

CARLOS ANDRÉ ZAVADINACK

**RACIOCÍNIO BASEADO EM CASOS COMO
RECOMENDADOR DE CONTEÚDO PEDAGÓGICO**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. Fabiano Silva

CURITIBA

2016

CARLOS ANDRÉ ZAVADINACK

**RACIOCÍNIO BASEADO EM CASOS COMO
RECOMENDADOR DE CONTEÚDO PEDAGÓGICO**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. Fabiano Silva

CURITIBA

2016

V39

Zavadinack, Carlos Andre

Raciocínio baseado em casos como recomendador de conteúdo pedagógico. / Carlos Andre Zavadinack. – Curitiba, 2016.
67: il. [algumas color.]; tabs. : color. : 30 cm.

Dissertação (mestrado) - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática.

Orientador: Prof. Dr. Fabiano Silva.

Bibliografia: p.63-67.

1. Raciocínio. 2. Educação. 3. Tecnologia educacional. I. Universidade Federal do Paraná. II. Silva, Fabiano. III. Título.
CDD 371.334



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
Setor CIÊNCIAS EXATAS
Programa de Pós Graduação em INFORMÁTICA
Código CAPES: 40001016034P5

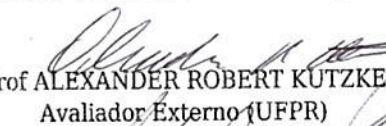
TERMO DE APROVAÇÃO

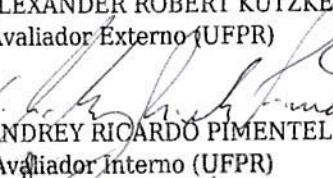
Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **CARLOS ANDRE ZAVADINACK**, intitulada: "**RACIOCÍNIO BASEADO EM CASOS COMO RECOMENDADOR DE CONTEÚDO PEDAGÓGICO**", após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua APROVAÇÃO.

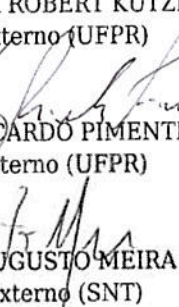
Curitiba, 31 de Agosto de 2016.


Prof FABIANO SILVA

Presidente da Banca Examinadora (UFPR)


Prof ALEXANDER ROBERT KUTZKE
Avaliador Externo (UFPR)


Prof ANDREY RICARDO PIMENTEL
Avaliador Interno (UFPR)


Prof JORGE AUGUSTO MEIRA
Avaliador Externo (SNT)



AGRADECIMENTOS

Sou muito grato ao meu orientador e amigo Prof.Dr. Fabiano Silva por ter compartilhado a sabedoria em muitas conversas formais e informais onde cresci em todos os sentidos durante essa fase da minha vida. Pela confiança em meu trabalho e nos meus questionamentos. Muito obrigado, Prof.Fabiano.

Agradeço aos Profs. Drs. Alexander Kutzke, Alexandre Ibrahim Direne do Departamento de Informática pelo apoio na pesquisa. Kutzke esteve sempre à disposição para ajudar durante a pesquisa e elaboração deste documento. Inspiração e conhecimento obtido em boas conversas com o Direne. Um forte abraço, Prof. Direne.

Um agradecimento aos mestres Ricardo T. de Oliveira e Guilherme Derinievicz. Com o apoio desses dois excelentes cientistas da computação pude aprimorar minhas competências em Programação, Lógica e Matemática Discreta. Valeu, amigos.

Quero agradecer ao Centro de Computação Científica e Software Livre (C3SL) pelo apoio. Grupo de pesquisa com o qual pude contar para o desenvolvimento de meus experimentos. Foi uma honra, pessoal.

Agradecer imensamente pelo incentivo e amizade de meus amigos de mestrado: Elisabete, Edgar, Renato, Razer, e Clariane. Foi um grande prazer, amigos.

Agradeço às três cadelas lá de casa, Milka, Sascha e Luna por me lembrarem que a vida é mais leve do que parece.

Quero agradecer à minha família. Minha irmã Maria Fernanda e aos meus pais Carlos e Eunice. Vocês criaram um filho curioso e por isso estou aqui, muito alegre, escrevendo essas páginas.

Gratidão imensa por ter do meu lado a minha esposa Mariana, por todo apoio para a conclusão da pesquisa e também deste documento. Também muito grato por meu filho que compreendeu o momento que o pai estava passando e aproveitou cada minuto que pudemos compartilhar entre um experimento e outro. Vocês são o meu maior incentivo.

SUMÁRIO

LISTA DE FIGURAS	v
LISTA DE TABELAS	vi
RESUMO	vii
ABSTRACT	viii
1 INTRODUÇÃO	1
1.1 Objetivos	5
1.1.1 Objetivo Geral	5
1.1.2 Objetivos Específicos	5
1.2 Estrutura do Documento	5
2 RACIOCÍNIO BASEADO EM CASOS	6
2.1 Raciocínio Baseado em Casos	6
2.2 Sistemas e Frameworks genéricos para CBR	14
2.2.1 jColibri	14
2.2.2 Tuuurbine	15
2.2.3 myCBR	16
2.3 O CBR nos Sistemas de apoio à aprendizagem	17
2.3.1 Sistemas de Apoio ao Ensino	17
2.3.2 O CBR nos Sistemas de apoio à aprendizagem	18
2.4 Considerações	20
3 REPRESENTAÇÃO DE DADOS	21
3.1 Linked Data	21
3.2 Ontologias	24
3.2.1 Ontologias OWL	27

3.3	RDF	28
3.3.1	SPARQL	31
3.4	Considerações	31
4	LEVANTAMENTO DE DADOS	33
4.1	FARMA-ALG	33
4.1.1	Mediação do Erro	33
4.1.2	FARMA-ALG: Arcabouço de Sistema para Auxílio à Mediação do Erro	35
4.2	Levantamento de dados	36
4.2.1	Ontologia com base nos erros	39
4.3	Considerações	43
5	SISTEMA CBR COMO RECOMENDADOR DE CONTEÚDO PEDAGÓGICO	44
5.1	Representação de Conhecimento em CBR	44
5.1.1	Vocabulário	45
5.1.2	Métrica de Similaridade	45
5.1.3	Base de Casos	47
5.1.4	Adaptação	47
5.2	Representação de conhecimento no myCBR 3	48
5.3	Descrição do experimento	51
5.3.1	Vocabulário dos exercícios	51
5.3.2	Similaridade entre exercícios	51
5.3.3	Base de casos com lista de exercícios	54
5.3.4	Consultas (<i>Queries</i>)	55
5.4	Execução e resultados dos experimentos	55
5.5	Considerações Finais	59
6	CONCLUSÃO	60
6.1	Contribuições	61

	iv
6.2 Dificuldades	61
6.3 Trabalhos futuros	62
BIBLIOGRAFIA	67

LISTA DE FIGURAS

2.1	Ciclo CBR Aamodt e Plaza (1994) [1]	9
2.2	Tecnologias e o Ciclo CBR [2]	12
3.1	Exemplo de Ontologia utilizando a coleção do Timóteo	26
3.2	Exemplo de um Grafo $G = (V, A)$	29
3.3	Um grafo RDF com dois nós [3]	29
3.4	Exemplo de um grafo RDF (Coleção do Timóteo)	30
4.1	Ontologia de Objetos Educacionais	38
4.2	Ontologia das Taxonomias	39
4.3	Função do código que constrói um dicionário de exercícios e a quantidade de respostas erradas.	40
4.4	Função que retorna dicionário contendo exercícios e a quantidade de respostas erradas abaixo de uma média dada como parâmetro	41
4.5	Ontologia dos exercícios divididos de acordo com os erros	42
4.6	Uma amostra da representação computacional (XML/RDF) da Ontologia construída.	42
5.1	Interface do myCBR 3.1 Workbench	49
5.2	Amostra da Classe Alta da Ontologia	53
5.3	Parte da tabela da função de similaridade para o myCBR	54
5.4	Exemplo de uma consulta do <i>Grupo 1</i> feita no myCBR.	57
5.5	Exemplo de uma recuperação do <i>Grupo 4</i> feita no myCBR.	58

LISTA DE TABELAS

5.1	Um exemplo de um caso de lista de exercícios adicionado na base de casos testada no myCBR 3.	55
5.2	Tabela de consultas ordenadas quanto à similaridade.	57
5.3	Uma das consultas do <i>Grupo 5</i>	58
5.4	Dois resultados de uma consulta do <i>Grupo 5</i>	58

RESUMO

Raciocínio baseado em Casos (CBR) é um método para resolver problemas com recuperação de soluções para problemas anteriores. CBR demanda uma representação de conhecimento que permite o resolvidor a buscar casos similares através de uma pergunta e a taxa de similaridade é dado pela distância em uma estrutura de árvore, uma ontologia. O objetivo da presente pesquisa é utilizar CBR como uma ferramenta pedagógica através de quatro pilares: Raciocínio Baseado em Casos, Representação de conhecimento, Informática Educacional e a mediação do erro na educação. Considera-se erro como uma questão de importância no desenvolvimento pedagógico, então isso deve ser mediado. A mediação de erro é utilizada como um guia para uma classificação quantitativa de exercícios, levando em consideração quantas vezes um exercício foi respondido erroneamente, a distância entre os exercícios dá suas similaridades. Esse tipo de classificação automática para exercícios em sistemas de apoio educacional é uma das principais contribuições dessa pesquisa. Este trabalho sugere que o ciclo CBR é profícuo no desenvolvimento de uma ferramenta para criação automática de exames.

Palavras-chave: raciocínio baseado em casos, mediação por erros, representação do conhecimento, informática educacional.

ABSTRACT

Case-based Reasoning (CBR) is a method for solving problems with similar retained solutions. CBR demands a knowledge representation that allows the reasoner to find similar cases by a query and the similarity rate is given by a distance in hierarchical tree structure, an ontology. The main goal of this research is to use CBR as a pedagogical tool supported by three pillars: Case-based reasoning, Knowledge representation and Error Mediation in Education. It is considered that the error has a role of importance in the pedagogical development, so it has to be mediated. The error mediation is used as a rule for a quantitative way of exercises classification, it takes into account how many times an exercise have been uncorrected answered, the distance between exercises gives the similarity between them. This kind of automatically classification for exercises in a educational support systems is one of the main contributions of this research. This work suggests that the CBR cycle is useful in the designing of a tool for automatic creation of exams.

Keywords: case-based reasoning, error mediation, knowledge representation, educational informatics.

CAPÍTULO 1

INTRODUÇÃO

Criar listas de exercícios automaticamente, utilizando um método de resolução de problemas é o cerne desse trabalho. Nos próximos parágrafos, o leitor tem uma contextualização da dissertação que aqui se inicia. É apresentado como surgiu o tema, suas motivações e contribuições. Diante das motivações e também das justificativas apresentadas, chega-se ao objetivo, tratado na seção 1.1 e a seção 1.2 apresenta a estrutura do presente documento.

Nos últimos 20 anos houve uma evolução digital atrelada ao crescimento da *World Wide Web*. A informação se espalha e se multiplica por computadores e servidores ao redor do mundo. Aumenta a demanda por recursos de armazenamento e processamento de dados e também a preocupação com seus métodos de organização e processamento. A Ciência da Computação é uma disciplina fundamental nos estudos destes recursos, e nessa pesquisa elenca-se a Inteligência Artificial (IA) como o estudo de soluções para problemas representados computacionalmente.

Russel e Norvig [4] apresentam quatro abordagens na definição de IA, todas encaminhadas na ideia do comportamento computacional se aproximar do comportamento humano, sendo elas: pensar humanamente, agir humanamente, pensar racionalmente e agir racionalmente¹. Esta última abordagem, agir racionalmente, introduz o conceito de agente racional: “Um agente racional é aquele que age pra efetuar o melhor resultado ou, quando há incerteza, o melhor resultado esperado” [4]. O que vai de encontro ao conceito dado por Dean *et al* [5] no livro “Artificial Intelligence: Theory and practice”: “IA é o desenvolvimento e estudo de programas de computador que se comportam de forma inteligente².”

¹O racional em contraponto com o humano apenas confere que o ser humano é “instável emocionalmente” [4].

²Embora a definição de *inteligência* provoque uma discussão nos conceitos, o presente estudo considera inteligência como a melhor forma de se resolver um determinado problema.

A pesquisa aqui apresentada é desenvolvida no Laboratório de Inteligência Artificial e Métodos Formais (LIAMF)³, dentro do Departamento de Informática (DINF) da Universidade Federal do Paraná (UFPR). Uma das principais linhas de pesquisa dentro do LIAMF é sobre os Sistemas Tutores Inteligentes⁴, o que inclui estudos sobre ambientes inteligentes de apoio à aprendizagem. Considera-se relevante dentro dessa linha a busca por métodos no contexto da IA que contribuam com sistemas de apoio à aprendizagem.

Dentro da linha de pesquisa citada, desenvolveu-se a tese de doutorado de Alexander Kutzke [6], onde um arcabouço de sistema para auxílio na promoção do erro no ensino de programação de computadores é desenvolvido, o FARMA-ALG. Kutzke levanta algumas demandas a saber: análise da recomendação de erros no processo de ensino e aprendizagem; pesquisa e desenvolvimento de uma ferramenta para a criação automática de provas e análise dos dados de erros coletados. E experimentar algum método ainda não explorado no LIAMF que pudesse colaborar com as demandas apontadas por Kutzke em [6] impulsionou a elaboração da proposta de pesquisa. O método escolhido para experimentar é o Raciocínio Baseado em Casos (*Case-Based Reasoning* - CBR), método de resolução de problemas através da recuperação e adaptação de casos resolvidos anteriormente.

O raciocínio baseado em casos como abordagem teórica foi concebido nos anos 80 por Roger Schank [7] e Janet Kolodner [8]. Atualmente, no Brasil, um único livro foi publicado tratando sobre o assunto, o Raciocínio Baseado em Casos, de 2004 de autoria dos pesquisadores Christiane Gresse von Wangenheim e Aldo von Wangenheim [9]. O estudo do CBR é orientado por duas motivações primordiais. A primeira é o da ciência cognitiva, o desejo por modelar o comportamento humano. O segundo, do ponto de vista da IA, é o desejo de desenvolver sistemas de IA mais efetivos, conforme pode ser visto no artigo [10] publicado em 1996.

Compreender o que é caso em CBR é profícuo nesse instante introdutório do trabalho. Janet Kolodner [8] conceitua caso (*case*) como um pedaço contextualizado de conhecimento representando uma experiência onde um problema foi resolvido e ensina uma lição determinante para que o resolvidor alcance o objetivo determinado.

³<http://www.inf.ufpr.br/liamf/>

⁴<http://www.inf.ufpr.br/liamf/sti.html>

Um resolvidor baseado em casos utiliza de um ciclo de atividades conhecido por ciclo CBR. O ciclo CBR é proposto por Aamodt e Plaza [1], contém quatro ações: recuperar, reutilizar, revisar e armazenar. Na fase de recuperação, há a busca por casos de acordo com um problema proposto. Se o problema já foi resolvido anteriormente, ele já está na base de casos e a recuperação traz uma resposta exata para o problema e reutiliza-se a solução. Porém se isso não ocorre, há uma possível adaptação da solução na reutilização de acordo com o problema, essa adaptação se dá pela similaridade. Após a revisão, onde se avalia se a solução proposta está de acordo com determinados critérios, o caso é armazenado.

A adaptação de soluções para problemas similares desperta atenção e levanta a hipótese de que o CBR como método pode ser útil em sistemas de apoio à aprendizagem. No presente trabalho os casos serão formados por listas de exercícios e a busca por listas similares é o foco da pesquisa, possibilitando a geração de listas similares e distintas para distintos alunos. Na tese de Kutzke [6], o erro no processo de aprendizagem tem papel de importância, e utilizando a base de dados levantadas durante a pesquisa com o FARMA-ALG, classificam-se exercícios resolvidos utilizando o erro como parâmetro no agrupamento.

Organizar informação é salutar tanto quando se trata de raciocínio baseado em casos quanto em sistemas de apoio à aprendizagem. Durante as investigações em CBR, percebe-se que a representação de conhecimento é um dos fatores preponderantes, o que também faz parte de uma das linhas de pesquisa do LIAMF - Lógica e representação do conhecimento⁵. A classificação na base de conhecimento utiliza de conceitos oriundos dos estudos de ontologia, semântica web, *linked data*, Arcabouço de representação de Recursos (Resource Description Framework - RDF). A representação de recursos no RDF se estrutura em tuplas (sujeito, propriedade, objeto). Os elementos vão se relacionando através de grafos que permitem a conexão entre objetos distintos, isso possibilita uma geração de inferências eficiente, permitindo contextos mais complexos.

Dá-se a formação dos pilares teóricos apresentados nos parágrafos anteriores: raciocínio

⁵<http://www.inf.ufpr.br/liamf/rc.html>

baseado em casos, representação de conhecimento, sistema de apoio à aprendizagem, mediação do erro na educação. Na elaboração da proposta de pesquisa, opta-se por um sistema CBR genérico para investigação, o TUUURBINE. Desenvolvido nos laboratórios LIRIS⁶ e LORIA⁷, TUUURBINE é um sistema CBR genérico que utiliza do RDF na representação de dados. Elencando-se na proposta de pesquisa as perguntas:

1. Diante dos sistemas CBR observados, quais as motivações para a utilização do RDF na representação?
2. Qual é a eficiência, no que tange ao uso de recursos computacionais, de um sistema CBR baseado em RDF na recomendação de conteúdo pedagógico com relação aos existentes?
3. A utilização de um sistema CBR baseado em RDF pode ser mais eficaz, no que confere à qualidade das recomendações, diante de outros sistemas recomendadores para educação?

Os dados colhidos do FARMA-ALG são estruturados para uma representação por meio de ontologias. A dissertação tem o objetivo ajustado à mediação do erros, que intenta utilizar da capacidade dos sistemas CBR de adaptar soluções a problemas distintos. Onde é possível propor um sistema CBR como um módulo dentro de sistemas de apoio à aprendizagem, e experimentar-se-ia o CBR como método para elaboração de lista de exercícios distintas para distintos alunos.

Então, resta determinar onde o trabalho pretende chegar. Um norte ao qual a pesquisa aponta. O objetivo foi pensado, conforme apresentado anteriormente, de forma a colaborar com a linha de pesquisa de Sistemas Tutores Inteligentes e também, durante a estruturação dos dados, com a linha sobre Lógica e Representação do conhecimento. A próxima seção apresenta os objetivos dessa pesquisa.

⁶<http://liris.cnrs.fr/>

⁷<http://www.loria.fr>

1.1 Objetivos

1.1.1 Objetivo Geral

Experimentar o método do Raciocínio Baseado em Casos (CBR) na recomendação de conteúdo pedagógico mediado pelo erro.

1.1.2 Objetivos Específicos

- Observar ferramentas CBR e suas aplicações.
- Estudar ferramentas CBR na recomendação de conteúdos através de técnicas empíricas.
- Trazer a compreensão sobre a mediação do erro na educação escolar.
- Experimentar uma ferramenta CBR para gerar lista de exercícios utilizando a mediação do erro como parâmetro.
- Perceber a contribuição da representação do conhecimento na consulta de base de casos.

1.2 Estrutura do Documento

O capítulo 2 aborda didaticamente o tema do Raciocínio Baseado em Casos. O capítulo 3 trata sobre representação de dados, trazendo a compreensão de conceitos de *linked data*. O capítulo 4 fecha a parte central do trabalho e consta os dados que serão utilizados. O capítulo 5 apresenta como é efetuada a pesquisa, apresentando a estruturação dos casos e como serão feitas as consultas. E finalmente, a conclusão do documento no capítulo 6.

CAPÍTULO 2

RACIOCÍNIO BASEADO EM CASOS

Raciocínio baseado em casos (*Case-Based Reasoning* - CBR) é o método de resolução de problemas que se baseia em casos anteriores. Essa compreensão pode se dar tanto na área pedagógica no aspecto cognitivo quanto na Ciência da Computação. Como método computacional, o CBR é uma linha dentro da área de Inteligência Artificial. Segundo Adrian Furnham [11], se há um problema e uma solução possível, devem-se criar agentes inteligentes de software para tal, um agente que aprende e resolve problemas. Este é o cerne do CBR.

Esse Capítulo busca contextualizar o leitor sobre CBR no âmbito computacional. Traz a concepção da ideia, os eventos, competições, as principais publicações e seus autores. O capítulo não pretende compará-lo paulatinamente com técnicas similares na IA. A seção 2.1 apresenta o que é CBR. A seção 2.2 trata sobre os três principais sistemas genéricos CBR identificados na literatura; na seção 4 há pesquisas que mostram a interface de Sistemas Tutores Inteligente com o CBR. A seção 2.3 trata sobre a interface entre sistemas de apoio à educação e o raciocínio baseado em casos.

2.1 Raciocínio Baseado em Casos

Furnham [11] traz que resolver problemas é uma ação intelectual com uma meta e um propósito definido.

“Alguns “problemas” são resolvidos com relativa rapidez, quase automaticamente, porque são encontrados o tempo todo. Outros problemas, contudo, requerem reestruturação, visão e recálculo” [11].

Ou seja, conforme o tempo vai passando em nossa vida e, geralmente, mais experiências vamos adquirindo, mais fácil e rápido fica tomar determinadas decisões, resolvendo pro-

blemas e também os precavendo. O conceito de *heurística* usado na psicologia descreve um método que as pessoas utilizam para tentar resolver os problemas [11]. Resolver problemas com base em outros problemas similares é um método heurístico.

Aprendemos a não colocar o dedo na tomada, pois dá choque, e se ninguém nos tivesse falado isso, aprenderíamos de forma dolorosa. Dada a situação “dedo na tomada” acarreta em um problema chamado “choque”, podemos começar o esboço de uma representação computacional para um caso: Raciocinar com base em casos é considerar o sucesso e também o fracasso. Para melhor compreender o conceito, pode-se ampliar a complexidade dos casos. Traz-se um personagem chamado Timóteo, rapaz que acaba de sair da casa dos pais e tem que se adaptar com a vida de morar sozinho. Ele enfrenta alguns desafios nessa nova empreitada e aprende a resolver problemas cotidianos.

Quando está fazendo café, Timóteo derruba uma boa quantidade do pó, ainda seco, no chão, sendo assim, o Problema é representado por **chão sujo de café**. Ele suspeita que a solução é utilizar uma toalha de papel seca, temos a **Solução1:passar toalha de papel**. Porém, ele avalia que **Solução1** falhou, uma parte considerável de pó ainda permanece no chão.

Janet Kolodner, uma das pioneiras nas pesquisas em raciocínio baseado em casos, traz em seu artigo de 1992 “Uma Introdução ao Raciocínio baseado em casos” [8], onde CBR pode significar a adaptação de soluções antigas de acordo com novas demandas; utilização de casos antigos para explicar novas situações e também para criticá-las. A noção de caso (*case*) é fundamental para iniciar o estudo em CBR, já tratado na introdução, porém a repetição é dada como necessária na presente seção. Kolodner define caso como um pedaço contextualizado de conhecimento representando uma experiência que ensina uma lição fundamental para que o resolvidor alcance o objetivo determinado, experiência cobre tanto situações bem sucedidas ou que falharam. Aldo von Wangenheim *et al* [9] no livro “Raciocínio baseado em casos”, conceituam caso como “uma peça de conhecimento contextualizado que registra um episódio em que um problema ou situação problemática foi total ou parcialmente resolvida”. Quando Timóteo tenta uma solução para o problema do pó de café, mesmo falhando, temos um caso.

De acordo com Kolodner, a qualidade de um resolvidor baseado em casos depende de quatro fatores [8]:

- as experiências que ele tem;
- sua habilidade em entender novas situações utilizando antigas experiências;
- a competência na adaptação; e
- a competência na avaliação.

Kolodner [8] reforça que resolvidores menos experientes terão um número menor de soluções do que os mais experientes. Não significa, necessariamente, que o menos experiente trará soluções piores, apenas que tal fator sugere que um resolvidor mais especialista pode ser melhor qualificado. No que tange a habilidade de compreensão de situações novas, duas partes se apresentam: *recordar* experiências e *interpretar* a nova situação em termos de experiências lembradas. A adaptação é o processo de se ajustar uma solução antiga às demandas de uma nova situação. E a competência de avaliação é o *feedback* dado ao resolvidor, que por sua vez deverá avaliar se o que foi proposto como solução de fato resolveu o problema.

Aamodt e Plaza, em 1994, [1] irão demonstrar o CBR como um processo cíclico na resolução e aprendizado de problemas. A Figura 2.1 representa este ciclo contando com os processos:

- Recuperação (*Retrieve*);
- Reutilização (*Reuse*);
- Revisão (*Revise*);
- Armazenamento (*Retain*).

De acordo com Aamodt e Plaza [1], o sistema recebe como entrada um novo caso e o resolve recuperando um ou mais casos antigos. Após isso, o caso será reutilizado. Ocorre a revisão da solução com base nesta reutilização e finalmente armazenará o novo caso,

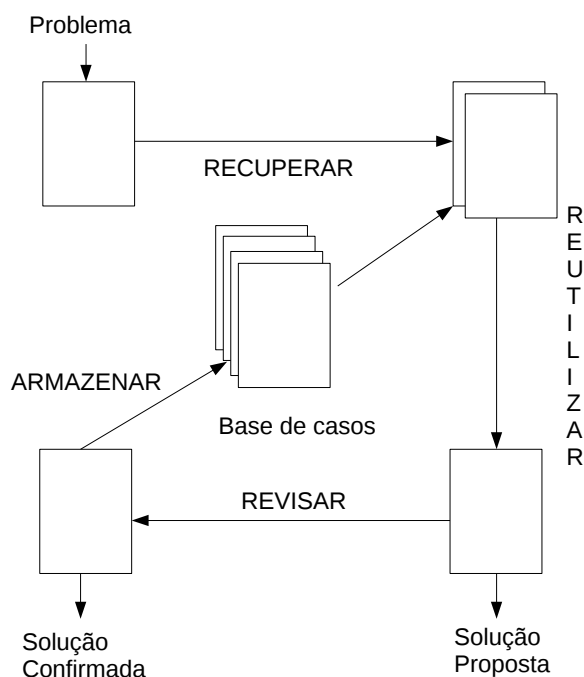


Figura 2.1: Ciclo CBR Aamodt e Plaza (1994) [1]

incorporando-o com a base de conhecimento. Cada processo do ciclo pode contar com tarefas mais restritas, tratadas abaixo.

No processo de *Recuperação* ocorre a identificação das características de um determinado caso. Timóteo armazena o caso `Problema:chão sujo de café + Solução1:passar toalha de papel`, estabelece um Domínio de conhecimento `D: Sujeira` e algumas categorias de classificação: `pó, café, chão, seco, toalha, papel`. Assim que o resultado for encontrado, há uma subdivisão por um correspondente inicial, recuperando um número de candidatos possíveis e um processo mais elaborado que seleciona o melhor dentre eles, e por fim seleciona os casos semelhantes [1]. No exemplo, o sistema buscaria por casos semelhantes no Domínio de `Sujeira`, buscando resultados que se assemelham.

Na *Reutilização* da solução, o foco é dado nas diferenças entre o caso recuperado e o novo caso, identificando qual parte do caso antigo pode ser utilizado. Conforme o exemplo, o resolvidor baseado em casos do rapaz começará a fazer as devidas substituições para colocar mais um caso dentro da base. As tarefas pertencentes ao processo de reutilização são o de cópia e a adaptação. Neste momento há de se observar dois caminhos na reuti-

lização de casos passados, a resolução e também o método que construiu a solução. No exemplo, o resolvidor sugere, no Domínio D: Sujeira, um caso: Problema:sal no chão + Solução100:pano úmido com um pingo de detergente. Diante de outros métodos de resolução de problemas, a adaptação do CBR se diferencia, entre outros aspectos, pela sua capacidade de conseguir, de acordo com algumas técnicas, substituir elementos semelhantes.

A fase de *Revisão* é o momento para identificar se a solução é correta e caso não seja, a falha será aprendida. Duas tarefas acontecem nessa fase: avaliar a solução e a de reparar os erros da solução atual. Finalmente a fase de *Armazenamento*, onde o caso é aprendido, incorporando o que é útil, onde o problema resolvido será retido na base de conhecimento. O sistema irá buscar qual informação do caso será retida, em qual formato, sua representação e integrá-lo na estrutura de memória. Como a limpeza com o pano úmido e um pingo de detergente funciona, há um novo caso armazenado: Problema:chão sujo de café + Solução100:pano úmido, um pingo de detergente.

O primeiro evento internacional focado em CBR aconteceu em Outubro de 1995 com a Conferência Internacional de Raciocínio Baseado em Casos (*International Conference of Case-based Reasoning - ICCBR*). Publicações traziam abordagens para tratar da adaptação, representação, similaridade de casos no CBR. Em 1997, começam a ser apontadas técnicas para os distintos processos dentro do Ciclo CBR. Neste ano, durante a segunda ICCBR, são apresentados estudos sobre modelagem fuzzy, modelos probabilísticos e também sobre representação da base de casos de um sistema. Até 2009, o evento era bienal e intercalava como a Conferência Europeia de Raciocínio Baseado em Casos (*European Conference of Case-based Reasoning - ECCBR*), mas em 2010 as duas se uniram em uma única série, e assim o evento se tornou anual.

Richter contribui, em 1998, com uma publicação a respeito de CBR [12]. Ele traz, além de conceitos básicos aqui anteriormente tratados, a abordagem quanto à representação de conhecimento no CBR. Ele aborda a noção de containers de conhecimento (*knowledge containers*). Segundo Richter [12], o conhecimento utilizado durante o ciclo CBR é representado formalmente em quatro containers:

- *Vocabulário* definido por atributos e o conjunto de valores permitidos.
- *Métricas de Similaridade* são funções que calculam a similaridade entre os atributos individualmente e também entre problemas e casos. Em geral, são funções que calculam a distância entre elementos em tabelas comparativas.
- *Casos* são as descrições de episódios de experiências pode ser dividido em pares de problemas e soluções, contendo os atributos valorados utilizados para a descrição do problema e também da solução.
- *Adaptações ou transformações da solução* é o conhecimento representado por regras utilizadas para a adaptação da solução nos casos recuperados.

“O conteúdo dos containers podem ser alterados localmente, tendo que manipulações em um container tem consequências nos outros” [12]. Conhecer representação modernas de dados é fundamental para a representação do conhecimento em CBR, aquela será dada no capítulo 3 e esta será tratada no experimento do capítulo 5.

Watson, em 1999 [2], percebe o CBR como uma metodologia¹ para resolução de problemas e não como uma tecnologia² [2]. Watson trouxe que CBR utiliza de outras técnicas - programação lógica, redes neurais, algoritmos genéticos, lógica fuzzy dentre outras mais - para resolver problemas. Watson traz a definição de Schank [7]: “Um resolvedor baseado em casos soluciona problemas utilizando ou adaptando soluções para problemas anteriores”. Watson mostra que Schank define “o que” um resolvedor baseado em casos faz e não “como” ele faz. “Conceitualmente CBR é comumente descrito pelo ciclo-CBR” [2]. A figura 2.2 ilustra uma forma de ver como o ciclo CBR interage com outras tecnologias. Elas atuam diretamente em dois processos fundamentais para o raciocínio: na recuperação e revisão.

Através de uma análise de alguns sistemas de CBR que utilizam distintas tecnologias, Watson percebe que eles seguem um conjunto de princípios orientadores:

- resolver problemas através da reutilização de soluções de problemas anteriores;

¹Segundo [13], Metodologia é o processo para se atingir um determinado fim ou para se chegar ao conhecimento.

²Tecnologia [13] é a aplicação prática do conhecimento científico em diversas áreas de pesquisa.

- a recuperação de problemas passados (casos) consiste em acessar através das similaridades dos problemas e
- assim que um novo problema é resolvido, ele é adicionado à biblioteca de casos, passível de utilização.

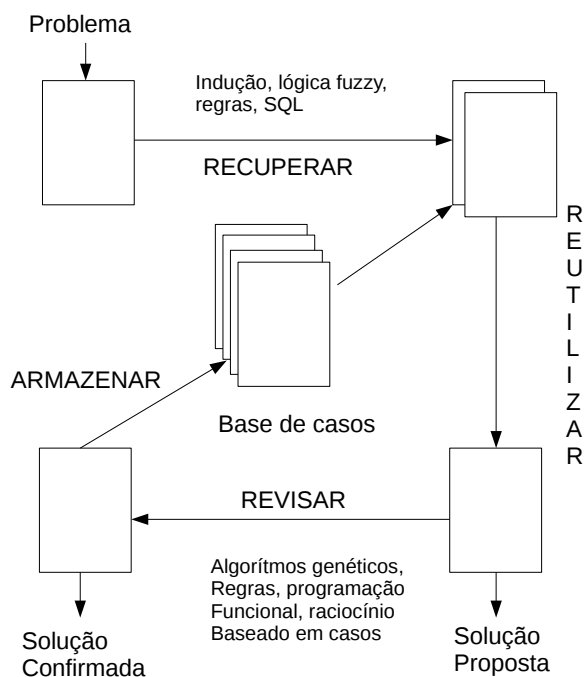


Figura 2.2: Tecnologias e o Ciclo CBR [2]

Em 2001, a aplicabilidade do CBR se apresenta como assunto acadêmico, assim, algumas abordagens surgem no intuito de aprimorar a metodologia em questão. Tais abordagens tratam sobre aquisição, representação e organização do conhecimento.

Aha e Breslow [14], em 2001, elaboram um artigo sobre raciocínio baseado em casos de conversação, onde a aquisição conhecimento se dá através do diálogo entre cliente e agente. Não há um modelo de domínio e nem uma estrutura padronizada para todos os casos. Embora aparente uma pesquisa fragilizada pela não delimitação de um domínio, ela é profícua para domínios onde um volume alto de pequenos problemas devem ser resolvidos várias vezes.

O artigo de 2003 de Bergmann e Schaaf [15] trata sobre CBR estrutural, abordagem

dada nos casos descritos com atributos e valores pré-definidos. Atributos podem ser organizados como tabelas, conjunto de tabelas ou também orientados a objeto. Reflexão útil aos domínios onde conhecimentos adicionais são utilizados no intuito de aprimorar os resultados.

Em 2005, Kolodner et al.[16] ajustam o Porém, quando era adolescente, Fulvia já sentia que seu corpo não correspondia à imagem que ela tinha de si mesma. Na verdade, ela nasceu um menino, batizado de Fulvio. Porém, secretamente, seu maior desejo era ser uma mulher.foco na representação do raciocínio baseado em casos. Segundo Aldo von Wangenheim [9], a representação é um dos elementos básicos de um sistema CBR.

“A experiência representada em um caso pode ser estruturada de várias maneiras. Geralmente, isto é descrito com a subdivisão em problema e solução, mas de acordo com o tipo da reutilização, um conhecimento adicional pode ser necessário” [16].

Dentro dos sistemas apresentados nas próximas seções, observa-se tal quesito também.

Em 2006, Craw et al.[17] trazem métodos que buscam fazer as melhores adaptações. Como caso de estudo, o artigo em questão utiliza um resolvidor, com domínio farmacêutico, para manipulação de comprimidos. Os sistemas CBR que apenas recuperavam informação tiveram melhor desempenho depois de aprenderem a adaptação de conhecimento na forma de casos ou regras. Na recuperação de casos anteriores e similares o método mais utilizado foi o *algoritmo do vizinho mais próximo* - método que, no intuito de andar por um grafo³, considera a distância entre os vértices, optando sempre por visitar o mais próximo, muito bom para instâncias pequenas porém não muito confiável para dados de dimensões elevadas [4]. Percebe-se que uma escolha dos algoritmos de aprendizagem é um caminho estratégico para a generalização apropriada na tarefa.

É muito comum dentro das comunidades de Inteligência Artificial a ocorrência de competições com domínios ou aplicações genéricas, profícuas como ambiente para testes de novas técnicas e avanços nas metodologias. A comunidade CBR criou o Desafio do Computador Cozinheiro (*Computer Cooking Contest - CCC*). A sua primeira edição foi

³Há uma explicação sobre grafos na seção 3.3

em 2008, no ECCBR e, a partir de então, o ICCBR conta sempre com a competição. Dentro das ferramentas desenvolvidas é possível observar os funcionamentos e avanços das técnicas de recuperação, reutilização, revisão e armazenamento.

2.2 Sistemas e Frameworks genéricos para CBR

Três sistemas são elencados para conhecimento do leitor. O primeiro é o jColibri [18], um framework orientado a objetos disponível para o desenvolvimento de sistemas CBR (utilizado no JadaCook), subseção 2.2.1. O segundo, tratado na subseção 2.2.2 é o TUUURBINE [19], um sistema CBR genérico que utiliza tecnologias de web semânticas em sua representação. E a última parte dessa seção (5.1) é o myCBR [20]. Os pontos destacados na presente seção: o ciclo CBR e a representação.

2.2.1 jColibri

O jColibri⁴, apresentado em [18, 21], é um arcabouço orientado a objetos, desenvolvido em Java para a obtenção de sistemas CBR, um projeto da Universidade Complutense de Madrid. A arquitetura do jColibri contém uma estrutura para os casos que permite o acesso direto ao banco de casos, os quais são representados através de *Java Beans*. *Java Beans* é uma representação utilizada em Java caracterizada pelo encapsulamento de vários objetos em um só. O sistema também tem uma biblioteca de métodos para guardar as soluções. O objetivo principal da pesquisa que tem como produto o jColibri, é atender a necessidade de um software aberto para CBR.

jColibri representa um caso de maneira genérica, onde cada caso é um indivíduo que pode ter qualquer número de relações com outros indivíduos (atributos de caso)[18]. A base de casos do jColibri oferece um conjunto de conectores para o gerenciamento das bases. Conectores são objetos que permitem o acesso e recuperação de casos do meio de armazenamento e retornar ao sistema CBR de uma forma uniforme. jColibri inclui conectores que trabalham com arquivos de textos, arquivos XML, base de dados relacionais e sistemas de lógica de descrição.

⁴<http://gaia.fdi.ucm.es/research/colibri>

A aplicação divide-se em pré-ciclo, ciclo e pós-ciclo. A tarefa de pré-ciclo está dedicada a computação da estrutura de índices ou processamentos dos textos. Posteriormente, se dá o ciclo, com as quatro fases do CBR, na busca de casos similares, reutilização de casos anteriores, revisão da solução proposta e armazenamento da experiência. O pós-ciclo tratará sobre as tarefas de manutenção, como lembrar de soluções inutilizadas ou recalculando a estrutura dos índices.

A versão atual é a jColibri2, teve a arquitetura dividida em duas camadas, onde a inferior, chamada de jColibri, detalhada em [21], utiliza de uma recuperação mais aprimorada no gerenciamento da base de casos. Enquanto na camada de cima, a plataforma se encontra o Colibri Studio, até 2013 constava como em desenvolvimento, mas utilizava das mesmas propriedades da primeira versão, a qual cuida das tarefas e dos métodos de resolução de problemas. A estrutura de casos é organizada em: descrição, solução, resultado e justificativa da solução.

A ferramenta tem grande aceitabilidade na comunidade acadêmica, já atingiu a marca de 10.000 downloads com usuários em 100 países (até 2015). Busca-se ainda o desenvolvimento de ferramentas de alto nível, por meio da disponibilidade de *templates* que possam ser utilizados em distintas aplicações, estes modelos podem ditar o comportamento do sistema CBR.

2.2.2 Tuurbine

Foi com o intuito de desenvolver um sistema genérico capaz de resolver problemas de outros domínios que o mesmo grupo que desenvolveu o TAAABLE por cinco anos fez o TUURBINE⁵. Demonstrando que a participação no CCC pode ser produtiva no avanço dos sistemas CBR como um todo. O TUURBINE implementa inferências baseadas em casos genéricos. A busca por casos similares se dá pelos custos de generalização e nas regras de adaptação originárias no TAAABLE. Além da motivação de se desenvolver uma ferramenta genérica, outro estímulo é trazer um sistema que possa dar conta da grande e crescente quantidade de conhecimento disponível na Web [19].

⁵<http://tuurbine.loria.fr/>

O TUUURBINE foi desenvolvido de acordo com as padronizações RDF (*Resource Description Framework*)⁶, SPARQL (linguagem de busca para RDF)⁷, RDFS (*RDF schema*)⁸ para facilitar a interoperação com a Web Semântica. O conhecimento é arquivado em um *triple store* (base de dados voltadas especialmente à RDF). Tal local pode ser instalado tanto na máquina onde está o sistema ou em alguma máquina remota. Ao contrário do TAAABLE, o conhecimento não é mais gerenciado pelo sistema CBR, mas será uma ferramenta externa e baseada nas linguagens padrões. Tem-se somente o conhecimento necessário recuperado para resolver um problema novo.

Considera-se profícuo observar representações de conhecimento independentes do arcabouço CBR. Especificamente, o TUUURBINE utiliza o RDF, representação eficiente no que confere ao armazenamento de domínios de conhecimento mais complexos.

2.2.3 myCBR

O myCBR⁹ é um arcabouço desenvolvido através da parceria entre um centro de competência de CBR no DFKI¹⁰ e a escola de Inteligência Artificial da UWL¹¹. A versão mais recente é 3.1 beta e é descrita como uma ferramenta *open-source* para recuperação de dados através da similaridade. Ainda não se apresenta como um sistema CBR completo, pois não é possível efetuar as fases de Reutilização e Armazenamento previstos no ciclo CBR. Embora, Hundt *et al* em [20] afirmem que a adaptação estaria presente nas próximas versões, a versão disponível não está atualizada nesse quesito.

O myCBR possibilita uma construção da base de conhecimento através de um interface simplificada e direta. O usuário modela o conhecimento através de um conjuntos de atributos possíveis e valores possíveis para esses atributos. É possível também estruturar os casos ou carregar casos através de arquivos no formato .csv. Os valores que fazem parte do domínio de atributos são relacionados através da similaridade, determinante na

⁶<http://www.w3.org/standards/techs/rdf>

⁷<http://www.w3.org/standards/techs/sparql>

⁸<http://www.w3.org/TR/rdf-schema/>

⁹<http://mycbr-project.net/>

¹⁰<http://www.dfki.de/web>

¹¹<http://www.uwl.ac.uk/>

recuperação de casos. O myCBR é o arcabouço de CBR utilizado para a execução dos experimentos dessa dissertação, e uma abordagem detalhada sobre os conceitos e teorias que o cerceiam são dados no capítulo 5.

2.3 O CBR nos Sistemas de apoio à aprendizagem

Aprimorar sistemas que auxiliem na aquisição de conhecimento por parte dos indivíduos é uma atividade inerente ao campo de pesquisa de Sistemas Tutores Inteligentes (STI). Compreende-se que soluções propostas em estudos da Inteligência Artificial podem contribuir com teorias subjacentes aos STI. A próxima subseção confere uma abordagem objetiva ao campo de sistemas de apoio ao ensino. E para que se possa perceber como se dá a interface entre raciocínio baseado em casos e STI, objetivamente na utilização daquele neste, a subseção 2.3.2 traz pesquisas que mostram aplicações nesse contexto.

2.3.1 Sistemas de Apoio ao Ensino

Dentro do campo de sistemas voltados à educação, destacam-se alguns ambientes de aprendizagem a serem observados e compreendidos: sistemas de ensino, sistemas tutores inteligentes. Diante dos dois grupos de sistemas apresentados temos a utilização dos sistemas recomendadores voltados para educação.

Para compreender o conceito de Sistemas Tutores Inteligentes (Intelligent tutoring systems) o livro de Etienne Wenger, *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge* [22], datado de 1987, traz conceitos fundamentais para estudos nesta área em questão. No livro, identifica-se uma transição entre a utilização de computadores como ferramentas auxiliares na educação de sistemas - Instruções Assistidas por Computador (*Computer-Aided Instructions* - CAI) - para as que adquirem conhecimento passíveis de interação e instrução, os STIs.

Nessa visão transitória, Wenger traz a nomenclatura de sistema de comunicação de conhecimento: sistema que acessa representações de aprendizagens, no intuito de co-

municá-las. “Os Sistemas Tutores Inteligentes são sistemas instrucionais baseados em computador com modelos de conteúdo instrucional que especificam *o que* ensinar, e estratégias de ensino que especificam *como* ensinar” [22]. Os projetos de STI podem variar significativamente pelo nível relativo de inteligência desses componentes. Considera-se como uma categoria de sistemas que também pode ser chamada de agentes pedagógicos.

De acordo com McGreal [23], Objetos de Aprendizagem (OAs), são recursos educacionais que podem ser empregados em aprendizados suportados por tecnologias (e.g. exercícios, vídeos, tutorais). OAs podem ser organizados de acordo com o domínio de aprendizagem dentro de repositórios que poderão armazenar tanto filmes quanto jogos educacionais. Os STIs utilizam o repositório de OAs como recurso dentro dos processos.

2.3.2 O CBR nos Sistemas de apoio à aprendizagem

Dentro dos modelos estruturais da arquitetura básica dos STI, pesquisas se desenvolveram na busca por soluções, metodologias e técnicas diversas que possam melhorar os principais aspectos dos STI. Dentre elas, o uso do Raciocínio Baseado em Casos pode ser aplicado à educação, na pesquisa que aqui se apresenta foi identificada a utilização da metodologia na adaptabilidade de ambientes de aprendizagem.

Funk e Conlan [24] trazem uma visão geral de alto nível de uma técnica de monitoramento personalizado para adequar e melhorar a qualidade das recomendações utilizando CBR de baixa latência e técnicas de filtragem. A pesquisa se enquadra dentro da área de Gestão de Conhecimento (GC), área que busca o estudo de métodos, modelos e estratégias para busca, reutilização e manutenção de conhecimento. Considera-se a GC altamente relevante no estudo dos STI e principalmente quando o desenvolvimento daquele aborda diferentes métodos e técnicas da IA, como o CBR, *clustering* e técnicas de filtragem colaborativa.

Considerando que um modelo é um processo parcialmente ordenado composto de componentes (e.g. objetos educacionais) com pedidos e resultados (e.g conhecimento pedido e conhecimento aprendido) [24]. Partes dos modelos podem ser salvos com informações que entram através do *feedback* do usuário. Quando se trata de um grande espaço de

busca, sugere-se sistemas CBR no apoio dentro de STI, possibilitando corrigir problemas e reutilizar modelos parcial e totalmente [24]. “Propõe-se abordagem de raciocínio baseado em casos para identificar e corrigir os problemas no que confere à personalização do conteúdo, podendo ser por correspondência, reutilização, validação e armazenamento de casos, onde os casos podem ser modelos de aprendizagem individuais, narrativos ou modelos de conteúdo individual” [24].

Os sistemas recomendadores são ferramentas, software e técnicas capazes de sugerir itens de acordo com o perfil de cada usuário, como qual música escutar, qual produto comprar, que notícia ler, etc [25]. Utilizando este conceito dentro dos sistemas voltados à educação, um sistemas recomendador para educação é uma ferramenta auxiliar que compreende certas regras, agrupa os usuários e recomenda conteúdos educacionais. Porém quem receberá a recomendação varia de acordo com o ambiente de aprendizagem. No caso dos STI, quem recebe as recomendações será o estudante. Já no caso de sistemas que apoiam o ensino, a recomendação pode ser feita tanto para professores quanto para os alunos.

Em 2009, Gómez-Albarrán e Jiménez-Dias [26] descreveram a utilização de um resolvidor baseado em casos para a recomendação de OAs armazenados em repositórios educacionais. A abordagem combina o filtro baseado em conteúdo através de processos de filtragem por mecanismos colaborativos. O filtro baseado em conteúdo leva em conta o conhecimento do aluno e qual o objetivo de aprendizagem.

Os mecanismos de filtro colaborativo ajudam a estabelecer a utilidade que um OA concreto tem para um estudante, baseado na classificação (i.e., relevância, preferências e opiniões) que estudantes similares (estudantes com objetivos e nível de conhecimento semelhantes) fizeram com este OA [26].

A incorporação colaborativa de recomendações consiste na avaliação dos Objetos Educacionais, dada pelos alunos. Do ponto de vista do CBR, tal informação é vista como um aprendizado que será utilizada para melhorar o processo de recomendação. Os autores sugerem a utilização de ontologias na indexação dos OAs, pois são consideradas cruciais nos estágios de recuperação e avaliação dos objetos de aprendizagem [27].

2.4 Considerações

Por uma perspectiva da psicologia, resolver problemas é uma ação intelectual com um objetivo específico. Um método para resolver problemas é utilizar casos anteriores semelhantes a um novo problema que aparecer [11]. Sugere-se aqui o método CBR e aplica-se ele no desenvolvimento de sistemas de IA. Conforme o agente inteligente interage com novos problemas, ele vai aprendendo e aplicando soluções de acordo com aquilo que aprendeu.

Sistemas CBR utilizam de técnicas de estrutura de dados e de IA, por isso deve ser visto como um método e não como uma tecnologia [2]. O objeto principal é um caso, que contém um problema e como este foi ou não resolvido. É um método cíclico dividido em quatro grandes tarefas: recuperar, reutilizar, revisar e armazenar.

Membros da comunidade acadêmica dedicada ao estudo do CBR organizaram um desafio no qual poderiam ser testadas as técnicas aplicadas em sistemas CBR. É uma competição onde sistemas computacionais devem sugerir receitas de acordo com ingredientes ou características de cada prato. É o Desafio do Computador Cozinheiro (*Computer Cooking Contest - CCC*). Esta competição contribui com muitos avanços na área de CBR.

O TUUURBINE faz parte da classe de outras ferramentas de CBR genéricas, como o myCBR e o jColibri. É proveitoso ressaltar que, diante das ferramentas aqui apresentadas, a originalidade do TUUURBINE se dá por ser uma implementação de um sistema CBR apoiado em RDF e exploração de semânticas dos modelos RDF e RDFS [19]. O myCBR é um sistema possível modelar o conhecimento de maneira simples e direta, possibilitando testar a recuperação de casos através da relação entre os valores determinados para os atributos dos casos.

O presente capítulo aborda sistemas voltados à educação, focando nos sistemas tutores inteligentes. Destaca-se a utilização do jColibri na recomendação de objetos de aprendizagem armazenados em repositórios educacionais. Embora o tema de aplicações do raciocínio baseado em casos na área de educação seja assunto com relevância, ainda não se percebe uma movimentação na utilização de alguns sistemas genéricos dedicados aos domínios de ensino.

CAPÍTULO 3

REPRESENTAÇÃO DE DADOS

Com o crescimento dos usuários e dados na rede mundial de computadores, tem-se um aumento da preocupação com a disponibilidade da informação, tanto no que confere ao acesso quanto à organização e manutenção dos dados. É um aumento exponencial de dados, dessa forma, técnicas para tratamento de dados são estudadas. Elas se referem: à seleção, ao processamento, à recuperação e à disseminação de dados. Os dados podem, no contexto da rede mundial, ser organizados de acordo com os termos, classificação de categorias ou a partir de conceitos e seus relacionamentos.

A próxima seção (3.1) é introdutória e cria o alicerce ao presente capítulo, tratará sobre *Linked Data*. E para um melhor entendimento deste, é fundamental o conceito de Ontologias tratado na seção 3.2. A representação das informações para o linked data se dá através do RDF, tratado na seção 3.3.

3.1 Linked Data

De acordo com Bizer *et al* [28] *Linked Data* é o termo que se refere ao conjunto de melhores práticas para publicação e aos dados conectados dentro da Web. A rede mundial de computadores transformou radicalmente a forma como o conhecimento é compartilhado, democratizando o acesso ao que é publicado por meio de documentos, onde usuários utilizam links de hipertexto através de navegadores *web* e também mecanismos de busca indexam documentos e analisam a estrutura de links entre eles. Nesse contexto, também surgiram formatos de representação de dados com CSV, XML, tabelas HTML. Porém, tais formatos são consideradas, por Bizer *et al* em [28], inexpressivas para possibilitar entidades individuais descritas em um documento particular para ser conectado através de links com entidades relacionadas.

No cenário que apresenta uma internet onde os documentos não se relacionam facil-

mente é que o *Linked Data* surge. O objetivo é adotar melhores práticas que possibilitem uma conexão global entre todos os tipos de documentos (filmes, publicações científicas, música, programas de TV, etc). *Linked Data* é, resumidamente, a utilização da Web para criar conexões entre dados de origens distintas. “Tecnicamente, *Linked Data* se refere aos dados publicados na rede de uma maneira que possa ser lido pela máquina, seu significado pode ser definido explicitamente, sendo conectado a outros conjuntos externos de dados, e pode por sua vez ser conexão de outros conjuntos externos [28].”

Tim Berners-Lee [29] considera que a semântica web (*Web Semantics*) não é apenas colocar dados na Web. É sobre fazer *links*, pois dessa forma pessoa ou máquina pode explorar dados relacionados na web. Com base nesse raciocínio, Berners-Lee determinou ‘regras’ para a publicação de dados na Web. Ele trata por expectativas de comportamento na publicação de dados na Web que a seguir estão listadas:

1. Utilizar URIs como nome para coisas. URI é a sigla para Identificador Padrão de Recursos (*Uniform Resource Identifier*) e, segundo o autor, o que não estiver nomeado nesse padrão não é chamado de Semântica Web.
2. Utilizar HTTP URIs possibilitando a busca por seus nomes. É necessário que se compreenda que HTTP URIs são nomes (não endereços), pesquisa por nome HTTP é um complexo, poderoso e evoluído conjunto de padrões.
3. Quando alguém procurar um URI, fornecerá informação útil, utilizando os padrões (RDF, SPARQL¹). A busca poderá ser feita utilizando as propriedades e classes, pode achar dados utilizando de informações através do RDF, RDFS, ontologias OWL, incluindo as relações entre os termos na Ontologia.
4. Incluir links para outros URIs, possibilitando a descoberta de outros objetos.

Algumas bases de dados são criadas utilizando os princípios do linked data. O *Wikidata* é um projeto de gestão das informações factuais do *Wikipédia*, também conhecido como “Wikipedia for data” [30]. É uma base de conhecimento colaborativa, assim como o

¹Ambos RDF e SPARQL serão tratados posteriormente, na seção 3.3

Wikipédia. É um projeto que ajuda na compreensão das intenções e abordagens possíveis com Linked Data. Suas propriedades têm influências tanto do Linked Data quanto da Wikipédia:

- Edição aberta: todo o usuário pode editar.
- Controle comunitário: não apenas os dados podem ser controlados pelos usuários mas também o esquema dos dados.
- Pluralidade: permite dados conflituosos e fornece mecanismos para organizá-los.
- Dados secundários: o Wikidata reúne fatos publicados de acordo em fontes primárias, junto com referências e tais fontes, um exemplo é não há um “número de habitantes do Brasil Colônia”, mas sim “o número de habitantes do Brasil Colônia publicado pela UNB em 2011”.
- Dados multilíngues: dados em várias línguas são aceitos.
- Fácil acesso: o objetivo do Wikidata é permitir que os dados a serem utilizados tanto no Wikipédia quanto em aplicações externas, com a utilização de dados exportados para o uso geral de serviços web.
- Evolução contínua: segue como o Wikipédia no crescimento de seus usuários e desenvolvedores.

Algumas pesquisas buscam agregar as possibilidades de compartilhamento de informação que se baseiam nos princípios do Linked Data, porém mais voltadas à pesquisa científica. É o caso da pesquisa apresentada por Bechhofer *et al* no artigo [31], que trata do conceito de *Research Object* (RO), que tem como proposta principal fornecer uma classe de artefatos que podem encapsular conhecimento digital e também fornecer um mecanismo para dividir e descobrir ativos de pesquisas reutilizadas e conhecimento científico. RO pode agregar uma camada estrutural que é consistente com a visão do Linked Data. ROs são tanto recursos acessíveis através dos princípios do linked data e que também pode agregar outros recursos.

3.2 Ontologias

O artigo de Almeida e Bax [32] auxilia na compreensão sobre as classificações, principalmente no que tange a uma visão geral sobre ontologias. Nele, pode-se compreender suas aplicações. Aqui ele será a base teórica no exemplo a seguir. Timóteo quer organizar seus discos, livros e filmes sob as estantes do seu pequeno apartamento. Algumas estruturas de dados organizam as informações de acordo com a utilização dos termos, como um dicionário, um glossário. Então está lá nosso personagem tentando essa estrutura em seus conteúdos culturais, colocando em ordem alfabética.

Timóteo utilizará os títulos dos livros e dos filmes, já para os discos, os artistas. Ele começará com o livro *À espera dos Bárbaros* - J.M Cotzee, logo depois terá outro livro: *A Insustentável Leveza do Ser* - Milan Kundera. O terceiro item seria um disco do *Brazilian Girls* com o álbum *New York City*, e assim seria organizada a coleção de Timóteo. Uma mistura estranha, o que parece não colaborar com a organização dos objetos dentro do seu humilde apartamento, aparentemente não é a melhor solução.

Uma outra forma de organizar os dados seria através das categorias, através do assunto e esquemas de classificação. Continuando com nosso exemplo, Timóteo quer aproveitar melhor o espaço, ele divide em três categorias maiores: discos, livros e DVDs. Em cada categoria haverão outras classificações. Os livros serão divididos por assuntos: romances, esotéricos, técnicos etc. Os discos por estilo musical: clássico, *rock*, *jazz trip hop* etc. Os filmes em *comédia*, *drama*, *suspense* etc.

O nosso personagem gosta um pouco mais da última forma de organizar os itens comparado com a ordem alfabética. Porém, ele ainda tem dificuldade de relacioná-los conceitualmente, tendo em vista que alguns filmes se relacionam com alguns livros, ou discos. Timóteo percebe uma similaridade conceitual entre os livros de Jack Kerouac com os discos do Bob Dylan. Ele tenta organizar a estante de forma que os que se relacionam conceitualmente fiquem próximos, de alguma maneira. Estruturas com a organização fundamentada nos conceitos e suas relações são as *Ontologias* e as *redes semânticas*.

Ontologias permitem um esquema de indexação geral que contempla o conhecimento através da similaridade entre conceitos. Necessita-se de especialistas para a criação de

uma ontologia, eles definirão regras e regularão a combinação entre termos e relações em um domínio do conhecimento. As perguntas serão feitas de acordo com os conceitos definidos dentro da ontologia utilizada.

A organização de Timóteo deve levar em consideração as classes: livro, filmes e discos. Depois teria uma classe de propriedade dentro de cada classe. Onde um objeto pode pertencer a mais de uma propriedade, esta pode acontecer de acordo com o ano que foi lançado, com o movimento cultural, país de origem, etc. Um livro e um filme podem pertencer a uma mesma propriedade porém não a uma mesma classe. “O que se busca, em última instância, são melhorias nos processos de recuperação da informação” [32].

De acordo com Cimiano [33], a origem do termo *Ontologia* vem do grego e é o estudo (-logia) sobre o ser (onto-)[33]. Platão (427 - 347 AC) foi um dos primeiros filósofos a mencionar sobre o mundo das ideias em contraste com o real ou objetos observáveis. Mais tarde, seu aluno, Aristóteles (384 - 322 AC) moldou uma fundação lógica das ontologias e trouxe noções de *categoria* e *subassunto*. A diferença e proximidade entre as coisas era o cerne de uma nova noção, é considerado como o fundador da taxonomia, a ciência da classificação das coisas. Atualmente, na Ciência da Computação, conforme Gruber [34], não se fala mais de ‘ontologia’ como a ciência da existência, mas de ‘ontologias’ como especificações formais de uma conceitualização, ou como a representação de modelos conceituais em um determinado domínio.

Classes fornecem um mecanismo de abstração para agrupar recursos com características semelhantes, agrupando conjunto de indivíduos com propriedades compartilhadas. A apresentação de classes irá representar um modelo de domínio através de uma ontologia. Timóteo agrupa os discos juntos em uma estante, na estante de baixo os livros ficam próximos de alguns discos por compartilharem alguma propriedade (o movimento cultural, por exemplo [35]).

O exemplo de Timóteo pode ser representado graficamente na figura 3.1. Ocorrem os **Discos**, **Livros** e **Filmes**, todas no nível 0. Para cada uma dessas classes, as subclasses terão um nome, por exemplo, em **Livros** se subdividirá de acordo com os autores no seu primeiro nível(1) de subdivisão, já os filmes irão se diferenciar de acordo com o estilo

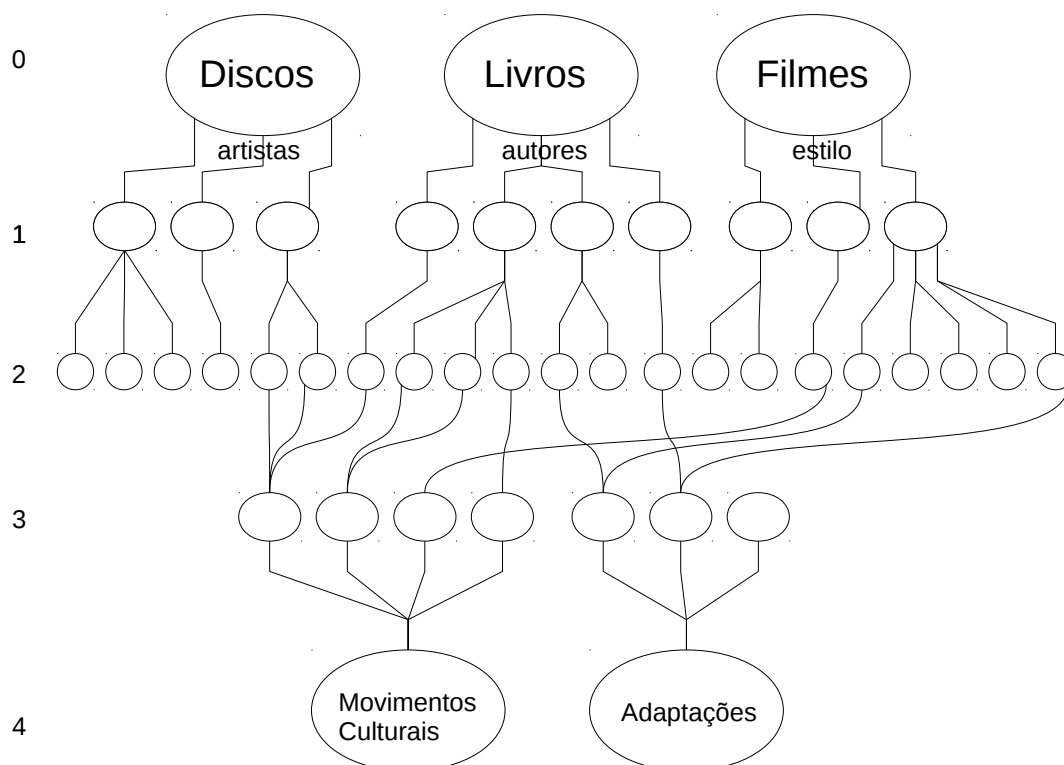


Figura 3.1: Exemplo de Ontologia utilizando a coleção do Timóteo

(ação, drama, terror, etc). Os objetos estão representados no nível 2, perceba que eles se relacionam entre si através de duas subclasses. A do nível 1, diretamente ligadas às mídias. E também se relacionam através de duas propriedades **Movimentos Culturais** e **Adaptações**, sendo a primeira a relação entre os itens que têm essa propriedade em comum, a segunda trata de livros que viraram filmes ou vice-versa.

Cimiano [33] define formalmente ontologia como uma estrutura

$$\mathcal{O} := (\mathcal{C}, \leq_{\mathcal{C}}, \mathcal{R}, \sigma_{\mathcal{R}}, \leq_{\mathcal{R}}, \mathcal{A}, \sigma_{\mathcal{A}}, \mathcal{T})$$

onde:

- \mathcal{C} , \mathcal{R} , \mathcal{A} e \mathcal{T} são conjuntos disjuntos onde seus elementos são chamados de identificadores conceituais(\mathcal{C}), identificadores relacionais(\mathcal{R}), identificadores de atributos(\mathcal{A})

e tipos de dados, respectivamente(\mathcal{T}),

- \leq_C é a representação de uma grade feita sobre C , chamado de hierarquia conceitual ou taxonomia,
- σ_R é uma função $\sigma_R : R \rightarrow C^+$ chamado de assinatura de relação,
- \leq_R uma ordem parcial em R , chamada relação de hierarquia, onde $r_1 \leq_R r_2$ implica $|\sigma_R(r_1)| = |\sigma_R(r_2)|$ e $\pi_i(\sigma_R(r_1)) \leq_C \pi_i(\sigma_R(r_2))$, para cada $1 \leq i \leq |\sigma_R(r_1)|$,
- $\sigma_A : \mathcal{A} \rightarrow \mathcal{C} \times \mathcal{T}$ é uma função chamada de assinatura de atributo
- \mathcal{T} é um conjunto de tipos de dados, como *strings*, *inteiros*, *etc.*

$\pi_i(t)$ é o i -ésimo componente de uma tupla t .

E a definição formal para Domínio:

Para uma relação $r \in R$ com $|\sigma(r)| = 2$, definimos seu domínio por $\text{dom}(r) := \pi_1(\sigma(r))$.

Thomas C. Jepsen [36], em 2009, percebeu que o verdadeiro potencial das ontologias está na habilidade de criar relacionamentos entre classes e instâncias, além de atribuir propriedades para estes relacionamentos, o que nos permite inferir conclusões. Jepsen sugeriu também a utilização de um resolvidor (*reasoner*) para classificar automaticamente os indivíduos e conjuntos em um determinado domínio, colocando, por exemplo, se uma classe é subclasse de outra.

3.2.1 Ontologias OWL

OWL é uma linguagem utilizada para definição e instanciação das ontologias Web e foi definida pelo W3C². McGuinness e van Harmeien com o artigo [35] fundamentam o conceito. Dada uma Ontologia OWL, a semântica formal OWL irá determinar suas consequências lógicas, considerando fatos que não estão literalmente presentes na ontologia, mas vinculados pela semântica [35]. Os vínculos podem ser utilizados em um ou em vários documentos distribuídos que foram combinados através de mecanismos OWL[37].

²O consórcio W3C (*World Wide Web Consortium*) é uma comunidade internacional que desenvolve padrões abertos para garantir o bom crescimento da rede mundial de computadores: <http://www.w3.org/>

De acordo ainda com o McGuinness et al [35], a linguagem OWL subdivide-se em três sub-linguagens distintas que variam suas características de acordo com a aplicação a que se destinam: *OWL Lite* é uma sintaxe simples com capacidades de classificação; *OWL DL*, uma versão computacionalmente mais completa que suporta descrições lógicas; e *OWL Full*, versão que possibilita compatibilidade com RDF *schema*.

Retomando Jepsen [36], as representações visuais são fundamentais para apresentar as ontologias, um dos motivos é que uma ontologia pode conter relações complicadas para a expressão textual ou através da lógica formal. É mais fácil representar relações mais complexas em uma imagem que pode ser compreendida em um nível intuitivo. Linguagens visuais para representação de conhecimento geralmente utilizam nós ligados por arcos. Uma dessas formas é a rede semântica, um grafo que formado por nós que representam objetos ou conceitos em um domínio, e os arcos representam as relações entre os nós. Um método que se estrutura em forma de grafos é o RDF, tratado na próxima seção.

3.3 RDF

RDF (*Resource Description Framework*) é um arcabouço desenvolvido para representar informação na web. É um modelo de dados que representa informações sobre recursos, dentro disso, qualquer objeto pode ser um recurso (documentos, pessoas, conceitos abstratos, números e cadeia de caracteres). Sua sintaxe utiliza do conceito de tripla ³ na forma (**sujeito**, **predicado**, **objeto**), cada tripla representa a relação entre os recursos. E esta pode ser interpretada como uma entidade (**sujeito**) que tem uma propriedade (**predicado**) com um determinado valor (**objeto**)[3].

Primeiramente, uma compreensão sobre grafos para dar continuidade na presente seção. Com base na publicação de Netto [38], que traz uma abordagem didática sobre grafos. Um grafo é uma estrutura que permite várias maneiras de ser representada, mas para uma melhor definição, um grafo $G = (V, A)$, onde o V é conjunto de vértices e A um conjunto de arestas. Estas serão representadas por duplas daquelas. A figura 3.3

³Tupla é uma sequência ordenada de objetos, logo Tripla é uma sequência ordenada de três objetos.

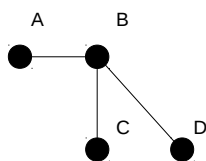


Figura 3.2: Exemplo de um Grafo $G = (V, A)$

mostra os conjuntos dentro de G :

$$V(G) = \{A, B, C, D\}$$

e

$$A(G) = \{(A; B), (B; C), (B; D)\}.$$

Então, se as arestas ligam os vértices, podemos apenas com a representação textual conseguir desenhar o grafo.

No caso de uma base RDF, os vértices serão os sujeitos e os objetos, enquanto as arestas serão os predicados que relacionam os vértices. Tal representação é dada através de um conjunto finito de tuplas. E se Timóteo quiser representar a sua coleção computacionalmente? Supondo que ele utilize o RDF, o que nos ajuda na compreensão dessa linguagem e também na pesquisa que aqui se apresenta. Timóteo com seu conhecimento computacional tentará passar parte de sua coleção para um grafo RDF. Para isso, ele terá que dar nome a cada vértice e aresta utilizados. No caso, Timóteo opta por distribuir parte de sua coleção para ver como alguns itens se relacionam. A ontologia dele é definida

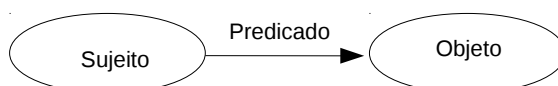


Figura 3.3: Um grafo RDF com dois nós [3]

por uma raiz \top de nome coleção. A partir dela saem três arestas Tipo que apontarão para os vértices livros, discos e filmes. A figura 3.4⁴ mostra que os vértices agem como sujeito e também momentos como objeto. Por exemplo: livros é objeto do sujeito coleção na tupla (coleção, tipo, livros) e sujeito perante o objeto 0 estrangeiro na tupla (livros, Albert Camus, 0 estrangeiro).

Um vocabulário RDF é uma coleção de IRIs (*International Resource Identifier*) (ver seção 3) destinada a ser utilizada em grafos RDF. Um deles é o RDF-*Schema* que possibilita a definir e documentar vocabulários RDF adicionais. RDF-*Schema* é uma extensão semântica com a qual é possível declarar classes, estabelecer propriedades por meio de triplas e também hierarquizar os elementos dentro de classes [39].

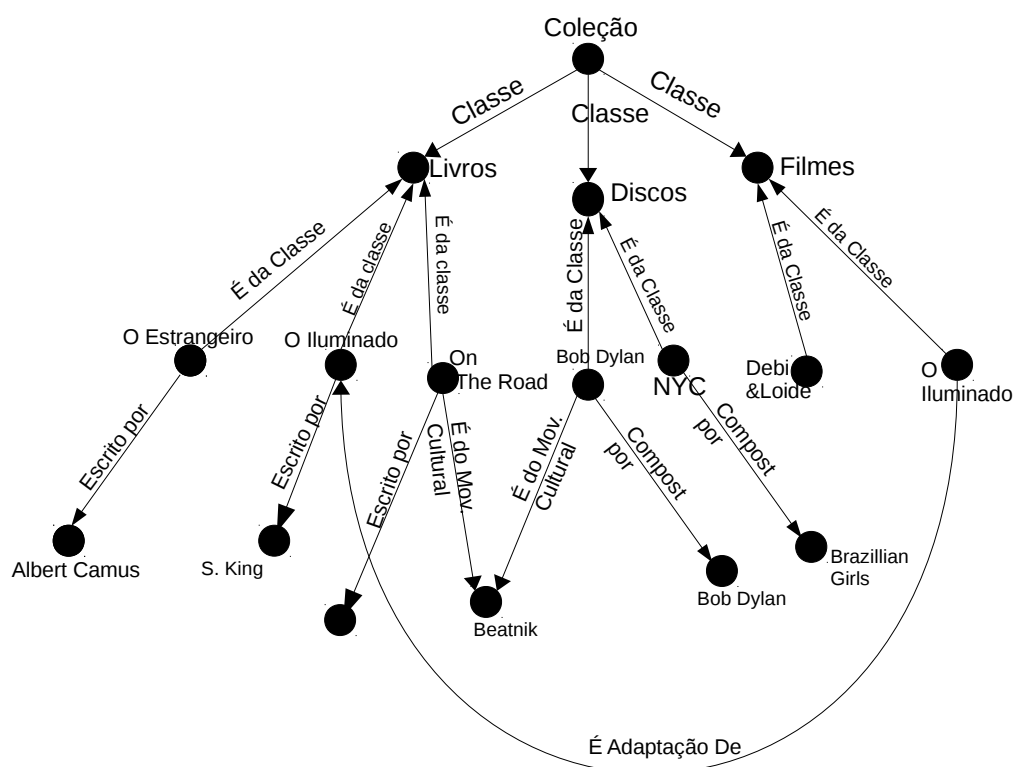


Figura 3.4: Exemplo de um grafo RDF (Coleção do Timóteo)

⁴Este grafo RDF segue as associações dadas na representação gráfica apresentada na figura 3.1.

3.3.1 SPARQL

Para fazer consultas nas bases RDF, utiliza-se a linguagem recomendada pelo W3C, o SPARQL, nome recursivo dado por *SPARQL protocol and RDF Query Language*. SPARQL utiliza de uma sintaxe baseada em triplas, assim como o RDF. Com o SPARQL 1.1 é possível elaborar perguntas com base no grafo RDF para consulta e também alterar o grafo [40].

Sintaticamente, a estrutura da linguagem se dá por três cláusulas: SELECT especifica como os resultados serão retornados; FROM determina qual, ou quais, base de dados pode ser pesquisada; e WHERE dará qual o padrão de grafo que será pesquisado. Cada uma dessas cláusulas se divide em algumas opções, que podem ser vistas em [40].

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT * WHERE
{ ?s ?p ?o
  }
LIMIT 10
```

3.4 Considerações

O capítulo atual trata sobre linked data, constando sobre ontologia, semântica web, RDF e busca em RDF. As ontologias permitem um esquema de indexação que contempla a similaridade entre os conceitos. Através de regras para combinar e classificar o que foi indexado em uma determinada estrutura.

A linguagem OWL é utilizada para definição e instanciação das ontologias *Web*. Sua semântica formal determina consequências lógicas, considerando fatos vinculados através daquela. De acordo com a aplicação a que se destina, a linguagem OWL é subdividida em três sub-linguagens: *OWL Lite*, *OWL DL*, *OWL Full*. Esta possibilita compatibilidade com a representação através do RDF.

O RDF é um modelo de dados que representa informações sobre recursos, com uma sintaxe que utiliza o conceito de triplas (**sujeito**, **predicado**, **objeto**), é um arcabouço utilizado em ferramentas que buscam similaridades através de inferências. O SPARQL

é o método utilizado para fazer consulta nas bases RDF (O SQL está para banco de dados, assim como SPARQL para RDF). São termos simples localizados, tecnicamente, em três cláusulas: SELECT, FROM, WHERE, é possível também determinar algumas particularidades.

Linked data é uma abordagem em discussão e sendo aplicada em algumas vertentes da Ciência da Computação. Berners-Lee crê que ela pode ser fundamental na formação de uma nova rede dados que se conectarão e poderão facilitar a elaboração de bases complexas. Esse presente trabalho pretende observar como utilizar tais dados conectados dentro de uma aplicação de resolução de problema baseado em casos. O próximo capítulo traz a utilização do RDF como método de representação de conhecimento para uma aplicação específica, o TUUURBINE.

CAPÍTULO 4

LEVANTAMENTO DE DADOS

Este capítulo contempla a coleta e tratamento dos dados elencados para a presente pesquisa. Apresenta o FARMA-ALG, ferramenta de apoio ao ensino de algoritmos, de onde os dados são coletados.

O capítulo mostra como os dados foram gerados pelo FARMA-ALG, a Seção 4.1 se dedica a descrever o desenvolvimento técnico e teórico do FARMA-ALG. E apresenta, na seção 4.2 a reestruturação dos dados para uma ontologia representada computacionalmente.

4.1 Farma-Alg

A tese de Kutzke [6] tem como objetivo “instrumentalizar professores e alunos na busca de mediação do erro durante o trabalho educativo”. Para isso, desenvolve o FARMA-ALG, um arcabouço de sistema para manipulação de registros de respostas, permitindo a professores e alunos o “armazenamento e o acesso facilitado a registro de erros”. O Farma-Alg é composto, basicamente, por dois componentes: uma base de dados e uma máquina de semelhança.

Esta seção é subdividida de forma a organizar três alicerces considerados relevantes para compreender o FARMA-ALG. O primeiro alicerce é o conceitual, tratado na subseção 4.1.1, trata sobre a mediação do erro, as fundamentações teóricas que o cerceiam e também como funciona o arcabouço em questão.

4.1.1 Mediação do Erro

Considera-se erro as respostas incorretas ou que não atendem às expectativas de quem determinou uma dada pergunta. Distintas concepções de erro foram desenvolvidas, e é necessário compreendê-las. Teorias não-críticas baseiam-se em um ideal de que a educação,

por ser independente das relações sociais, pode ser o caminho para trazer solução aos problemas da sociedade ([41] apud [6]). Kutzke [6] descreve três teorias não-críticas que trarão uma perspectiva sobre o erro: tradicional, escola nova e tecnicista. Cada uma das perspectivas traz uma concepção sobre o erro.

A pedagogia tradicional trata o erro como um empecilho que deve ser extirpado do processo, não há correção das falhas. Na perspectiva escolanovista, a qual considera que o processo de aprendizagem deve se sobrepor ao conteúdo ([41] apud [6]), o erro é um indicador de que o aluno não está pronto. A Teoria Tecnicista, que se apoia nos pressupostos da neutralidade científica, a presença do erro indica falha, ineficácia, improdutividade. Kutzke [6] com base nos preceitos da Psicologia Histórico-Cultural observa o erro como objeto de estudo tanto para professores como para alunos. Onde a compreensão do trajeto do desenvolvimento psíquico específico do aluno é primordial para então interpretar o erro não como um desvio do objetivo educacional, mas sim parte do caminho para a compreensão pedagógica desse.

Kutzke [6] afirma que “a importância do erro no processo de ensino e aprendizagem, de acordo com a abordagem histórico-cultural, se funda justamente nas diferenças encerradas entre os desenvolvimentos de conceitos espontâneos e científicos”. Compreende-se que o desenvolvimento de conceitos científicos não possui um início e nem um fim específico, isto é, perante uma mediação apropriada, está sempre acontecendo. Como não há assimilação de um conceito em sua forma acabada, “o erro é parte integrante e fundamental do processo de formação de conceitos científicos” (idem). Kutzke considera que o erro deve ser *mediado*, tornando-o objeto de uma atividade de ensino, uma parte do processo de assimilação de conceitos. Considerando a pedagogia histórico-crítica, percebe-se uma demanda por ferramentas que auxiliem os professores na análise de erros dos alunos, por isso Kutzke propõe um arcabouço de mediação de erro na educação, tratado na seção 4.1.2.

4.1.2 FARMA-ALG: Arcabouço de Sistema para Auxílio à Mediação do Erro

Identificando carências encontradas na literatura no que concerne a ferramentas que auxiliem professores na mediação do erro, Kutze e Direne em [42] propõem um arcabouço de sistema para manipulação de registro de respostas baseado em suas relações de similaridade. Kutze e Direne [42] Argumentam “que um arcabouço de sistema que contenha o armazenamento automático, a recuperação, a visualização e a manipulação de registros de respostas pode promover a mediação do erro”. São quatro funcionalidades fundamentais dadas pelo arcabouço em questão, listados e explicados a seguir.

1. **Armazenamento automático dos registros de resposta** tenta evitar que respostas de alunos sejam perdidas ou ignoradas. A base de dados possui todas as repostas e a máquina de semelhança verifica a proximidade entre as respostas. A partir destas relações geradas pela máquina e também as definições do professor, o sistema gera um grafo de similaridade. Onde os vértices representam as respostas e as arestas que as ligam possuem um peso que vai de acordo com o grau de semelhança. Em cima desse grafo é que a estrutura de dados base do arcabouço se forma.
2. A **Manipulação dos registros de resposta** “se caracteriza pelas operações de categorização de respostas e inserção e remoção de relações entre respostas”. O professor insere ou retira as relações entre as respostas, inserido as quais não foram atribuídas pela máquina e retirando as que foram feitas erroneamente. A partir de tais manipulações, o sistema agrega subsídios para realizar generalizações automáticas, processo denominado por *propagação de tags*.
3. **Recuperação facilitada dos registros de resposta** é feita através da busca por palavras-chave e meta-dados como nome do aluno, turma, exercício, data e horário da reposta. O que possibilita a busca por dados específicos que sejam do interesse de ambos em prol da mediação do erro. O sistema infere respostas e questões relevantes

para um grupo de alunos. Através das similaridades entre as respostas, o sistema sugere semelhanças entre alunos, agrupando-os.

4. **Visualização dos registros de resposta** que possibilita dois tipos de visualização das respostas armazenadas. Uma onde o professor pode visualizar o grafo de similaridade e nele realizar operações de manipulação. E também é possível professores e alunos visualizarem as respostas sobre uma linha do tempo, ou seja, em uma ordem cronológica.

Kutzke e Direne [42] citam o sistema FARMA (Ferramenta de Autoria para a Remediação de erros com Mobilidade na Aprendizagem) [43] como base à aplicação do arcabouço proposto. Eles apresentam um arcabouço direcionado para o ensino de programação de computadores. Embora o FARMA não “realiza nenhum processamento sobre os registros de erro armazenados (...) a ferramenta apresenta uma boa forma de autoria de Objetos de Aprendizagem aos professores” [42]. Cria-se um sistema avaliador capaz de determinar se uma resposta está correta ou não, no contexto da implementação proposta, as respostas são códigos-fonte de programas.

“A semelhança entre duas respostas é baseada em vários fatores, entre eles: grau de semelhança (de texto) entre a saída obtida para cada conjunto de testes e a saída esperada; grau de semelhança entre as saídas obtidas para cada caso de teste; grau de semelhança entre o código-fonte das respostas” [42].

A participação do professor na máquina de semelhança se dá através de um grafo de similaridade que o professor sugere a entrada ou retirada de arestas entre os nós.

4.2 Levantamento de dados

Esta pesquisa utiliza um conjunto de dados levantado através da pesquisa desenvolvida por Kutzke em [6]. Os dados foram compostos através de 80 alunos, que responderam 23 exercícios, obtendo-se 3723 respostas. Esta seção apresenta a mineração destes dados

e, a partir deles, a criação de uma ontologia que considerou erros e acertos dos alunos. Hipoteticamente, um resolvidor baseado em casos pode ser útil na recomendação de conteúdos, considerando as demandas apresentadas na Introdução e o objetivo na seção 1.1, foi criada uma base de casos a ser utilizada em um sistema CBR.

O modelo de dados do FARMA-ALG contempla classes pertinentes a um arcabouço de ensino, entre elas: respostas, usuários, turmas, recomendações, exercícios e comentários. O sistema avaliador trata sobre armazenamento de dados através das respostas e, as similaridades são geradas de acordo com a mediação dos erros pelo professor, conforme visto na seção 4.1. Os dados do FARMA-ALG foram gerados no formato JSON¹ (abreviatura para *Java Script Object Notation*).

JSON é um formato de texto composto por uma coleção de pares (chave, valor), sendo que cada chave só pode aparecer uma vez em cada coleção. Cada coleção é um objeto em JSON, o objeto inicia e termina com chaves (`{ }`). Na estrutura estarão as chaves que são relacionadas com seus valores através dos dois pontos (`:`). Os pares são separados por vírgulas. Vetores aparecem entre colchetes (`[]`) Exemplo de um objeto em JSON:

```
1 { 'chave_1': 'nome', 'chave_2': [0, 1, 2, 3], 'chave_3': { 'chave_1': [0, 2] } }
```

A primeira chave (`'chave_1'`) do objeto é uma *string* que tem um valor `'nome'` como seu par. A segunda chave já tem um vetor como valor. E a terceira chave (`'chave_3'`) tem um outro objeto.

Torna-se necessário mapear esses dados e construir uma estrutura em ontologias. O banco é formado por 18 arquivos no formato `“.json”` como `answers.json` com 3723 objetos; `users.json` com 80 objetos; `exercises.json` com 23 objetos. Cada objeto terá um identificador único.

A partir dos dados do FARMA-ALG, códigos em Python² mapearam os dados para formar uma ontologia na semântica OWL (ver seção 3.2). Uma entrada contém todas as repostas dadas pelos alunos. Cada resposta contém, além da resposta escrita em linguagem de programação, o título e a descrição do exercício; o aluno que a respondeu; a data de criação; os testes que verificam a assertividade do código; entre outros dados.

¹<http://rfc7159.net/rfc7159>

²<https://www.python.org/>

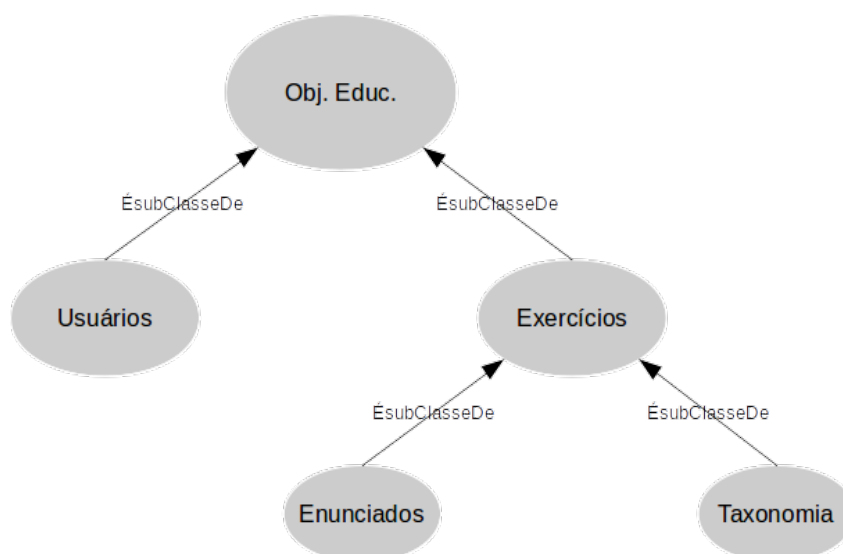


Figura 4.1: Ontologia de Objetos Educacionais

O arquivo de entrada já contribuiu na construção de uma ontologia que contempla como classe principal, ou raiz da ontologia a classe “Objetos Educacionais” e como classes subjacentes: “Usuários” e “Exercícios” (Figura 4.1). A classe de usuários é subdividida de acordo com o que a máquina de semelhanças dadas no trabalho de Kutzke em [44]. No presente trabalho, será dedicado à classe “Exercícios” e uma forma de classificá-los no que tange à dificuldade. Os exercícios são divididos em duas subclasses conceituais:

- Enunciados: onde são posicionados os títulos em pares com os respectivos enunciados dos exercícios.
- Taxonomia: agrupamento que contém ontologias de classificação dos exercícios no que tange à dificuldade de resolução do exercício.

A Figura 4.2 ilustra como esta classe se divide nas duas subclasses. Considerando que aqui se espera perceber a contribuição da mediação do erro no processo de aprendizado, o foco é dado nos erros dos alunos para os exercícios. Um algoritmo para classificação de erros foi feito e ele é apresentado na subseção seguinte.

4.2.1 Ontologia com base nos erros

Nesta subsecção é apresentada a classe conceitual que foi estruturada durante a pesquisa. Além da apresentação formal da ontologia, é apresentado o funcionamento do algoritmo e o código em Python que minera os dados. É importante compreender que a classe “Errados” é a raiz da ontologia, desconsidera-se, funcionalmente, as outras classes já citadas e dar-se-á a formação da ontologia que é a estrutura conceitual da base de casos utilizada no experimento.

A ontologia agrupa exercícios de acordo com o número de erros em suas respostas. Na pesquisa de Kutzke [6] a similaridade entre as respostas é mediada pelos professores. No presente trabalho, a proximidade entre os exercícios é dada automaticamente, sem a necessidade de um especialista.

Os exercícios foram divididos em quatro classes. São quatro níveis de exercícios considerando a quantidade de erros no estabelecimento nível de dificuldade. Aqui não se considera, ainda, a curva de aprendizagem, apenas é ajustado o foco na quantidade de

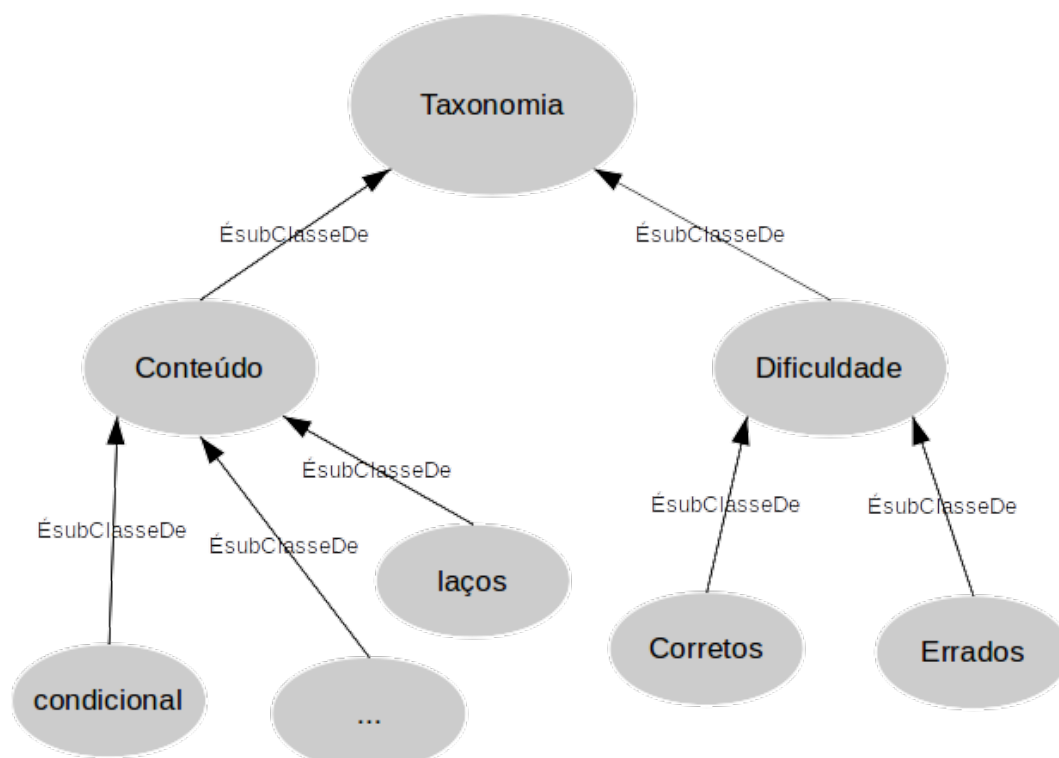


Figura 4.2: Ontologia das Taxonomias

```

1     def make_dict_of(key, list_, bool_):
2         dict_ = {}
3         for d in list_:
4             if (d['correct']) == bool_:
5                 if d[key][id] in dict_:
6                     dict_[d[key][id]] = dict_[d[key][id]] + 1
7                 else:
8                     dict_[d[key][id]] = 1
9         return dict_
10    dict_wrong = make_dict_of('question_id', list_answer, False)
11    dict_correct = make_dict_of('question_id', list_answer, True)

```

Figura 4.3: Função do código que constrói um dicionário de exercícios e a quantidade de respostas erradas.

vezes que um exercício foi errado. Há, então, um equilíbrio para cada lado da ontologia e que isso pode contribuir nos testes do sistema CBR. A estrutura pode ser ainda mais complexa, com mais níveis hierárquicos e classes, porém, como se trata de um universo de 23 exercícios se torna cauteloso trabalhar com uma ontologia reduzida.

As respostas possuem similaridade entre elas, conforme apresenta Kutzke em [44], porém, aqui é dada a importância para aproximação entre os exercícios devido a quantidade de erros dos mesmos, ou seja a similaridade é dada pelo número de vezes que um exercício foi respondido erroneamente.

O algoritmo de classificação após carregar o arquivo de entrada (*answers.json*), cria-se uma lista contendo 3723 objetos. Uma função cria um dicionário que contará a quantidade de erros ou acertos de cada questão. A função precisa de três parâmetros: **key** é a chave do elemento que será contabilizada; **list_** é a lista onde estão representadas as respostas e um valor booleano, em caso de **True** considera-se as questões corretas, em caso de **False** considera-se as questões erradas. A Figura mostra a função que faz a construção do dicionário que contabiliza a quantidade de erros e acertos. Entre as linhas 5 e 8, especificamente, acontece a contagem de ocorrência dos erros.

Após essa atribuição, a variável `dict_wrong` é do tipo `dict` com pares contendo identificador do exercício como chave e o valor é a quantidade de vezes que o exercício foi errado. Pode-se representar essa estrutura temporária, com a devida substituição dos IDs pelos títulos dos exercícios, parcialmente, por `dict_wrong := {'Multiplicação': 104, 'Conversão_de_velocidade': 130, 'Quadrado_mágico': 101, 'Números_repetidos': 271,...}`.

```

1  def make_low(average, dict_):
2      low_dict = {}
3      k = 0
4      while k < len(dict_.keys()):
5          if dict_[dict_.keys()[k]] <= average:
6              low_dict[dict_.keys()[k]] = dict_[dict_.keys()[k]]
7              k = k + 1
8          else:
9              k = k + 1
10     return low_dict

```

Figura 4.4: Função que retorna dicionário contendo exercícios e a quantidade de respostas erradas abaixo de uma média dada como parâmetro

Faz-se uma média (`average`) que contém exercícios que tem erros abaixo da média de erros, somando o total de erros ($104 + 130 + 101 + 271 + \dots$) e dividindo pela quantidade de exercícios do dicionário (no caso, 23). Havendo esse corte, formam-se dois objetos: um dicionário contendo exercícios com quantidade de respostas erradas acima da média calculada (`high_dict`) e um dicionário contendo exercícios com quantidade de respostas erradas abaixo da média calculada (`low_dict`).

Há duas funções que distribuem os exercícios em dicionários. A Figura 4.4 mostra a função que cria um dicionário contendo questões com uma quantidade de respostas erradas abaixo da média calculada. Utiliza-se as mesmas funções para sub-dividir os dois dicionários resultantes, porém agora de acordo com a média de cada um.

Através de atribuições que criam a base de conhecimento, os exercícios se transformam em instâncias de classes. A figura 4.5 representa a estrutura. Formalmente, a ontologia \mathcal{O} dos erros possui um conjunto C de conceitos: $C := \{ Errados, Alta, Baixa, Alta_alta, Alta_baixa, Baixa_alta, Baixa_baixa, Id \}$. A classe *Errados* é considerada a raiz da ontologia. O conjunto R de relações é: $R := \{ \acute{e}SubClasseDe, \acute{e}InstânciaDe \}$. E o conjunto A de atributos: $A := \{ título, enunciado \}$. As relações e atributos são assinaladas, formalmente, assim:

$$\sigma_R(\acute{e}SubClasseDe) = \{(Alta, Errados), (Baixa, Errados), \dots, (Baixa_baixa, Baixa)\}$$

$$\sigma_R(\acute{e}InstânciaDe) = \{(Id, Alta_Alta), (Id, Alta_baixa)\}$$

$$\sigma_A(título) = (Id, string) \sigma_A(enunciado) = (Id, string)$$

A ontologia construída é representada computacionalmente utilizando da semântica

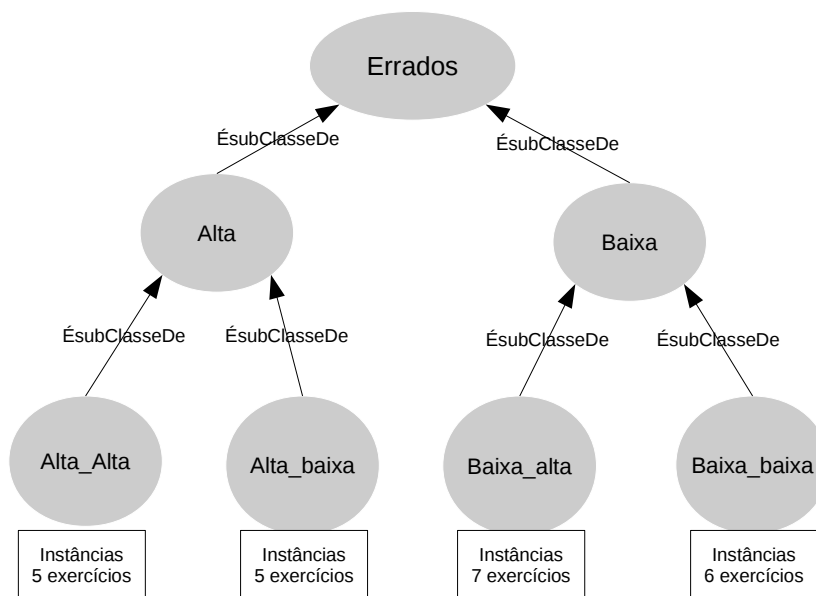


Figura 4.5: Ontologia dos exercícios divididos de acordo com os erros

```

1 <?xml version="1.0" encoding="UTF-8"?>
2   <rdf:RDF
3     xmlns:property="http://cbr.c3sl.ufpr.br/farma/Properties/"
4     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
5     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
6     <rdf:Description rdf:about="http://cbr.c3sl.ufpr.br/farma/class/
7       question/Graus,_minutos_e_segundos">
8       <property:isTitleOf rdf:resource="http://cbr.c3sl.ufpr.br/farma/
9         class/question_id/530243f03cc450d130000004"/>
10      </rdf:Description>
11      <rdf:Description rdf:about="http://cbr.c3sl.ufpr.br/farma/class/low">
12        <rdfs:subClassOf rdf:resource="http://cbr.c3sl.ufpr.br/farma/class
13          /question"/>
14      </rdf:Description>

```

Figura 4.6: Uma amostra da representação computacional (XML/RDF) da Ontologia construída.

web tratada no capítulo 3. Concebe-se arquivos em uma estrutura XML/RDF constando as relações entre as classes com a propriedade `subClassOf`; entre as instâncias - `isTitleOf`, `hasTitle` e `hasContent`; e das instâncias com as classes somente com a propriedade `type`. A Figura 4.6 apresenta parte da estrutura codificada.

Após essa estruturação dos exercícios, inicia-se a construção de uma base de casos para a execução dos testes. Essa base de casos conterà listas de exercícios distintas e a

estrutura apresentada aqui dá a similaridade entre eles. O próximo capítulo explica como se dá a concepção da base, a validação dos testes e a execução dos experimentos.

4.3 Considerações

Esse capítulo almeja contextualizar o leitor nos dados utilizados na presente pesquisa. Sistemas de apoio ao ensino são sistemas ainda em pleno avanço no seu campo de pesquisa. Um desses sistemas é o FARMA-ALG [6], ele é desenvolvido para dar apoio ao ensino de programação de computadores à luz da Psicologia Histórico-Cultural e utiliza da mediação dos erros através da participação do professor.

Tendo o erro como fator fundamental no desenvolvimento de conceitos científicos por estudantes, concebe-se uma abordagem que considera o erro determinante na elaboração dos dados para análise. Utilizando a base do FARMA-ALG, o leitor entra em contato com a reestruturação da Ontologia que organiza e aproxima exercícios através dos erros.

No FARMA-ALG, a similaridade entre os exercícios é dada por meio da participação do Professor, conferindo caráter qualitativo à análise. Complementando com método apresentado por Kutzke [6], o presente trabalho propõe similaridade através da quantidade de vezes que um exercício foi errado, uma análise quantitativa.

CAPÍTULO 5

SISTEMA CBR COMO RECOMENDADOR DE CONTEÚDO PEDAGÓGICO

Experimentar o CBR na recomendação de lista de exercícios com dados estruturados em ontologias é o objetivo desta pesquisa e no presente capítulo será feito o relato desse experimento. Descreve como o sistema CBR determinado para os testes é utilizado com os dados apresentados no capítulo 4. Torna-se necessário aqui compreender como é o funcionamento do sistema utilizado para o experimento: o myCBR. E relata como foram dadas as consultas, apresentando limitações dadas e possíveis resultados.

Na seção 5.1 intenta-se explicar ao leitor como os dados apresentados no capítulo 4 foram estruturados para se adequar ao método de raciocínio baseado em casos. Descreve-se como foi planejado e validado o experimento na seção 5.3. A seção 5.2 apresenta as características e limitações encontradas no myCBR e, por consequência, na execução dos experimentos. A seção 5.4 relata os resultados obtidos e também aborda sobre as limitações encontradas. E a seção de fechamento constando as conclusões diante do experimento.

5.1 Representação de Conhecimento em CBR

Um artigo que trata sobre modelagem e manutenção de conhecimento no myCBR [20] é utilizado como primeira referência para a presente seção. E, à luz de Richter [12], explana-se alguns conceitos que demandam observação na elaboração dos dados utilizados para o experimento.

Um conjunto de dados estruturados e operações de inferências é conhecido por base de conhecimento [12]. É possível representá-lo em estruturas específicas que são adequadas para determinados tipos de conhecimento, nomeados por containers de conhecimento, brevemente apresentados na seção 2.1: Vocabulário, Similaridade, Base de Casos,

Adaptações. Para estruturar os dados no conhecimento passível de leitura por resolvedores baseados em casos é preciso estruturá-los em quatro containers. As subseções a seguir trazem especificações formais sobre os quatro containers.

5.1.1 Vocabulário

O vocabulário se dá através da definição dos conceitos, atributos e a coleção de valores permitidos. Podem ser atributos numéricos ou uma lista de símbolos permitidos [20]. A representação do atributo-valor é considerada por Richter [12] como uma estrutura importante em CBR. E, conforme o autor, um atributo consiste em:

- um nome A
- um conjunto finito $DOM(A)$ com o conjunto de valores de A .
- uma variável x_A

Determinados valores podem ser utilizados para um atributo A . Caso esses valores sejam apenas números, o atributo A é chamado de numérico e simbólico se é determinado por um conjunto finito e arbitrário de valores. Os valores determinados para um atributo A podem ser relacionados através de graus de similaridades.

5.1.2 Métrica de Similaridade

Uma recuperação de casos em uma base de dados só se dá através de buscas exatas. A *similaridade* é o ponto fundamental para recuperação de buscas inexatas. Segundo Richter [12], a similaridade pode ser formalizada através de uma abordagem relacional que utiliza uma relação R de quatro elementos $R(x, y, u, v)$ que diz que x e y são tão similares quanto u e v são. O que, de acordo com Richter [12], permite a definição da noção do *vizinho mais próximo*:

$$NN(x, z) \Leftrightarrow \forall_y R(x, z, x, y)$$

onde z é o vizinho mais próximo de x . Se o vizinho mais próximo é único, então NN é utilizado para simbolizar a função. A relação de similaridade é dada como uma ordenação parcial. As ordenações parciais podem ser percebidas por funções numéricas chamadas *Métricas de Similaridades* $sim(a, b)$ e duplamente de *Distânciadist* (a, b) . “Se $R_{sim}(x, y, u, v) \Leftrightarrow R_{dist}(x, y, u, v)$, sim e $dist$ são chamados de compatíveis” [12].

Cimiano[33] traz definições sobre similaridade e distância, colaborando com o presente estudo: *Métrica de Similaridade é uma função $sim : \mathbb{R}^n \times \mathbb{R}^n \rightarrow [0, 1]$ desde que atenda as condições:*

$$\forall \mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^n, sim(\mathbf{v}_1, \mathbf{v}_2) = 0 \Leftrightarrow \mathbf{v}_1 \cdot \mathbf{v}_2 = 0$$

$$\forall \mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^n, sim(\mathbf{v}_1, \mathbf{v}_2) > 0 \Leftrightarrow \mathbf{v}_1 \cdot \mathbf{v}_2 > 0$$

$$\forall \mathbf{v} \in \mathbb{R}^n, sim(\mathbf{v}, \mathbf{v}) = 1$$

A primeira condição determina que a similaridade entre dois vetores é zero caso não exista nenhuma característica em comum. Logo, a similaridade será maior que zero caso eles tenham ao menos uma característica em comum. A última propriedade aponta que um vetor é totalmente similar a ele mesmo.

Cabe também a definição de Métrica de Distância: *Métrica de Distância é uma função $d: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}_0^+$ que atenda a seguinte condição:*

$$\forall \mathbf{v} \ dist(\mathbf{v}, \mathbf{v}) = 0, \text{ significa que a distância de um vetor dele mesmo é } 0.$$

Uma métrica de distância $dist$ pode ser transformada para uma medição de similaridade sim por uma função decrescente bijetiva e monótona $f(dist)$. É necessário que a função de transformação de distância para similaridade atenda as condições [33]:

$$dist(x, y) = +\infty \Leftrightarrow sim(x, y) = 0$$

$$dist(x, y) = 0 \Leftrightarrow sim(x, y) = 1$$

A distância entre dois elementos x e y tende ao infinito se e somente se a similaridade entre eles for igual à zero. A distância entre dois elementos x e y é 0 se e somente se a similaridade entre eles for igual à 1.

Na determinação da função de similaridade de acordo com a distância é sugerido a

determinação de uma distância máxima *maxdist*.

Richter [12] traz que o container com as similaridades pode armazenar conhecimentos mais ou menos sofisticados sobre classe de problemas. As similaridades podem ser de superfície, que consideram apenas propriedades sintáticas da representação. *Similaridades locais* tratam apenas dos valores de um único atributo. *Similaridade global* são as similaridades entre os casos e podem incorporar métricas locais, onde a relação entre atributos pode ser dada por pesos mas também a posição relativa em uma hierarquia.

5.1.3 Base de Casos

Um container de informação dedicado a armazenar os casos pode ser chamado de base de casos. Considera-se a representação de caso, em [12], por

(problema, solução).

Uma *base de casos CB* é um conjunto de casos que na proposta de melhorar a recuperação “é comumente equipado com estruturas adicionais; bases de casos estruturadas existem também sob o nome de *memória de casos (case memory)*” [12].

A recuperação é uma operação básica tanto em bases de casos como de dados. Enquanto na base de dados a consulta recupera respostas exatas, uma consulta faz uma busca por soluções utilizando respostas inexatas [12]. Ainda de acordo com Richter, como nas bases de dados, estruturar em árvores colabora na melhoria da recuperação.

5.1.4 Adaptação

Um container de conhecimento utilizado também em CBR é o de *Adaptações*. Já tratado no capítulo 2, adaptação de um caso é o ato de fazer as mudanças apropriadas na solução do caso. O container de adaptações pode ser composto por regras de adaptação.

Dada a compreensão de que os quatro containers se estruturam dentro do ciclo CBR, o sistema CBR que utiliza esse método de resolução de problemas poderá, finalmente, recuperar casos e possibilitar adaptações. A próxima seção utiliza das formalidades e conceitos

tratados nas subseções anteriores para mostrar como os dados podem ser estruturados no myCBR 3.

5.2 Representação de conhecimento no myCBR 3

O objetivo da presente seção é apresentar como se dá a formalização do domínio de conhecimento para o sistema escolhido. Intenta-se que o leitor possa entender a escolha do myCBR diante de outros sistemas vistos no capítulo 2.

Considera-se três sistemas para os experimentos da presente dissertação: TUUURBINE, jColibri e o myCBR. Todos os três foram instalados em máquinas com os requisitos determinados nas respectivas documentações. O TUUURBINE não funcionou através do *web service*, embora os dados para a presente pesquisa tenham sido modelados na representação em RDF exigido pelo sistema, possibilitando apenas uma consulta por correspondente exato através do SPARQL, o que pouco colabora com o presente experimento. O jColibri e o myCBR foram comparados em [45], donde se retira a conclusão de que o jColibri é um arcabouço que pode ser utilizado como base para aplicações complexas utilizando várias bases de dados, demandando conhecimento avançado em programação e também tempo para desenvolvimento da aplicação. A plataforma myCBR, de acordo com Atanassov *et al* [45], é indicada para aplicações não complexas, e contra indicado para aplicações com um número muito alto de atributos com soluções em texto. Diante de uma quantidade de atributos pequena, onde se interessa mais pelos resultados das consultas de acordo com a similaridade, optou-se pelo myCBR.

E a plataforma utilizada é o myCBR *Workbench* na versão beta 3.1 que agiliza modelar o conhecimento para CBR, especialmente os containers de vocabulário e de métrica de similaridade. São três containers de conhecimento possíveis na plataforma: base de casos, vocabulário e métrica de similaridade. Embora o artigo [20] cria uma expectativa de que uma versão atualizada possibilitaria adaptações, não foi encontrada nenhuma versão com essa propriedade.

A figura 5.1 mostra do lado esquerdo qual projeto está aberto, os conceitos que contém os atributos e nestes estão relacionados o conjunto de valores possíveis. No lado esquerdo

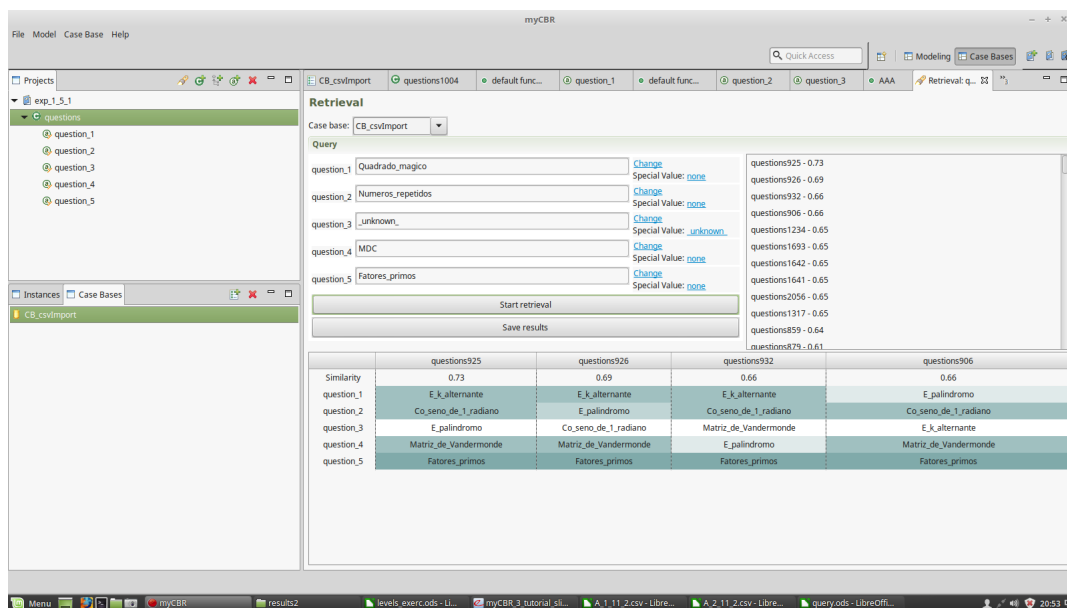


Figura 5.1: Interface do myCBR 3.1 Workbench

inferior estão as instâncias e as bases de casos. A parte central é a principal e aparece o que o usuário determina, no momento dessa captura de imagem estava sendo feita uma recuperação de casos.

O *Vocabulário* pode ser dado por *símbolos* ou *valores numéricos* que cada atributo pode aceitar. Os valores possíveis são adicionados manualmente. No caso de valores numéricos, determina-se o intervalo aceito.

A *Similaridade* pode ser dada tanto entre os conceitos, quanto locais, ou seja, entre os valores dos atributos. De acordo com o Tutorial, seguem os modos de edição para métricas de similaridade no myCBR¹:

Tabela é o modo padrão de edição da similaridade entre dois valores que se dá através de, inicialmente, uma tabela com linhas e colunas constando os valores do vocabulário. Inicia-se com uma matriz identidade, onde a diagonal principal contém o valor 1 para representar a similaridade entre os mesmos valores $sim(x, x) = 1$. Pode-se determinar a relação de similaridade entre os outros elementos colocando valores no intervalo $[0,1]$.

Taxonomia é o modo de edição por hierarquia entre os elementos somente possível

¹De acordo com o tutorial de Laura Zilles disponível em <http://mycbr-project.net/tutorials.html>

de utilizar apenas em símbolos, onde conjuntos no mesmo nível são disjuntos; nós no último nível são objetos reais.

Modo avançado é possível determinar detalhes para os valores dos tipos `integer` e `float`, é indicado no caso da similaridade do modo padrão (*Tabela*) não seja suficiente para a função pretendida pelo usuário.

Similaridade Global é o modo de visão geral dos atributos, possibilitando colocação de peso de um atributo perante outro, determinar o modo de similaridade utilizado para cada atributo. A similaridade global pode ser calculada utilizando quatro abordagens: soma de pesos, euclidiano, mínimo, máximo.

A *Base de casos* pode ser criada manualmente na própria ferramenta ou através da exportação de um arquivo `.csv` contendo as instâncias, ou casos. A criação manual se dá através da indicação de um base de casos e da inclusão de instâncias no mesmo. As instâncias podem ser determinadas através do preenchimento de cada atributo e de acordo com o valor previsto no vocabulário. Na exportação, os casos deverão estar com os valores permitidos no vocabulário.

A *Recuperação* é dada através de uma consulta (*query*) constando valores determinados para os atributos. Caso um atributo consultado não é conhecido, é dado o valor `_unknown_` a ele. Caso seja indiferente o atributo na recuperação, deveser dado o valor `_undefined_`. A recuperação dada através da similaridade entre os elementos e, por consequência, a similaridade entre os casos. Ela é dada por uma lista ordenada com o caso mais similar à consulta no topo e com menor grau de similaridade em baixo.

Após as medidas de similaridade necessárias para o domínio de conhecimento e a base de dados minimamente estruturadas, já é possível dar início à Recuperação. Necessita-se, então, determinar em qual base ocorrerá a consulta. Na interface é possível preencher os campos com os valores pré-determinados no vocabulário.

5.3 Descrição do experimento

Os dados apresentados na seção 4.2 são estruturados nos containers de conhecimento do myCBR. O vocabulário é composto pelos nomes dos exercícios e utiliza-se a ontologia que determina a distância e, por consequência, a similaridade entre os elementos.

A subseção 5.3.1 apresenta quais os valores determinados para cada atributo. A medida de similaridade entre os exercícios é apresentada na subseção 5.3.2. A base de casos e sua concepção é apresentada na subseção 5.3.3. Recomendam-se consultas que possam auxiliar na tirada de conclusões na recuperação de casos que serão tratadas na subseção 5.3.4.

5.3.1 Vocabulário dos exercícios

Para se ter uma lista de exercícios é necessário criar cinco atributos distintos, um para cada questão. Diante do fato de que o myCBR não possibilita consultas que diferenciem os valores possíveis na busca de similaridade, opta-se por uma lista formada por cinco atributos, sendo então um conjunto conceitual

$$questions = \{question_1, question_2, question_3, question_4, question_5\}.$$

Todos os atributos possuem o mesmo conjunto de valores previstos,

$$question_n = \{Aniversario, Co_seno_de_1_radiano, \dots, Conversao_de_Temperaturas\}.$$

Os valores são do tipo símbolos. O $DOM(question_n)$ de cada atributo contém os 23 exercícios.

Cada atributo demanda uma função de similaridade entre os valores. A próxima subseção trata da relação entre os elementos no que tange à similaridade.

5.3.2 Similaridade entre exercícios

Cada atributo tem a mesma função de similaridade. Como na base de casos os exercícios podem aparecer em qualquer campo das questões, opta-se por ter uma mesma

função de similaridade. Conforme apresentado na seção 5.2, a função de similaridade entre os elementos pode ser dada por uma tabela que determina o quão similar é um símbolo a outro.

Para compreender a relação de similaridade entre os exercícios, é necessário utilizar a ontologia apresentada no capítulo 4, pois é através dela que a tabela de similaridade será construída. O grafo que estrutura a relação entre os elementos dará a distância entre um elemento e outro.

A distância d entre dois elementos x e y é dada pela soma dos pesos w das arestas passadas no grafo. Dado pela fórmula

$$d(x, y) = \sum_{i=1}^n w_i$$

onde i é o valor dado cada vez que se passa por uma aresta, iniciando em 1 até chegar ao último vértice, ou seja, o próprio y . Se saindo de x para chegar em y visitam-se 4 vértices (sem contar o x como um deles), passando por 4 arestas, então $n = 4$. O peso w_i é dado em cada aresta.

Considera-se, para a ontologia em questão, que cada aresta terá o peso de 0,125. Trivialmente, ao final da caminhada, multiplica-se o número de vértices visitados pelo peso padrão w . Considerando que w tem um peso único, a distância pode ser representada por:

$$d(x, y) = \text{vértices_visitados} \times w$$

Por exemplo, para achar a distância entre os exercícios ‘Conversão de Temperaturas’ e ‘União’, de acordo com a Figura 5.2 Tendo-se um contador v , que dará o número de vértices visitados, é preciso considerar a visita em **Alta_Alta** ($v = 1$), posteriormente a visita de **Alta** ($v = 2$), **Alta_Baixa** ($v = 3$) e o vértice ‘Uniao’ ($v = 4$). Tendo que o número de vértices visitados é dado por $v = 4$, o peso w é igual à 0,125, tem-se a distância $d = 4 * 0,125$, ou seja, $d(\text{‘Conversao.de.Temperaturas’}, \text{‘Uniao’}) = 0,50$.

Para encontrar a similaridade sim entre dois elementos x e y , determina-se uma função

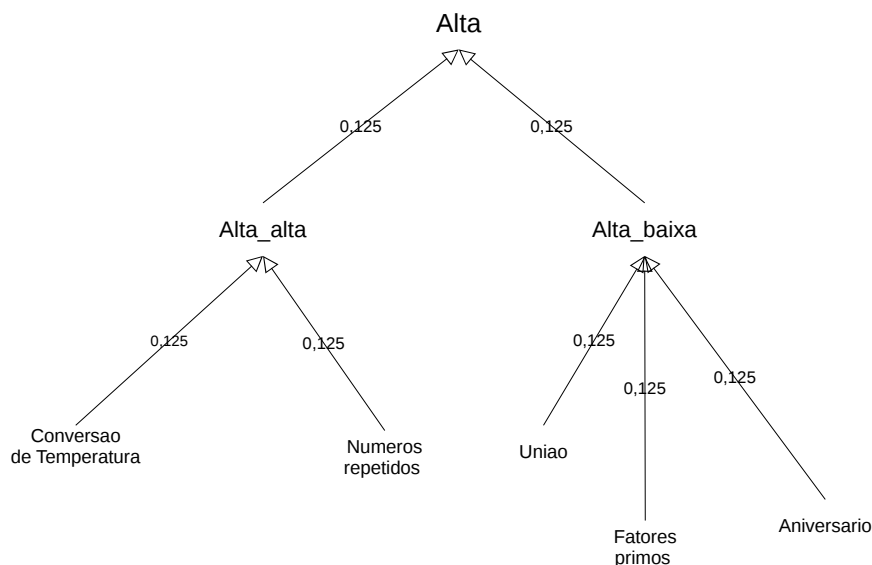


Figura 5.2: Amostra da Classe Alta da Ontologia

de transformação, segundo Cimiano [33]:

$$sim(x, y) = k - d(x, y)$$

onde k é uma constante adequada.

Para os dados aqui trabalhados, considera-se $k = 1$. Para achar a similaridade entre os exercícios 'Fatores_primos' e 'Aniversario': tendo que a distância d é dada por $d('Fatores_primos', 'Aniversario') = 0,25$, (figura 5.2) logo, a similaridade é dada por $sim('Fatores_primos', 'Aniversario') = 1 - 0,25 = 0,75$.

Para determinar a similaridade de cada atributo no myCBR 3 é dada uma tabela com as similaridades entre os exercícios com linhas e colunas constando todos os exercícios do vocabulário. A Figura 5.3 apresenta parte da tabela construída para alimentar a função padrão no myCBR 3. Os exercícios mais próximos, que pertencem a uma mesma sub-classe, têm similaridade de 0,75, o mínimo de similaridade entre dois exercícios é 0,25.

	Conversão_de_temperaturas	Números_repetidos	Co-seno_de_1_radiano	Graus_minutos_e_segundos	União	Aniversário	Fatores_primos	É_palindromo?	Conversão_de_velocidade
Conversão_de_temperaturas	1.00	0.75	0.75	0.75	0.50	0.50	0.50	0.50	0.50
Números_repetidos	0.75	1.00	0.75	0.75	0.50	0.50	0.50	0.50	0.50
Co-seno_de_1_radiano	0.75	0.75	1.00	0.75	0.50	0.50	0.50	0.50	0.50
Graus_minutos_e_segundos	0.75	0.75	0.75	1.00	0.50	0.50	0.50	0.50	0.50
União	0.50	0.50	0.50	0.50	1.00	0.75	0.75	0.75	0.75
Aniversário	0.50	0.50	0.50	0.50	0.75	1.00	0.75	0.75	0.75
Fatores_primos	0.50	0.50	0.50	0.50	0.75	0.75	1.00	0.75	0.75
É_palindromo?	0.50	0.50	0.50	0.50	0.75	0.75	0.75	1.00	0.75
Conversão_de_velocidade	0.50	0.50	0.50	0.50	0.75	0.75	0.75	0.75	1.00

Figura 5.3: Parte da tabela da função de similaridade para o myCBR

5.3.3 Base de casos com lista de exercícios

A base de casos é composta por listas de exercícios, cada caso contém cinco exercícios. Para construir essa base de casos, necessita-se compreender que a ordem nas listas faz diferença na recuperação. Por isso, foi gerada uma base de casos que faz a permutação das listas propostas, para aprimorar as consultas.

Determina-se um conjunto $F = \{ \text{'É } k\text{-alternante'}, \text{'Co-seno de 1 radiano'}, \text{'Matriz de Vandermonde'}, \text{'É palindromo?'} \}$ onde quatro valores que estarão em todas as listas, cada um foi retirado de uma sub-classe mais abaixo na ontologia, para criar uma base de casos válida com listas de exercícios equilibradas. Foi gerado um total de 2160 casos constando os quatro valores fixos do conjunto F que se alternam com os outros 19 valores variáveis, não ocorrendo a repetição de exercícios dentro de uma mesma lista.

A tabela 5.1 é um exemplo de um caso adicionado ao myCBR 3, cada questão é um atributo. Quatro atributos contém os exercícios do conjunto F (question_1, question_2, question_4, question_5). Esta é a estrutura da base de casos para os experimentos na forma de consultas (*queries*) apresentados na próxima subseção.

question_1	E_k alternante
question_2	Co_seno_de_1_radiano
question_3	Vigesimo_terceiro_numero_primo
question_4	Matriz_de_Vandermonde
question_5	E_palindromo

Tabela 5.1: Um exemplo de um caso de lista de exercícios adicionado na base de casos testada no myCBR 3.

5.3.4 Consultas (*Queries*)

As consultas são estabelecidas para que se possa perceber como a automatização na sugestão de lista de exercícios pode ser profícua. Tenta-se utilizar recuperações que possam tirar listas distintas como resultados e que possam ser equilibradas. As consultas foram determinadas a não ir de encontro direto aos casos, ou seja, buscas exatas. Pois o principal objetivo é perceber como uma plataforma CBR pode sugerir outras listas através de uma pergunta constando uma lista possível.

Foram feitas 30 consultas em quatro grupos distintos: 12 consultas não constando nenhum valor fixo; 6 com a presença de um mais elementos dos valores fixos; 8 com a presença de um valor desconhecido (`_unknown_` no myCBR); e 4 consultas onde todos os exercícios fazem parte da mesma sub-classe.

Diante das bases de conhecimento e de casos estabelecidas, dá-se o início às consultas e a obtenção dos resultados através das similaridades entre as questões. A próxima seção apresenta exemplos de algumas consultas, os resultados e considerações a respeito.

5.4 Execução e resultados dos experimentos

Tendo que as consultas são feitas no intuito de entender a dinâmica do CBR na recomendação de lista de exercícios, opta-se por elencar consultas que abordem as propriedades do CBR e, especificamente, da ferramenta utilizada no experimento, o myCBR 3. Como tratado na Seção 5.3, a base de casos foi determinada de forma a colaborar com o objetivo almejado nas consultas, constando quatro valores fixos determinados através do conjunto F e os outros 19 que se alternam.

Quando se faz a recuperação de casos (*retrieve*) no myCBR 3, os casos encontrados na

base são listados na ordem de similaridade, o usuário determina se quer mais ao topo o caso com maior similaridade ou com menor similaridade. No presente estudo, consideram-se os casos com mais alto grau de similaridade na consulta.

As consultas no experimento foram estruturadas de acordo com três parâmetros considerados como fatores críticos em uma busca de listas de exercícios na base em questão:

Equilíbrio: é uma variável booleana na presente pesquisa, considera-se uma consulta equilibrada se quatro dos cinco exercícios da lista pertencem, cada um, à sub-classes distintas na ontologia. Caso contrário, não é uma lista equilibrada.

Valores do conjunto F : é a quantidade de valores do conjunto F utilizados na consulta.

Valores desconhecidos: simula a ausência de um dos elementos da lista. Considera-se apenas a presença de um dos atributos no valor `__unknow__`.

A tabela 5.2 apresenta os resultados contando com os parâmetros explanados acima. Na primeira coluna são determinados os grupos de consultas para interpretação posterior. A quantidade de consultas de cada grupo é dado na segunda coluna por *# de consultas*. A terceira coluna consta se nas consultas foi considerado o equilíbrio (1) ou não (0). A presença de `__unknow__` é demonstrada de forma booleana na tabela, caso exista um elemento `__unknow__` na consulta é dado o valor 1, caso contrário, 0. A coluna encabeçada por *Valores de F* é quantidade de valores do conjunto F que foram utilizados, podendo assumir valores inteiros no intervalo $[0,4]$. A última coluna traz o mais alto grau de similaridade encontrado na recuperação de cada grupo.

A Figura 5.4 é um exemplo de consulta do *Grupo 1*. Os casos com mais alto grau de similaridade estão demonstrados na Figura 5.5, onde percebe-se que a presença de ‘Ocorrencias’ no caso ‘questions1591’ é o único elemento que tem correspondente direto, o resto se dá pela similaridade entre as questões da lista.

Obviamente, quanto mais os elementos na consulta estiverem parecidos com a base de casos, maior a similaridade dos casos recuperados com a consulta, é o que acontece no *Grupo 1*, quando se obtém 0,89 em consultas com quatro valores que estarão em qualquer

Query		
question_1	Conversao_de_temperaturas	Change Special Value: none
question_2	Uniao	Change Special Value: none
question_3	Fracoes	Change Special Value: none
question_4	Ocorrencias	Change Special Value: none
question_5	Vigesimo_terceiro_numero_primo	Change Special Value: none

Figura 5.4: Exemplo de uma consulta do *Grupo 1* feita no myCBR.

caso na base, ou seja, o conjunto F . É interessante observar o resultado das 12 consultas do *Grupo 4*, onde não é dado nenhum valor do conjunto F , a similaridade mais alta obtida é de 0,73. O *Grupo 3* também obtém o mesmo grau de similaridade mesmo contendo dois elementos do grupo F .

É pertinente perceber que é possível utilizar o CBR para, a partir de uma lista de exemplo, recuperar listas com altos índices de similaridade. O *Grupo 1* é a prova de que com um caso que não está presente no banco de casos, pode-se recuperar casos similares válidos.

Percebe-se que consultas desequilibradas conseguem uma similaridade alta pelo fato da base de casos ser formada por listas de equilibradas de exercícios. A Tabela 5.3 apresenta uma consulta feita à base de casos, constam apenas valores da sub-classe Alta-baixa. Parte das listas resultantes aparecem na Tabela 5.4.

O grau de similaridade é muito sutil entre os casos, deve-se portanto perceber cada

Grupo	# de consultas	Equilíbrio	..unknow..	Valores de F	Maior Similaridade
<i>Grupo 1</i>	3	1	0	4	0,89
<i>Grupo 2</i>	3	1	0	1	0,77
<i>Grupo 3</i>	4	1	1	2	0,73
<i>Grupo 4</i>	12	1	0	0	0,73
<i>Grupo 5</i>	4	0	0	0	0,67
<i>Grupo 6</i>	4	1	0	1	0,63

Tabela 5.2: Tabela de consultas ordenadas quanto à similaridade.

	questions1591	questions1508
Similarity	0.73	0.73
question_1	Co_seno_de_1_radiano	Co_seno_de_1_radiano
question_2	E_palindromo	E_palindromo
question_3	E_k_alternante	E_k_alternante
question_4	Ocorrencias	Matriz_de_Vandermonde
question_5	Matriz_de_Vandermonde	Vigesimo_terceiro_numero_primo

Figura 5.5: Exemplo de uma recuperação do *Grupo 4* feita no myCBR.

question_1	E.k.alternante
question_2	Sistema_de_equacoes
question_3	Vigesimo_terceiro_numero_primo
question_4	Deslocamento_de_Imagem
question_5	Quadrado_magico

Tabela 5.3: Uma das consultas do *Grupo 5*.

	questions1446	questions1424
Similarity	0.67	0.67
question_1	E.k.alternante	E.k.alternante
question_2	Matriz_de_Vandermonde	Matriz_de_Vandermonde
question_3	Vigesimo_terceiro_numero_primo	E_palindromo
question_4	E_palindromo	Co_seno_de_1_radiano
question_5	Co_seno_de_1_radiano	Quadrado_magico

Tabela 5.4: Dois resultados de uma consulta do *Grupo 5*.

décimo como relevante na leitura dos resultados. Concluí-se que a recomendação de lista de exercícios através de parte do ciclo CBR é relevante e possibilita a automatização nesta tarefa, onde, por exemplo, uma ferramenta de geração de lista de exercícios para um grupo de alunos consegue criar listas distintas de acordo com uma base de listas pré-determinadas e previamente identificadas. Embora não exista a fase de adaptação por parte do myCBR, pode-se utilizar a recuperação de casos, a representação de conhecimento e relação de similaridade entre elementos como subterfúgios para a relevância da pesquisa. Com a presença da adaptação em uma próxima versão do myCBR, amplia-se as possibilidades de resultados obtidos a partir de base de casos ainda menores.

5.5 Considerações Finais

A presente seção trata sobre a representação do conhecimento especificamente nas aplicações de raciocínio baseado em casos. Considera-se a observação sobre distância vetorial entre elementos e a similaridade dada através da distância. O foco principal é dado nas consultas à base de casos construída no intuito de atingir o objetivo da presente pesquisa.

É possível, então, através de uma lista de exercícios equilibrada se recuperar listas equivalentes através do raciocínio baseado em casos e das recuperações de similaridade. Infelizmente, não foi possível, perceber como poderiam ocorrer adaptações de casos, pois a ferramenta utilizada não disponibiliza essa tarefa. Sugere-se a utilização automatizada de recuperação de casos. Em uma situação real, é possível utilizar uma plataforma CBR junto a um sistema de ensino, onde distintos alunos terão distintas listas de exercícios geradas através de recuperações com o método do CBR.

CAPÍTULO 6

CONCLUSÃO

Investigar métodos computacionais capazes de resolver problemas complexos é uma prática pertinente, relevante e fundamental para o bom desenvolvimento dos estudos em Ciência da Computação. Essa prática contempla avaliar de diversas formas como um problema pode ser resolvido por uma solução computacional, angariando melhorar um determinado método de solução de problemas computacionalmente representáveis ou apenas descartá-lo para futuras pesquisas. O método aqui investigado é o Raciocínio Baseado em Casos, o qual, em um determinado domínio, busca resolver problemas utilizando de soluções anteriores. A solução se caracteriza por um ciclo: Recuperação, Reutilização, Revisão e Armazenamento. Durante a fase de reutilização podem ocorrer adaptações, esta é a atividade que se dispõe interesse na pesquisa aqui apresentada, pois aparenta ser uma forma econômica computacionalmente para tratar de soluções complexas, onde é possível arranjar respostas à perguntas genéricas em domínios com uma gama grande de conceitos.

Diante de um arcabouço de apoio ao ensino que utiliza a mediação de erro, uniu-se a investigação de um método pedagógico de ensino mediado por erro com a investigação de um método de resolução de problemas baseado em casos. É nessa interface que se destaca a relevância desse trabalho que aqui se apresenta. Observando se o sistema CBR pode ser profícuo na recomendação de lista de exercícios equilibradas e, como parâmetro, utiliza da proximidade entre exercícios concretizada pela quantidade de vezes que foram respondidas erroneamente. Esse trabalho almeja atender à demanda apontada por Kutzke [6] sobre a análise da recomendação de erros no processo de ensino e também busca investigar o potencial do CBR no contexto pedagógico.

6.1 Contribuições

A criação de uma ontologia baseada em respostas é uma das contribuições que esse trabalho dá a ambas as áreas aqui dadas como alicerces acadêmicos.

A base de casos do sistema CBR proposto é composta por listas de exercícios distintas. Cada caso é uma lista e os elementos que a compõe estão distribuídos dentro da ontologia desenvolvida.

Com a ferramenta aqui proposta é possível criar listas de exercícios equilibradas e distintas para uma turma de alunos. As listas vão se adaptando conforme os exercícios são respondidos, pois a mediação dos erros é utilizada na atual criação da ontologia. Podendo, por exemplo, criar-se um número de listas em cursos de Ensino a Distância que dificulta a troca de informações entre alunos, qualificando o método de ensino. Propõe-se a automatização do processo de geração de lista de exercícios, desde a classificação dos exercícios no que confere ao número de erros até a sugestão de listas através de consultas e recuperações automáticas.

6.2 Dificuldades

Durante as prévias da pesquisa elencou-se um sistema CBR genérico com grande importância dentro dos eventos especializados na área, o TUUURBINE, porém diversos problemas foram encontrados para fazer com que o sistema funcionasse. Diante de alguns desses problemas, optou-se por uma ontologia e um domínio de tamanho reduzido, podendo isso ser escalonado conforme os resultados fossem aparecendo.

O primeiro problema foi a falta de arquivos citados no manual encontrado no site oficial da ferramenta, havendo a necessidade do contato direto com os pesquisadores do laboratório responsável. A comunicação aconteceu e as respostas, na maior parte das vezes, eram respondidas imediatamente. Porém, outros problemas começaram a aparecer no que confere a ausência de bibliotecas e também na dificuldade de compilação do código. Houve a busca por outras ferramentas genéricas de CBR. O myCBR é utilizado sem a presença da fase de adaptação no sistema, dificultando a exploração do total potencial do

CBR.

6.3 Trabalhos futuros

Demandas podem ser levantadas tanto no que confere aos estudos sobre CBR quanto no que se refere aos estudos de sistemas de apoio ao ensino. Aponta-se como uma relevante possibilidade de estudos: uma comparação entre os principais sistemas CBR disponíveis, utilizando de *benchmark* ou de pautas determinadas para avaliação destes sistemas.

Em congruência com o raciocínio utilizado para o desenvolvimento da ontologia aqui apresentada, pesquisas podem ampliar a complexidade e qualificar a ontologia. Podendo ocorrer a criação de novas classes e subclasses, verificando outras relações entre as instâncias. Uma abordagem no que tange ao nível de dificuldade de cada lista de exercício é uma brecha aberta por essa pesquisa.

A base de casos pode ser ampliada com um número maior de exercícios, para isso, podem ser utilizadas outras ferramentas que trabalhem de forma semelhante ao FARMA-ALG. Além de se esperar a atualização do myCBR e, finalmente, perceber as adaptações das listas de exercícios, onde será possível fazer perguntas à base de casos de tamanhos menores com grande poder de representação, e a partir de uma lista, gerar outras listas similares.

BIBLIOGRAFIA

- [1] A. Aamodt and E. Plaza, “Case-based reasoning, foundational issues, methodological variations, and system approaches,” in *AI COMMUNICATIONS*, vol. 7, pp. 39–59, 1994.
- [2] I. Watson, “Case-based reasoning is a methodology not a technology,” *Knowledge-Based Systems*, vol. 12, no. 5, pp. 303 – 308, 1999.
- [3] W. W. W. Consortium *et al.*, “Rdf 1.1 concepts and abstract syntax,” 2014.
- [4] S. Russel and P. Norvig, *Inteligência Artificial*. Rio de Janeiro: Campus, 2004.
- [5] T. Dean, J. Allen, and J. Aloimononos, *Artificial Intelligence: theory and practice*. The Benjamin/Cummings Publishing Company, Inc., 1995.
- [6] A. R. Kutzke, “Informática educacional e a mediação do erro na educação: um estudo teórico-crítico e uma proposta de instrumento computacional,” 2015.
- [7] R. C.K. Riesbeck, *Inside Case-Based Reasoning*, vol. 1989. Erlbaum, Northvale, NJ, 1989.
- [8] J. L. Kolodner, “An introduction to case-based reasoning,” *Artif. Intell. Rev.*, vol. 6, no. 1, pp. 3–34, 1992.
- [9] C. G. von Wangenheim and A. von Wangenheim, *Raciocínio Baseado em Casos*. Barueri,SP: Manole, 2003.
- [10] D. B. Leake, “Cbr in context: The present and future,” *Case-Based Reasoning, Experiences, Lessons & Future Directions*, pp. 1–30, 1996.
- [11] A. Furnham, *50 ideias de psicologia que você precisa conhecer*. São Paulo:Planeta, 2015.

- [12] M. M. Richter, “Introduction,” in *Case-based reasoning technology: from foundations to applications*, vol. 1400, Springer, 2003.
- [13] “Significados.” <http://www.significados.com.br/metodologia/>. Accessed: 2015-05-15.
- [14] D. W. Aha, L. Breslow, and H. Muñoz-Avila, “Conversational case-based reasoning,” *Appl. Intell.*, vol. 14, no. 1, pp. 9–32, 2001.
- [15] R. Bergmann and M. Schaaf, “Structural case-based reasoning and ontology-based knowledge management: A perfect match?,” *j-jucs*, vol. 9, pp. 608–626, jul 2003. http://www.jucs.org/jucs97/structural_case_based_reasoning.
- [16] R. Bergmann, J. L. Kolodner, and E. Plaza, “Representation in case-based reasoning,” *Knowledge Eng. Review*, vol. 20, no. 3, pp. 209–213, 2005.
- [17] S. Craw, N. Wiratunga, and R. Rowe, “Learning adaptation knowledge to improve case-based reasoning,” *Artif. Intell.*, vol. 170, no. 16-17, pp. 1175–1192, 2006.
- [18] B. Díaz-Agudo, P. A. González-Calero, J. A. Recio-García, and A. A. Sánchez-Ruiz-Granados, “Building cbr systems with jcolibri,” vol. 69, pp. 68–75, Elsevier, 2007.
- [19] E. Gaillard, L. Infante-Blanco, J. Lieber, and E. Nauer, “Tuuurbine: A generic CBR engine over RDFS,” in *Case-Based Reasoning Research and Development - 22nd International Conference, ICCBR 2014, Cork, Ireland, September 29, 2014 - October 1, 2014. Proceedings*, pp. 140–154, 2014.
- [20] E. Hundt, P. Reuss, and C. Sauer, “Knowledge modelling and maintenance in mycbr3?,” 2014.
- [21] J. A. Recio-García, P. A. González-Calero, and B. Díaz-Agudo, “jcolibri2: A framework for building case-based reasoning systems,” vol. 0, pp. –, 2013.
- [22] E. Wenger, *Artificial intelligence and tutoring systems: computational and cognitive approaches to the communication of knowledge*. Morgan Kaufmann, 2014.

- [23] R. McGreal, “Learning objects: A practical definition,” *International Journal of Instructional Technology and Distance Learning (IJITDL)*, vol. 9, no. 1, 2004.
- [24] P. Funk and O. Conlan, “Case-based reasoning to improve adaptability of intelligent tutoring systems.,” in *ECCBR Workshops*, pp. 15–24, Citeseer, 2002.
- [25] F. Ricci, L. Rokach, and B. Shapira, *Introduction to recommender systems handbook*. Springer, 2011.
- [26] M. Gomez-Albarran and G. Jimenez-Diaz, “Recommendation and students’ authoring in repositories of learning objects: A case-based reasoning approach,” *International Journal of Emerging Technologies in Learning (iJET)*, vol. 4, no. 2009, pp. 35–40, 2009.
- [27] R. M. Vicari, A. Ribeiro, J. M. C. da Silva, E. R. Santos, T. Primo, and M. Bez, “Brazilian proposal for agent-based learning objects metadata standard-obao,” in *Metadata and Semantic Research*, pp. 300–311, Springer, 2010.
- [28] C. Bizer, T. Heath, and T. Berners-Lee, “Linked data-the story so far,” *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, pp. 205–227, 2009.
- [29] T. Berners-Lee, “Design issues: Linked data,” 2006.
- [30] D. Vrandečić and M. Krotzsch, “Wikidata: a free collaborative knowledgebase,” *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.
- [31] S. Bechhofer, I. Buchan, D. De Roure, P. Missier, J. Ainsworth, J. Bhagat, P. Couch, D. Cruickshank, M. Delderfield, I. Dunlop, *et al.*, “Why linked data is not enough for scientists,” *Future Generation Computer Systems*, vol. 29, no. 2, pp. 599–611, 2013.
- [32] M. B. Almeida and M. P. Bax, “Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção,” *Ciência da Informação, Brasília*, vol. 32, no. 3, pp. 7–20, 2003.
- [33] P. Cimiano, *Ontology learning from text*. Springer, 2006.

- [34] T. R. Gruber, “Toward principles for the design of ontologies used for knowledge sharing?,” *International journal of human-computer studies*, vol. 43, no. 5, pp. 907–928, 1995.
- [35] D. L. McGuinness, F. Van Harmelen, *et al.*, “Owl web ontology language overview,” *W3C recommendation*, vol. 10, no. 10, p. 2004, 2004.
- [36] T. C. Jepsen, “Just what is an ontology, anyway?,” *IT professional*, vol. 11, no. 5, pp. 22–27, 2009.
- [37] C. Welty, D. L. McGuinness, and M. K. Smith, “Owl web ontology language guide,” *W3C recommendation, W3C (February 2004) <http://www.w3.org/TR/2004/REC-owl-guide-20040210>*, 2004.
- [38] P. O. B. Netto, *Grafos: teoria, modelos, algoritmos*. Edgard Blücher, 2003.
- [39] D. Brickley and R. Guha, “Rdf schema 1.1. w3c recommendation (25 february 2014),” *World Wide Web Consortium*, 2014.
- [40] W. S. W. Group *et al.*, “Sparql 1.1 overview. w3c recommendation 21 march 2013,” 2012.
- [41] D. SAVIANI, “Escola e democracia. rev,” *Campinas: Autores Associados*, 1997.
- [42] A. R. Kutzke and A. Direne, “Mediação do erro na educação: um arcabouço de sistema para a instrumentalização de professores e alunos,” in *Anais do Simpósio Brasileiro de Informática na Educação*, vol. 25, pp. 737–746, 2014.
- [43] D. Marczal, A. Direne, A. Pimentel, and E. M. Krynski, “Farma: Uma ferramenta de autoria para objetos de aprendizagem de conceitos matemáticos,” in *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, vol. 4, p. 23, 2015.
- [44] A. R. Kutzke and A. I. Direne, “Farma-*alg*: An application for error mediation in computer programming skill acquisition,” in *Artificial Intelligence in Education*, pp. 690–693, Springer, 2015.

- [45] A. Atanassov and L. Antonov, “Comparative analysis of case based reasoning software frameworks jcolibri and mycbr,” *Journal of the University of Chemical Technology & Metallurgy*, vol. 47, no. 1, pp. 83–90, 2012.

CARLOS ANDRÉ ZAVADINACK

**RACIOCÍNIO BASEADO EM CASOS COMO
RECOMENDADOR DE CONTEÚDO PEDAGÓGICO**

Proposta de Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. Fabiano Silva

CURITIBA

2016