passivas, tanto de um documento quanto de seus elementos, podem ser acessadas também individualmente;
 3. *Sobre as versões:* através da operação de reconstrução temporal dos dispositivos legais, é possível pesquisar todo seu histórico, incluindo seus elementos e suas referências;
 4. *Sobre a validade dos elementos:* através da operação de verificação da consistência temporal, o *JudLaw* analisa todas as referências de documentos jurídicos e normas em busca de inconsistências temporais.

A base de dados do protótipo é composta pelas fontes descritas no apêndice A, inseridas manualmente e de forma estruturada na base de dados, de acordo com a modelagem apresentada neste trabalho. Espera-se automatizar a operação de povoamento com os módulos de *Coleta de Dados e Extração de Informação*.

6.3 Considerações Finais

Neste capítulo, foi apresentado o projeto do sistema *JudLaw*, incluindo a descrição da sua arquitetura e de seus principais módulos, dos quais dois estão em fase de desenvolvimento. Além disso, no capítulo foram descritos os aspectos gerais do protótipo do sistema, fase atual na qual está o *JudLaw*. No próximo capítulo, serão apresentadas as considerações finais sobre este trabalho, além da descrição dos trabalhos futuros.

Capítulo 7

Considerações Finais e Trabalhos Futuros

Na Seção 7.1 serão feitas as considerações finais acerca deste trabalho. Será apresentado um resumo do atual panorama de como são feitas as pesquisas normativo-jurisprudenciais e as principais contribuições desta pesquisa para facilitar as tarefas dos operadores do Direito. Em seguida, na Seção 7.2, serão apresentados os trabalhos futuros em relação ao *JudLaw*, dos quais alguns já estão em desenvolvimento.

7.1 Considerações Finais

Atualmente, a pesquisa normativo-jurisprudencial é feita de forma manual em quase sua totalidade. São escassas as ferramentas que os operadores do Direito dispõem para a execução dessa tarefa. As únicas ferramentas disponíveis estão em órgãos governamentais, suas licenças custam caro e a única funcionalidade delas é o armazenamento dos documentos, de forma *textual*.

Os operadores perdem muito tempo procurando fontes em diversos sítios, os quais não oferecem uma busca adequada que retorne apenas os documentos relevantes. A forma como os documentos são publicados também não é a mais adequada, pois tratam-se de textos sem qualquer semântica agregada, quando na verdade, esses documentos são ricos tanto estruturalmente quanto em seu conteúdo.

Pode-se afirmar que no Brasil ainda há um vasto campo a ser explorado em relação aos sistemas de gerenciamento de conhecimento legal. Apesar da existência do *LexMLBR*, este ainda apresenta falhas em sua busca, além de não implementar todas as idéias que foram des-

critas em sua documentação. Não foram encontrados outros SGCLs voltados para as fontes brasileiras, e os sistemas de outros países também não contemplam todas as funcionalidades desejadas, segundo Riveret [64], deixando uma grande lacuna para pesquisas na área.

Com a utilização de sistemas gestores de conhecimento legal, é possível tornar o trabalho dos especialistas mais eficiente, ao destacar aspectos referenciais e temporais dos documentos, poupando verificações manuais, a exemplo da consistência temporal das fontes. Outra forma de facilitar a pesquisa normativo-jurisprudencial é através da centralização das fontes em um único repositório, aliada com a funcionalidade de navegação pelas referências dos documentos.

Esta pesquisa não visa a resolver todos os problemas levantados nesse cenário, mas sim dar o primeiro passo, provendo a infraestrutura necessária para isso. Em linhas gerais, as principais contribuições do trabalho podem ser enumeradas como: (i) modelos *TTMDoc*, *ITMJudLaw* e as operações associadas a este, (ii) gramática formal das fontes do Direito e (iii) projeto de um sistema de gerenciamento de conhecimento legal (*JudLaw*), do qual já foram obtidos resultados com dois dos módulos propostos e foi desenvolvido um protótipo com algumas funcionalidades implementadas e testadas.

Uma parte fundamental no desenvolvimento de um SGCL é a modelagem das suas fontes. A partir dessa premissa, foi desenvolvido o modelo genérico de dados para bancos textuais-temporais, o *TTMDoc*. No âmbito desta pesquisa, ele foi aplicado no domínio legal e jurídico, porém a proposta é que ele seja aplicável na modelagem de quaisquer tipos de documentos versionados. Exemplos de outras aplicações poderiam ser a escrita de um livro por diversos atores ou a elaboração de relatórios incrementais da execução de um projeto.

A partir da análise dos trabalhos relacionados à modelagem de fontes do Direito, e tendo como base o modelo *TTMDoc*, foi desenvolvido o modelo *ITMJudLaw*. O objetivo do modelo não foi apenas reunir as melhores práticas encontradas na literatura, mas também preencher as lacunas existentes. A ênfase do modelo foram as *estruturas*, *referências* e os aspectos *temporais* das fontes.

Nem todos os modelos representaram a integração entre as fontes do Direito, apesar de ser um comportamento básico dessas. Foram definidas também operações de consulta a uma base *ITMJudLaw*, possibilitando a navegação *referencial* e *temporal* (versionamento) entre os documentos, além de verificações temporais (reconstrução e consistência). Alguns

aspectos não foram incluídos, a exemplo das ontologias, pois a semântica das fontes não foi objeto de estudo da pesquisa.

Uma das contribuições desta pesquisa é inédita em âmbito nacional, ou seja, não foram encontrados outros trabalhos nacionais que abordassem essa temática. Trata-se da *formalização* das fontes do Direito, utilizando a Forma Estendida de Backus-Naur. Com a descrição de uma gramática formal para esses documentos, é possível realizar a extração automática de informações, visando ao seu armazenamento estruturado.

Como última contribuição, foi apresentado o projeto de um SGCL que atende aos requisitos descritos por Riveret [64]. Apesar de trabalhos isolados tratarem desses requisitos, dos sistemas atualmente em funcionamento, nenhum deles implementa todos. Dos principais módulos descritos, dois estão em desenvolvimento (Coleta de Dados e Extração de Informação) e já apresentaram resultados, citados anteriormente.

O âmbito de aplicação de sistemas gestores de conhecimento legal é bastante vasto. Todos os profissionais que trabalham com as fontes do Direito podem ter seu trabalho facilitado com um SGCL, o qual pode ser aplicado em escritórios de advocacia, fóruns, tribunais e órgãos governamentais. Inclusive, vários governos da Europa estão patrocinando projetos nessa área para melhor documentar e acessar suas informações.

7.2 Trabalhos Futuros

Devido à complexidade e porte de um SGCL, não foi possível implementar todas as funcionalidades desejadas durante esta pesquisa. Sendo assim, foram determinadas algumas proposições para trabalhos futuros relacionados com a primeira versão do *JudLaw*:

- Conclusão dos módulos em desenvolvimento: Coleta de Dados e Extração da Informação;
- Desenvolvimento do módulo de Recuperação de Informação;
- Desenvolvimento da interface do *JudLaw*;
- Aplicação e testes de usabilidade em ambientes de trabalho de operadores do Direito;
- Validação estatística dos resultados das consultas, com a utilização de métricas como precisão, cobertura e *f-score*.

Após o lançamento da primeira versão do sistema, concluindo a abordagem estrutural das fontes, pretende-se desenvolver a interpretação semântica das fontes para dar suporte a consultas mais elaboradas e processos de tomada de decisão. Para isso, será feito um estudo aprofundado na semântica das fontes, provavelmente sendo utilizadas ontologias e técnicas de Inteligência Artificial, como o Raciocínio Baseado em Casos e o estudo de formalismos lógicos aplicados ao Direito. A partir da abordagem semântica, será lançada a segunda versão do *JudLaw*. Outros trabalhos possíveis podem estar relacionados com a aplicação do modelo *TTMDoc* a outros domínios como normas empresariais e outros textos fortemente versionados.

Bibliografia

- [1] AKOMA NTOSO 1.0 Schema. Disponível em <http://www.akomantoso.org/release-notes/akoma-ntoso-1.0-schema/referencemanual-all-pages>, Dezembro 2009. Acesso em Junho 2010.
- [2] Araújo, I., Drumond L., Mariano R. and Girardi, R. ONTOJURIS E ONTOTRIB - Ontologias para a modelagem do conhecimento jurídico. In *Seminário de Pesquisa em Ontologia no Brasil*, 2008.
- [3] Arnold-Moore, T. Automatically processing amendments to legislation. In *ICAAIL '95: Proceedings of the 5th international conference on Artificial intelligence and law*, pages 297–306, New York, NY, USA, 1995. ACM.
- [4] Arnold-Moore, T. Automatic generation of amendment legislation. In *ICAAIL '97: Proceedings of the 6th international conference on Artificial intelligence and law*, pages 56–62, New York, NY, USA, 1997. ACM.
- [5] Arnold-Moore, T. About time: Legislation’s forgotten dimension. In *Proceedings of the 3rd AustLII Law via the Internet Conference*, Sydney, Australia, 2001.
- [6] Arnold-Moore, T. Point in time publication for legislation (xml and legislation). In *Proceedings of the 6th Conference on Computerisation of Law via the Internet*, Paris, France, 2004.
- [7] Arnold-Moore, T. and Cledes, J. Connected to the Law: Tasmanian Legislation Using EnAct. *Journal of Information, Law and Technology*, 2000.
- [8] Ashley, K. D. Ontological requirements for analogical, teleological, and hypothetical legal reasoning. In *ICAAIL '09: Proceedings of the 12th International Conference on Artificial Intelligence and Law*, pages 1–10, New York, NY, USA, 2009. ACM.

- [9] Atkinson, K. Introduction to special issue on modelling legal cases. *Artif. Intell. Law*, 16(4):329–331, 2008.
- [10] Bekiari, C., Doerr M. and Le Boeuf. P. FRBR - object-oriented definition and mapping to FRBRer. Disponível em http://cidoc.ics.forth.gr/docs/frbr_oo/frbr_docs/FRBRoo_V1.0_draft_2009_may_.pdf, Maio 2009. Acesso em Junho 2010.
- [11] Benjamins, V., Contreras, J., Casanovas, P., Ayuso, M., Becue, M., Lemus, L. and Urios, C. Ontologies of professional legal knowledge as the basis for intelligent it support for judges. *Artificial Intelligence and Law*, 12(4):359–378, December 2004.
- [12] Bergel, J.L. *Teoria Geral do Direito*. Martins Fontes, São Paulo, 2001.
- [13] Biagioli, C. and Grossi, D. Formal aspects of legislative meta-drafting. In *Proceeding of the 2008 conference on Legal Knowledge and Information Systems*, pages 192–201, Amsterdam, The Netherlands, 2008. IOS Press.
- [14] Biagioli, C., Francesconi, E., Passerini, A., Montemagni, S. and Soria, C. Automatic semantics extraction in law documents. In *ICAAIL '05: Proceedings of the 10th international conference on Artificial intelligence and law*, pages 133–140, New York, NY, USA, 2005. ACM.
- [15] Bobbio, N. *Teoria do Ordenamento Jurídico*. UnB, Brasília, 10th edition, 1999.
- [16] Boer, A. and Winkels, R. and Vitali, F. Proposed XML Standards for Law: MetaLex and LKIF. In *Proceeding of the 2007 conference on Legal Knowledge and Information Systems*, pages 19–28, Amsterdam, The Netherlands, 2007. IOS Press.
- [17] Boer, A., Gordon, T. F., van den Berg, K., Di Bello, M., Föhréc, A. and Vas, R. *Specification of the legal knowledge interchange format*, 2007. Deliverable 1.1, European project for Standardized Transparent Representations in order to Extend Legal Accessibility (ESTRELLA, IST-4-027655).
- [18] Boer, A., Hoekstra, R., de Maat, E., Hupkes, E., Vitali, F., Palmirani, M., Rátai, B. CEN MetaLex Workshop Agreement. <http://www.metalex.eu/WA/proposal>, 2009. Acesso em Junho 2010.

- [19] Boer, A., Hoekstra, R., Winkels, R., van Engers, T. and Willaert, F. METAlex: Legislation in XML. In *Proceeding of the 2002 conference on Legal Knowledge and Information Systems*, pages 1–10, Amsterdam, The Netherlands, 2002. IOS Press.
- [20] Boer A., Winkels R., van Engers T. and de Maat, E. A content management system based on an event-based model of version management information in legislation. In *Proceeding of the 2004 conference on Legal Knowledge and Information Systems*, pages 19–28, Amsterdam, The Netherlands, 2004. IOS Press.
- [21] Boer A., Winkels R., van Engers T. and de Maat, E. Time and versions in METAlex XML. In *Proceeding of the Workshop on Legislative XML*, 2004.
- [22] Breuker, J., Valente, A., and Winkels, R. Legal ontologies in knowledge engineering and information management. *Artificial Intelligence and Law*, 12(4):241–277, December 2004.
- [23] Brighi, R. and Palmirani, M. Legal text analysis of the modification provisions: a pattern oriented approach. In *ICAAIL '09: Proceedings of the 12th International Conference on Artificial Intelligence and Law*, pages 238–239, New York, NY, USA, 2009. ACM.
- [24] Bueno, T. C. D´A. Recuperação da informação jurídica em sistemas baseado em casos. Master's thesis, Pós-graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, Brasil, 1999.
- [25] Carvalho Junior, J. Definição de um Padrão Baseado em XML para Descrição de Normas Jurídicas Brasileiras. Master's thesis, Pós-graduação em Informática, Universidade Federal do Paraná, Curitiba, Brasil, 2004.
- [26] Carvalho, K. G. *Técnica Legislativa*. Del Rey, Belo Horizonte, 2001.
- [27] Casanovas, P., Casellas, N., Tempich, C. and Vrandečić, D. and Benjamins, R. OPJK and DILIGENT: ontology modeling in a distributed environment. *Artif. Intell. Law*, 15(2):171–186, 2007.

- [28] Casellas, N., Casanovas, P., Vallbé, J., Poblet, M., Ramos, F., Gorroñoigoitia, J., Contreras, J., Blázquez, M. and Benjamins, R. Juriservice ii: ontology development and architectural design. In *ICAAIL '05: Proceedings of the 10th international conference on Artificial intelligence and law*, pages 188–194, New York, NY, USA, 2005. ACM.
- [29] Couto Filho, R. S. Considerações sobre a validade, a vigência e a eficácia das normas jurídicas. Online at <http://jus2.uol.com.br/doutrina/texto.asp?id=21>, Dezembro 2000. Acesso em Junho 2010.
- [30] de Maat, E. and Winkels, R. Categorisation of norms. In *Proceeding of the 2007 conference on Legal Knowledge and Information Systems*, pages 79–88, Amsterdam, The Netherlands, 2007. IOS Press.
- [31] de Maat, E. and Winkels, R. Automatic classification of sentences in dutch laws. In *Proceeding of the 2008 conference on Legal Knowledge and Information Systems*, pages 207–216, Amsterdam, The Netherlands, 2008. IOS Press.
- [32] de Maat, E. and Winkels, R. A next step towards automated modelling of sources of law. In *ICAAIL '09: Proceedings of the 12th International Conference on Artificial Intelligence and Law*, pages 31–39, New York, NY, USA, 2009. ACM.
- [33] de Maat, E., Winkels, R. and van Engers, T. Automated detection of reference structures in law. In *Proceeding of the 2006 conference on Legal Knowledge and Information Systems*, pages 41–50, Amsterdam, The Netherlands, 2006. IOS Press.
- [34] Dunham, M. H. *Data Mining: Introductory and Advanced Topics*. Prentice Hall, August 2002.
- [35] Farah, F. and Rousselot, F. DARES: documents annotation and recombining system-application to the European law. *Artif. Intell. Law*, 15(2):83–102, 2007.
- [36] International Organization for Standardization. ISO/IEC 14977:1996. Information technology – Syntactic metalanguage – Extended BNF, 1996.
- [37] França, R. L. *Lei (sua eficácia no tempo)*, volume 31. Borsoi, Rio de Janeiro, 1947.

- [38] Francesconi, E. and Peruginelli, G. Searching and retrieving legal literature through automated semantic indexing. In *ICAAIL '07: Proceedings of the 11th international conference on Artificial intelligence and law*, pages 131–139, New York, NY, USA, 2007. ACM.
- [39] Gaizauskas, R. and Wilks, Y. Information Extraction: Beyond Document Retrieval. *Computational Linguistics and Chinese Language Processing*, 3(2):17–60, Agosto 1998.
- [40] Gordon, T. F. and Walton, D. The carneades argumentation framework using presumptions and exceptions to model critical questions. In *Proceeding of the 2006 conference on Computational Models of Argument*, pages 195–207, Amsterdam, The Netherlands, 2006. IOS Press.
- [41] Governatori, G., Rotolo, A., Riveret, R., Palmirani, M. and Sartor, G. Variants of temporal defeasible logics for modelling norm modifications. In *ICAAIL '07: Proceedings of the 11th international conference on Artificial intelligence and law*, pages 155–159, New York, NY, USA, 2007. ACM.
- [42] Grandi, F., Mandreoli, F. and Tiberio, P. Temporal modelling and management of normative documents in xml format. *Data Knowl. Eng.*, 54(3):327–354, 2005.
- [43] Grishman, R. Information extraction: Techniques and challenges. In *SCIE '97: International Summer School on Information Extraction*, pages 10–27, London, UK, 1997. Springer-Verlag.
- [44] Hachey, B. and Grover, C. Extractive summarisation of legal texts. *Artif. Intell. Law*, 14(4):305–345, 2006.
- [45] Jensen, C. S. and Snodgrass, R. T. Temporal data management. In *IEEE Transactions on Knowledge and Data Engineering*, volume 11, pages 36–44, January/February 1999.
- [46] Kelsen, H. *Teoria Pura do Direito*. Martins Fontes, São Paulo, third edition, 1999. Tradução: João Baptista Machado.

- [47] Kock, N. LexDania: An XML Meta-schema for Legislative Documents. In *Proceeding of the 2003 International Workshop on the Development of Standards for Describing Legal Documents*, Utrecht, The Netherlands, December 2003.
- [48] Lauritsen, M. and Gordon, T. F. Toward a general theory of document modeling. In *ICAIL '09: Proceedings of the 12th International Conference on Artificial Intelligence and Law*, pages 202–211, New York, NY, USA, 2009. ACM.
- [49] Laxman, S. and Sastry, P. A survey of temporal data mining. *Sadhana*, 31(2):173–198, April 2006.
- [50] Projeto LexML Brasil. Apresentação. Disponível em <http://projeto.lexml.gov.br/documentacao/Apresentacao.pdf>, Dezembro 2008. Acesso em Junho 2010.
- [51] ----- Modelo de Referência. Disponível em <http://projeto.lexml.gov.br/documentacao/Parte-1-Modelo-de-Referencia.pdf>, Dezembro 2008. Acesso em Junho 2010.
- [52] ----- LexML URN. Disponível em <http://projeto.lexml.gov.br/documentacao/Parte-2-LexML-URN.pdf>, Dezembro 2008. Acesso em Junho 2010.
- [53] ----- XML Schema. Disponível em <http://projeto.lexml.gov.br/documentacao/Parte-3-XML-Schema.pdf>, Dezembro 2008. Acesso em Junho 2010.
- [54] Lima, J. A. de O. *Modelo Genérico de Relacionamentos na Organização da Informação Legislativa e Jurídica*. PhD thesis, Doutorado em Ciência da Informação, Universidade de Brasília, Brasília, Março 2008.
- [55] Logghe, M., Van de Kerchove, K. and Moens, M. F. Automatic Version Management of Legislation: The Agora-Lex Project. In *DEXA '00: Proceedings of the 11th International Workshop on Database and Expert Systems Applications*, page 1051, Washington, DC, USA, 2000. IEEE Computer Society.
- [56] Lupo, C. Standard for legal documents : the Italian experience within the Normeinrete project. In *Proceeding of the 2003 International Workshop on the Development*

- of Standards for Describing Legal Documents*, Utrecht, The Netherlands, December 2003.
- [57] Martinek, J. and Cybulka, J. Dynamics of legal provisions and its representation. In *ICAAIL '05 Proceedings of the 10th international conference on Artificial intelligence and law*, pages 20–24, New York, NY, USA, 2005. ACM.
- [58] Maximiliano, C. *Hermenêutica e Aplicação do Direito*. Forense, Rio de Janeiro, 2006.
- [59] Mencía, E. L. Segmentation of legal documents. In *ICAAIL '09: Proceedings of the 12th International Conference on Artificial Intelligence and Law*, pages 88–97, New York, NY, USA, 2009. ACM.
- [60] Mendes, G. and Júnior, N. J. F. *Manual de Redação da Presidência da República*. Presidência da República, Brasília, 2002.
- [61] Moens, M.F. and Logghe, M. Digital legislation: reflections on the Agora-Lex project. In *Proceedings of Business Information Systems*, pages 228–233, Poznan, Poland, 2002. The Poznan University of Economics.
- [62] Palmirani, M., Brighi, R. and Massini, M. Automated extraction of normative references in legal texts. In *ICAAIL '03: Proceedings of the 9th international conference on Artificial intelligence and law*, pages 105–106, New York, NY, USA, 2003. ACM.
- [63] Schweizerischer Verein Für Rechtsinformatik. CHLexML - Data Standard for the Representation of Swiss Law Texts, 2008. Technical XML Reference, CHLexML Schema 1.0.
- [64] Riveret, R. *Interactions between normative systems and software cognitive agents. A formalization in temporal modal defeasible logic and its implementation*. PhD thesis, Università Degli Studi Di Bologna, Bologna, Italy, December 2008.
- [65] Ráo, V. *O Direito e a vida dos direitos*. Revista dos Tribunais, São Paulo, 2004.
- [66] Schilder, F. and McCulloh, A. Temporal information extraction from legal documents. In Graham Katz, James Pustejovsky, and Frank Schilder, editors, *Annotating, Extracting and Reasoning about Time and Events*, number 05151 in Dagstuhl Seminar Procee-

- dings, Dagstuhl, Germany, 2005. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.
- [67] Siena, A., Ghanavati, S., Perini, A., Amyot, D., Peyton, L. and Susi, A. A legal perspective on business: Modeling the impact of law. In Gilbert Babin, Peter Kropf, and Michael Weiss, editors, *E-Technologies: Innovation in an Open World*, volume 26, chapter 23, pages 267–278. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [68] Siena, A., Mylopoulos, J., Perini, A. and Susi, A. Designing law-compliant software requirements. In *Conceptual Modeling - ER 2009*, volume 5829, chapter 35, pages 472–486. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [69] Siena, A., Mylopoulos, J., Perini, A. and Susi, A. Towards a framework for law-compliant software requirements. In *Software Engineering - Companion Volume, 2009. ICSE-Companion 2009. 31st International Conference on*, pages 251–254, May 2009.
- [70] Sutton, S. A. The role of attorney mental models of law in case relevance determinations: an exploratory analysis. *J. Am. Soc. Inf. Sci.*, 45(3):186–200, 1994.
- [71] Tansel, A. U., Clifford, J., Gadia, S., Jajodia, S., Segev, A. and Snodgrass, R., editor. *Temporal databases: theory, design, and implementation*. Benjamin-Cummings Publishing Co., Inc., Redwood City, CA, USA, 1993.
- [72] Tansel, A. U., Clifford, J., Gadia, S., Jajodia, S., Segev, A. and Snodgrass, R., editor. *Temporal databases: theory, design, and implementation*. Benjamin-Cummings Publishing Co., Inc., Redwood City, CA, USA, 1993.
- [73] van Engers, T., van Gog, R. and Sayah, K. A case study on automated norm extraction. In *Proceeding of the 2004 conference on Legal Knowledge and Information Systems*, pages 49–58, Amsterdam, The Netherlands, 2004. IOS Press.
- [74] Vanzin, M., Abel, M. and Heuser, C. A. Extração e representação de estruturas de documentos jurídicos. In *Simpósio de Tecnologias de Documentos (STD 2002)*, volume 1, pages 92–99, 2002.

- [75] Wyner, A. Z. Approaches to text mining arguments from legal cases. Lecture Notes in Computer Science, 2010. Series Editors: Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J., Kobsa, A., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O.M., Pandu Rangan, C., Steffen, B., Madhu, S., Terzopoulos, D., Tygar, J.D., Weikum, G.
- [76] Zhang, P. and Koppaka, L. Semantics-based legal citation network. In *ICAIL '07: Proceedings of the 11th international conference on Artificial intelligence and law*, pages 123–130, New York, NY, USA, 2007. ACM.

Apêndice A

Fontes do Estudo de Caso

- *Códigos*: Comercial (Lei nº 556, 25.6.1850), Civil (Lei nº 10.406, 10.1.2002), Penal (Del nº 2.848, 7.12.1940), Processo Penal (Del nº 3.689, 3.10.1941), Eleitoral (Lei nº 4.737, 15.7.1965) e Processo Civil (Del nº 3.689, 3.10.1941);
- *Constituição da República Federativa do Brasil de 1988*;
- *Decretos*: 7.020 de 27.11.2009, 7.019 de 27.11.2009, 7.018 de 27.11.2009, 7.017 de 26.11.2009, 7.016 de 26.11.2009, 7.015 de 24.11.2009, 7.014 de 23.11.2009, 7.013 de 19.11.2009, 7.012 de 19.11.2009 e 7.011 de 18.11.2009;
- *Leis Complementares*: 132 de 7.10.2009, 131 de 27.5.2009, 130 de 17.4.2009, 129 de 8.1.2009, 128 de 19.12.2008, 127 de 14.8.2007, 126 de 15.1.2007, 125 de 3.1.2007, 124 de 3.1.2007 e 123 de 14.12.2006;
- *Leis Delegadas*: 13 de 27.8.92 e 12 de 7.8.92;
- *Leis Ordinárias*: 12.101 de 27.11.2009, 12.100 de 27.11.2009, 12.099 de 27.11.2009, 12.098 de 24.11.2009, 12.097 de 24.11.2009, 12.096 de 24.11.2009, 12.095 de 19.11.2009, 12.094 de 19.11.2009, 12.093 de 16.11.2009 e 12.092 de 16.11.2009, 12.091 de 11.11.2009;
- *Medidas Provisórias*: 471 de 20.11.2009, 470 de 13.10.2009, 469 de 5.10.2009, 468 de 31.8.2009, 467 de 30.7.2009, 466 de 29.7.2009, 465 de 29.6.2009, 464 de 9.6.2009, 463 de 20.5.2009 e 462 de 14.5.2009;

- *Emendas Constitucionais*: 61 de 11.11.2009, 60 de 11.11.2009, 59 de 11.11.2009, 58 de 23.9.2009, 57 de 18.12.2008, 56 de 20.12.2007, 55 de 20.9.2007, 54 de 20.9.2007, 53 de 19.12.2006, 52 de 8.3.2006.
- *Acórdãos do STF*: ADPF 96 AgR/DF, Rcl 3094 AgR/RN, Rcl 3456 AgR/PI, Inq 2815 ED-AgR/DF, MS 28401 AgR/RJ, Pet 4673 AgR/SP, HD 87 AgR/DF, RE 584100/SP, Inq 2567 AgR/SP, AI 460098 ED/PE, AI 503093 AgR/SP, AI 699770 AgR/RS, HC 94888/SP, HC 96307/GO, HC 99671/DF, RE 267599 AgR/MG, ACO 11912 AgR/RO, RE 572205 AgR-AgR/RS, AI 552318 ED/SP e RHC 95108/ES.
- *Acórdãos do STJ*: REsp 1120616/PR, EREsp 501248/RS, EDcl no AgRg no CC 98613/RS, EDcl no AgRg nos EREsp 209320/DF, EREsp 835537/MG, AgRg nos EDcl nos EREsp 875823/MG, AgRg nos EREsp 944984/RS, EDcl nos EDcl no AgRg no Ag 956500/SP, AgRg no REsp 1120790/PR, AgRg no REsp 1135296/RS, AgRg no REsp 1109302/RS, AgRg no RMS 30390/SP, AgRg no Ag 1191569/RJ, REsp 1157286/RS, RMS 20987/AM, REsp 508377/MS, AgRg no Ag 928652/RS, EREsp 649809/SP, AgRg no REsp 976171/MG e CC 73028/MA.

Apêndice B

Aplicação do Modelo ITMJudLaw

Aplicando o *ITMJudLaw* ao RECURSO ESPECIAL Nº 698.900 - SP (2004/0153389-4), proveniente do Superior Tribunal de Justiça:

DocumentoJuridico

- *IdentificadorUnico*: REsp698900SP;
- *dataJulgamento*: 01/09/2009;
- *dataPublicação*: 08/09/2009;

Cabeçalho

- *tribunal*: Superior Tribunal de Justiça;
- *tipoDocJud*: Recurso Especial;
- *numero*: 698900;
- *local*: SP;
- *codRegistro*: 2004/0153389-4;

Parte

- *titulo*: RELATOR;
- *nome*: MINISTRO HERMAN BENJAMIN;
- *titulo*: RECORRENTE;
- *nome*: ESTADO DE SÃO PAULO;

- *titulo:* PROCURADOR;
- *nome:* FATIMA FERNANDES CATELLANI E OUTRO(S);
- *titulo:* RECORRIDO;
- *nome:* ANIBAL ZACHARIAS E CÔNJUGE;
- *titulo:* ADVOGADO;
- *nome:* JOSÉ ALCIDES DE QUEIROZ ALVES;

Ementa: PROCESSUAL CIVIL. SUPRESSÃO DE INSTÂNCIA. OMISSÃO. ART. 535 DO CPC. OFENSA. 1. Hipótese em que o juiz de origem extinguiu a Ação de Indenização por Desapropriação Indireta por falta de interesse de agir. 2. O Tribunal de origem reformou a decisão e determinou o retorno dos autos para que o magistrado analise o mérito da demanda. (...)

Relatório: O EXMO. SR. MINISTRO HERMAN BENJAMIN (Relator): Trata-se de Recurso Especial interposto, com fundamento no art. 105, III, “a” e “b”, da Constituição da República, contra acórdão assim ementado (fl. 408): (...)

Voto: O EXMO. SR. MINISTRO HERMAN BENJAMIN (Relator): Os particulares propuseram Ação de Indenização por Desapropriação Indireta, relativa a imóvel de 206.000 m², localizado em Ilhabela-SP, por conta da criação de Parque Estadual e de tombamento da área (fl. 22). (...)

Encerramento

- *decisão:* Certifico que a egrégia SEGUNDA TURMA, ao apreciar o processo em epígrafe na sessão realizada nesta data, proferiu a seguinte decisão: “A Turma, por unanimidade, deu provimento ao recurso, nos termos do voto do(a) Sr(a). Ministro(a)-Relator(a).” Os Srs. Ministros Mauro Campbell Marques, Eliana Calmon, Castro Meira e Humberto Martins votaram com o Sr. Ministro Relator.;
- *acórdão:* Vistos, relatados e discutidos os autos em que são partes as acima indicadas, acordam os Ministros da Segunda Turma do Superior Tribunal de Justiça: “A Turma, por unanimidade, deu provimento ao recurso,

nos termos do voto do(a) Sr(a). Ministro(a)-Relator(a).” Os Srs. Ministros Mauro Campbell Marques, Eliana Calmon, Castro Meira e Humberto Martins votaram com o Sr. Ministro Relator. Brasília, 1º de setembro de 2009(data do julgamento).;

- *orgaoJulgador*: SEGUNDA TURMA;

Aplicando agora o *ITMJudLaw* à LEI Nº 12.244 DE 24 DE MAIO DE 2010:

Norma

- *dataPublicação*: 14/05/2010;
- *identificadorUnico*: br:federal:lei;2010-05-14;12244;
- *validade*: [14/05/2010-*];
- *versao*: 1;
- *vigência*: [14/05/2010-*];
- *apelido*: null;
- *ementa*: Dispõe sobre a universalização das bibliotecas nas instituições de ensino do País.;
- *epígrafe*: LEI Nº 12.244 DE 24 DE MAIO DE 2010;
- *esfera*: federal;
- *local*: Brasília;
- *numero*: 12244;
- *preâmbulo*: O PRESIDENTE DA REPÚBLICA Faço saber que o Congresso Nacional decreta e eu sanciono a seguinte Lei.;
- *tipo*: Lei_ordinaria;
- *elementosNorma*: art1_br:federal:lei;2010-05-14;12244,
art2_br:federal:lei;2010-05-14;12244, art3_br:federal:lei;2010-05-14;12244, art4_br:federal:lei;2010-05-14;12244.

Os elementos que compõem a norma:

ElementoNorma

- *dataPublicação*: 14/05/2010;
- *identificadorUnico*: art1_br:federal:lei;2010-05-14;12244;
- *validade*: [14/05/2010-*];
- *versao*: 1;
- *vigência*: [14/05/2010-*];
- *tipo*: Artigo
- *texto*: As instituições de ensino públicas e privadas de todos os sistemas de ensino do País contarão com bibliotecas, nos termos desta Lei.

ElementoNorma

- *dataPublicação*: 14/05/2010;
- *identificadorUnico*: art2_br:federal:lei;2010-05-14;12244;
- *validade*: [14/05/2010-*];
- *versao*: 1;
- *vigência*: [14/05/2010-*];
- *tipo*: Artigo;
- *texto*: Para os fins desta Lei, considera-se biblioteca escolar a coleção de livros, materiais videográficos e documentos registrados em qualquer suporte destinados a consulta, pesquisa, estudo ou leitura.;
- *elementosNorma*: parunico_art2_br:federal:lei;2010-05-14;12244.

ElementoNorma

- *dataPublicação*: 14/05/2010;
- *identificadorUnico*: parunico_art2_br:federal:lei;2010-05-14;12244;
- *validade*: [14/05/2010-*];
- *versao*: 1;

- *vigência*: [14/05/2010-*];
- *tipo*: Parágrafo
- *texto*: Será obrigatório um acervo de livros na biblioteca de, no mínimo, um título para cada aluno matriculado, cabendo ao respectivo sistema de ensino determinar a ampliação deste acervo conforme sua realidade (...)

ElementoNorma

- *dataPublicação*: 14/05/2010;
- *identificadorUnico*: art3_br:federal:lei;2010-05-14;12244;
- *validade*: [14/05/2010-*];
- *versao*: 1;
- *vigência*: [14/05/2010-*];
- *tipo*: Artigo
- *texto*: Os sistemas de ensino do País deverão desenvolver esforços progressivos para que a universalização das bibliotecas escolares, nos termos previstos nesta Lei, seja efetivada num prazo máximo de dez anos, respeitada a profissão de Bibliotecário, disciplinada pelas Leis nos 4.084, de 30 de junho de 1962, e 9.674, de 25 de junho de 1998.;
- *citacoesFeitas*: br:federal:lei;1962-06-30;4084, br:federal:lei;1998-06-25;9674.

ElementoNorma

- *dataPublicação*: 14/05/2010;
- *identificadorUnico*: art4_br:federal:lei;2010-05-14;12244;
- *validade*: [14/05/2010-*];
- *versao*: 1;
- *vigência*: [14/05/2010-*];
- *tipo*: Artigo
- *texto*: Esta Lei entra em vigor na data de sua publicação.

Apêndice C

Descrição Formal Completa das Fontes

norma = cabeçalho , epígrafe , ementa , preâmbulo , {elemento_norma}- , local_data ;

cabeçalho = “Presidência da República”, “Casa Civil” , “Subchefia para Assuntos Jurídicos” ;

epígrafe = tipo_dispositivo_legal , “N^o” , *número_norma* , “,” , data_norma_epígrafe , [“.”] ;

tipo_dispositivo_legal = tipo_lei | “DECRETO” | “EMENDA CONSTITUCIONAL” | “MEDIDA PROVISÓRIA” ;

tipo_lei = “LEI” | “LEI” , tipo_lei_específico ;

tipo_lei_específico = “COMPLEMENTAR” | “DELEGADA” ;

data_norma_epígrafe = “DE” , dia_data , “DE” , mês_data_extenso_maiúsculo , “DE” , ano_data ;

mês_data_extenso_maiúsculo = “JANEIRO” | “FEVEREIRO” | “MARÇO” | “ABRIL” | “MAIO” | “JUNHO” | “JULHO” | “AGOSTO” | “SETEMBRO” | “OUTUBRO” | “NOVEMBRO” | “DEZEMBRO” ;

ementa = texto_ref ;

preâmbulo = autoridade , texto_ref ;

autoridade = “O PRESIDENTE DA REPÚBLICA” | “O VICE-PRESIDENTE DA REPÚBLICA” | “As Mesas da Câmara dos Deputados e do Senado Federal” ;

elemento_norma = id_elemento_norma , texto_ref , {referência_inversa_TL} , {elemento_norma} ;

id_elemento_norma = id_parte | id_livro | id_título | id_capítulo | id_seção | id_subseção | id_artigo | id_parágrafo | id_inciso | id_alínea | id_item ;

id_parte = “PARTE” , id_parte_específico ;
id_parte_específico = “GERAL” | “ESPECIAL” | numeração_parte ;
id_livro = “LIVRO” , numeração_romana ;
id_título = “TÍTULO” , numeração_romana ;
id_capítulo = “CAPÍTULO” , numeração_romana ;
id_seção = “SEÇÃO” , numeração_romana ;
id_subseção = “SUBSEÇÃO” , numeração_romana ;
id_artigo = “Art.” , numeração_art_par , [“-” | “.”] ;
id_parágrafo = (id_parágrafo_único | id_parágrafo_comum) , “-” ;
id_parágrafo_único = “Parágrafo único” ;
id_parágrafo_comum = “§” , numeração_art_par ;
id_inciso = numeração_romana , “-” ;
id_alínea = numeração_alínea , “)” ;
id_item = numeração_arábica , “.” ;
texto_ref = texto , {referência_direta_TL} ;
referência_TL = referência_direta_TL | referência_inversa_TL ;
referência_inversa_TL = “(” , termo_ref_inversa , (“pelo” | “pela”) ,
tipo_dispositivo_legal , “no” , *número_norma* , “,” , [“de”] , data_referência_numeral ,
“)” ;
termo_ref_inversa = termo_inclusão | termo_modificação | termo_revogação ;
termo_inclusão = “Incluído” | “Incluídos” ;
termo_modificação = “Vide” | “Redação dada” ;
termo_revogação = “Revogado” | “Revogados” ;
data_referência_numeral = data_referência_numeral_completa | ano_data ;
data_referência_numeral_completa = dia_data , separador_data , mês_data_numeral ,
separador_data , ano_data ;
separador_data = “.” | “/” ;
referência_direta_TL = referência_direta_TL_interna | referência_direta_TL_externa ;
referência_direta_TL_interna = referência_direta_TL_interna_simples | re-
ferência_direta_TL_interna_composta ;

referência_direta_TL_interna_simples = {id_elemento_norma_ref_simples ,
{separador_referência_simples}}- ;

id_elemento_norma_ref_simples = id_parte | id_livro | id_título | id_capítulo | id_seção
| id_subseção | id_artigo_ref_simples | id_parágrafo_ref_simples | id_inciso_ref_simples |
id_alínea_ref_simples | id_item_ref_simples ;

id_artigo_ref_simples = (“art.” | “artigo”) , numeração_art_par , [“,” , “caput”] ;

id_parágrafo_ref_simples = (id_parágrafo_único | id_parágrafo_comum_ref) ;

id_parágrafo_comum_ref_simples = (“§” | “parágrafo”) , numeração_art_par ;

id_inciso_ref_simples = [“inciso” | “inc.”] , numeração_romana ;

id_alínea_ref_simples = [“alínea”] , ‘ ‘ ‘ , numeração_alínea , ‘ ’ ’ ;

id_item_ref_simples = “item” , numeração_arábica ;

separador_referência_simples = “de” | “da” | “do” | “,” ;

referência_direta_TL_interna_composta = {id_elemento_norma_ref_composta ,
{separador_referência_simples , referência_direta_TL_interna_simples}}- ;

id_elemento_norma_ref_composta = id_parte_ref_composta | id_livro_ref_composta
| id_título_ref_composta | id_capítulo_ref_composta | id_seção_ref_composta |
id_subseção_ref_composta | id_artigo_ref_composta | id_parágrafo_ref_composta |
id_inciso_ref_composta | id_alínea_ref_composta | id_item_ref_composta ;

id_parte_ref_composta = “Partes” , id_parte_específico_ref ;

id_parte_específico_ref = (“GERAL” | “ESPECIAL” | numeração_parte) ,
{separador_referência_composta , id_parte_específico_ref}- ;

id_livro_ref_composta = “livros” , numeração_romana , {separador_referência_composta
, numeração_romana}- ;

id_título_ref_composta = “títulos” , numeração_romana ,
{separador_referência_composta , numeração_romana}- ;

id_capítulo_ref_composta = “capítulos” , numeração_romana ,
{separador_referência_composta , numeração_romana}- ;

id_seção_ref_composta = “seções” , numeração_romana ,
{separador_referência_composta , numeração_romana}- ;

id_subseção_ref_composta = “subseções” , numeração_romana ,
{separador_referência_composta , numeração_romana}- ;

id_artigo_ref_composta = (“arts.” | “artigos”) , numeração_art_par ,
{separador_referência_composta , numeração_art_par}- ;

id_parágrafo_ref_composta = (“parágrafos” | {“§”}-) , numeração_art_par ,
{separador_referência_composta , numeração_art_par}- ;

id_inciso_ref_composta = [“incisos”] , numeração_romana ,
{separador_referência_composta , numeração_romana}- ;

id_alínea_ref_composta = [“alíneas”] , ‘ “ ’ , numeração_alínea , ‘ ” ’ ,
{separador_referência_composta , ‘ “ ’ , numeração_alínea , ‘ ” ’}- ;

id_item_ref_composta = “itens” , numeração_arábica , {separador_referência ,
numeração_arábica}- ;

separador_referência_composta = “e” , “;” ;

referência_direta_TL_externa = referência_direta_TL_interna ,
{separador_referência_simples}- , tipo_dispositivo_legal_ref , [“nº”] , *número_norma* ,
separador_referência_simples , data_referência] ;

tipo_dispositivo_legal_ref = tipo_lei_ref | “Decreto” | “Decreto-lei” | “Emenda Constitu-
cional” | (“Medida Provisória” | “MP”) | códigos ;

tipo_lei_ref = “Lei” | “Lei” , tipo_lei_específico | “LCP” ;

tipo_lei_específico = “Complementar” | “Delegada” ;

códigos = (“Código Penal” | “CP”) | (“Código de Processo Penal” | “CPP”) |
(“Constituição” | “Constituição Federal” | “CF”) ;

data_referência = data_referência_completa | ano_data ;

data_referência_completa = “de” , dia_data , “de” , mês_data_extenso , “de” , ano_data ;

local_data = “Brasília” , “;” , [“em”] , dia_data , “de” , mês_data_extenso , “de” , ano_data
 , “;” ;

mês_data_extenso = “Janeiro” | “Fevereiro” | “Março” | “Abril” | “Maio” | “Junho” |
“Julho” | “Agosto” | “Setembro” | “Outubro” | “Novembro” | “Dezembro” ;

documento_jurídico_STF = cabeçalho1 , cabeçalho2 , {parte}- , ementa , acórdão ,
relatório , {voto}- , extrato_da_ata ;

cabeçalho1 = “Superior Tribunal Federal” , “Coordenadoria de Análise de Juris-
prudência” , “DJe nº” , número_dje , “Divulgação” , data_divulgação , “Publicação” ,
data_publicação , “Ementário nº” , número_ementário ;

cabeçalho2 = *data_julgamento* , *órgão_julgador* , *cabeçalho2_docjudinfo* ;
cabeçalho2_docjudinfo = *tipo_dj* , *número_dj* , *local* ;
parte = *título* , “:” , *nome* ;
ementa = “EMENTA:” , *texto_ref_dj* ;
acórdão = “A C Ó R D Ã O” , *texto* ;
relatório = “R E L A T Ó R I O” , *texto_ref_dj* , [“É o relatório.”] ;
voto = “V O T O” , *texto_ref_dj* , [“É como voto.”] ;
texto_ref_dj = *texto* , {*referência_DJ*} ;
extrato_da_ata = “EXTRATO DA ATA” , *tipo_dj* , *número_dj* , “PROCED.:” , *local* ,
{*parte*}- , “Decisão:” , *decisão* ;
documento_jurídico_STJ = “Superior Tribunal de Justiça” , *cabeçalho* , {*parte*}- ,
ementa , *acórdão* , *relatório* , {*voto*}- , *certidão_de_julgamento* ;
cabeçalho = *tipo_dj* , “N^o” , *número_dj* , “-” , *sigla_local* , “(” , *cod_registro_dj* , “)” ;
cod_registro_dj = *ano_data* , “/” , *código* ;
ementa = “EMENTA” , *texto_ref_dj* ;
acórdão = “ACÓRDÃO” , *texto* , *local_acordão_stj* , *data_julgamento_extenso* ;
local_acordão_stj = *local* | *local* , “(” , *sigla_local* , “)” | *local* , “/” , *sigla_local* ;
data_julgamento_extenso = *dia_data* , “,” , “de” , *mês_data_extenso* , “de” , *ano_data* ;
relatório = “RELATÓRIO” , *texto_ref_dj* , [“É o relatório.”] ;
voto = “VOTO” , *texto_ref_dj* , [“É como voto.”] ;
texto_ref_dj = *texto* , {*referência_DJ*} ;
certidão_de_julgamento = “CERTIDÃO DE JULGAMENTO” , *órgão_julgador* ,
“Número Registro:” , *cod_registro_dj* , *sigla_tipo_dj* , *número_dj* , “/” , *sigla_local* , “JUL-
GADO:” , *data_julgamento_numeral* , “CERTIDÃO” , *decisão* ;
data_julgamento_numeral = *dia_data* , “/” , *mês_data_numeral* , “/” , *ano_data* ;
referência_DJ = *referência_DJ_DJ* | *referência_DJ_TL* | *referência_DJ_DO* ;
referência_DJ_DJ = [*tribunal_abrev*] , *tipo_DJ_abrev* , [*exp_num*] , *número_dj* , [-*AgR*] ,
“/” , *sigla_estado* ;
tribunal_abrev = “STF” | “STJ” ;
sigla_estado = “AC” | “AL” | “AP” | “AM” | “BA” | “CE” | “DF” | “ES” | “GO” | “MA”
| “MT” | “MS” | “MG” | “PA” | “PB” | “PR” | “PE” | “PI” | “RJ” | “RN” | “RS” | “RO” |

“RR” | “SP” | “SC” | “SE” | “TO” | sigla_país;

referência_DJ_TL = (referência_direta_TL_interna_simples | referência_DJ_TL_composta) , referência_DJ_TL_final_norma_data | referência_DJ_TL_final_códigos ;

referência_DJ_TL_final_norma_data = {separador_referência_simples}- , tipo_dispositivo_legal_ref , exp_num , número_norma , {separador_referência_simples_DJ}- , data_referência_DJ ;

referência_DJ_TL_final_códigos = “;” , [“todos”] , “do” , códigos ;

referência_DJ_TL_composta = referência_direta_TL_interna_composta | referência_DJ_TL_composta_especializada ;

referência_DJ_TL_composta_especializada = {id_elemento_norma_ref_simples | id_elemento_norma_ref_composta , [referência_DJ_TL_final_norma_data] , {separador_referência_composta_DJ}-}- , id_elemento_norma_ref_simples | id_elemento_norma_ref_composta , referência_DJ_TL_final_norma_data | referência_DJ_TL_final_códigos ;

separador_referência_composta_DJ = separador_referência_simples | “e” | “c/c” | “no” | “na” ;

exp_num = “no” | “n.” | “n.º” ;

separador_referência_simples_DJ = separador_referência_simples | “/” ;

data_referência_DJ = data_referência_completa_DJ | ano_data ;

data_referência_completa_DJ = “de” , dia_data , separador_data , mês_data_numeral , separador_data , ano_data ;

referência_DJ_DO = “(” , nome_obra , “;” , edição , editora , local , ano , página , “)” ;