

UNIVERSIDADE FEDERAL DO PARÁ  
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Renan Cristiano Rocha Rodrigues

**WEBCMTOOL: AMBIENTE WEB PARA FACILITAR A AVALIAÇÃO DA  
APRENDIZAGEM SIGNIFICATIVA, BASEADO EM MAPAS  
CONCEITUAIS E ONTOLOGIAS DE DOMÍNIO**

Belém  
2012

**Renan Cristiano Rocha Rodrigues**

**WEBCMTOOL: AMBIENTE WEB PARA FACILITAR A AVALIAÇÃO DA  
APRENDIZAGEM SIGNIFICATIVA, BASEADO EM MAPAS  
CONCEITUAIS E ONTOLOGIAS DE DOMÍNIO**

Dissertação de Mestrado apresentada para obtenção do grau de Mestre em Ciência da Computação. Programa de Pós-Graduação em Ciência da Computação. Instituto de Ciências Exatas e Naturais. Universidade Federal do Pará. Área de Concentração: Sistemas de Informação. Orientador: Prof. Dr. Francisco Edson Lopes da Rocha

Belém  
2012

Rodrigues, Renan Cristiano Rocha

WebCMTool: Ambiente Web para Facilitar a Aprendizagem Significativa, Baseado em Mapas Conceituais e Ontologias de Domínio / (Renan Cristiano Rocha Rodrigues); orientador, Francisco Edson Lopes da Rocha. - 2012.

70 f. il. 28 cm

Dissertação (Mestrado) - Universidade Federal do Pará. Instituto de Ciências Exatas e Naturais. Programa de Pós-Graduação em Ciência da Computação. Belém, 2012.

1. Educação-Processamento de dados. 2. Sistemas de informação gerencial. I. Rocha, Francisco Edson Lopes da, orient. II. Universidade Federal do Pará, Instituto de Ciências Exatas e Naturais, Programa de Pós-Graduação em Ciência da Computação. III. Título.

CDD 22. ed. 371.334

**FOLHA  
DE  
APROVAÇÃO**

Dedico esta obra, principalmente, aos meus grandes conselheiros e leais amigos: Papai e Mamãe... E a todos, cuja raiz do conhecimento, um dia, foi fecundada pela Educação.

## AGRADECIMENTOS

Agradeço a Deus pela saúde e, sobretudo, pela sabedoria em mim depositada, fator determinante que contribuiu significativamente para a realização deste trabalho. Sem dúvida nenhuma, esta obra é o fruto legítimo de uma dedicação suprema, marcada pelo esforço e contribuição científica riquíssimos.

Ao meu pai, José Joaquim Melo Rodrigues, pelas conversas e conselhos que foram fundamentais para minha vida. Obrigado pelo profundo e intenso amor, pela dedicação no seu trabalho, quando precisei estudar e me dedicar neste mestrado, bem como pelo pão de cada dia, pela atenção e contínua felicidade que proporciona desde o dia primeiro de abril até os dias de hoje, saiba que não seria feliz completamente se você; meu pai companheiro, meu pai amigo, meu pai conselheiro, meu pai engraçado, meu pai amável, meu pai confiável, meu inigualável pai... não estivesse na minha vida. Este sucesso é dedicado a você!

À minha maravilhosa mãe, Osley Maria Rocha Rodrigues, pelos agradáveis conselhos, pela dedicação e por ter me fornecido amor e carinho durante a realização do meu mestrado, pela segurança e conforto confiáveis, mas principalmente, por sempre acreditar que um dia estaria completando este trabalho. Um dos objetivos que devemos ter na vida, segundo a minha mãezinha, é a dedicação em qualquer trabalho que nos comprometemos a fazer, mas sempre dando o melhor de si, pois assim, com certeza, não haverá reconhecimento que não será atraído durante a vida. Tal objetivo deve estar, alicerçado ao amor com Deus, que é nossa fonte de felicidade. Esta fonte de conhecimento impresso nas páginas desta dissertação é dedicada a você, minha mamãe!

Aos meus queridos irmãos: Ao Igor Oliveira Rodrigues, pelas visitas e conselhos sobre nunca me desviar do caminho do bem e me guiar para o caminho do estudo a fim de chegar sobre degraus cada vez mais altos. Ao Bruno Abner Pereira Rodrigues (em memória) por me incentivar a fazer Ciência da Computação e ensinar a gostar de programação, fatores cruciais para o desenvolvimento deste trabalho, ademais pelos momentos ímpares vividos juntos. Ao Diego Alan Rocha Rodrigues pelo exemplo concreto de perspicácia e dicas que me ajudaram neste mestrado, pelo apoio e companheirismo, saiba que você é um dos espelhos de atitude, iniciativa e coragem que tenho na vida meu “manozão”. Ao Patrick José Rocha Rodrigues pelo companheirismo diário nas diversas situações encontradas e por me ajudar a combater obstáculos que, de alguma forma, me atrapalhavam de estudar, trabalhar ou mesmo me dedicar

a esta pós; às conversas sábias sobre nossos sonhos que juntos tivemos e por ser um exemplo nato de vontade de crescimento, sua ajuda saudável foi um dos condicionadores para este sucesso, espero sempre contar com você “maninho”. Obrigado a todos e saibam que amo demais vocês!

À minha companheira, Danielle Leal Sampaio – “minha sereia”, por ter estado sempre ao meu lado nos momentos tristes como na perda da minha madrinha e de minha vovó, sua ausência poderia ter influenciado nas minhas metas. Obrigado por ter estado comigo e me dado forças que, quando preenchidos pela sua presença, tornaram-se bem mais especiais e únicos para mim. Obrigado também pela lealdade, fidelidade e companheirismo, pelo seu amor, por me fazer feliz com seu sorriso maroto e por seu carinho cativante, por acreditar neste sucesso, mesmo quando a esperança assusta com sua ausência. Agradeço a sua dedicação e, principalmente, por crer em mim, ainda quando atribulações acontecem e problemas surgem. Desculpa pelas faltas como namorado e companheiro na sua vida, esta vitória, entretanto, reflete o esforço desses momentos... Ainda quero tê-la durante muitos outros sucessos que vierem a ocorrer depois deste. Amo você!

A todos os meus amigos e familiares maternos e paternos pela força depositada. Em especial, agradeço a Jorge Rodrigues, Henderson Costa, Débora Costa, Arivaldo Brito, Fabrizio Farias, Aryane Silva, Luciana Sampaio, Manoel Monteiro, Murillo Cordeiro, Hendel Costa e Gustavo Vale pelos conselhos profissionais e pessoais que frutificaram meu foco nesta pesquisa; às minhas tias Terezinha Costa, Oslecy Garcia e aos meus tios Joaquim Rodrigues e Thomaz Rocha por compartilharem comigo histórias fabulosas sobre suas vidas que durante certo tempo me serviram como caminho das pedras para seguir rumo a uma conclusão; ao meu tio Haroldo Costa pelas boas conversas sobre Deus, fé e credibilidade na vitória; a minha “dindinha”, Ângela Rodrigues (em memória) por sempre acreditar em mim, pelo amor e por me ensinar que a desistência nunca deve esta, acompanhada à vontade de crescimento; à minha maravilhosa vovó Francisca Rocha (em memória) pelas inteligentes conversas de valores e integridade, pela companhia, por estimular em mim sempre o prazer pela leitura e busca pelo conhecimento; à minha vovó Maria Rodrigues a ‘Maroca’, pelo carinho e atenção notáveis em sua companhia e, claro, pelos almoços e jantas surpresas em sua casa. Obrigado a todos vocês!

Ao Manoel Sampaio e a Rute Sampaio pelos ricos conselhos, também por me acolherem como um filho durante este propósito. Obrigado, sobretudo, pela amizade de vocês.

Ao meu orientador, Prof. Francisco Edson Lopes da Rocha, pela orientação, confiança

no meu trabalho e ajuda nesta pesquisa. Obrigado pela amizade e por todas as confissões passadas nesse período de convívio. Espero conseguir agregar metade do conhecimento que hoje agrega e preenche no intelecto de seus alunos e orientandos em sala de aula.

À Universidade Federal do Pará, principalmente ao Programa de Pós-Graduação em Ciência da Computação (PPGCC) pelo apoio com a infraestrutura, materiais e laboratórios que foram cláusulas fundamentais para conclusão desta obra. Obrigado também a todos os professores e a coordenação.

Aos amigos do LABIE - UFPA pelo companheirismo até altas horas da noite, dúvidas sanadas, caronas, diversões, além do cafezinho, é claro...!

Aos grandes amigos e/ou fabulosos programadores pelas ajudas nas entrelinhas da programação ou ainda por uma simples palavra amiga de conforto diante de um desespero ou pelas conversas para fortalecimento, mas precisamente por nunca me fazerem desistir em hipótese alguma, são eles: Gabriel Souza, Glauber Monteiro, Cleyton Dim, Ramon Falsoni, Rodrigo Lisboa, Rodrigo Barbalho, David Lopes e Daniel Alencar.

A Tatiane Borges e André Gusmão desde a parceria inicial no desenvolvimento de seu TCC até sua conclusão com dedicação e muito esforço. Foi um prazer imenso trabalhar com excelentes profissionais como vocês.

Aos amigos do CESUPA, da FabSoft e do grupo LCN, principalmente ao Prof. Otávio Noura por enfatizar que depois de cada luta há sempre a recompensa, ao Prof. Fagundes de Moraes por ter estimulado em mim o caminho à pesquisa e a todos os antigos membros da FabSoft com quem tive a oportunidade de trabalhar.

*Renan Rodrigues*



*“Tudo o que sua mão encontrar para fazer, faça-o com todo o seu coração.”*

**Jesus de Nazaré**

*“Sempre há uma forma de vencer um obstáculo na vida, eu não acredito que um obstáculo não possa ser vencido de forma alguma...”*

**(Durante uma das conversas com o meu pai em 2011)**

**José Joaquim Melo Rodrigues**

*“Não há problema que não possa ser resolvido com a paciência.”*

**Chico Xavier**

*“O estudo e a leitura são a garantia do sucesso ‘dindinho’, eles vêm primeiro que o sucesso e, não o contrário.”*

**(Comentário da minha querida madrinha – ‘a minha dindinha’ – quando morava com ela)**

**Angela Maria Melo Rodrigues**

*“Se antes a terra, e depois o capital eram os fatores decisivos de produção, hoje o fator decisivo é cada vez mais, o homem em si, ou seja, o seu conhecimento.”*

**Papa João Paulo II**

*“O bom é o pior inimigo do melhor. O comodismo das boas virtudes são o grande empecilho para nos tornarmos melhores.”*

**Jim Homberguer**

*“É bem melhor pensar sem falar, do que falar sem pensar e, cometer o erro de perder algo ou até mesmo machucar alguém.”*

**(Um dos ensinamentos da minha mãe)**

**Osley Maria Rocha Rodrigues**

*“Se cheguei até aqui foi porque me apoiei no ombro dos gigantes.”*

**Isaac Newton**

*“De mil maneiras somos pressionados, mas não esmagados. Vivemos perplexos, mas não desesperamos, perseguidos, mas não desamparados. Somos abatidos até ao chão, mas não aniquilados.”*

**2CORÍNTIOS 4:8-10**

*"Se a educação sozinha não transforma a sociedade, sem ela, tampouco, a sociedade muda."*

**Paulo Freire**

*“Se você quer ser bem-sucedido, precisa ter dedicação total, buscar seu último limite e dar o melhor de si mesmo”*

**Ayrton Senna**

*“Ter uma ideia realmente original. É a única forma de me destacar. É a única forma de... ser importante.”*

**(Fala do personagem de Russel Crowe, John Nash, no filme Uma Mente Brilhante)**

**John Forbes Nash Jr.**

## RESUMO

Este trabalho apresenta o *WebCMTool*, que é um ambiente *Web* que visa facilitar a avaliação da Aprendizagem Significativa baseado em Mapas Conceituais (MCs) e Ontologias de Domínio, através da inclusão de um Algoritmo Genético na ferramenta de edição de Mapas Conceituais incorporado ao ambiente, que possibilita a diminuição da sobrecarga de trabalho do professor na tarefa de avaliação. Após esta tarefa é gerado um relatório detalhado para o aluno sobre sua avaliação auxiliando-o na valorização de sua Idiossincrasia ao aprender. A avaliação automática é realizada por meio de um processo computacional que recebe três insumos de entrada: o mapa do aluno, a ontologia do professor e uma taxonomia de frases de ligação. Durante o processo de avaliação são geradas populações de MCs, de modo que cada mapa contém um valor de adaptação que lhe é característico. Os MCs com valor de adaptação maior ou igual à média da população são escolhidos para dar continuidade ao processo de avaliação. A continuidade ocorre porque esses MCs são utilizados como pais em operações de cruzamentos para a geração de descendentes. Este processo ocorre até que os mapas gerados por descendências futuras possuam um valor de adaptação ótima. Quando isto ocorre, esses mapas resultantes da avaliação são utilizados para análise e compreensão de resultados pelo aluno por meio de um relatório que verifica o valor atribuído às proposições de seu mapa, verifica também a nota em percentual que valida cada proposição considerada correta e o resultado de seu desempenho no mapa. Um protótipo do ambiente foi aplicado e testado num curso de Programação em uma Universidade.

**PALAVRAS-CHAVE:** Mapas Conceituais, Ontologias de Domínio, Algoritmos Genéticos, Idiossincrasia, Aprendizagem Significativa.

## **ABSTRACT**

This work presents the WebCMTool, which is a web environment to facilitate the assessment of Meaningful Learning based on Concept Maps (CMs) and Domain Ontology, through the inclusion of a Genetic Algorithm in editing tool Concept Maps embedded environment, which enables the reduction of the workload of the teacher in the assessment task. After this task is to generate a detailed report to the student about his assessment assisting him in appreciation of his Idiosyncrasy to learn. The automatic assessment is performed by a computational process that takes three inputs input: the map of the student, the teacher's ontology and taxonomy linking phrases. During the assessment process are generated populations of CMs, so that each map contains one adaptation value which is characteristic. The CMs with adaptation value greater than or equal to the average of the population are chosen to continue the assessment process. The continuity is because these CMs are used as parents in breeding operations to generate offspring. This process occurs until the maps generated by future offspring have a great amount of adaptation. When this occurs, these maps from the assessment are used for analysis and understanding of results by the student through a report that checks the value assigned to the propositions of your map, it also notes that validates percentage in each proposition considered correct and the result of their performance on the map. A prototype environment has been implemented and tested in a programming course at a University.

**KEYWORDS:** Concept Maps, Domain Ontologies, Genetic Algorithms, Idiosyncrasy, Meaningful Learning.

## LISTA DE FIGURAS

<i>Figura 1.</i>	<i>Diagrama geral da aprendizagem</i>	30
<i>Figura 2.</i>	<i>Fase inicial da aprendizagem significativa</i>	30
<i>Figura 3.</i>	<i>Fase de assimilação</i>	31
<i>Figura 4.</i>	<i>Fase de ancoragem e diferenciação</i>	31
<i>Figura 5.</i>	<i>Fase de reconciliação e consolidação</i>	32
<i>Figura 6.</i>	<i>Aprendizagem mecânica</i>	33
<i>Figura 7.</i>	<i>Exemplo de diferenciação progressiva e reconciliação integrativa</i>	34
<i>Figura 8.</i>	<i>Uma proposição de um mapa conceitual da árvore</i>	36
<i>Figura 9.</i>	<i>Exemplo de um mapa conceitual do carboidrato no webcmeditor</i>	37
<i>Figura 10.</i>	<i>Componentes da avaliação da aprendizagem</i>	38
<i>Figura 11.</i>	<i>Técnicas de mapeamento conceitual em diferentes pontos num continuum</i>	39
<i>Figura 12.</i>	<i>Componentes de um algoritmo genético</i>	42
<i>Figura 13.</i>	<i>Exemplo de cromossomo através da codificação binária</i>	43
<i>Figura 14.</i>	<i>Exemplo de cromossomo através da codificação por permutação</i>	43
<i>Figura 15.</i>	<i>Exemplo de cromossomo através da codificação de valores</i>	44
<i>Figura 16.</i>	<i>Exemplos de cromossomos através da codificação em árvore</i>	45
<i>Figura 17.</i>	<i>Taxonomia das frases de ligação do ambiente webcmtool</i>	47
<i>Figura 18.</i>	<i>Ontologia sobre o cromossomo</i>	49
<i>Figura 19.</i>	<i>Processamento do gaadt-cm</i>	53
<i>Figura 20.</i>	<i>Arquitetura cliente e servidor do javaserver faces</i>	55
<i>Figura 21.</i>	<i>Visão geral do processo scrum</i>	56
<i>Figura 22.</i>	<i>Arquitetura de módulos do webcmtool</i>	57
<i>Figura 23.</i>	<i>Editor webcmeditor durante a construção de um mc contextualizado</i>	59
<i>Figura 24.</i>	<i>Editor webon_tool na construção de uma ontologia e suas relações binárias</i>	60
<i>Figura 25.</i>	<i>Relação binária de uma ontologia no webon_tool</i>	61

<i>Figura 26.</i>	<i>Conjunto base b originado pela união de conceitos e relações binárias</i>	62
<i>Figura 27.</i>	<i>Alfabeto gerado da figura 25</i>	63
<i>Figura 28.</i>	<i>Função para o cálculo da adaptação de um cromossomo</i>	65
<i>Figura 29.</i>	<i>Função para o cálculo do grau de adaptação de um gene</i>	65
<i>Figura 30.</i>	<i>Organização do módulo repositório</i>	67
<i>Figura 31.</i>	<i>Página de acesso interno</i>	68
<i>Figura 32.</i>	<i>Fluxograma do funcionamento geral do spring security</i>	69
<i>Figura 33.</i>	<i>Diagrama de casos de uso geral da avaliação</i>	70
<i>Figura 34.</i>	<i>Classe ambiente com os principais métodos do gaadt-cm</i>	71
<i>Figura 35.</i>	<i>Construção de uma mapa conceitual contextualizado por um aluno no curso no webcmtool</i>	77
<i>Figura 36.</i>	<i>Exemplo do mapa conceitual construído por um dos grupos</i>	81
<i>Figura 37.</i>	<i>Ontologia utilizada para o exercício com o webcmtool</i>	82
<i>Figura 38.</i>	<i>Construção do mapa contextualizado de um aluno de sistemas de informação</i>	83
<i>Figura 39.</i>	<i>Parte do relatório da avaliação do mapa conceitual avaliado pelo aluno</i>	84
<i>Figura 40.</i>	<i>Visão geral da ferramenta ergolist</i>	97
<i>Figura 41.</i>	<i>Aspecto de uma questão na ferramenta ergolist</i>	99

## LISTA DE QUADROS

Quadro 1.	<i>Indivíduos de uma população: principais tipos de codificação</i>	42
Quadro 2.	<i>Comparação entre o cmtool e o webcmttool</i>	57
Quadro 3.	<i>Operadores lógicos e regras de inferência</i>	64
Quadro 4.	<i>Valores que o peso <math>\theta</math> pode assumir</i>	65
Quadro 5.	<i>Valores admissíveis dos parâmetros do grau do gene</i>	66
Quadro 6.	<i>Conteúdo programático do curso linguagem Java</i>	67
Quadro 7.	<i>Quantidade de alunos que já conheciam os conceitos estudados em sala</i>	76
Quadro 8.	<i>Grau de severidade definido na avaliação</i>	94
Quadro 9.	<i>Estrutura dos resultados da avaliação realizada</i>	94
Quadro 10.	<i>Lista de parte dos problemas e os resultados dos avaliadores</i>	95
Quadro 11.	<i>Resultado da avaliação do princípio estética e projeto minimalista</i>	96

## LISTA DE GRÁFICOS

<i>Gráfico 1.</i>	<i>Quantidade de alunos que utilizaram o ambiente por graduação</i>	<i>78</i>
<i>Gráfico 2.</i>	<i>Quantidade de alunos distribuídos por universidades</i>	<i>79</i>
<i>Gráfico 3.</i>	<i>Evolução da aprendizagem em relação a média da turma no módulo 1</i>	<i>86</i>
<i>Gráfico 4.</i>	<i>Evolução da aprendizagem em relação a média da turma no módulo 2</i>	<i>87</i>
<i>Gráfico 5.</i>	<i>Evolução da aprendizagem em relação a média da turma no módulo 3</i>	<i>89</i>

## **LISTA DE ABREVIATURAS**

<i>AS</i>	<i>Aprendizagem Significativa</i>
<i>AG</i>	<i>Algoritmo Genético</i>
<i>MC</i>	<i>Mapa Conceitual</i>
<i>GAADT-CM</i>	<i>Genetic Algorithm based on Abstract Data Types applied to Concept Maps</i>
<i>MCEst</i>	<i>Mapa Conceitual do Estudante</i>
<i>JSF</i>	<i>JavaServer Faces</i>
<i>UML</i>	<i>Unified Modeling Language</i>



# SUMÁRIO

1	INTRODUÇÃO .....	18
1.1	IDENTIFICAÇÃO DO PROBLEMA .....	20
1.2	OBJETIVOS .....	21
1.2.1	Objetivos Específicos .....	21
1.3	DESAFIOS .....	22
1.4	CONTRIBUIÇÕES .....	22
1.5	METODOLOGIA DO TRABALHO .....	23
1.6	ORGANIZAÇÃO DA DISSERTAÇÃO .....	24
2	FUNDAMENTAÇÃO TEÓRICA .....	26
2.1	A IMPORTÂNCIA DA WEB PARA A EDUCAÇÃO .....	26
2.2	APRENDIZAGEM SIGNIFICATIVA.....	28
2.2.1	Processos da aprendizagem significativa .....	33
2.2.2	Avaliação da aprendizagem significativa.....	34
2.3	MAPAS CONCEITUAIS.....	35
2.3.1	Avaliação da aprendizagem com mapas conceituais .....	37
2.4	ALGORITMOS GENÉTICOS.....	41
2.5	TAXONOMIA DAS FRASES DE LIGAÇÃO .....	46
2.6	ONTOLOGIAS DE DOMÍNIO .....	47
2.7	ESTADO DA ARTE: AVALIAÇÃO AUTOMÁTICA DA APRENDIZAGEM COM MAPAS CONCEITUAIS .....	49
3	WEBCMTOOL COMO FACILITADOR DA AVALIAÇÃO AUTOMÁTICA DA APRENDIZAGEM POR MAPAS CONCEITUAIS E ONTOLOGIAS DE DOMÍNIO.....	53
3.1	INTRODUÇÃO .....	53
3.2	ARQUITETURA DO WEBCMTOOL.....	54
3.2.1	Módulo WebCMEditor .....	58
3.2.2	Módulo <i>WebOn_Tool</i> .....	59
3.2.3	Módulo <i>GAADT-CM</i> .....	60
3.2.4	Módulo Avaliador .....	66
3.2.5	Módulo Repositório.....	67
3.2.6	Módulo Administrador.....	68
3.3	ESPECIFICAÇÕES TÉCNICAS.....	69
3.3.1	Diagrama de casos de usos .....	70
3.3.2	Visão geral da avaliação no GAADT-CM .....	71

4	ANÁLISE DOS RESULTADOS .....	73
4.1	PRIMEIRA ETAPA: ANÁLISE DA APLICAÇÃO DO AMBIENTE .....	73
4.1.1	Perfil dos Alunos.....	77
4.1.2	Exercícios em grupo e individual sem o webcmtool.....	80
4.1.3	Exercício com o webcmtool.....	81
4.1.4	Os resultados da aprendizagem dos alunos .....	84
4.1.5	Dificuldades relatadas pelos alunos .....	89
4.1.6	Entrevista e questionário de satisfação do ambiente .....	89
4.2	SEGUNDA ETAPA: ANÁLISE DE USABILIDADE .....	90
4.2.1	Usabilidade .....	90
4.2.2	Resultados encontrados na avaliação heurística.....	92
4.2.3	Resultados encontrados na avaliação por checklist pela ferramenta ergolist .....	96
5	CONSIDERAÇÕES FINAIS.....	101
5.1	CONTRIBUIÇÕES CIENTÍFICAS.....	103
5.2	TRABALHOS FUTUROS .....	103
	APÊNDICE A – DETALHES DA IMPLEMENTAÇÃO .....	113
	APÊNDICE B – RELATÓRIO DA AVALIAÇÃO DO MÓDULO AVALIADOR DO WEBCMTOOL .....	133
	APÊNDICE C – AJUDA NO WEBCMTOOL.....	138
	APÊNDICE D – QUESTIONÁRIO DE SATISFAÇÃO .....	141
	APÊNDICE E – ESTUDO DE VIABILIDADE .....	145
	E.1 Tempo e escopo .....	147
	E.2 Riscos e benefícios .....	149
	ANEXO A – RELATÓRIO DA AVALIAÇÃO HEURÍSTICA DO WEBCMTOOL.....	151
	ANEXO B – RELATÓRIO DA AVALIAÇÃO POR CHECKLIST DO WEBCMTOOL.....	162

## 1 INTRODUÇÃO

O advento da Internet trouxe para a sala de aula inúmeros benefícios, ajudando a melhorar o processo de ensino/aprendizagem, facilitando a construção do conhecimento com o apoio de ferramentas automáticas e sua aplicabilidade na Educação (OLIVEIRA, 2008). Os avanços nas áreas de Computação e Educação deslocaram as pesquisas das antigas teorias educacionais *comportamentalistas* para o *construtivismo*, havendo o consenso de que no construtivismo o conhecimento não seja transmitido, mas sim construído pelo aluno, trabalhando-se com a ideia de que aprendizagem envolve a construção de conhecimento e mudança conceitual (ROCHA, 2007).

Aprendizagem Significativa<sup>1</sup> (AS) é uma teoria construtivista em grande evidência. Ela é compreendida como um processo no qual uma nova informação é relacionada a um aspecto relevante específico existente na estrutura cognitiva de um indivíduo (AUSUBEL, 2000). Mapa Conceitual<sup>2</sup> (MC) é a principal ferramenta cognitiva de desenvolvimento da aprendizagem significativa, sendo utilizada para representação do conhecimento por meio de diagramas de conceitos (NOVAK, 2010).

O crescimento da Educação a Distância (EaD) e sua tentativa em aplicar as ideias construtivistas da AS por meio de Ambientes Virtuais de Aprendizagens (AVAs), bem como o uso de MCs em outros sistemas ajudam na construção do conhecimento com o desafio de se ter, por exemplo, milhares de estudantes compartilhando recursos de um mesmo ambiente. É difícil, entretanto, avaliar a aprendizagem de grupos com estas dimensões, pois as ferramentas automáticas de ensino e avaliação disponíveis ainda são incompletas. Segundo Rocha (op. cit.), do ponto de vista construtivista, uma lacuna permanece aberta: na maioria destes ambientes, a forma de avaliar a aprendizagem é insatisfatória por falta de uma clara base epistemológica norteando sua concepção e desenvolvimento. Neste sentido, observa-se que a avaliação automática em AVAs é focada, sobretudo, na avaliação de questões fechadas (objetivas). Porém, estas questões são insuficientes para cobrir todos os fundamentos presentes na taxonomia dos objetivos educacionais proposto por Bloom (AIRASIAN, 2001), a qual define uma classificação para o conhecimento que pode ser aprendido em sala de aula e os processos cognitivos que auxiliam a aprendizagem. Por exemplo, Bloom sugere que a criação de

---

<sup>1</sup> Aprendizagem Significativa será referenciada, neste trabalho, como AS.

<sup>2</sup> Neste trabalho, os termos mapa conceitual e mapas conceituais serão representados como MC e MCs, respectivamente.

redações e textos que abstraíam o conhecimento só pode ser avaliada por meio de questões abertas (ROCHA, 2007).

Além de apresentar conceitos que ajudem o processo de ensino e aprendizagem na área da Educação, esta pesquisa envolve a aplicação de conceitos de Inteligência Artificial Simbólica (IAS) e Inteligência Artificial Evolucionária (IAE). IAS se fundamenta na hipótese de que inteligência resulta da manipulação formal de símbolos, sendo as ontologias de domínios aplicações de IAS utilizada em várias áreas de conhecimento (O'LEARY, 1998). Elas podem ser usadas para propósitos de navegação, adaptação, compartilhamento e cooperação (ROCHA et. al., 2005). Por outro lado, a IAE se inspira na Biologia, sendo sistematizada a partir da observação de como a Natureza age para solucionar seus problemas. Nesta inteligência, o Algoritmo Genético (SCHWEFEL, 1987) – AG – é o principal paradigma da Computação Evolucionária. Segundo Holland (1975), seu criador, os AGs são utilizados para resolver problemas gerais de otimização ligados ao conceito de adaptação em diversas áreas de engenharia e ciências, como otimização numérica e combinatória, em que são descritos com base em conceitos de *fitness*, noção de herança e população de indivíduos.

A pesquisa apresentada nesta dissertação é fundada no desenvolvimento de um ambiente computacional para a Web denominado *Web Concept Mapping Tool (WebCMTTool)*, sendo uma extensão do *Concept Mapping Tool (CMTTool)* (ROCHA et. al., 2004). O objetivo do ambiente é de contribuir significativamente para a avaliação automática na Web de questões abertas baseadas em MCs. Composto a ferramenta há um AG, desenvolvido para rodar em paralelo e cuja principal característica educacional implementada é a valorização dos aspectos idiossincráticos da aprendizagem do estudante, e uma ontologia de domínio produzida pelo professor. Do ponto de vista educacional, a ontologia funciona como um repositório de conceitos cujo aprendizado garante ao aprendiz o domínio da estrutura do conhecimento envolvido e, do ponto de vista computacional, a ontologia é o espaço de busca do AG, ou seja, é o repositório de onde ele tira a matéria prima para construir os indivíduos das populações de MCs. Compõe também o ambiente uma taxonomia de frases de ligação definida em Jr. et. al. (2004), cujo uso serve para esclarecer o significado que o aprendiz atribui à ligação entre conceitos.

As ideias importantes subjacentes a esta pesquisa são: (i) facilitação da aprendizagem e do ensino construtivista, baseada em MCs e ontologias de domínio; e, (ii) diminuição da sobrecarga de trabalho do professor durante o processo de avaliação, suprimindo a deficiência de avaliação de questões abertas envolvendo MCs.

## 1.1 IDENTIFICAÇÃO DO PROBLEMA

Classicamente, a avaliação de trabalhos escolares tem sido feita de forma manual ou semi-automática, raramente de forma completamente automática. Neste último caso, o que é avaliado são questões fechadas, que embora úteis em determinados contextos, não atendem completamente às necessidades educacionais, deixando de lado a avaliação de questões abertas que, normalmente, se referem ao julgamento do conhecimento estrutural presente, por exemplo, em trabalhos de conclusão de curso, dissertações, teses e monografias (ROCHA, 2007).

O conhecimento estrutural pode ser avaliado por meio da avaliação automática de textos ou por meio de MCs (RUIZ-PRIMO; SHAVELSON, 1996), entre outras possibilidades. A avaliação do conhecimento estrutural por meio de MCs oferece uma grande gama de possibilidades, começando pela metodologia *fill in the map* e, no outro extremo, sem qualquer ajuda, apresentado o texto ao aprendiz para ser mapeado. Entre estes dois extremos há um contínuo de possibilidades. A forma escolhida no projeto *WebCMTool* foi apresentar ao aprendiz uma coleção de conceitos armazenados numa ontologia de domínio e uma taxonomia de frases organizadas em diferentes dimensões semânticas para que ele possa, sem ambiguidade, explicitar o significado atribuído às unidades de pensamento, chamadas de proposições no contexto da AS (AUSUBEL, 2000).

Para se ter uma ideia da dificuldade envolvida, observa-se que a proposição <homem tem filhos> e <homem tem perna> não se referem ambas à dimensão <parte> porque <perna> é parte de <homem> mas <filho> não é parte de <homem>. Então como esclarecer o significado atribuído à palavra de ligação <tem>? No ambiente *CMTool* isto pode ser feito porque estas proposições podem ser escritas como <homem.parte.tem.pernas> e <homem.tutoria.tem.filho>, ou seja, o aprendiz pode dizer que no primeiro caso ele entende que a relação semântica entre <homem> e <perna> é uma relação de partição e a palavra de ligação <tem> representa esta dimensão semântica, enquanto que no segundo caso a palavra de ligação <tem> representa a dimensão semântica de tutoria. Complementarmente, observa-se que o *CMTool* é um poderoso instrumento de explicitação das dimensões semânticas de relacionamento entre entidades que normalmente aparecem na prática da Engenharia de Software.

O grande desafio desta pesquisa foi propor um ambiente, na *Web*, que auxiliasse a avaliação de MCs desenvolvidos por alunos de modo a valorizar o aspecto idiossincrático de sua aprendizagem. Neste cenário, optou-se por estender o *CMTool* (ROCHA, 2007) com o objetivo de fornecê-lo num ambiente totalmente integrado por meio de permissões de usuários, visando enriquecer a aprendizagem.

## 1.2 OBJETIVOS

O objetivo geral deste trabalho consistiu em modelar e desenvolver um ambiente computacional na *Web* com foco na avaliação da aprendizagem, sendo esta mediada por MCs. Este ambiente utiliza um modelo de aprendizagem e avaliação segundo o consenso da AS, sendo adequado para avaliação e acompanhamento individual passo a passo, avaliação e acompanhamento de grupos de estudantes e para a geração de comparações da aprendizagem entre estudantes.

### 1.2.1 Objetivos Específicos

Os seguintes tópicos destacam-se como os objetivos específicos deste trabalho:

- reformular a arquitetura de módulos do *CMTool*;
- adaptar e adicionar o motor de inferência ao AG;
- adicionando novas regras de lógicas matemáticas;
- projetar os editores *CMEditor* e *On\_Tool* na *Web*;
- implementar mecanismos especializados na arquitetura, que permitam integrar aprendizagem e avaliação da aprendizagem, segundo o modelo construtivista da AS;
- integrar a avaliação do algoritmo genético ao editor de MCs, para que estudantes possam avaliar seus mapas<sup>3</sup> pelo editor;
- desenvolver o repositório no ambiente para dados dos usuários e arquivos *XML*;
- modelar um ambiente amigável para acesso aos usuários na *Web*.

---

<sup>3</sup> Neste trabalho, são consideradas sinônimas as expressões mapa e mapa conceitual.

### 1.3 DESAFIOS

Foram identificados três desafios na pesquisa, determinação: (1) de um mecanismo de avaliação no editor de MCs que permita ao estudante iniciar sua avaliação com base no AG; (2) do ambiente na *Web* baseada na arquitetura de módulos do *CMTool*; e, (3) de um novo modelo com base em regras de lógicas matemáticas que fomentasse o motor de inferência na criação de proposições inferidas implicitamente.

Na literatura (RUIZ-PRIMO; SHAVELSON, 1996; RUIZ-PRIMO ET AL., 1997), trabalhos prévios sobre avaliação da aprendizagem eram baseados no modelo clássico de comparação entre as respostas do estudante e um gabarito de referência geralmente definido por um professor ou especialista da área. Este modelo de avaliação está de acordo com a escola tradicional, referente à “crença” comportamentalista. Neste cenário foi desenvolvido o *CMTool* com o desafio de se utilizar em sala de aula o modelo construtivista, incluindo um mecanismo de avaliação da aprendizagem significativo automático. Com base nestas ideias, tem-se como desafio atual encontrar maneiras viáveis de utilizar em sala de aula a avaliação na *Web* pelos alunos, além da possibilidade e futuro uso também a distância, segundo o modelo construtivista, permitindo notoriamente que estudantes avaliem seus próprios MCs.

Ademais, a modelagem de um ambiente na *Web*, contendo tais características, representa por si só um difícil problema. Para tanto, optou-se por mapear cada módulo do ambiente original para então traçar seu progresso durante a extensão. Para isto foram utilizadas técnicas ágeis de gestão de projetos.

Simultaneamente ao processo de extensão do ambiente, encontrou-se como desafio a introdução de regras baseadas nas lógicas matemáticas como sendo fundamental para aproximar um resultado correto após a avaliação de um MC.

### 1.4 CONTRIBUIÇÕES

O *WebCMTool* possui as mesmas características que o ambiente *CMTool*, além de avançar o estado da arte da IAS e da avaliação automática de MCs pela web, acrescentando:

- i. um módulo específico para avaliação dentro do editor de MCs com base na sessão do aluno;

- ii. um *GAADT-CM* (GUSMÃO; BORGES, 2011), tendo como principal relevância sua otimização e integração a um motor de inferência.

A contribuição principal desta obra é a avaliação da AS na *Web*, iniciada pelo editor de MCs com base em um AG que utiliza ontologias de domínio como espaço de busca para formar sua população inicial. Neste sentido, o *WebCMTool* dá partida também a um modelo de ambiente que contribui com as funcionalidades descritas:

- viabiliza um modelo de *aprender a aprender* (NOVAK; GOWIN, 1984), estabelecido na linguagem visual no *WebCMEditor* e *WebOn\_Tool* (capítulo 3);
- viabiliza um modelo de avaliação da aprendizagem (ZABALA, 1998), que respeita o modo individual de aprender de cada aluno, sem carga adicional ao professor;
- viabiliza a aplicação em sala de aula de um modelo construtivista, baseado na AS, diminuindo a sobrecarga do professor na tarefa de avaliação, além de permitir que alunos construam conhecimento significativo na modalidade não-presencial;
- viabiliza a inserção no contexto a distância, permitindo que cada vez mais alunos e professores construam conhecimento com as práticas da AS;
- permite gerar relatório individualizados dos alunos comparando seu progresso após cada avaliação.

## 1.5 METODOLOGIA DO TRABALHO

A metodologia utilizada está fundamentada na execução de uma pesquisa experimental baseada em um ambiente no contexto da educação em computação, envolvendo AS, MC e ontologia de domínio. Para isto, foi utilizado o *WebCMTool* com o propósito de auxiliar os alunos na construção de significados, analisados por meio de uma avaliação idiossincrática, baseada na estrutura cognitiva de cada aluno em seu MC construído.

O ambiente foi implementado na linguagem de programação Java, integrando a este ferramentas de edição de MCs, de avaliação e de ontologias de domínios, no âmbito do Laboratório de Informática Educativa (LABIE) da Universidade Federal do Pará (UFPA).

A fim de testar a viabilidade do ambiente, optou-se por experimentá-lo em um Curso de Programação ofertado pela Empresa Júnior de Informática na Universidade. A aplicação foi organizada em três módulos aplicados durante quatro dias, em que cada módulo possuiu 4



horas de duração, totalizando-se dezesseis horas de aplicação. Em ambos os módulos foram apresentados conceitos, lista de exemplos e lista de exercícios referentes, por exemplo, sobre a construção de MCs. Logo, este curso fez parte de um estudo inicial, que se preocupou em verificar se o ambiente ajuda na construção do conhecimento significativo bem como se o mesmo ajuda na avaliação da aprendizagem, analisando quais aspectos podem ser aproveitados e quais as melhorias devem ser realizadas em trabalhos posteriores. Apesar do pouco tempo de análise para determinar se o ambiente é apropriado, ressalta-se que o mesmo refere-se a um protótipo e não a um modelo conclusivo já maduro, portanto, esta pesquisa apresenta um ambiente ainda em crescimento e com um desafiador valor agregado ao ceio científico.

## 1.6 ORGANIZAÇÃO DA DISSERTAÇÃO

Além do capítulo introdutório esta dissertação está estruturada em quatro capítulos, quatro apêndices e dois anexos os quais foram distribuídos na forma descrita a seguir:

O capítulo 2 apresenta o referencial teórico, tratado nesta dissertação. Desta forma, são explicados conceitos sobre a Teoria da Assimilação de David Paul Ausubel, ressaltando as características da AS, além de mostrar sua ferramenta cognitiva, os MCs de Joseph Novak. É abordado também os AGs que são os responsáveis pela avaliação automática do ambiente e seu espaço de busca, as ontologias de domínio. É apresentada a taxonomia de frases de ligação presente no ambiente, utilizada tanto nos MCs como nas ontologias de domínio construídas. Procura-se ressaltar neste capítulo sobre a avaliação automática da aprendizagem mostrando o estado da arte em trabalhos correlatos sobre este tema.

O capítulo 3 descreve a estrutura e arquitetura de módulos do ambiente *WebCMTool*, que facilita a avaliação da aprendizagem mediada por MCs e ontologias de domínio.

O capítulo 4 apresenta a análise dos resultados obtidos após validação e testes realizados na avaliação dos alunos pelo AG.

O capítulo 5 apresenta as considerações finais e discussões da dissertação; aborda as principais contribuições e discute como a pesquisa proposta na dissertação pode avançar em trabalhos futuros.

O apêndice A contém os detalhes da implementação de cada módulo do ambiente durante a fase de análise e projeto do *WebCMTool*.

O apêndice B contém detalhes do relatório gerado na avaliação pelo AG, especificando os atributos presentes tendo como base a ontologia sobre Linguagem de Programação.

O apêndice C apresenta o ajuda do *WebCMTool* esclarecendo os principais pontos de dúvidas durante o uso dos editores.

O apêndice D mostra o questionário de satisfação submetido aos alunos para avaliação do ambiente, este questionário foi importante na avaliação, principalmente pelas considerações sugeridas pelos alunos que utilizaram na prática o *WebCMTool*.

O relatório de avaliação Heurística fundamentada na regras de Nielsen e o relatório de avaliação por Checklist com modelos de questões utilizadas e uma lista das perguntas analisadas pelos profissionais encontra-se no Anexo A e Anexo B, respectivamente. Ambos realizados por três especialistas de computação.

## 2 FUNDAMENTAÇÃO TEÓRICA

Na educação, os trabalhos de diversos autores (BELLONI, 2008; CALDAS et. al., 2009; NOVAK, 2010; CRUZ et. al., 2010; RODRIGUES et. al., 2011; GONÇALVEZ et. al., 2011) mostram que os avanços tecnológicos têm contribuído, de modo significativo, em todo o sistema educacional, procedendo no sentido de cumprir um papel sociopedagógico que possibilite a inserção de indivíduos com pensamentos construtivistas no mundo competitivo. Para isto, ambientes foram desenvolvidos para propósito de colaboração (ARAÚJO et. al., 2002, 2003; LIMA et. al., 2006) e avaliação (MACHADO et. al., 2004; ROCHA, 2007; RODRIGUES et. al., op. cit.) na educação, com a finalidade de auxiliar de forma automática os envolvidos na construção do conhecimento tendo, assim, um papel de melhora no processo educacional.

Segundo Rocha (2007), houve o deslocamento nas pesquisas das antigas teorias educacionais comportamentalistas para o construtivismo e, com isto, a AS se destacou principalmente por ser a teoria construtivista em evidência que se preocupa com a aplicabilidade na prática em sala de aula tendo o MC como o seu principal mediador.

Logo, este trabalho apresenta uma forma de avaliar automaticamente os MCs no cenário da *Web*. Dessa maneira, este capítulo cobre o referencial teórico de todos os conceitos tratados nesta obra, estando dividido da seguinte maneira: a Seção 2.2 descreve a importância da *Web* para a educação; a Seção 2.3 apresenta a AS e seu contexto na educação levando em conta, a princípio, as ideias de David P. Ausubel; na Seção 2.4, é mostrado o MC; a Seção 2.5 abre um espaço para a discussão sobre AGs e, em seguida na Seção 2.6, é explicado sobre as ontologias no contexto computacional e sua aplicação, por fim, a Seção 2.7, relata os trabalhos a respeito da avaliação da aprendizagem mediada por MCs de acordo com a direção tratada.

### 2.1 A IMPORTÂNCIA DA WEB PARA A EDUCAÇÃO

A utilização da *World Wide Web* (WWW ou simplesmente *Web*), como principal serviço de comunicação e disseminação da informação, é a garantia do concreto sucesso da Internet no mundo atual. Este serviço permitiu que a troca da informação entre pessoas no mundo crescesse a uma escala exponencial (BBC NEWS, 2010), frisando, assim, a comunicação virtual.

Adicionalmente, a *Web* surge na educação como um meio viabilizador da aprendizagem, seja de um lado, promovendo a interação contínua e ajuda na construção do conhecimento por meio de ferramentas de aprendizagem, que facilitam a troca de informações no contexto do ensino/aprendizagem, ou seja, por outro lado, aproximando professores e alunos separados geograficamente a uma única interseção: a educação.

Isto implica que não é mais necessário estar sentado ao lado do outro para poder desenvolver tarefas em grupo. A disponibilização dos artefatos produzidos em um ambiente comum, que possa ser acessado de qualquer lugar e a qualquer hora, viabiliza as interações assíncronas. Ademais, os docentes podem estar em contato constante com seus estudantes, acompanhando e orientando suas atividades, sem que com isso tenha que esperar por um encontro em uma hora e um local determinado. Estas tecnologias e suas possibilidades de uso na *Web* abriram os horizontes para se repensar, por exemplo, na educação à distância, antes baseada no uso de materiais específicos, para estudantes que teriam que estudar praticamente isolados (AGUIAR, 2006).

Para Silva (2005), as ferramentas usuais são imprescindíveis, porém não suficientes, sendo assim, não basta apenas usar correio, *chat*, fóruns, videoconferências, *downloads* e *uploads*. É necessário que se criem ambientes para o uso articulado destas ferramentas, associados a uma proposta pedagógica.

Na educação, sistemas computacionais disponíveis na *Web*, denominados Ambientes *E-learning* (EAs), têm como objetivo a ampliação do acesso à educação e à informação, de onde o professor e o aluno podem estar separados fisicamente no espaço e/ou tempo (FERLIN et. al., 2008). Como exemplo desses tipos de sistemas, citam-se o *Modular Object-Oriented Dynamic Learning Environment 1* (Moodle), *Sakai2*, Ambiente de Ensino-Aprendizagem Adaptativo na *Web 3* (AdaptWeb), *TelEduc4*, o projeto Tecnologia da Informação para o Desenvolvimento da Internet Avançada - Aprendizado Eletrônico<sup>5</sup> (TIDIA-AE) entre outros, que foram desenvolvidos em plataforma de software livre e são destinados ao suporte de atividades educacionais via Internet (ALMEIDA, 2003). A utilização de ambientes virtuais de aprendizagem leva a uma reflexão (LIMA et. al., 2011): Como o conhecimento é construído nesses ambientes?

Inquestionavelmente, a forma de fazer a avaliação da aprendizagem tornou-se um foco importante da pesquisa em educação e pesquisas recentes (MOREIRA, 2006; ROCHA, 2007; GRUNDSPENKIS et. al., 2009; NOVAK, 2010; RODRIGUES et. al., op. cit.) têm contribuído para o desenvolvimento de estratégias cognitivas com foco no ensino e na avaliação da

aprendizagem. Procurando contribuir nesta direção, este trabalho apresenta um ambiente que ajuda na construção do conhecimento, valorizando a forma pessoal de aprender de cada pessoa, o que fica mais evidenciado, quando reforçada pelo uso de instrumentos pedagógicos como o MC. As seções que seguem fazem um breve apanhado sobre os temas pedagógicos que envolvem este trabalho.

## 2.2 APRENDIZAGEM SIGNIFICATIVA

Em meados de 1963, David Paul Ausubel publicou o livro *The Psychology of Meaningful Verbal Learning* (AUSUBEL, 1963) em que demonstrou os resultados obtidos em suas pesquisas. A prioridade de Ausubel foi, principalmente, a aprendizagem que ocorre na sala de aula, apresentando uma teoria cognitivista que procura explicar os mecanismos internos que ocorrem na mente humana com relação ao aprendizado e à estruturação do conhecimento. Batizou suas ideias de Teoria da Assimilação.

A Teoria da Assimilação têm como ideia principal a AS que, para acontecer, precisa que a aprendizagem de um novo conceito esteja relacionada a um *subsunçor* (“âncora”), ou seja, um conceito relevante e prévio da estrutura cognitiva do indivíduo. Desta forma, o *subsunçor* pode ser entendido como uma estrutura específica de conhecimento a qual uma nova informação pode se integrar no cérebro humano, que é altamente organizado e detentor de uma hierarquia conceitual que armazena experiências prévias do estudante (AUSUBEL, 2000).

Para Rocha (2007), a AS é definida como:

[uma teoria cognitiva de construção de significados. Ela propõe que a aquisição e retenção do conhecimento é o produto de um processo interacional ativo, integrador, entre a matéria de ensino e ideias relevantes presentes na estrutura cognitiva do estudante, às quais novas ideias são relacionáveis de maneira bem particulares].

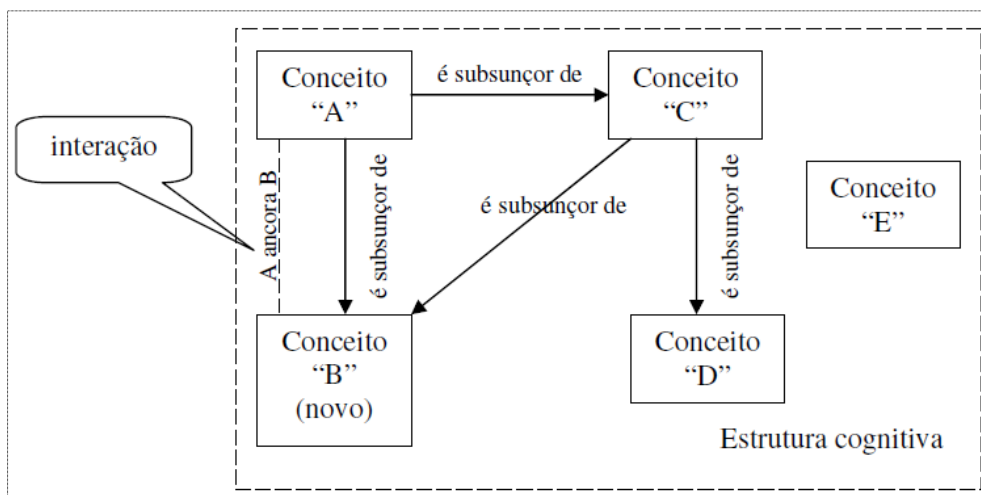
Neste contexto, poder-se-ia perguntar: se os *subsunçores* estiverem ausentes para viabilizar a AS, como seria possível criá-los?

Segundo Ausubel (2000), a AS é precedida por outro tipo de aprendizagem existente num *continuum* com a aprendizagem significativa: a aprendizagem mecânica, também

conhecida como aprendizagem memorística. Ela é definida como sendo a aprendizagem de novas informações sem produzir qualquer conhecimento novo, visto que sua relação com o conhecimento existente na estrutura cognitiva é arbitrária, não substantiva, literal, periférica e, geralmente, de curta duração (AUSUBEL, 2000). Tipicamente, esta aprendizagem (p.ex. números de ceps) tem utilidade limitada a pequenas práticas que economizam tempo e esforço (ROCHA, 2007). Desta forma, os subsunçores podem aparecer durante a fase de aprendizagem memorística e serem mais adiante apropriados para a construção de significados.

A Figura 1 representa graficamente a ideia explicada. Segundo Ferreira (2008), sob a visão da teoria ausubeliana a estrutura cognitiva do aluno, na interação escolar, recebe um conceito (informação) novo (B) e um conceito existente (A) ancora o novo, por assimilação, na estrutura, modificando-a, de forma a criar novas ligações (subsunções) entre os já existentes (C e D). Tal assimilação (subsunção) é designada como significativa por Ausubel. A interação entre o conhecimento prévio com o novo possibilita a ancoragem necessária à aprendizagem significativa. Um conceito incorporado de forma mecânica (E) permanece na estrutura como indicado na figura.

**Figura 1. Diagrama geral da aprendizagem**

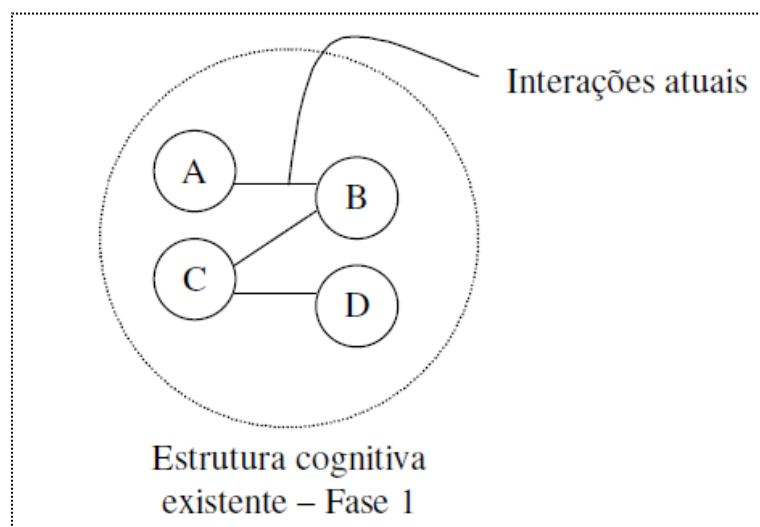


Fonte: (FERREIRA, 2008).

Este processo é descrito por Ausubel como sendo ordenado e sequencial, ou seja, a internalização de conceitos pelo aprendiz ocorre como um processo, organizado em fases distintas e que podem ser facilitadas pedagogicamente pelo professor. A aprendizagem significativa se dá na seguinte sequência ou fases (Figura 2 a 6), conforme descrito por Ausubel (FERREIRA, 2008).

Fase 1 (Estágio Inicial) – A estrutura cognitiva do aprendente possui conceitos previamente formados, com suas respectivas ligações (AUSUBEL, 1963).

**Figura 2. Fase inicial da aprendizagem significativa**

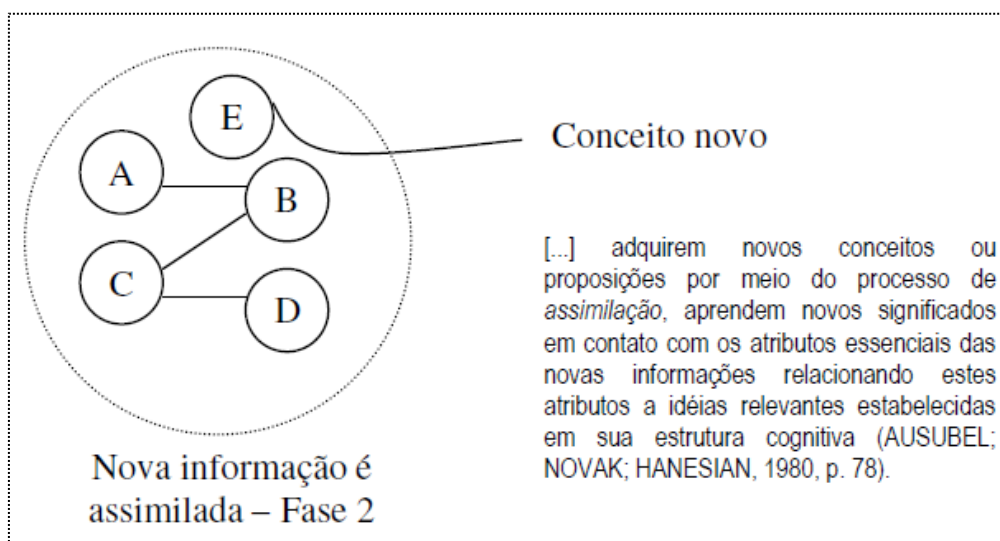


Fonte: (FEIRREIRA, 2008).

Na primeira fase, representada na Figura 2, a estrutura cognitiva do estudante está previamente organizada e consolidada. Partindo desse conjunto de conceitos formados, com suas respectivas ligações (interações), inicia-se o processo de AS.

Fase 2 (Apresentação do novo conceito) – A estrutura cognitiva do aprendente assimila a nova informação.

**Figura 3. Fase de assimilação**

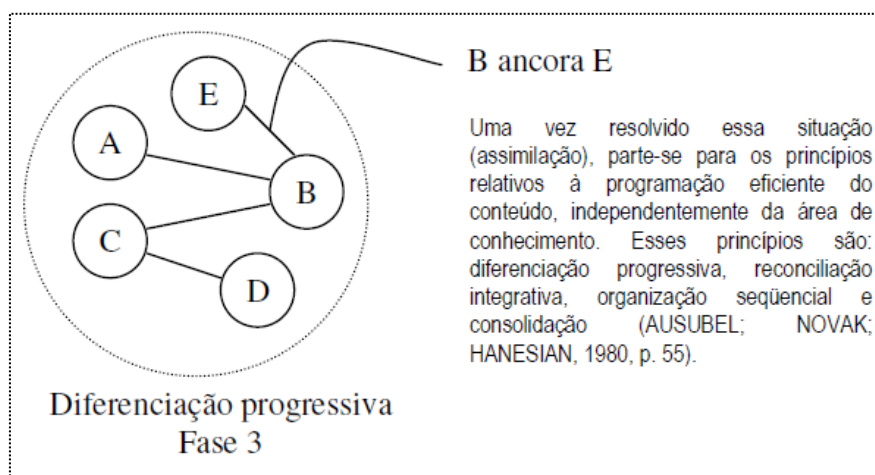


Fonte: (FERREIRA, 2008).

A nova informação é apresentada ao aluno, conforme a Figura 3, que a relaciona de forma não-arbitrária e não-literal ao conjunto de conceitos já formados em sua estrutura cognitiva. Ausubel denominou esse relacionamento de assimilação.

Fase 3 (Diferenciação progressiva) – Um conceito existente ancora a nova informação.

**Figura 4. Fase de ancoragem e diferenciação**

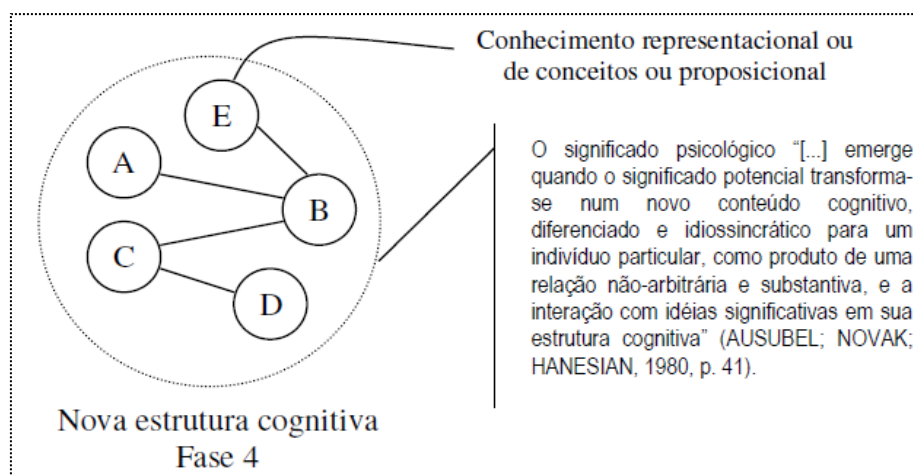


Fonte: (FERREIRA, 2008).

Mesmo ancorado em conceito já existente, na terceira fase esquematizada na Figura 4, o novo conceito é diferenciado progressivamente (ver Seção 2.2.1) em seus detalhes e especificidades por meio de discriminações sucessivas, para que seja atingida a “consolidação”.

Fase 4 (Reconciliação integrativa) – A estrutura cognitiva do aprendente se reorganiza, caracterizando, assim, a aprendizagem significativa, de um novo conceito.

**Figura 5. Fase de reconciliação e consolidação**



Fonte: (FERREIRA, 2008).

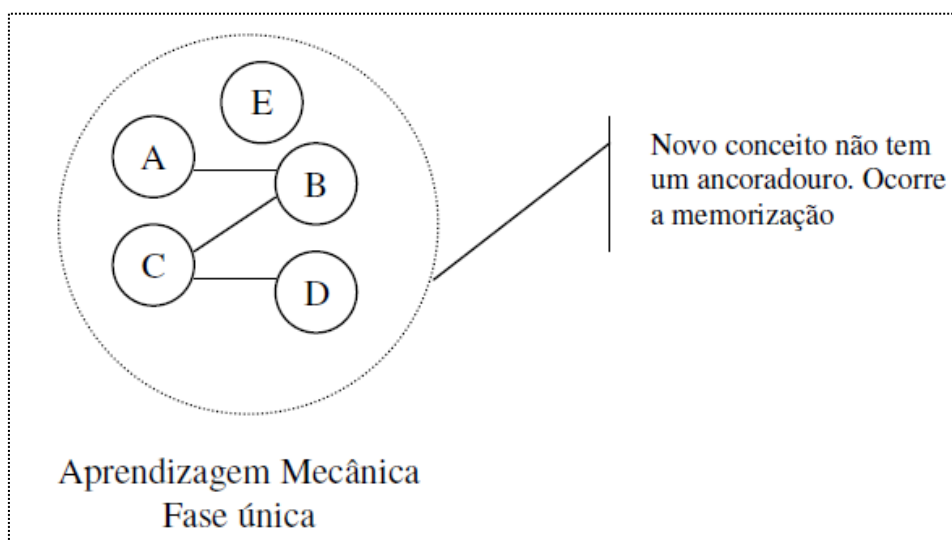


Na última etapa da AS, Fase 4 esquematizada na Figura 5, ocorre o processo de síntese pela diferenciação progressiva e consolidação, em que o aprendiz estabelece conexões entre os conceitos correlatos de sua estrutura cognitiva, adquirindo deste modo maior estabilidade e clareza. A nova estrutura cognitiva consolidada fica então pronta para reiniciar o processo e ancorar novos conceitos, encerrando o ciclo de AS.

Quando a ação pedagógica apresenta o conceito já pronto ao aprendiz, por exemplo, ocorre a Aprendizagem Mecânica, que é realizada em fase única, por meio da memorização. A Figura 6 esquematiza a Aprendizagem Mecânica:

Fase Única (Aprendizagem Mecânica) – quando os conceitos e princípios são apresentados prontos ao aluno, basta memorizá-los ou “decorá-los” (AUSUBEL, 1963b).

**Figura 6. Aprendizagem mecânica**



Fonte: (FERREIRA, 2008).

Finalmente, para que a aprendizagem significativa ocorra, é importante que a estrutura cognitiva do estudante contenha subsunçores claros, estáveis e diferenciáveis que sirvam como ponto de ancoragem para novas aprendizagens (ROCHA, 2005). Além disso, o estudante deve ser capaz de aplicar continuamente os processos mentais que medeiam a aprendizagem, com o objetivo de formar novos significados.

### 2.2.1 Processos da aprendizagem significativa

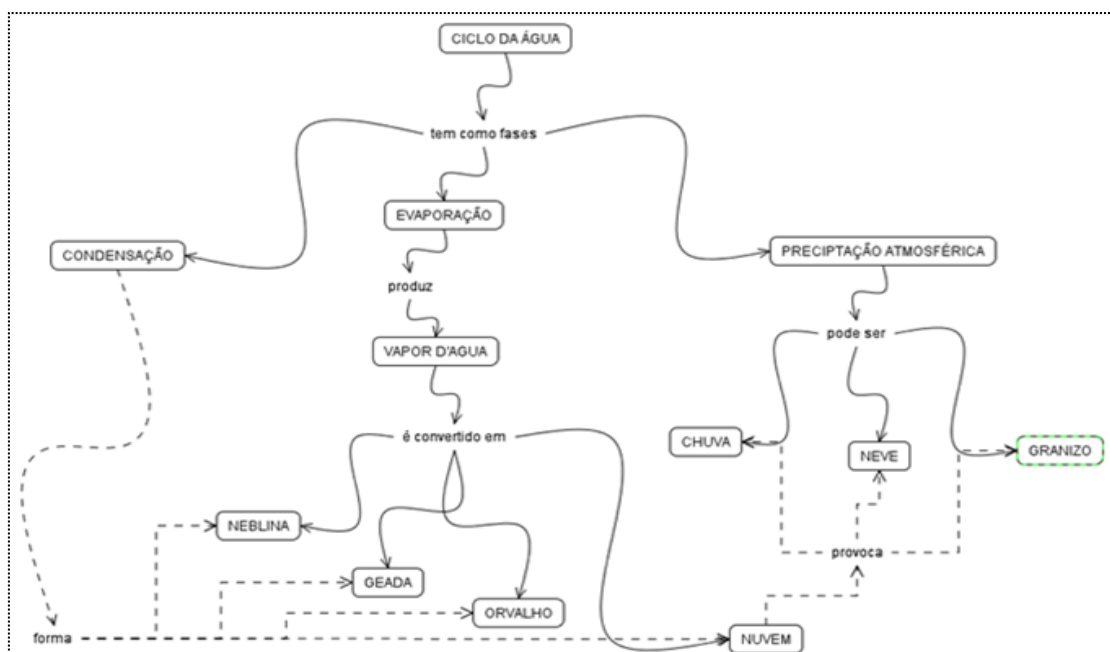
Na aprendizagem significativa, há dois processos que norteiam o aprendizado de novos conhecimentos a serem trabalhados, são: i) diferenciação progressiva e, ii) reconciliação integrativa.

Na diferenciação progressiva, os conceitos apresentados aos alunos são programados de maneira que o conceito mais geral de um conteúdo é apresentado em primeiro lugar e, posteriormente, são apresentados os conceitos que lhe são mais específicos. Este processo conduz à ampliação da estrutura cognitiva (adição de novas ideias por meio de novas aprendizagens). Neste processo, o novo conceito ou proposição é aprendido por subordinação, ou seja, por um processo de interação e ancoragem em um conceito superordenado (*subsunçor*) onde tanto o conceito que está sendo adquirido como o conceito já aprendido (o *subsunçor*) se modifica (ROCHA, 2007).

A reconciliação integrativa ocorre numa aprendizagem de característica superordenada ou combinatorial, em que os conceitos originais buscam associações entre si, interligando de forma expansiva e sintética. Neste processo, as ideias mais amplas relacionam subsunçores que a princípio estavam separados na estrutura cognitiva.

A Figura 7 apresenta exemplos de conceitos diferenciados progressivamente e reconciliados integrativamente.

**Figura 7. Exemplo de diferenciação progressiva e reconciliação integrativa**



Fonte: (Adaptado de ROCHA, 2007).

A reconciliação integrativa é vista como a finalização do processo de mapeamento, em que são analisadas a determinação de ligações entre dois ramos diferentes ou a introdução de níveis intermediários de inclusividade entre dois ou mais conceitos existentes.

O conceito CICLO DA ÁGUA está diferenciado progressivamente (setas contínuas) porque ele tem conceitos que lhe são subordinados (CONDENSAÇÃO, EVAPORAÇÃO e PRECIPITAÇÃO ATMOSFÉRICA), ou seja, CICLO DA ÁGUA é o *subsunçor* e os outros são assimilados por ele. Por outro lado, os conceitos NUVEM, NEVE, GRANIZO e CHUVA são conceitos que já existem na estrutura cognitiva do indivíduo e que inicialmente não estavam conectados, mas por meio da própria capacidade criativa do estudante se ligam gerando um novo aprendizado por meio da reconciliação integrativa apresentada por setas tracejadas.

### **2.2.2 Avaliação da aprendizagem significativa**

Ausubel especifica três maneiras para se realizar a avaliação da aprendizagem significativa, que são: 1) testes de compreensão; 2) resolução de problemas; e, 3) aprendizagem dependente de sequência.

Os testes de compreensão para serem eficazes devem ser mostrados em uma linguagem e em um contexto diferente daqueles encontrados no material utilizado para apresentar um novo conteúdo (ROCHA, 2007), pois se não for feito desta forma, as respostas podem apenas exprimir um aprendizado decorrente do processo de memorização.

A resolução de problemas é outra forma para testar a aprendizagem significativa, entretanto para que essa técnica seja eficaz é necessário o conhecimento de outras habilidades, como: raciocínio, familiaridade com o problema e perseverança. Desta forma, o insucesso da avaliação pode ser decorrente da falta de conhecimento de uma ou mais dessas habilidades e não da falta de compreensão do conhecimento apresentado. Contrariamente, o sucesso da avaliação não garante que seja um indicativo da compreensão do conteúdo, pois respostas podem ser acertadas na sorte ou decorrente do processo de memorização.

A última técnica é a aprendizagem dependente de sequência que consiste em apresentar ao estudante uma tarefa de aprendizagem que é dependente de uma tarefa anterior (ROCHA, 2007). Entretanto, como uma abordagem alternativa à questão da avaliação Joseph D. Novak

propõe os mapas conceituais, explicado na próxima seção, como uma maneira adicional de teste da aprendizagem significativa.

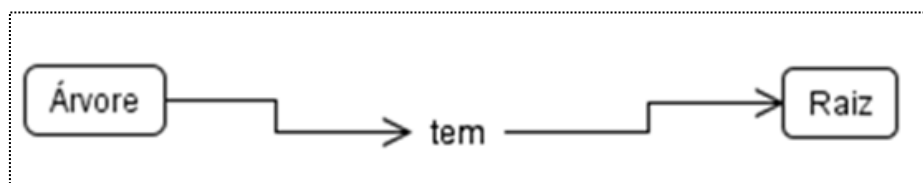
### 2.3 MAPAS CONCEITUAIS

Preocupado com a aplicabilidade das teorias de Ausubel, Novak desenvolveu a ferramenta cognitiva da Aprendizagem Significativa: o Mapa Conceitual (MC) (NOVAK, 2010). Um MC é um diagrama consistindo de conceitos (nodos) interligados por linhas, que indicam as relações entre conceitos mais abrangentes (mais inclusivos) e os conceitos menos abrangentes (menos inclusivos), em que esses conceitos procuram refletir a organização conceitual de uma disciplina ou de uma parte dela.

Embora normalmente tenham uma organização hierárquica e, muitas vezes, incluam setas, tais diagramas não devem ser confundidos com organogramas ou diagramas de fluxo, pois não implicam sequência, temporalidade ou direcionalidade, nem hierarquias organizacionais ou de poder. Mapas conceituais são diagramas de significados, de relações significativas (MOREIRA, 1993).

Mapas conceituais são formados por conceitos, frases de ligação e proposições. Uma proposição em um mapa é formada por pelo menos dois conceitos e uma palavra ou frase de ligação que une esses dois conceitos conforme demonstra a Figura 8 com os conceitos árvore e raiz e unindo esses conceitos há a frase de ligação ‘tem’.

**Figura 8. Uma proposição de um mapa conceitual da árvore**



Fonte: (Elaborada pelo autor).

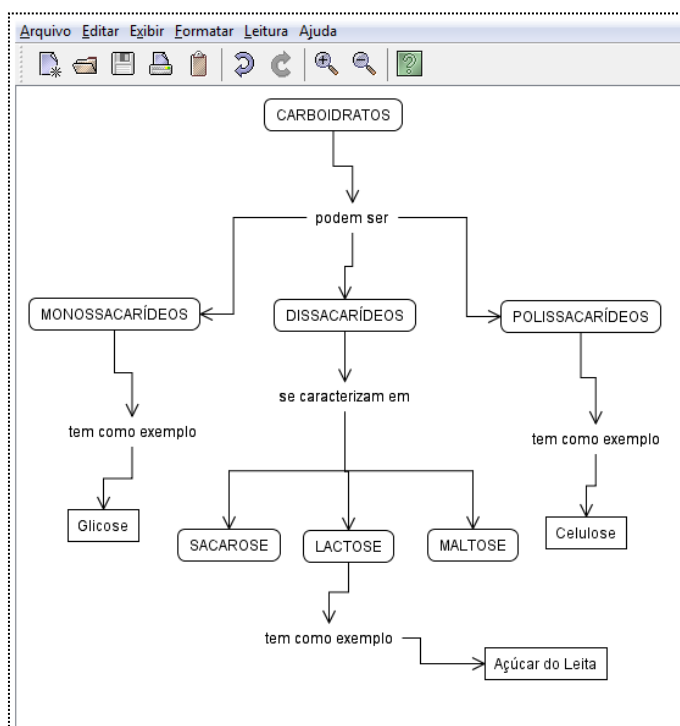
A estrutura hierarquizada destes diagramas é baseada nos conceitos de *subsunção* definido na teoria da AS de Ausubel em que uma nova informação se relaciona e se integra a conceitos mais inclusivos (NOVAK; GROWIN, 1999).

O MC como ferramenta cognitiva pode ser desenhado tanto de forma manual (com o uso de um papel e um lápis) como de forma automática com o uso de ferramentas de apoio que facilitem a aprendizagem (Seção 2.6). Frisa-se que MCs estão cada vez mais sendo utilizados

para propósitos de: suporte para navegação em hipertexto (NOVAK, 2010), resolução de problemas; ferramenta para análise de currículo, apresentação de um assunto, recurso para avaliação da aprendizagem, estruturação de um índice ou sumário e outras aplicabilidades como no ensino de diferentes níveis escolares, incluindo o Jardim de Infância (MANCINELLI et. al., 2004).

A Figura 9 mostra um mapa conceitual sobre o Carboidrato da Biologia, desenhado no *WebCMEditor* (Seção 3.2.1).

**Figura 9. Exemplo de um mapa conceitual do carboidrato no *webcmeditor***



Fonte: (Elaborada pelo autor).

Este mapa, de acordo com o modelo proposto por Novak, contém os conceitos:

{CARBOIDRATOS, MONOSSACARÍDEOS, DISSACARÍDEOS, POLISSACARÍDEOS,  
SACAROSE, LACTOSE e MALTOSE}

As proposições:

{<CARBOIDRATOS podem ser MONOSSACARÍDEOS>, <CARBOIDRATOS podem ser  
DISSACARÍDEOS>, <CARBOIDRATOS podem ser POLISSACARÍDEOS>},

<DISSACARÍDEOS se caracterizam em SACAROSE>, <DISSACARÍDEOS se caracterizam em LACTOSE>, <DISSACARÍDEOS se caracterizam em MALTOSE>}

E, os exemplos:

{Glicose, Celulose e Açúcar do Leite}

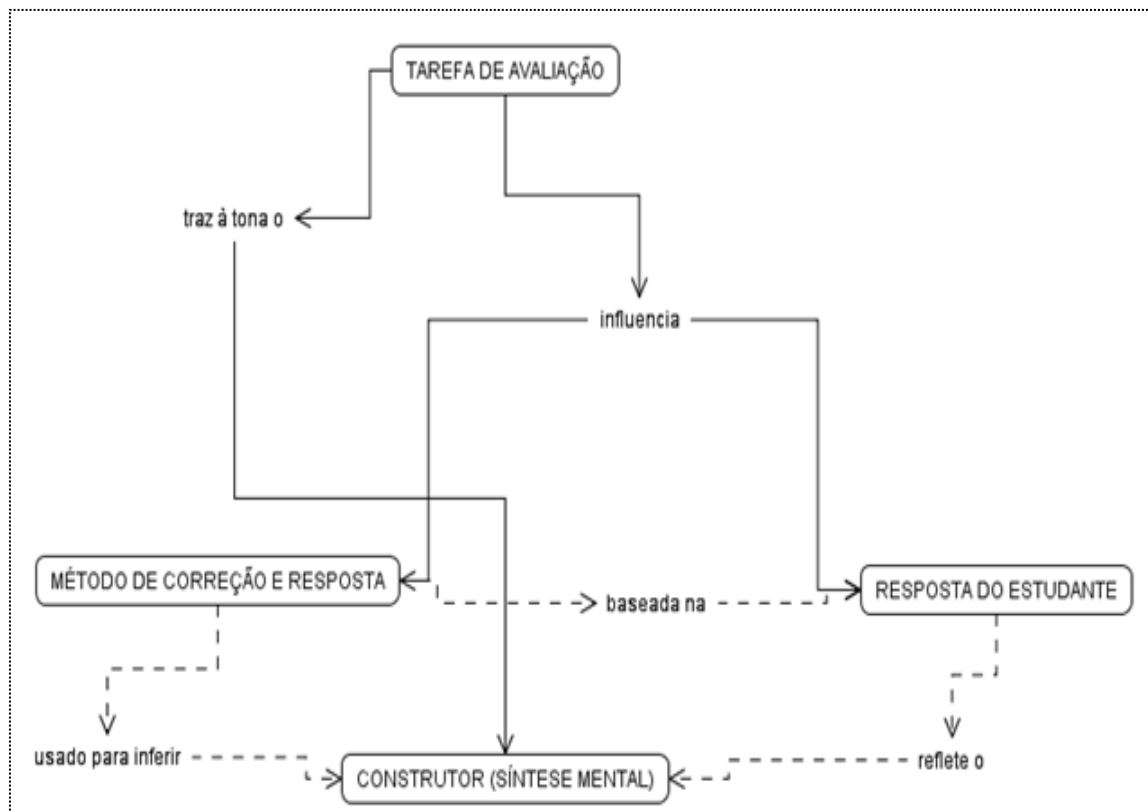
O conceito CARBOIDRATO é um exemplo de conceito mais inclusivo, enquanto que os conceitos MONOSSACARÍDEOS, DISSACARÍDEOS e POLISSACARÍDEOS são conceitos menos inclusivos, ou seja, mais específicos a ele na estrutura do mapa. A próxima seção descreve a avaliação da aprendizagem, utilizando mapas conceituais.

### **2.3.1 Avaliação da aprendizagem com mapas conceituais**

Entre os períodos de 1996 a 2004, pesquisas foram feitas para investigar e testar os mapas conceituais com base no conhecimento estrutural de um estudante e como uma ferramenta de avaliação (RUIZ-PRIMO e SHAVELSON, 1996; RUIZ-PRIMO et. al., 1996; RUIZ-PRIMO et. al., 1997; RUIZ-PRIMO et. al., 2001; RUIZ-PRIMO et. al., 2001; RUIZ-PRIMO e SHAVELSON, 2004).

À medida que Ruiz-Primo e Shavelson (2004) demonstravam claramente como ocorre a avaliação quando se utiliza MCs como o principal mediador, também ressaltavam que a avaliação da aprendizagem é caracterizada por: a) uma tarefa de avaliação que estimula o estudante a mostrar detalhes da estrutura de seu conhecimento; b) um formato para a resposta do estudante; e, c) um modelo de correção por meio do qual o MC do estudante pode ser avaliado com precisão e consistência. Tais características podem ser evidenciadas no MC da Figura 10.

Figura 10. Componentes da avaliação da aprendizagem

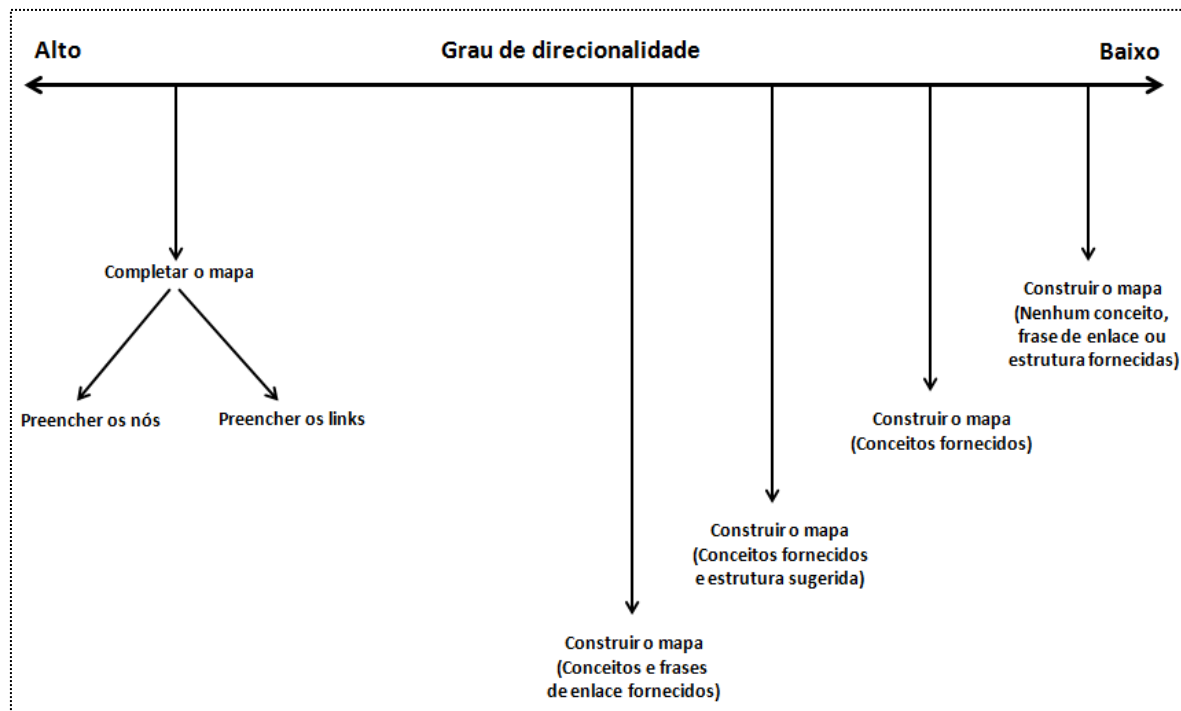


Fonte: (Adaptado de ROCHA, 2007).

Em relação a esta Figura 10, TAREFA DE AVALIAÇÃO pode ser caracterizada por diferentes técnicas de mapeamento, selecionadas ao longo de um *continuum*. Para tarefas de alto grau de direcionalidade, pode-se usar a técnica *fill-in-the-map* (semelhante a *fill-in-the-blanks*), em que são fornecidos ao estudante os conceitos, as conexões entre os conceitos, as frases de enlace e a estrutura do mapa, restando a este completar o preenchimento do mapa com os conceitos, as ligações e as frases de enlace onde estes elementos estiverem ausentes. Por outro lado, para tarefas de baixa direcionalidade, pode-se usar a técnica de mapeamento conceitual conhecida por *construct-a-map* - os estudantes ficam livres para decidir quantos e quais conceitos colocar no mapa, quais conceitos são relacionados e que palavras de enlace usar para explicar a relação entre conceitos relacionados (ROCHA, 2007).

A Figura 11 mostra as técnicas de mapeamento conceitual em alguns pontos de um *continuum*, onde a direcionalidade refere-se ao grau de conhecimento disponível para o aluno construir seu mapa conceitual.

Figura 11. Técnicas de mapeamento conceitual em diferentes pontos num *continuum*



Fonte: (RUIZ-PRIMO, 2004).

Adicionalmente às ideias de Ruiz-Primo, Rocha (2007) relembra os objetivos apresentados por Zeilik e seus colegas (ZEILIK et. al., 1997) sobre o processo de ensino-aprendizagem, sendo eles: a) aprender termos, fatos e conceitos sobre o assunto tratado; b) organizar informação em categorias significativas; c) sintetizar e integrar informação, ideias e conceitos; d) pensar sobre a estrutura geral e descobrir as conexões entre os conceitos; e) pensar criativamente sobre o assunto tratado; f) melhorar as habilidades de acesso ao conhecimento armazenado na memória de longo prazo; g) desenvolver as habilidades, as estratégias e o hábito de pensar de modo elevado (*high level thinking*); e, h) usar a linguagem gráfica de modo eficaz. Estes objetivos propostos podem ser alcançados com o uso dos seguintes tipos de testes:

- Mapa conceitual colaborativo: os estudantes trabalham em grupo (três ou quatro por grupo), negociando os significados que eles atribuem aos conceitos, procurando atingir um consenso ou demarcar pontos de vista diferentes.
- Mapeamento conceitual do tipo *fill in the blanks*: os estudantes recebem um MC com todos os *links* e somente as caixas sem os *labels* dos conceitos (os significados); esta é uma das tarefas mais difíceis, em que se procura determinar se o estudante tem o domínio da estrutura do assunto tratado.



- Mapeamento conceitual do tipo *fill in the blanks* parcial: os estudantes recebem um MC parcialmente completo, ou seja, contendo todos os links e alguns conceitos com seus *labels* (aproximadamente  $\frac{2}{3}$  dos conceitos). Os conceitos que faltarem para completar o MC podem ser colocados em lista de múltipla escolha; a hipótese em que se baseia este tipo de tese é que o pensamento do estudante é tão mais próximo do pensamento do professor na medida em que o conhecimento representado no seu MC se aproxima daquele do professor ou do especialista, conforme o caso.
- Mapeamento conceitual de termos selecionados: os estudantes recebem uma lista de *labels* (significados) (dez a vinte) para produzir seu próprio MC usando somente os elementos da lista. O foco aqui é na determinação da dimensão semântica que relaciona os conceitos (e, conseqüentemente, na escolha da palavra (frase) de enlace correspondente) e na evolução da estrutura cognitiva dos estudantes.
- Mapeamento conceitual baseado em sementes: nesta abordagem, também conhecida como *micromapping* (TROWBRIDGE; WANDERSEE, 1996), os estudantes recebem um pequeno conjunto de *labels* (cinco a dez) e completam o conjunto com um número igual de conceitos de sua própria escolha. Em seguida mapeiam conceitualmente o conjunto final.
- Mapeamento conceitual, baseado em escolha guiada: os estudantes recebem uma lista de vinte *labels* de conceitos e selecionam dez para construir os seus MCs. Esta atividade repetida diversas vezes num período de tempo mostra que os conceitos que aparecem e desaparecem em cada uma das versões do mesmo MC. A hipótese de base é que estas mudanças representam reestruturações significativas da estrutura cognitiva dos estudantes.

Os mapas conceituais servem como técnicas de avaliação da aprendizagem que ajudam a eliminar algumas dificuldades encontradas na avaliação (CUNHA et. al., 2008), como por exemplo, conceitos mal formados (CAÑAS et. al., 1999) ou ainda conceitos verdadeiros e relacionamentos relevantes entre conceitos.

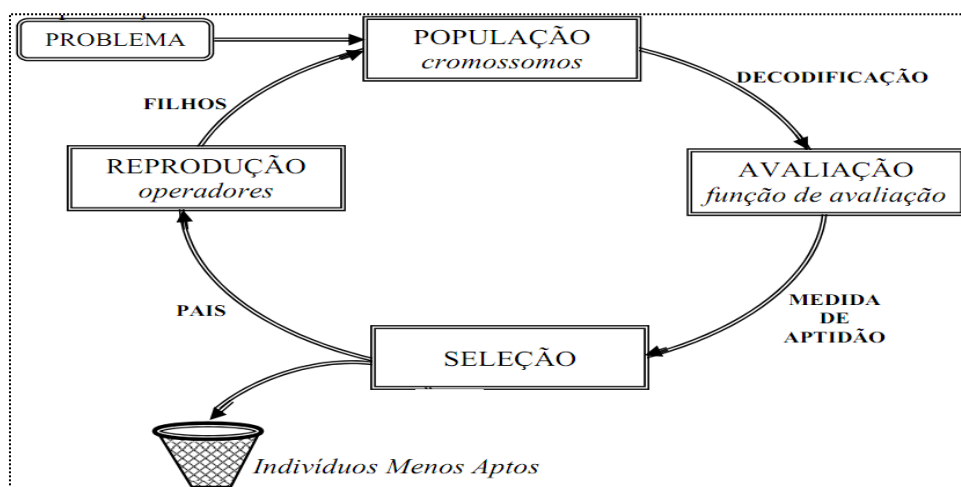
## 2.4 ALGORITMOS GENÉTICOS

Entre os estudos tecnológicos que enriqueceram a segunda metade do século XIX, há um que merece destaque que é a Teoria da Seleção Natural, proposta por Darwin (1994). Esta teoria é um dos mais importantes princípios no ramo da evolução, pois defende a ideia de que na natureza, aqueles seres vivos que melhor se adaptam tendem a sobreviver. A partir desse marco, os fundamentos da teoria evolucionista se constituem, nos princípios que norteiam as pesquisas sobre o desenvolvimento das espécies ao longo da vida na Terra (LOBO, 2005).

Neste contexto, um estudante de doutorado da Universidade de Michigan, chamado John H. Holland, propôs um método computacional para simular o processo de evolução das espécies, denominado de algoritmo genético (HOLLAND, 1992). Para Rocha (2007), o algoritmo genético (AG) é uma técnica de busca que permite a convergência em direção a melhor solução de um problema, dado um espaço de busca com soluções candidatas.

Entretanto, do ponto de vista computacional, um AG é um procedimento iterativo que mantém uma população de soluções candidatas,  $P(t)$  (ROCHA, 2007). Durante cada passo da iteração, os indivíduos da população corrente são avaliados quanto à sua adaptabilidade e, com base nesta avaliação, uma nova população é formada. Dessa forma, o AG começa trabalhando com uma população inicial,  $P(0)$ , que pode ser formada por regras heurísticas ou com métodos de seleção como: elitismo, roleta, estado fixo, *rank* ou torneio (TEIXEIRA, 2005). A Figura 12 mostra os componentes presentes no algoritmo genético e seu processo iterativo, sendo que a reprodução normalmente se processa por meio de operadores de cruzamento e mutação, segundo regras apropriadas.

**Figura 12. Componentes de um algoritmo genético**



Fonte: (TEIXEIRA, 2005).

Cada tipo de problema possível de ser resolvido por meio de algoritmos genéticos envolve um tipo de codificação dos indivíduos da população e um significado para os cromossomos desta mesma população. O Quadro 1 apresenta estas informações para alguns tipos de problemas.

**Quadro 1. Indivíduos de uma população: principais tipos de codificação**

<b>Codificação/Decodificação</b>	<b>Exemplo de Problema</b>	<b>Descrição</b>
Codificação Binária ( <i>Binary Encoding</i> )	Problema da Mochila	Cada bit é usado para dizer se o objeto correspondente está ou não na mochila
Codificação por Permutação	Caixeiro Viajante, Baseados em Ordem.	Os cromossomos descrevem a ordem em que o caixeiro visitará as cidades
Codificação de Valores	Cálculo de Pesos para uma Rede Neural	Valores reais dos cromossomos representam os pesos da rede neural.
Codificação em Árvore	Encontrar uma função que aproxime um dado par de valores	Cromossomos são funções representadas em uma árvore

Fonte: (Elaborado pelo autor)

A codificação binária é a mais comum entre as várias formas de codificação. Isto se deve, entre outros motivos, porque foi a codificação usada pelos primeiros pesquisadores devido à sua relativa simplicidade. Na sua estrutura, cada cromossomo é uma série de bits 0 ou 1, conforme mostra a Figura 13.

**Figura 13. Exemplo de cromossomo através da codificação binária**

<b>Cromossomo A</b>	<b>101100101100101011100101</b>
<b>Cromossomo B</b>	<b>111111100000110000011111</b>

Fonte: (TEIXEIRA, 2005).

A forma como as soluções candidatas são codificadas em um problema é importante para qualquer método de busca. Contudo, a maioria das soluções de AG utilizam sequências de bits (Bit String) de tamanho e ordem definidos. No entanto, vários experimentos têm sido feitos com outras formas de codificação neste trabalho, por exemplo, o AG apresentado contém soluções candidatas codificadas considerando que os indivíduos são mapas conceituais e não simples sequência de bits.

Apesar das facilidades de manipulação, o que permite ser aplicada a problemas mais simples, a codificação binária não é natural e é incapaz de ser aplicada a muitos problemas, sendo este tipo de codificação propício a ordenações arbitrárias, o que em alguns casos não é desejado.

A Codificação por permutação é muito utilizada em problemas de arrumação como o Problema do Caixeiro Viajante (PCV) e em outros que envolvem ordenação de tarefas. Nesta codificação, cada cromossomo é uma série de números que representam uma posição em uma sequência, conforme descreve a Figura 14.

**Figura 14. Exemplo de cromossomo através da codificação por permutação**

<b>Cromossomo A</b>	<b>3 4 1 2 5 9 8 7 6 3</b>
<b>Cromossomo B</b>	<b>3 8 2 9 1 6 4 7 5 3</b>

Fonte: (TEIXEIRA, 2005).

É comum, ao utilizar este tipo de codificação, que ajustes sejam feitos nas operações de cruzamento e mutação com o objetivo de manter a consistência do cromossomo utilizado.

Na Codificação de valores, cada cromossomo é uma sequência de alguns valores, podendo ser números reais, caracteres ou qualquer outro objeto, como mostra a Figura 15. É comum notar, quando se usa a Codificação de Valores em alguns problemas, a necessidade de desenvolver um método de cruzamento e mutação=específicos para o problema.

**Figura 15. Exemplo de cromossomo através da codificação de valores**

<b>Cromossomo A</b>	<b>1.2324 5.3243 0.4556 2.3293 2.4545</b>
<b>Cromossomo B</b>	<b>ABDJEIFJDHDIERJFDLDFLFEGTAZ</b>
<b>Cromossomo C</b>	<b>(voltar), (voltar), (direita), (a frente), (esquerda)</b>

Fonte: (TEIXEIRA, 2005).

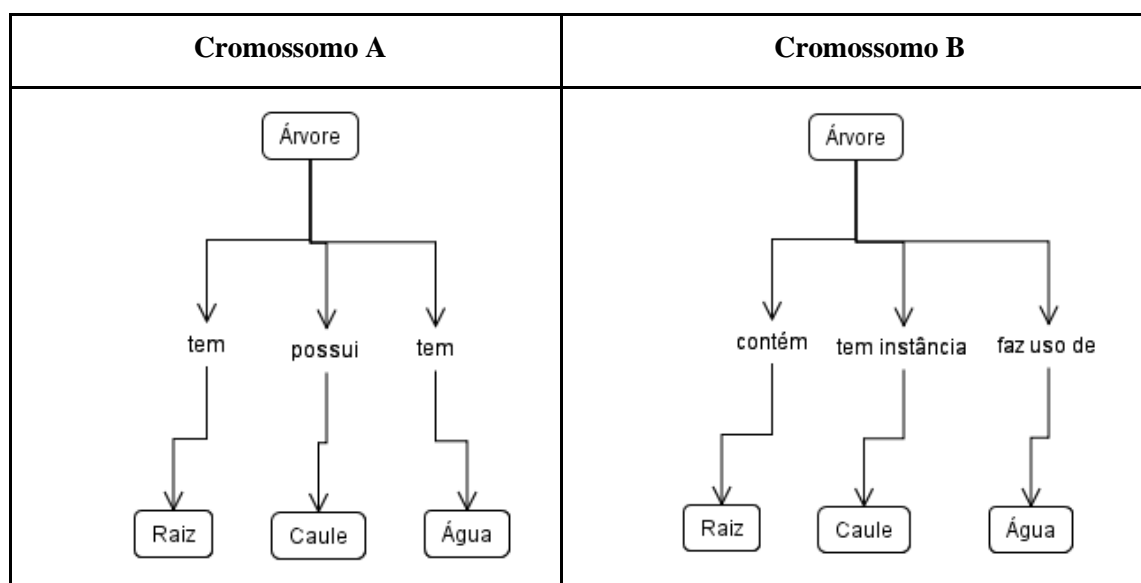
Embora Holland tenha afirmado que a performance dos AGs, ao utilizar a codificação de valores, seja pior do que com a codificação binária, algumas recentes comparações empíricas mostraram que isso não é bem assim. Apesar disso, a performance do algoritmo depende muito do problema que está sendo solucionado, e também do AG utilizado (TEIXEIRA, 2005).

O último, porém não menos importante, é a Codificação em árvore utilizada, por exemplo, para desenvolver programas de computador e expressões matemáticas. Nesta codificação, cada cromossomo é uma árvore de objetos, podendo estes objetos serem funções, comandos de linguagem de programação ou proposições de conceitos de um MC. A Figura 16 exemplifica esta codificação.

A Codificação em árvore tem a vantagem de permitir que o espaço de busca seja irrestrito, pois, a princípio, qualquer tamanho de árvore pode ser formado pelo uso de cruzamento e mutação. Atualmente, são desconhecidos guias que apresentem rigorosamente qual das codificações é a melhor. O problema em questão determina a codificação que será mais bem utilizada para que se gerem resultados ótimos.

Após a definição da codificação das soluções candidatas, é necessária, segundo Mitchel (1999), a escolha do método de seleção apropriado para os cromossomos serem selecionados ou reproduzidos conforme a iteração da Figura 12. Esta etapa é importante para que os cromossomos estejam prontos e selecionados para as operações de cruzamento (TEIXEIRA, 2005).

**Figura 16. Exemplos de cromossomos através da codificação em árvore**



Fonte: (Elaborada pelo autor).

A seleção enfatiza os melhores indivíduos de uma população com o intuito de produzir descendentes mais bem adaptados (maior valor de *fitness*). A operação de seleção, de acordo com Whitley (1994), escolhe cromossomos da população para a reprodução, utilizando na maioria das vezes a adaptabilidade do cromossomo, de modo que os indivíduos mais aptos sejam selecionados mais de uma vez, da mesma forma como ocorre nos mecanismos da seleção natural.

Para Mitchell (1999), o processo de seleção deve ser balanceado com a variação do cruzamento e da mutação. Métodos de seleção muito fortes são sinônimos de predominância na população, de indivíduos com valor de adaptação altamente sub-ótimo, reduzindo assim a diversidade necessária para provocar mudanças e progresso na população. De outra forma, métodos muito fracos provocam certa lentidão no processo evolucionário (TEIXEIRA, 2005).

O processo de seleção natural apresentado por Darwin é representado de forma artificial pelo operador de seleção de acordo com Goldberg (1989), entretanto, neste caso a sobrevivência é do mais apto entre criaturas representadas por sequência de caracteres. O algoritmo geral que define este processo é:

### **Algoritmo**

1. Cria a população inicial  $P(t)$  em  $t = 0$ ; esta população inicial é considerada a população corrente;
2. Aplica seleção: extraem da população corrente os melhores indivíduos; eles serão os pais da próxima população;
3. Aplica o cruzamento: os melhores indivíduos selecionados são combinados dando origem a uma prole;
4. Aplica mutação: para dar variabilidade genética à prole formada no passo 3. Partes dos seus componentes são modificadas; cria-se, assim, uma população descendente da população ancestral corrente; no novo passo de iteração, a população descendente se torna a nova população corrente;
5. Volta ao passo 2.

Esta seção descreve o AG e seus principais elementos no processo de reprodução e evolução de indivíduos de uma população corrente. Na Seção 3.2.3 é apresentado o módulo Algoritmo Genético para Tipos Abstratos de Dados aplicados a Mapas Conceituais (*GAADT-*

CM), utilizado neste trabalho como viabilizador do processo de avaliação no ambiente *WebCMTool*, que utiliza a codificação em árvore.

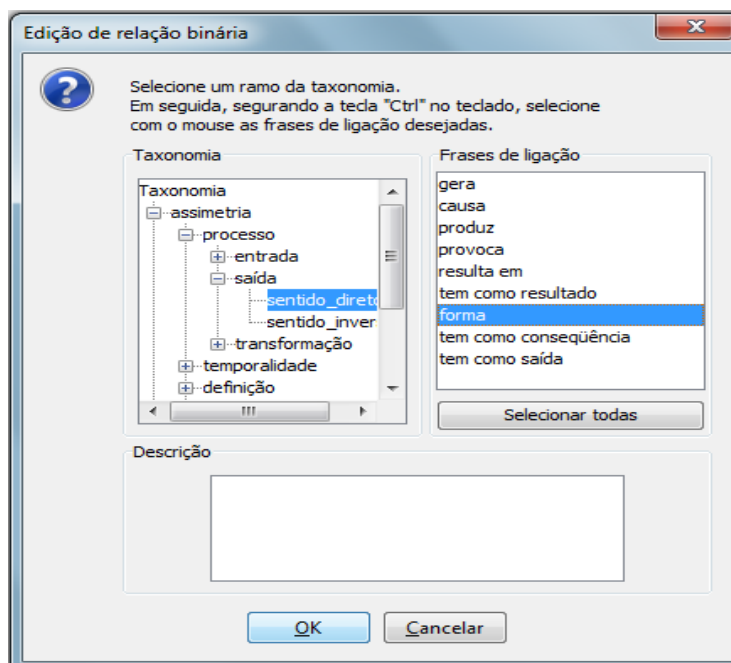
## 2.5 TAXONOMIA DAS FRASES DE LIGAÇÃO

A taxonomia de frases de ligação do ambiente, ou simplesmente taxonomia, é uma coleção de frases de ligação, organizadas hierarquicamente, segundo determinadas dimensões semânticas, ou ordens de significados. Usando um linguajar próprio da área de computação, pode-se dizer que a taxonomia do ambiente é organizada em supertipos que englobam outros supertipos, formando uma árvore cuja profundidade é determinada pela dimensão semântica que estiver sendo descrita. Esta taxonomia possibilita, segundo Rocha, (2007): “distinguir duas proposições diferentes, embora próximas, e atribuir valores de distância semântica proporcionais à diferença em significado entre proposições que estejam sendo comparadas”.

A taxonomia é dividida em dois ramos: o simétrico e o assimétrico. O simétrico contém as dimensões semânticas que servem para classificar as relações que possuem o mesmo significado lidas no sentido inverso ou direto; o assimétrico possui as dimensões semânticas que permitem escrever proposições que admitem inversas com mesma carga semântica, mas com mudança no grau de inclusividade (Rocha, 2007). Cada um desses ramos englobam diversos supertipos que serão utilizados para determinar uma dimensão semântica que conecta os conceitos em uma ontologia. Esta mesma taxonomia é usada para selecionar a frase de ligação adequada para relacionar conceitos no MC. A Figura 17 apresenta a interface da taxonomia.

Esta interface aparece quando o usuário abre a taxonomia das frases de ligação, durante o processo de escolha de uma frase após conectar dois conceitos em uma ontologia ou em um mapa conceitual. Ao desenhar um MC, o estudante seleciona a frase de ligação para conectar dois conceitos e ao desenhar uma ontologia o professor determina a dimensão semântica que irá conectar os conceitos. O professor poderá escolher a(s) frase(s) de ligação que ele considera mais adequadas para representar a dimensão semântica da ontologia, e essas frases são favorecidas pelo algoritmo genético da ferramenta.

**Figura 17. Taxonomia das frases de ligação do ambiente *webcmtool***



Fonte: (Elaborada pelo autor).

No caso dos MCs o estudante seleciona a frase de ligação que ele considera correta na ligação dos conceitos. Conforme pode ser observado na Figura 9, como exemplo, para conectar o conceito MONOSSACARÍDEOS e o exemplo Glicose foi selecionado na taxonomia da frase de ligação a frase “tem como exemplo”.

## 2.6 ONTOLOGIAS DE DOMÍNIO

Desde o século XVII, o termo “ontologia” segundo a “Filosofia de Aristóteles”, tem sido denominado como um ramo da Metafísica que estuda a existência, suas propriedades e classificações (LAYCOCK, 2006). Sendo, muitas vezes, encarada como um complemento à ideia de Epistemologia (ciência do conhecimento) (FREITAS et. al., 2009).

Para Guarino (1998), uma ontologia é definida como uma maneira de se conceitualizar de forma explícita e formal os conceitos e restrições relacionados a um vocabulário específico, sendo, frequentemente, restrita ao que se chama “ontologia formal”.

Carneiro e colaboradores (2005) definem ontologia como “um modelo abstrato usado para representar conceitos de forma objetiva”, dentro de um vocabulário de representação que captura esses conceitos, as relações entre os conceitos e propriedades de um domínio e um conjunto de axiomas, que restringem sua interpretação (FREITAS et. al., op. cit.).



Nas comunidades de inteligência artificial, engenharia de software e web semântica, o termo ontologia é, em geral, utilizado: (i) como artefatos concretos destinados a propósitos específicos sem muito cuidado ou atenção a questões fundacionais e (ii) como uma representação de um domínio específico (p. ex. programação genética, química orgânica, informática na educação ou política). Esse tipo de ontologia, denominada ontologia de domínio, é utilizado como base de conhecimento do domínio, dando suporte à tarefa de modelagem conceitual para sistemas do referido domínio (GUIZZARD et al., 2008).

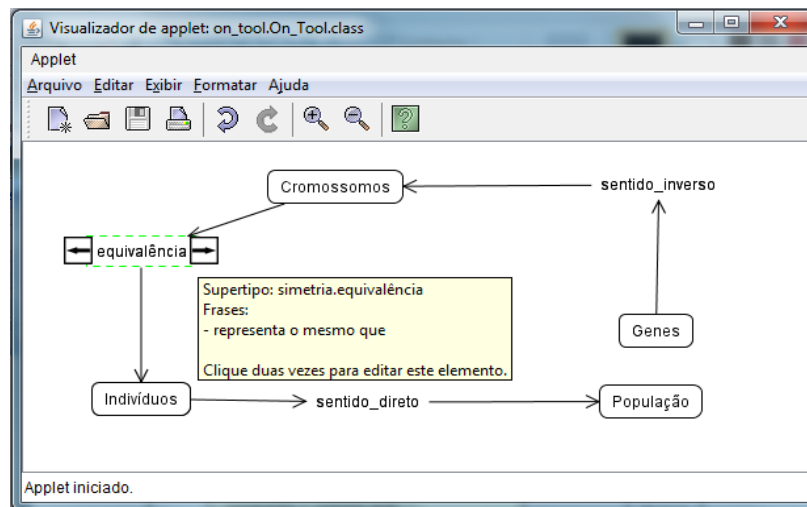
Ontologias de domínio podem ser aplicadas em várias áreas do conhecimento, como por exemplo: (a) na representação de modelos *topic maps*, dirigido por ontologia aplicada à disciplina de requisitos de um processo unificado (FARIAS; RODRIGUES, 2009); (b) para validação de diagramas de classes de domínio, baseado em análise ontológica (TAVARES, 2008); (c) no apoio a web semântica com uma linguagem ontológica (MCGUINNESS et al., 2002); (d) na melhora e cooperação em estudos ambientais em áreas produtoras de petróleo e gás-natural (SOUZA et al., 2005); e, por fim, (e) sendo geratrizes de proposições agrupadas, posteriormente, por um algoritmo genético para formar MCs (ROCHA, op. cit.).

Neste trabalho, ontologias são especificações formais de conceitos (vocabulário de um domínio) e dimensões semânticas de relacionamentos entre estes conceitos (ROCHA, op. cit.), estando estas dimensões de relacionamentos organizadas em uma taxonomia de frases de ligação especificada por Costa Jr. e colaboradores (2004).

As ontologias descritas no *WebCMTool*, conforme detalhado no capítulo 3, geram proposições que, quando agrupadas, formam MCs. O AG, especificado no ambiente, utiliza como uma de suas entradas uma ontologia de domínio para a tarefa de avaliação da aprendizagem. O professor, então, constrói as ontologias com o editor gráfico de ontologias, denominado *WebOn\_Tool* (Seção 3.2.2), permitindo sua exportação em arquivos XML, levando em conta as características presentes em uma gramática de inclusividade descrita em (Rocha, op. cit.). A Figura 18 mostra uma ontologia sobre o domínio cromossomo, construído no módulo *WebOn\_Tool* do ambiente.

O objetivo da ontologia descrita é, principalmente, estabelecer o domínio de MCs passíveis de serem criados, ou seja, todos os mapas conceituais que podem ser criados a partir da ontologia formam o espaço de busca do algoritmo genético e o contexto para a tarefa de aprendizagem dos estudantes.

**Figura 18. Ontologia sobre o cromossomo no *webon\_tool***



Fonte: (Elaborada pelo autor).

Na ontologia da Figura 18, a dimensão semântica selecionada para conectar os conceitos <<Cromossomos>> e <<Indivíduos>> é representada pela cadeia *simetria.equivalência*. Na ontologia aparece, simplificada, apenas o elemento final da dimensão semântica, neste caso “equivalência”, mas, ao colocar o mouse sobre a dimensão, é possível verificar toda dimensão semântica, cuja representação textual aparece abrangida pelo retângulo em amarelo na figura.

## 2.7 ESTADO DA ARTE: AVALIAÇÃO AUTOMÁTICA DA APRENDIZAGEM COM MAPAS CONCEITUAIS

No contexto da aprendizagem, a tarefa de avaliação é uma das atividades que mais sobrecarrega o professor, principalmente em países - como no Brasil – em que o professor especialista em avaliação não existe no âmbito escolar (ROCHA, 2007). A avaliação automática de questões do tipo verdadeiro-falso, combinação e múltipla-escolha não apresentam dificuldades para a automação do processo de correção. Entretanto, quando o cenário se refere a questões do tipo resposta-curta e ensaio, pesquisas nas áreas de processamento de linguagem natural (PLN) e de recuperação da informação (RI) se tornam necessárias, principalmente com o uso de algoritmos computacionais não triviais para auxiliarem nesta tarefa (HIRSCHMAN et. al., 2000; LANDAUER et. al., 2000).

A primeira pesquisa científica sobre o tema da avaliação de questões abertas foi realizada por Ellis Page em 1968 (PAGE, 1968), que reconheceu a pesada carga de trabalho do

professor e a possibilidade de usar programas de computador para avaliar textos escritos por estudantes. Para testar suas ideias, Page desenvolveu um sistema de correção e pontuação automática de ensaios denominada *Project Essay Grade* (PEG). O objetivo de Page era demonstrar a possibilidade de construir um sistema automático de correção da escrita que pudesse fazer o mesmo trabalho que um professor (ROCHA, op. cit.).

Seguindo a linha de pensamento de Page, Burstein e seus colegas (BURSTEIN, 1998) desenvolveram o *e-rater*, uma ferramenta capaz de avaliar analiticamente um texto com base em medidas mais diretas derivadas de qualidades gerais como variedade sintática, estruturação e conteúdo de tópicos e organização de ideias. O objetivo de Burstein era testar características linguísticas que representassem mais diretamente estas qualidades gerais e que pudessem ser automaticamente extraídas do texto, usando técnicas de PLN e RI (ROCHA, 2007).

Adicionalmente ao trabalho de Page (1968), MacDonald (1982) foi o primeiro a tratar de correção ortográfica, estilo e legibilidade, posteriormente, na segunda metade da década de 1990, surgiram outras contribuições para o avanço da avaliação automática de textos. Exemplos destas contribuições podem ser encontrados nos trabalhos de Lahan (2000), Chodorow e Leacock (2000), Miltsakaki e Kukich (2000) e, Bennet e Bejar (1998).

No contexto de questões abertas, a abordagem de MCs caracteriza-se como uma metodologia voltada para a avaliação que permite uma visão mais detalhada do aprendizado do estudante. Assim, na escola behaviorista (escola tradicional), por exemplo, grande parte de pesquisas sobre o uso de mapas conceituais na avaliação da aprendizagem repetem o esquema de comparar mapas de estudantes a mapas de referência, com o objetivo de quantificar a aprendizagem demonstrada pelo estudante. Os trabalhos de Bogden (1977), Diekhoff e Diekhoff (1982) e, Novak e Gowin (1984) são exemplos menos recentes desta abordagem. Além disto, o uso pelo computador no processo de avaliação mediada por mapas conceituais está ligado, segundo Rocha (2007), às possibilidades que se abrem de focar a avaliação com base no crescimento individual de construção de conhecimento que é desenvolvido pelo estudante, ou seja, nas mudanças conceituais que ele revela no decorrer deste processo, no seu aprender a aprender, relativizando a construção de mapas como produto de grupos que convergem/não convergem na direção de um mapa ideal, gerado por um professor ou por um perito (BEYERBACH e SMITH, 1990).

Na literatura, a avaliação automática da aprendizagem com mapas conceituais tornou-se cada vez mais frequente, seja por meio de ferramentas ou em Ambientes Virtuais de Aprendizagens (AVA), como determinado por Caldas e colaboradores (2009), que propõem

uma ferramenta integrada na plataforma à distância que utiliza técnicas de Inteligência Artificial (n-gramas e *KNN*) que realiza uma avaliação tanto quantitativa quanto qualitativa de questões abertas por meio de MCs. Ou ainda Machado et. al. (2004) que apresentaram um sistema de avaliação aplicado a Ciências Exatas para cobrir tanto questões fechadas quanto questões abertas por meio de um ambiente que permita a manipulação de símbolos matemáticos. Ainda sobre avaliação, Isotani e colaboradores (2004) apresentaram uma ferramenta de avaliação automática de questões abertas em exercícios aplicados inicialmente à disciplina de Geometria.

Outros ambientes foram elaborados com o propósito de auxiliar a avaliação automática, com destaque a pesquisas de Araújo et. al. (2003) que mostram um ambiente para apoiar a avaliação da aprendizagem baseada em MCs, de modo a fornecer uma interação entre aluno e professor na tarefa de construção e avaliação destes mapas. Além disto, nos últimos 20 anos, ferramentas foram desenvolvidas como auxílio de atividades cooperativas e de construção de MCs, melhorando o processo de colaboração educacional, como por exemplo: *jKSI mapper*, *Concept Map Tools (Cmap Tools)* e *Inspiration*. Entretanto, nenhuma valoriza o aspecto idiossincrático da aprendizagem como o *CMTool*. Assim, esta pesquisa contribui cientificamente com a disponibilização do *CMTool* na web, permitindo que alunos possam avaliar seus próprios MCs baseados no espaço de busca de conceitos de uma ontologia previamente disponibilizada por um professor.

Entretanto, ainda é comum notar, no ceio científico, abordagens de comparação automática de mapas conceituais com mapas de referências feitos previamente por um especialista ou perito, como uma maneira de avaliação. Citem-se como exemplos os trabalhos de Araújo et. al. (2002), Araújo et. al. (op. cit.), Cunha e Fernandes (2002), Fernandes et. al. (2004), Santos Junior et. al. (2005). Contudo, a pesquisa de Rocha et. al. (2004) mostra que modificações no conteúdo, organização e avaliação dos mapas é um patamar já atingido, além de desenvolver e validar um algoritmo genético responsável pelo processo de avaliação que tenta provar individualmente que o mapa desenvolvido está correto com base numa ontologia definida pelo professor no interior de um domínio específico tratado.

Apesar destes avanços, todas as tecnologias atuais para a avaliação automática de ensaios deixam considerável espaço para melhorias. Mais especificamente e segundo ideias de Rocha (2007), espera-se que métodos possam avaliar, dar *feedbacks* crítico e sugestões para melhoria do texto em termos de detalhes da lógica, da sintaxe e da semântica ao nível da

construção de sentenças; e da clareza, da compreensibilidade e das qualidades afetivas (humor, suspense e evocatividade) ao nível da sentença, do parágrafo e da organização do texto.

Este capítulo apresentou os diferentes fundamentos teóricos sobre os quais se fundamenta a pesquisa apresentada neste trabalho. Neste sentido, foram tratados temas sobre a importância da *Web* para a educação, a AS, MCs e ontologias de domínio, além do responsável pela avaliação: os AGs. Também foi apresentado um estudo sobre o estado da arte da avaliação da aprendizagem envolvendo MCs.

A compreensão de temas tão variados quanto os que foram apresentados e a reunião deles num trabalho que procura contribuir para o avanço da avaliação da aprendizagem, quando esta é mediada por MCs, foi um dos maiores desafios do desenvolvimento desta dissertação.

### 3 WEBCMTOOL COMO FACILITADOR DA AVALIAÇÃO AUTOMÁTICA DA APRENDIZAGEM POR MAPAS CONCEITUAIS E ONTOLOGIAS DE DOMÍNIO

Neste capítulo são apresentadas os pontos-chaves de desenvolvimento do *WebCMTool*, destacando suas características, sua estrutura, arquitetura e funcionalidade. É mostrado como ocorre a avaliação de um MC e quais impactos para a Educação.

#### 3.1 INTRODUÇÃO

No ambiente *WebCMTool*, o algoritmo genético é o principal componente no processo de avaliação da aprendizagem na *Web*. Este algoritmo recebe como entrada três insumos básicos conforme mostra a Figura 19: a Ontologia, o Mapa Conceitual do Estudante (MCEst) e a Taxonomia. A saída do algoritmo apresenta como resultado mapas com significados equivalentes ao do mapa do estudante.

**Figura 19. Processamento do *gaadt-cm***



Fonte: (ROCHA, 2007).

O insumo ontologia representa uma ontologia de domínio que contém o conhecimento estrutural de um determinado contexto da aprendizagem no interior de uma disciplina. O MCEst representa o conhecimento presente na estrutura cognitiva do estudante, ou seja, seu entendimento sobre as relações entre os conceitos presentes na ontologia, haja vista que este mapa conceitual foi construído levando em conta os conceitos presentes na ontologia de domínio. A taxonomia, por sua vez, representa a taxonomia de frases de ligação (ver Seção 2.6), que é a estrutura de uma árvore que contém as dimensões semânticas e as frases que serão utilizadas pelo estudante para a construção do seu mapa conceitual.

O *WebCMTool* possui três perfis de usuários, são eles: administrador, professor e aluno. O perfil de administrador tem acesso a informações específicas do sistema bem como a informações relacionadas aos outros perfis de usuários, podendo este, por exemplo, incluir permissões ou tirá-las de algum aluno ou professor específico. O perfil de professor e o perfil

de aluno têm acesso restrito a dois editores no ambiente: o *WebOn\_Tool* e o *WebCMEditor*, respectivamente. O ambiente conta com um AG baseado em axiomas matemáticos para avaliação dos MCs construídos pelos alunos. O AG é auxiliado por um motor de inferências capaz de identificar relações implícitas entre conceitos da ontologia, as quais podem aparecer nos mapas apresentados pelos estudantes para avaliação.

No ambiente, a tarefa de aprendizagem consiste na construção de MCs (tarefa do aluno), baseado em uma ontologia de domínio construída (tarefa do professor), tendo o apoio dos mecanismos do sistema para explicitar a estrutura de conhecimento envolvida no assunto da ontologia utilizada. Ao avaliar o MCEst, o sistema emprega todos os recursos disponíveis para provar que o aluno está correto, construindo populações de MCs que sejam comparáveis ao mapa conceitual do estudante, o qual pode conter relações explícitas entre conceitos, diretamente presentes na ontologia, ou relações implícitas, inferidas da ontologia. Não há mapas de referências de um especialista ou do próprio professor no sistema para serem comparados com o mapa do estudante. Se houver, por exemplo, 1000 alunos presentes no sistema e cada um deles construir um mapa diferente o *WebCMTTool* deve ser capaz de avaliar cada um deles, individualmente, construindo as populações de mapas adequadas para cada caso. É desta forma que a idiossincrasia da aprendizagem de cada um é levada em conta. A seção que segue descreve a arquitetura destacando cada módulo presente no ambiente.

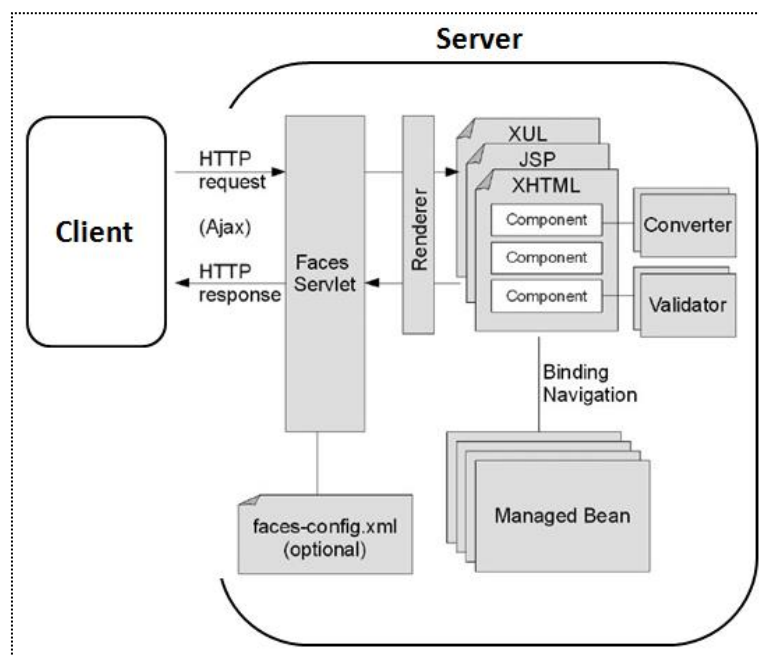
### 3.2 ARQUITETURA DO WEBCMTOOL

O *WebCMTTool* foi desenvolvido sobre a plataforma Java, utilizando, de um lado, a implementação *Mojarra* da especificação *JSF (JavaServer Faces)* para um *framework* de componentes para desenvolvimento *Web*, o que permitiu a utilização de todos os recursos padrões do *JSF*, como os componentes *HTML* de formulários, tabelas, *layout*, conversão e validação de dados e eventos, além de toda a inteligência para a interpretação dos arquivos e interação com o contêiner em Java. Por outro lado, foram utilizados editores escritos em *JApplet* para a edição e manipulação de MCs e ontologias de domínio.

O ambiente em si é um facilitador da aprendizagem significativa no contexto educacional e baseia-se na arquitetura de rede cliente/servidor conforme demonstra a Figura 20. Nesta arquitetura, as aplicações *JSF* utilizam interceptações *HTTP* via *Servlet Faces*, que é o principal *servlet* para a aplicação, produzindo, por exemplo, páginas *XHTML* por meio do *Renderer* que traduz a entrada de um valor do usuário em um componente específico para a

página, em que conversores são utilizados para converter o valor de um componente, dando-lhes novos formatos e validadores são utilizados para validar as entradas ocorridas no componente pelo usuário. Neste contexto, a lógica de negócios da aplicação é gerenciada pelos *managed beans*, que controlam inclusive a navegação entre páginas na camada de visualização.

**Figura 20.** Arquitetura cliente e servidor do *javaserver faces*



Fonte: (Adaptado de GONÇALVEZ, 2011).

Para a construção do ambiente, utilizaram-se regras e práticas do processo de desenvolvimento iterativo e incremental para gerenciamento de projetos e desenvolvimento ágil de software *Scrum*, com o intuito de gerar protótipos num tempo hábil e com uma qualidade garantida além de permitir o melhor gerenciamento entre os envolvidos. O fluxo de desenvolvimento do *Scrum* é mostrado pela Figura 21.

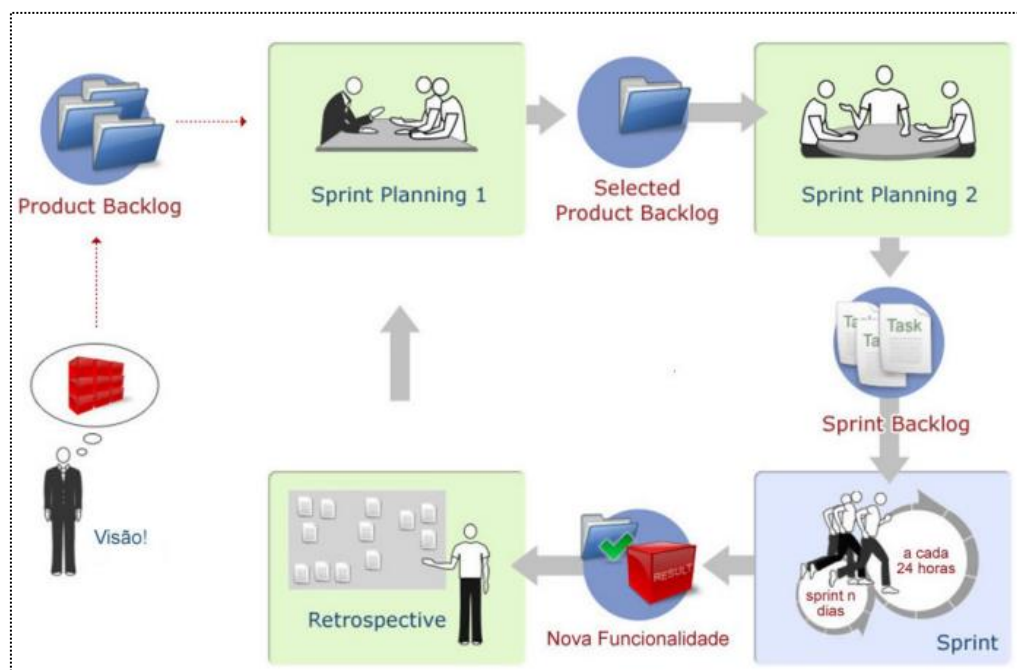
Um projeto no *Scrum* se inicia com uma visão do produto que será desenvolvido (SCHWABER, 2004). A visão contém uma lista das características do produto estabelecidas pelo cliente, além de algumas premissas e restrições. Em seguida, o *Product Backlog* é criado contendo a lista de todos os requisitos conhecidos. O *Product Backlog* é então priorizado e dividido em *releases*.

Todo o trabalho no *Scrum* é realizado em iterações chamadas de *sprints*, Schwaber (op. cit.) explica que cada *sprint* se inicia com uma reunião de planejamento (*Sprint Planning Meeting*), na qual o *Product Owner* e o time decidem em conjunto o que deverá ser



implementado (*Selected Product Backlog*). A reunião é dividida em duas partes. Na primeira parte (*Sprint Planning 1*), o *Product Owner* apresenta os requisitos de maior valor e prioriza aqueles que devem ser implementados. O time então define colaborativamente o que poderá entrar no desenvolvimento da próxima *sprint*, considerando sua capacidade de produção. Na segunda parte (*Sprint Planning 2*), o time planeja seu trabalho, definindo o *Sprint Backlog*, que são as tarefas necessárias para implementar as funcionalidades selecionadas a partir do *Product Backlog*. Nas primeiras *sprints*, é realizada a maioria dos trabalhos de arquitetura e de infraestrutura. A lista de tarefas pode ser modificada ao longo da *sprint* pelo time e as tarefas podem variar entre quatro a dezesseis horas para a sua conclusão (MARÇAL, 2009).

Figura 21. Visão geral do processo *scrum*



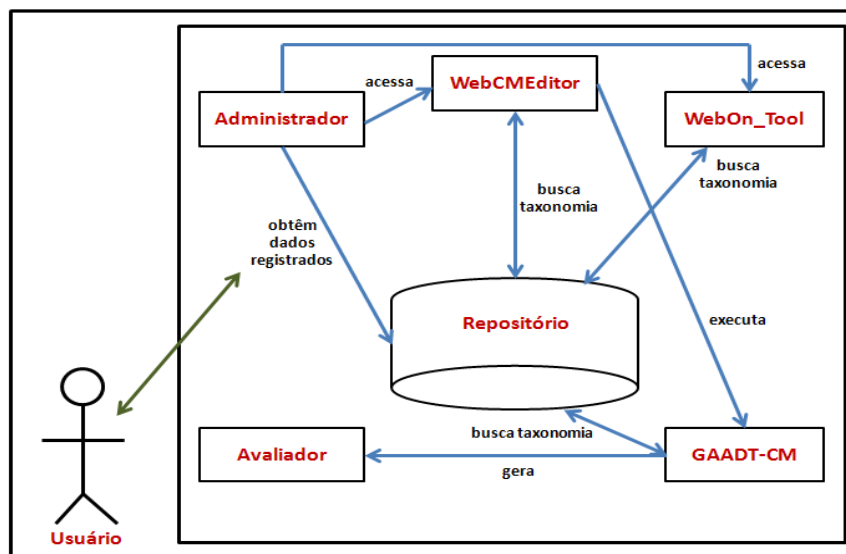
Fonte: (GLOGER, 2009).

O processo *Scrum* é composto por cinco cerimônias e três artefatos. As cerimônias são: Reunião de Planejamento do *Sprint* (*Sprint Planning*), Reunião de Planejamento de Entregas (*Release Planning*), Reunião Diária (*Daily Scrum*), Reunião de Revisão (*Sprint Review*) e Reunião de Retrospectiva (*Sprint Retrospective*). Os artefatos são: *Backlog* do *Sprint* (*Sprint Backlog*), *Backlog* do Produto (*Product Backlog*) e Gráfico *Burndown* (*Burndown Graph*). As cerimônias possuem baixo *overhead* para sua realização e podem ser realizadas independentemente de uma metodologia de trabalho, contudo, é necessário seguir a ordem na qual as mesmas são realizadas para organizar o trabalho de forma iterativa. A equipe de trabalho foi composta por quatro pessoas divididas nos seguintes papéis: um dono do produto,

dois programados e um mestre *Scrum* que atuou também como programador na equipe. O Apêndice A contém detalhes sobre este processo.

O *WebCMTool* é um *mindtool* (JONASSEN et. al., 1998) cujo objetivo principal é ajudar estudantes e professores a aplicar princípios construtivistas no decorrer do processo de aprendizagem. A Figura 22 destaca sua arquitetura de módulos.

Figura 22. Arquitetura de módulos do *webcmttool*



Fonte: (Elaborada pelo autor).

Por se tratar de um extensão, o *WebCMTool* apresenta uma arquitetura de módulos semelhante ao *CMTool* (ROCHA, 2007) havendo, entretanto, algumas diferenças (ver Tabela 3.1) peculiares para cada módulo.

Quadro 2. Comparação entre o *cmttool* e o *webcmttool*

Características	<i>CMTool</i>	<i>WebCMTool</i>
Possui editor para construir MCs e ontologias de domínio	X	X
Avalia MCs com um AG que valoriza a idiossincrasia	X	X
Permite que alunos avaliem seus MCs ao mesmo tempo		X
Utiliza o contexto da web para avaliação da aprendizagem		X
Contém um repositório com a taxonomia e usuários		X
Integra um GAADT-CM otimizado para avaliação na web		X
Integra o módulo administrador para acesso ao ambiente		X

Fonte: (Elaborado pelo autor).

As próximas subseções descrevem detalhadamente os módulos da arquitetura de acordo com a Figura 22, além de explicar o funcionamento no interior do ambiente.

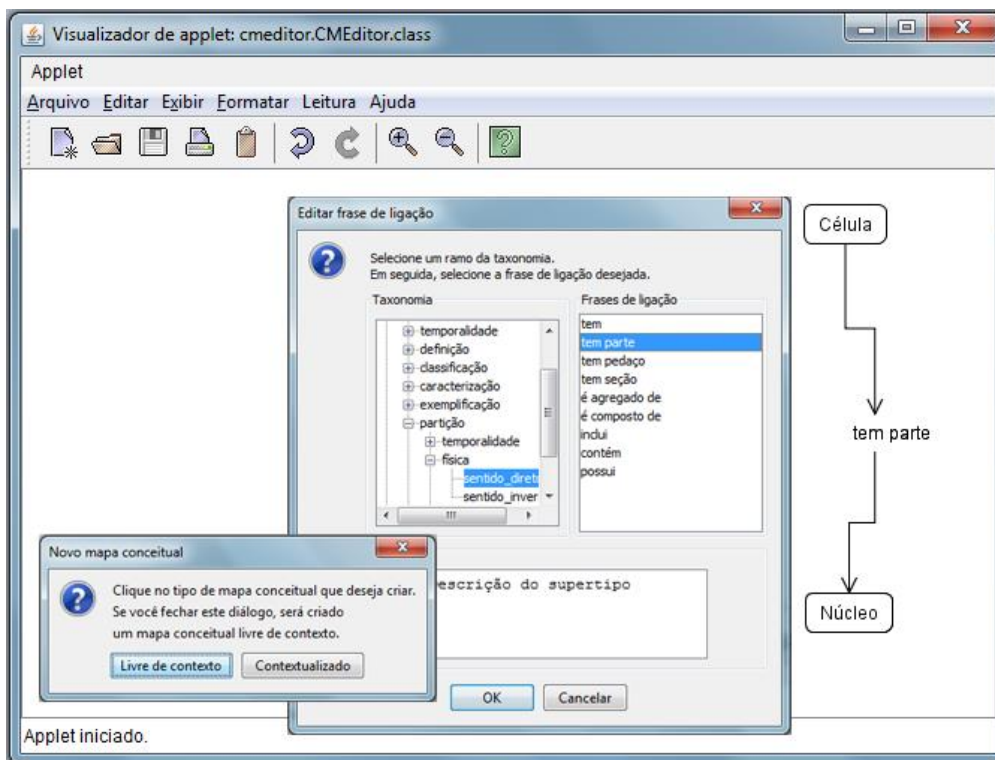
### 3.2.1 Módulo WebCMEditor

O *WebCMEditor* é o editor e manipulador de MCs do *WebCMTTool*. Este editor foi projetado com uma linguagem visual levando em conta técnicas de ergonomia e usabilidade (SILVA, 2006) cujo objetivo é facilitar a construção de MCs em conformidade com os princípios da Teoria da Assimilação (AUSUBEL, 2000). A Figura 23 mostra o aspecto visual do editor.

Basicamente, este editor pode ser usado para desenhar dois tipos de mapas: i) livres de contexto ou ii) contextualizados. Os mapas livres de contexto criam MCs para tarefas auxiliares ou preparatórias da aprendizagem como, por exemplo, desenhar mapas para escrita de artigos científicos ou ainda criar currículos profissionais, conforme demonstrado em Silva et al. (2005). Por outro lado, os mapas contextualizados abordam a construção de mapas com base em uma ontologia de domínio previamente fornecida pelo professor. A Figura 23 mostra, parcialmente, como construir um mapa contextualizado levando em conta conceitos presentes numa ontologia sobre a célula do corpo humano. Vale ressaltar que, durante a formação da primeira proposição, ou seja, <<Célula, tem parte, Núcleo>> foi utilizada uma taxonomia de frases de ligação específica que permitiu a escolha da frase de ligação apropriada para a formação da proposição (ver seção 2.4). A frase de ligação “tem parte” está armazenada em uma árvore de frases previamente definidas pela taxonomia, todas contidas em uma cadeia taxonômica específica, por exemplo, a frase da figura corresponde a cadeia *assimetria.partição.física.sentido\_direto*.

O módulo *WebCMEditor* é uma evolução do componente *CMEditor* do *CMTTool* (ROCHA, 2007), apresentando como principal progresso a sua execução em *JApplet* e o módulo de avaliação integrado ao próprio editor em que permite avaliar os MCs dos alunos. Assim, após a construção do mapa conceitual, o aluno submete seu mapa para avaliação no editor, que obtém mapas corretos próximos à ontologia de domínio desenhada pelo professor. A avaliação tenta sempre identificar o maior grau de proximidade semântica entre o mapa submetido e os mapas gerados pelo AG, tentando valorizar sempre a idiosincrasia do aluno e seu aprendizado.

**Figura 23. Editor *webcmEditor* durante a construção de um mc contextualizado**



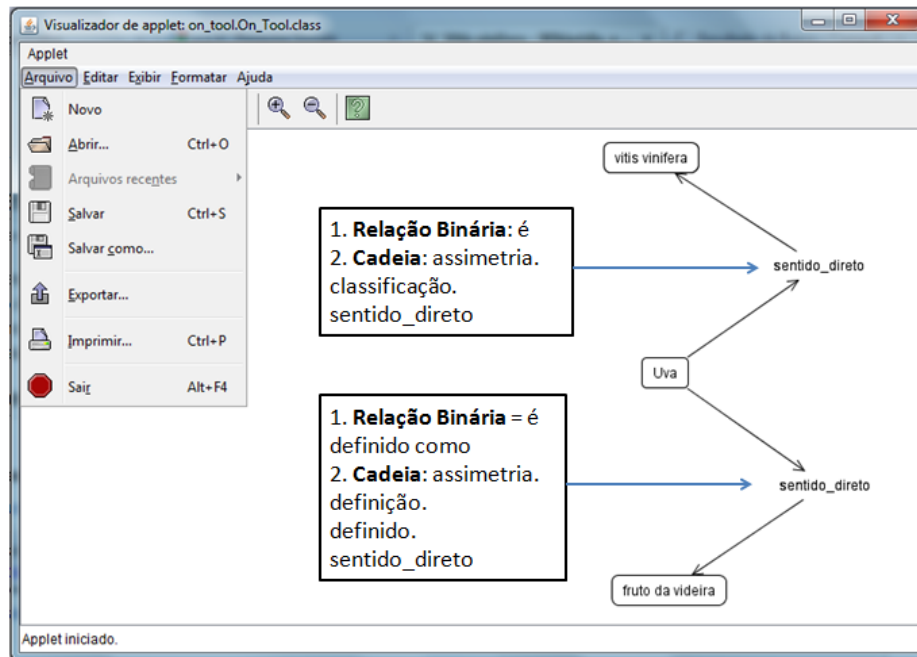
Fonte: (Elaborada pelo autor).

### 3.2.2 Módulo *WebOn\_Tool*

O módulo *WebOn\_Tool* é o editor e manipulador de ontologias de domínio do *WebCMTTool*. Ele foi projetado para ser similar ao *WebCMEditor* quanto ao aspecto visual, conforme mostra a Figura 24. Contudo, possui características específicas para a construção de ontologias, por exemplo, dois conceitos são ligados por uma relação binária e não por uma frase de ligação que forma uma proposição como ocorre no MC.

O *WebOn\_Tool*, cujo usuário é o professor, é usado para construir ontologias de domínio associadas às tarefas de aprendizagem colocadas no ambiente. Por exemplo, para desenhar uma ontologia de domínio, o professor desenha um diagrama semelhante a uma árvore (grafo) colocando nos vértices os conceitos e, nas ligações entre os vértices, frases previamente definidas com base na taxonomia de frases de ligação.

**Figura 24. Editor *webon\_tool* na construção de uma ontologia e suas relações binárias**



Fonte: (Elaborada pelo autor).

Uma ontologia é formada por relações binárias, diferente de um MC que é formado por proposições, ou seja, num mapa conceitual tem aprendizagem significativa representada, enquanto que numa ontologia há somente conhecimento para ser compartilhado (ROCHA, 2007).

Logo, quando o professor desenha uma ontologia com base em seu conhecimento sobre um determinado domínio, ele compartilha os conceitos deste conhecimento com o aluno, para que este possa criar proposições em seu mapa conceitual levando em conta os conceitos da ontologia e frases da taxonomia. Para melhor representar sua estrutura cognitiva, porém, o aluno não tem acesso às frases da relação binária escolhidas pelo professor, o que torna a tarefa de construção do MC muito difícil. Este modo de construção de mapas pelos alunos é conhecido por *construct a map* (Seção 2.4.1).

### 3.2.3 Módulo GAADT-CM

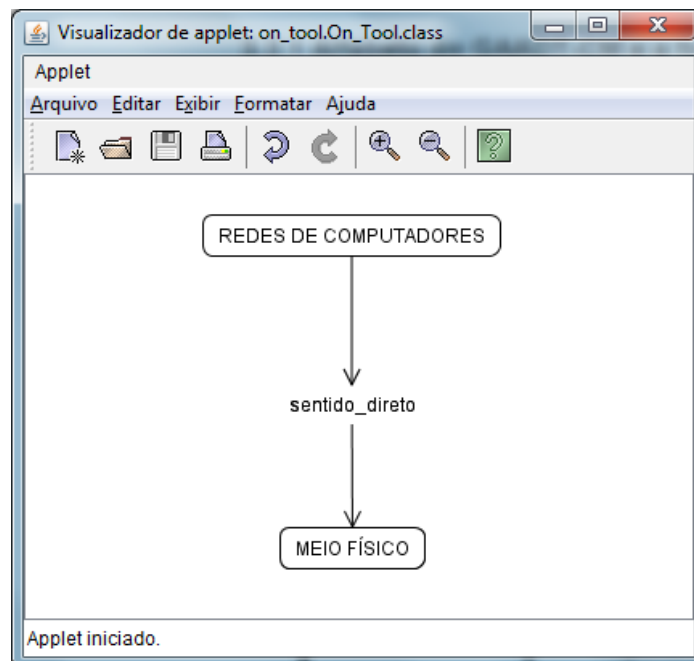
O módulo GAADT-CM<sup>4</sup> (*Genetic Algorithm based Abstract Data Types applied to Concept Maps*), incorporado ao ambiente, é a otimização do AG descrito no trabalho de Gusmão e Borges (2011) que é baseado na pesquisa seminal de Rocha e colaboradores (2004).

<sup>4</sup> A partir desta seção os termos GAADT-CM e AG serão considerados sinônimos

O *GAADT-CM* está definido em três níveis de abstração: base, gene e cromossomo, sendo a base o nível mais elementar composto por todos os conceitos definidos pela ontologia e frases usadas pelas proposições que podem ser construídas e usadas nos diversos mapas conceituais que são possíveis de serem gerados. Os genes, na linguagem da AS, são as proposições, ou seja, um gene é formado por dois conceitos e uma frase de ligação previamente definida pela taxonomia de frases de ligação. Por fim, os cromossomos (indivíduos da população), que são formados por um conjunto de genes representam os mapas conceituais na aprendizagem. O Diagrama de classes do *GAADT-CM* pode ser visualizado nos detalhes de implementação do Apêndice A.

Para ilustrar o funcionamento, considere-se que a ontologia simples da Figura 25 foi criada por um professor, o *GAADT-CM* determina um conjunto base B que contém todos os conceitos e relações (explícitas e implícitas), além de todas as frases de ligação que fazem parte da mesma dimensão semântica definida pelo professor, quando este escolheu a relação binária “é agregado de”, ou seja, as frases: tem, tem parte, tem pedaço, tem seção, é agregado de, é composto de, inclui, contém e possui, fazem parte da mesma dimensão semântica *assimetria.partição.física.sentido\_direto*, estando dessa forma, também, presentes no conjunto base B.

**Figura 25. Relação binária de uma ontologia no *webon\_tool***



Fonte: (Elaborada pelo autor).

Assim, o conjunto base gerado tendo como base esta ontologia se assemelha ao da Figura 26.

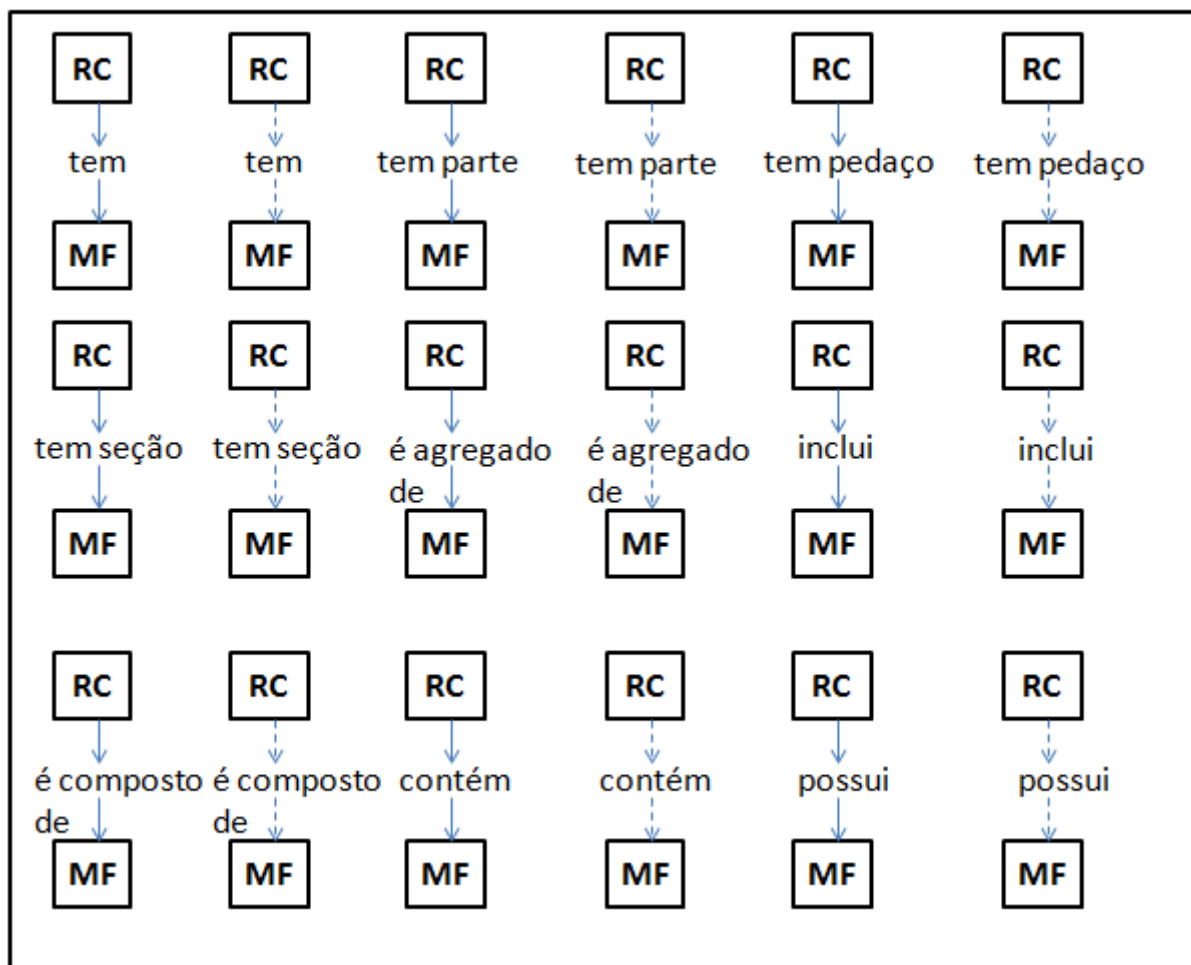
Figura 26. Conjunto base *b* originado pela união de conceitos e relações binárias

<p><b>Conceitos:</b> REDES DE COMPUTADORES e MEIO FÍSICO</p> <p><b>Relações:</b></p> <ul style="list-style-type: none"> <li>{assimetria.partição.física.sentido_direto, "tem", d}</li> <li>{assimetria.partição.física.sentido_direto, "tem", r}</li> <li>{assimetria.partição.física.sentido_direto, "tem parte", d}</li> <li>{assimetria.partição.física.sentido_direto, "tem parte", r}</li> <li>{assimetria.partição.física.sentido_direto, "tem pedaço", d}</li> <li>{assimetria.partição.física.sentido_direto, "tem pedaço", r}</li> <li>{assimetria.partição.física.sentido_direto, "tem seção", d}</li> <li>{assimetria.partição.física.sentido_direto, "tem seção", r}</li> <li>{assimetria.partição.física.sentido_direto, "é agregado de", d}</li> <li>{assimetria.partição.física.sentido_direto, "é agregado de", r}</li> <li>{assimetria.partição.física.sentido_direto, "é composto de", d}</li> <li>{assimetria.partição.física.sentido_direto, "é composto de", r}</li> <li>{assimetria.partição.física.sentido_direto, "inclui", d}</li> <li>{assimetria.partição.física.sentido_direto, "inclui", r}</li> <li>{assimetria.partição.física.sentido_direto, "contém", d}</li> <li>{assimetria.partição.física.sentido_direto, "contém", r}</li> <li>{assimetria.partição.física.sentido_direto, "possui", d}</li> <li>{assimetria.partição.física.sentido_direto, "possui", r}</li> </ul>
---

Fonte: (Elaborada pelo autor).

As letras **r** e **d** no conjunto **B** correspondem à aprendizagem por reconciliação integrativa e por diferenciação progressiva, respectivamente (ver Seção 2.3.1). A partir desse conjunto base, é formado o Alfabeto, que corresponde a todos os possíveis genes (proposições) que poderão compor os cromossomos (mapas conceituais). O AG possui um *Axioma Formadores de Genes* (AFGs) que define como esses genes serão formados a partir da combinação dos elementos da base do conjunto B (ROCHA et. al., 2004). Vale ressaltar que, em AGs tradicionais que utilizam a codificação binária, esse alfabeto é composto somente pelos genes 0 e 1, entretanto no *GAADT-CM* esse alfabeto é muito mais complexo. O alfabeto decorrente da ontologia da Figura 25 pode ser visto na Figura 27.

Figura 27. Alfabeto gerado da Figura 25



Fonte: (Elaborada pelo autor).

Ressalta-se que no alfabeto apresentado na Figura 27, convencionalmente as linhas cheias (ou contínuas) representam aprendizagem por diferenciação progressiva e as linhas tracejadas aprendizagem por reconciliação integrativa. O algoritmo foi projetado para construir MCs, simulando os processos cognitivos dessas duas AS, podendo, desta forma, auxiliar a avaliação dos MCs dos alunos construídos segundo a mesma teoria (ROCHA, 2007).

O módulo *GAADT-CM* usa como mecanismo auxiliar um motor de inferência capaz de interpretar e combinar axiomas representativos das dimensões semânticas da taxonomia, para responder se determinada proposição presente no mapa conceitual do aprendiz é válida ou não. Por exemplo, se a ontologia contém o conhecimento explícito <<CARRO, tem, MOTOR> e <MOTOR, usa, OLEO>>, o gene (proposição) <<CARRO, usa, OLEO>> é criado e validado pelo motor de inferência, porque axiomas referentes às dimensões utilizadas (partição e classificação) permitem inferir por silogismo hipotético este conhecimento.



O motor de inferência incluído no ambiente possui quatro operadores lógicos sobre proposições e quatro regras de inferência (ver Quadro 3.2) (FILHO, 1990). Neste módulo, as regras de inferência do motor permitem gerar genes inferidos que são adicionados ao alfabeto inicial que contém uma estrutura de dados somente com genes não inferidos, ou seja, que foram gerados explicitamente pelo AFG. Após esta etapa e com a inserção dos novos genes validados pelo motor de inferência, o alfabeto inicial passa a conter tanto genes inferidos quanto genes não inferidos, aumentando, a probabilidade de gerar um mapa correto mais próximo ao mapa do estudante, valorizando ainda mais o seu modo particular de aprender.

**Quadro 3. Operadores lógicos e regras de inferência**

<b>Operadores Lógicos sobre Proposições</b>	<b>Regras de inferência</b>
Negação	Modus tollens
Conjunção	Silogismo hipotético
Disjunção	Silogismo disjuntivo
Condicional	Inversão

Fonte: (Elaborado pelo autor).

Com o alfabeto completo (genes implícitos e genes explícitos) são gerados os cromossomos utilizando dois Axiomas Formadores de Cromossomos (AFCs) que são responsáveis pelo descarte de cromossomos desconexos e cíclicos. Os desconexos são excluídos por irem de encontro ao princípio da diferenciação progressiva, reconciliação integrativa e os cíclicos por infringirem as regras de hierarquia dos conceitos em um MC (ROCHA et. al., 2004). Os cromossomos gerados, obedecendo aos AFCs, constituem os cromossomos da população inicial, possuindo o mesmo tamanho do MC submetido pelo estudante haja vista que representam os cromossomos comparados ao MCEst.

Com a população inicial estabelecida, Rocha e colaboradores (ROCHA et. al., 2005) definiram uma função de adaptação (ver Figura 28) para permitir que a evolução ocorresse com base no cromossomo mais bem adaptado ao meio. Esta função encontra a adaptação de um cromossomo  $c$ , calculando a soma ponderada do grau de adaptação de cada um de seus genes específicos.

**Figura 28. Função para o cálculo da adaptação de um cromossomo**

$$adapt(c) = \left( \sum_{g \in c} \theta \times grau(g) \right) - \#(MCEst - c)$$

Fonte: (adaptado de ROCHA, 2007).

O peso  $\theta$  refere à importância de um determinado gene na construção de um MC próximo ao MCEst (ROCHA et. al., 2005). Os valores de  $\theta$  definem na Tabela 4.

**Quadro 4. Valores que o peso  $\theta$  pode assumir**

$\Theta$	Explicação
3	Se os dois conceitos do gene estiverem presentes no MCEst e conectados e entre si.
2	Se os dois conceitos do gene estiverem presentes no MCEst, mas não conectados entre si.
1	Se somente um conceito do gene estiver presente no MCEst
0	Se nenhum dos conceitos do gene existirem no MCEst

Fonte: (elaborado pelo autor).

Segundo Rocha (2007), para cada conceito que está presente no MCEst e não está presente no cromossomo (ou seja, no MC gerado pelo AG) é decrementado uma unidade do valor de adaptação calculado.

Para calcular o grau de adaptação de um cromossomo é necessário, primeiramente, calcular o grau de adaptação dos genes  $g$  presentes no cromossomo  $c$ , conforme mostra a Figura 29. Onde os valores  $\alpha$  e  $\beta$  podem assumir os valores definidos na Tabela 3.4.

**Figura 29. Função para o cálculo do grau de adaptação de um gene**

$$grau(g_k) = \alpha_k + 2\beta_k$$

Fonte: (adaptado de ROCHA, 2007).

Com base nos valores definidos para  $\alpha$  e  $\beta$ , verificam-se os princípios construtivistas em que o ambiente se baseia, pois os cromossomos cujos genes coincidam com a definição do aluno são favorecidos, obtendo o maior valor de adaptação, valorizando, desta forma, a maneira individual de o aluno construir conhecimento.

No capítulo 4, é descrita a aplicação da avaliação deste módulo, levando em conta as entradas descritas para o cálculo de adaptação dos cromossomos de acordo com a adaptação de cada um dos genes e sua formação final durante a análise da avaliação.

No *GAADT-CM*, foram utilizadas algumas técnicas adicionais na construção dos algoritmos genéticos: a técnica do estado estacionário, que define que os piores cromossomos da nova geração/população sejam substituídos pelos melhores cromossomos da geração anterior; e a utilização do repositório de genes dominantes (ROCHA, 2007).

**Quadro 5. Valores admissíveis dos parâmetros do grau do gene**

Parâmetro	Valor	Quando é usado
$\alpha$	1	A frase de ligação do gene não coincide com a escolha feita pelo professor ao desenhar a ontologia para nenhuma ligação envolvendo qualquer dos dois conceitos do gene.
	2	A frase de ligação do gene coincide com uma das frases de ligação consideradas como preferenciais pelo professor ao desenhar a ontologia, para alguma conexão envolvendo qualquer um dos dois conceitos do gene;
$\beta$	1	A frase de ligação do gene não coincide com a escolha feita pelo aluno ao desenhar MCEst em nenhuma ligação envolvendo qualquer dos dois conceitos do gene;
	2	A frase de ligação do gene coincide completamente com a frase de ligação usada pelo aluno em alguma ligação envolvendo qualquer um dos conceitos do gene existente em MC est

Fonte: (ROCHA, 2007).

### 3.2.4 Módulo Avaliador

O módulo avaliador representa o relatório que é gerado como produto do processo de avaliação da aprendizagem do aluno. Sua estrutura básica é composta pelo conjunto de mapas conceituais gerados pelo *GAADT-CM*, o MCEst submetido na avaliação, os níveis hierárquicos que formam o MCEst desde o conceito definido como raiz até seus filhos, a proximidade e

distância semântica dos conceitos presentes durante a comparação de cada proposição do MCEst aos dos MC gerados pelo *GAADT-CM* e uma nota sintetizada pelo percentual relativa a distância semântica do MCEst submetido com os MCs gerados pelo algoritmo.

Este relatório do módulo avaliador é uma síntese: i) da comunicação direta e visual que o aluno possui com os detalhes relevantes presentes em sua estrutura cognitiva por meio da aprendizagem significativa; ii) do momento em que o aluno autocorrigue seu MC, analisando os mapas corretos gerados pelo *GAADT-CM*; iii) do aprendizado por diferenciação progressiva e por reconciliação integrativa, verificando de forma coerente quais as frases de ligações usadas em cada uma de suas proposições, quando comparáveis aos MCs gerados no *GAADT-CM*. Detalhes sobre este relatório de avaliação são apresentados no Apêndice B.

### 3.2.5 Módulo Repositório

O módulo repositório representa no ambiente um diretório e um banco de dados. O diretório contém os mapas conceituais e ontologias salvos durante o acesso ao ambiente, além da taxonomia de frases de ligação, a qual implementa um modelo capaz de: i) distinguir duas proposições diferentes, embora próximas; e, ii) atribuir valores de distância semântica proporcionais à diferença em significado entre duas proposições quaisquer comparadas. A Figura 30 apresenta a organização do módulo repositório descrito no *WebCMTTool*.

Figura 30. Organização do módulo repositório



Fonte: (Elaborada pelo autor).

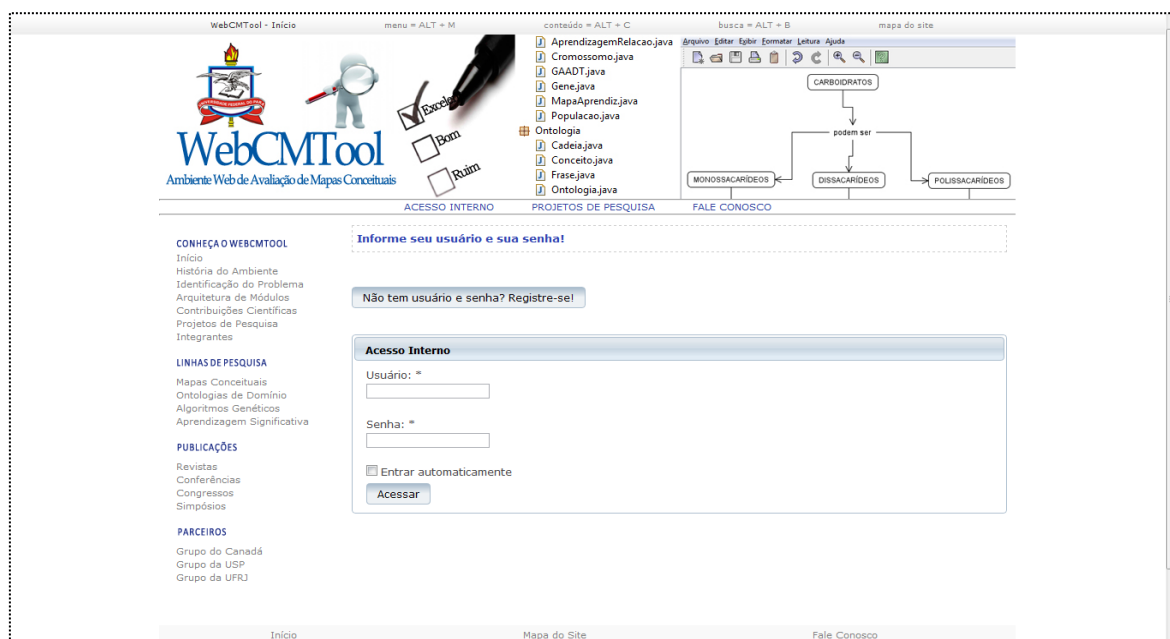
A outra representação descrita por este repositório, trata-se do banco de dados, em que o projetado refere-se ao relacional, sendo o *MySQL* o Sistema Gerenciador de Banco de Dados (SGBD), escolhido para gerenciamento dos dados dos usuários registrados (aluno, professor e administrador) que acessam o sistema *WebCMTool*.

### 3.2.6 Módulo Administrador

O módulo administrador é o responsável pelo controle de permissões para acesso ao ambiente. Estas permissões (papéis) distinguem, por exemplo, um aluno de um professor. Como resultado, o aluno tem acesso única e exclusivamente ao *WebCMEditor*, enquanto que o professor ao *WebOn\_Tool* e ao *WebCMEditor*.

A estrutura deste módulo foi baseada no *Spring Security* (SS) que é um dos projetos do *Spring Framework* capaz de garantir a segurança de acesso a sistemas. Assim, o SS garante a segurança do acesso a determinadas pastas, restringindo e permitindo somente acesso autorizado por usuários registrados no ambiente. Com efeito, sempre que alguém tentar acessar alguma pasta do ambiente que tenha a segurança garantida pelo SS será apresentado automaticamente à página de acesso interno para que o visitante se identifique, conforme mostra a Figura 31.

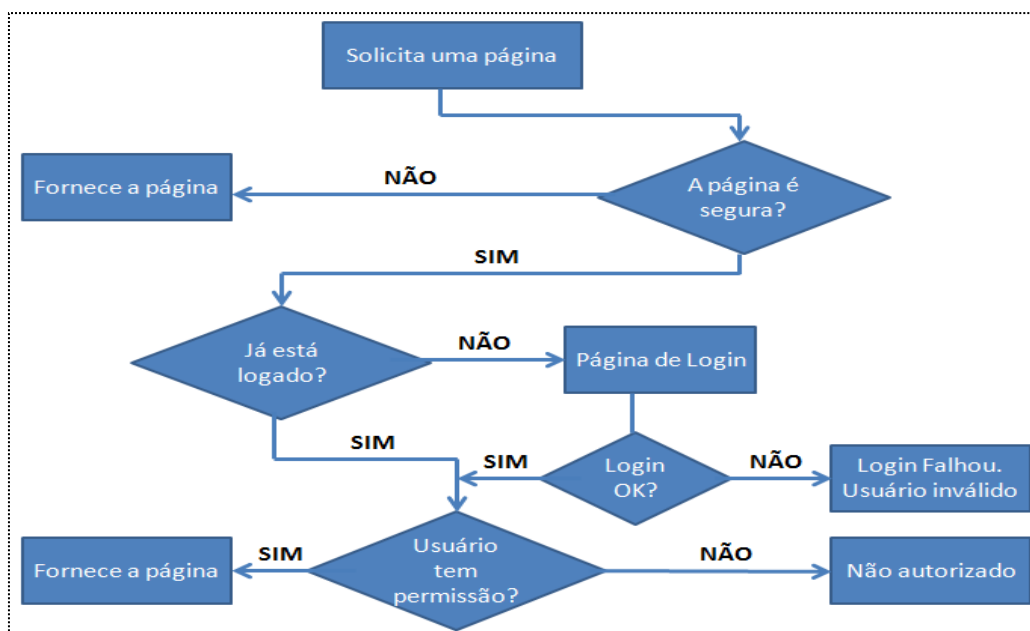
**Figura 31. Página de acesso interno**



Fonte: (Elaborada pelo autor)

Caso o visitante informe um login e senha inválida, ocorre uma validação no banco de dados, em que verifica por meio de uma busca qual usuário registrado possui aquele login e senha digitada na página, quando não encontrado nenhum registro o login e a senha são considerados inválidos, mostrando mensagens de erros referentes ao login e/ou senha digitada, além de ocorrer um redirecionamento à página de acesso interno. Em caso contrário, ou seja, após o login e a senha serem validados pelo ambiente é aberto uma sessão, permitindo o redirecionamento a(s) ferramenta(s) de edição. Esse esquema de login é parte da segurança fornecida pelo *Spring*. Um fluxograma demonstrando o funcionamento geral do SS pode ser visto na Figura 32.

**Figura 32. Fluxograma do funcionamento geral do *spring security***



Fonte: (Elaborada pelo autor).

Todas as configurações do SS estão baseadas em arquivos *XML*, que permitem restrições de acesso ao ambiente garantindo sua segurança, além de dividir os papéis para cada usuário.

### 3.3 ESPECIFICAÇÕES TÉCNICAS

Nesta seção, são apresentados os diagramas de *Unified Modeling Language (UML)* que representam a avaliação que contém detalhes das funcionalidades descritas pelo *WebCMTTool* além da estrutura em que foram implementadas no mesmo.

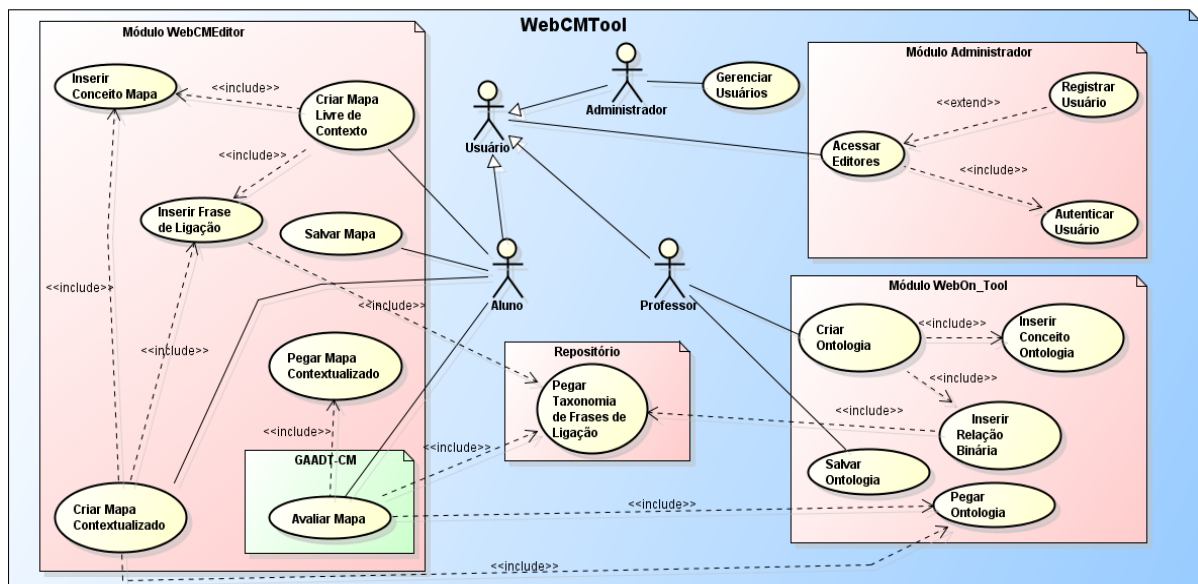
### 3.3.1 Diagrama de casos de usos

As funcionalidades desejáveis durante a realização da avaliação no *WebCMTool* e os papéis desempenhados por seus utilizadores são mostrados no diagrama de casos de uso geral da avaliação apresentado Figura 33.

Existem quatro atores no diagrama de casos de uso: usuário, administrador, professor e aluno. Os três últimos herdam as funcionalidades do usuário em relação ao acesso interno ao ambiente.

Em suma, o professor é responsável por criar uma ontologia de domínio, utilizando conceitos e relações binárias entre estes conceitos para compartilhamento de conhecimento. Esta ontologia<sup>5</sup>, por sua vez, é utilizada na criação de um mapa contextualizado por parte do aluno que utiliza os conceitos presentes na ontologia mais as frases de ligações apropriadas presentes na taxonomia de frase de ligação do módulo repositório, para relacionar estes conceitos, formando proposições que servirão para formar os mapas conceituais com base no domínio da ontologia do professor.

Figura 33. Diagrama de casos de uso geral da avaliação



Fonte: (Elaborada pelo autor).

O módulo *GAADT-CM* está integrado ao módulo *WebCMEEditor* de modo que o aluno pode avaliar seu mapa contextualizado com base nos três insumos básicos para a execução da avaliação: mapa, ontologia e taxonomia de frases de ligação. Por outro lado, o aluno poderá

<sup>5</sup> Neste capítulo, o termo ontologia e ontologia de domínio são considerados sinônimos.

também construir mapas livres de contexto (ou seja, não associados a uma ontologia de domínio), por exemplo, para propósito de construção de currículos, explicação de matéria em disciplinas escolares, especificação de conteúdos profissionais, entre outros, como melhor compreensão de um artigo científico.

O administrador é o ator responsável pela gerência dos usuários registrados no ambiente, bem como pela atribuição de permissão para os usuários. Este ator pode também cancelar o acesso interno a um determinado aluno ou professor registrado, deixando-o com acesso inativo ao ambiente.

### 3.3.2 Visão geral da avaliação no GAADT-CM

Um diagrama de classe é uma representação da estrutura e relações das classes que servem de modelo específico para os objetos que serão criados no ambiente *WebCMTool*. Assim, durante a etapa de avaliação da aprendizagem são utilizados, por exemplo, métodos fundamentais para a criação em tempo de execução e gerenciamento de objetos no *WebCMTool* como os métodos: *geraGenesAFC()*, responsável por gerar o alfabeto do *GAADT-CM*; *geraCromossomoAFC()*, que gera a população inicial, ou seja, a população ancestral; *verificaCiclo()* e *verificaDesconexao()*, que indicam quando há mapas conceituais construídos incorretamente; *Fecundação()*, *Cruzamento()*, *Mutacao()*, entre outros, que permitem a criação de novas populações a partir de uma população inicial. A Figura 34 mostra de forma completa os outros métodos utilizados pela classe *Ambiente*. O Apêndice A apresenta as classes utilizadas, suas relações, bem como a especificação dos pacotes utilizados na implementação do projeto.

**Figura 34. Classe ambiente com os principais métodos do *gaadt-cm***

<b>Ambiente</b>	
-	<code>prob_r : double = 0.40</code>
+	<code>Ambiente(ontologia : Ontologia, taxonomia : Taxonomia, mp_aprendiz : MapaAprendiz, tam_pop_inicial : int)</code>
+	<code>geraGenesAFC(taxonomia : Taxonomia) : void</code>
+	<code>geraCromossomoAFC(genes : List&lt;Gene&gt;) : Cromossomo</code>
-	<code>verificaCiclo(vetor_genes : List&lt;Gene&gt;) : boolean</code>
-	<code>verificaDesconexao(vetor_genes : List&lt;Gene&gt;) : boolean</code>
+	<code>getPopulacao_descendente() : Populacao</code>
+	<code>getPopulacao_intermediaria() : Populacao</code>
+	<code>getPopulacao_ancestral() : Populacao</code>
+	<code>RelatorioPopulacao(populacao : Populacao) : String</code>
+	<code>RelatorioBaseGenes() : void</code>
+	<code>aplicaSelecao() : boolean</code>
+	<code>Fecundacao() : void</code>
+	<code>identificaDominantes(ancestral_1 : Cromossomo, ancestral_2 : Cromossomo) : void</code>
+	<code>Cruzamento(tamanho_pop : int, populacao : Populacao, genes : ArrayList&lt;Gene&gt;) : void</code>
+	<code>Mutacao(prob_mutacao : double) : void</code>
+	<code>RenovaGeracao() : void</code>
+	<code>mantemMelhor(ontologia : Ontologia, mp_aprendiz : MapaAprendiz) : void</code>
+	<code>aplicaEstadoEstacionario(ontologia : Ontologia, mp_aprendiz : MapaAprendiz, gap : int) : void</code>
-	<code>getGeneAleatorioPri(genes : List&lt;Gene&gt;) : Gene</code>

Fonte: (Elaborada pelo autor).



De forma geral e levando em conta a etapa da avaliação, após serem detectados e validados os insumos, é iniciada a avaliação do mapa conceitual do aluno. Neste momento, é criada uma instância da classe *Ambiente* para a formação dos genes e cromossomos, utilizando os métodos descritos nesta classe, além de realizar as ações de seleção, cruzamento, mutação, identificar, por exemplo, genes dominantes e a fecundação.

Este capítulo representa a obra chave deste projeto, cujos detalhes foram levados em relevância ao descrever o ambiente *WebCMTool* como um facilitador da avaliação automática da aprendizagem significativa nas seções deste capítulo.

O aspecto crucial foi destacar principalmente a estrutura e arquitetura de módulos presentes no ambiente. Assim, a Seção 3.1 elucidou a visão geral do ambiente, sobretudo na importância da avaliação tendo como entradas insumos específicos com o intuito de valorizar o aspecto idiossincrático da aprendizagem do aluno perante seu aprendizado significativo descrito pelo seu mapa conceitual. Na Seção 3.2, foi descrita a arquitetura de módulos do *WebCMTool* onde foi esclarecido sobre a plataforma utilizada na implementação e framework utilizados, o processo de desenvolvimento projetado no ambiente. Na fase de análise e projeto e a arquitetura com seis módulos bem específicos é apresentada da Seção 3.2.1 à Seção 3.2.6.

Complementarmente e com o objetivo de esclarecer detalhes técnicos utilizados, foi apresentado o diagrama de casos de uso do *WebCMTool*, que molda as funcionalidades do ambiente sendo ressaltados pela Subseção 3.3.1. Além da descrição sobre a classe do *GAADT-CM* responsável pela criação do ambiente de avaliação. O processo de avaliação foi ressaltado, principalmente, no sentido de valorização da construção do conhecimento por parte do aluno, já que o *GAADT-CM* abordado, valoriza a idiossincrasia e a construção do conhecimento pelo aluno.

## 4 ANÁLISE DOS RESULTADOS

Este capítulo relata a análise dos resultados no ambiente *WebCMTool* desde a fase da aplicação numa Universidade até as avaliações heurísticas realizadas. Para o mesmo, foram feitas separadamente duas análises: uma da aplicação do ambiente e uma de usabilidade.

Os resultados coletados nas análises da aplicação e usabilidade por meio de avaliações heurísticas tiveram a finalidade de destacar, sobretudo, o valor que o ambiente agregou para a Educação. Um estudo de viabilidade também foi realizado para verificar se o projeto era viável inicialmente a sua aplicação e pode ser analisada no Apêndice E, neste contexto foram verificados aspectos quanto ao escopo, tempo, requisitos, estrutura, riscos e recursos utilizados.

A aplicação do ambiente permitiu que uma coleção de dados fosse coletada na qual, se obteve: i) informações a respeito do perfil dos alunos que participaram do curso durante a aplicação e como esses dados interferiram na aplicação; ii) as dificuldades encontrada pelos alunos; iii) os resultados das avaliações no ambiente pelos alunos e, por fim; iv) um questionário de satisfação dos alunos que utilizaram.

As avaliações heurísticas foram desenvolvidas por três profissionais, que garantiram a validade, conforme retrata Nielsen (1994), cuja meta foi encontrar problemas de usabilidade que interferem na relação com o usuário no ambiente. Foi submetido para a análise o ambiente e alguns módulos descritos pelo Capítulo 3.

Durante todo o projeto e desenvolvimento do ambiente, houve a preocupação com questões relacionadas à interface e acessibilidade. Entretanto, para justificar a continuação dos estudos, é necessário verificar se o projeto desenvolvido cumpre seu papel nos objetivos propostos viáveis à sua aplicação (MAFRA et. al., 2006). Desta forma, a análise de resultados do ambiente foi organizada em duas etapas:

1. primeira etapa: análise de aplicação;
2. segunda etapa: análise de usabilidade.

### 4.1 PRIMEIRA ETAPA: ANÁLISE DA APLICAÇÃO DO AMBIENTE

O Curso Linguagem Java – Desenvolvimento de Aplicações Multi-Plataformas, oferecido pela Empresa Júnior de Informática (EJI) da UFPA, foi selecionado para aplicação e testes de validação do ambiente *WebCMTool*. A escolha por este curso se deu por ser a área de atuação do professor que desenvolveu as ontologias de domínio para aplicação, e pela

necessidade de se aplicar a alunos que pudessem se comprometer com os testes, garantindo a Aprendizagem Significativa.

No curso, cinquenta e cinco alunos fizeram uma pré-inscrição, entretanto, somente vinte alunos efetivaram suas matrículas, dos quais quatorze alunos se matricularam antes da primeira aula e seis alunos após a primeira aula. Notou-se ainda que esses seis alunos sentiram grandes dificuldades no aprendizado, haja vista que não presenciaram a primeira aula que ocorreu no primeiro sábado. Para a matrícula, era necessário fazer uma inscrição antecipada na qual era pré-requisito possuir conhecimentos básicos de algoritmos para poder estudar a linguagem Java. A modalidade do curso foi presencial, pois tinha como meta acompanhar os alunos durante sua evolução da aprendizagem. A carga horária total do curso foi de trinta e duas horas que ocorreram aos sábados, contudo, para a aplicação foram utilizadas apenas dezesseis horas divididas em três módulos distintos.

O principal objetivo do *WebCMTool* no curso de Java foi auxiliar os alunos na construção de significados dentro do domínio de programação e de sedimentar o conhecimento que se estava aprendendo. Este auxílio foi garantido por meio de autoavaliações de MCs, desenhados com base na estrutura cognitiva de cada aluno, e por uma análise e compreensão dos resultados pelos próprios alunos após a avaliação dos seus MCs.

A duração do curso para aplicação consistiu de quatro sábados consecutivos, subdivididos em três módulos. O módulo 1 teve carga horária de oito horas, em que foram apresentados as primeiras noções de Java e alguns conteúdos iniciais para desenvolvimento, além de tarefas e exercícios práticos realizados em sala. O módulo 2 teve carga horária de quatro horas, que foram divididas da seguinte forma:

- 1 hora: foram apresentados novos conceitos da linguagem para agregar aos já estudados no módulo 1, além da realização de exercícios práticos, usando o computador;
- 1 hora: foram apresentados conceitos sobre a teoria da assimilação, aprendizagem significativa, mapas conceituais e ontologias de domínio para fomentar estudos posteriores;
- 2 horas: foram feitos exercícios envolvendo a construção de mapas conceituais livres de contexto e contextualizados, ambos sem o uso de computador, envolvendo apenas papel e caneta. O primeiro exercício se baseou na construção de minicurrículos dos alunos com base em mapas livres de

contextos. O segundo exercício se caracterizou pela construção de MCs contextualizados tomando como princípio conjuntos de conceitos fornecidos pelo professor com base em duas ontologias distintas e por uma lista de frases baseadas numa taxonomia que foram utilizadas na formação das proposições dos MCs. Este segundo exercício foi subdividido em duas tarefas: 1) representava a construção de MCs por grupos de cinco alunos, levando em conta um conjunto de conceitos sobre as características da linguagem Java, e; 2) representava a construção de MCs construídos individualmente, considerando conceitos relacionados à sintaxe e estrutura da linguagem Java vistos em aula.

O módulo 3, com carga horária de quatro horas, se dividiu da seguinte maneira:

- 1 hora: foram apresentados novos conceitos avançando o conteúdo de programação, acrescentando novas sintaxes ao que se tinha estudado nos módulos 1 e 2;
- 2 horas e 30 minutos: foi apresentado o ambiente *WebCMTool* e ensinado a construção de mapas livres de contextos e mapas contextualizados com o auxílio do computador. Foi relatado o uso do ambiente no contexto da educação, a geração do relatório de autoavaliação e características apresentadas após a avaliação idiossincrática do AG. Posteriormente, um exercício mais complexo foi realizado com base numa ontologia, envolvendo conceitos utilizados pelas ontologias dos exercícios do módulo 2 e outros conceitos adicionais utilizados com base na aula deste sábado. No ambiente, a ontologia disponibilizada tratou-se de um arquivo *XML* que, quando aberta pelo módulo *WebCMEditor*, automaticamente, carregava todos os conceitos da ontologia que o professor criou e que devem ser utilizados pelos alunos na formação das proposições dos seus MCs;
- 30 minutos: foi aplicado um questionários de satisfação para avaliação do ambiente pelos alunos que o utilizaram.

Cabe salientar, ainda, que a construção de MCs de forma manual e automática, construídos nos exercícios durante o curso, permitiu avaliar a facilidade de uso do ambiente por meio de testes realizado pelos alunos, a acurácia de avaliação durante a construção de

conhecimento e a diminuição da sobrecarga do professor. O Quadro 6 destaca de maneira detalhada os módulos, as horas e o conteúdo programático do curso na UFPA.

**Quadro 6. Conteúdo programático do curso linguagem Java**

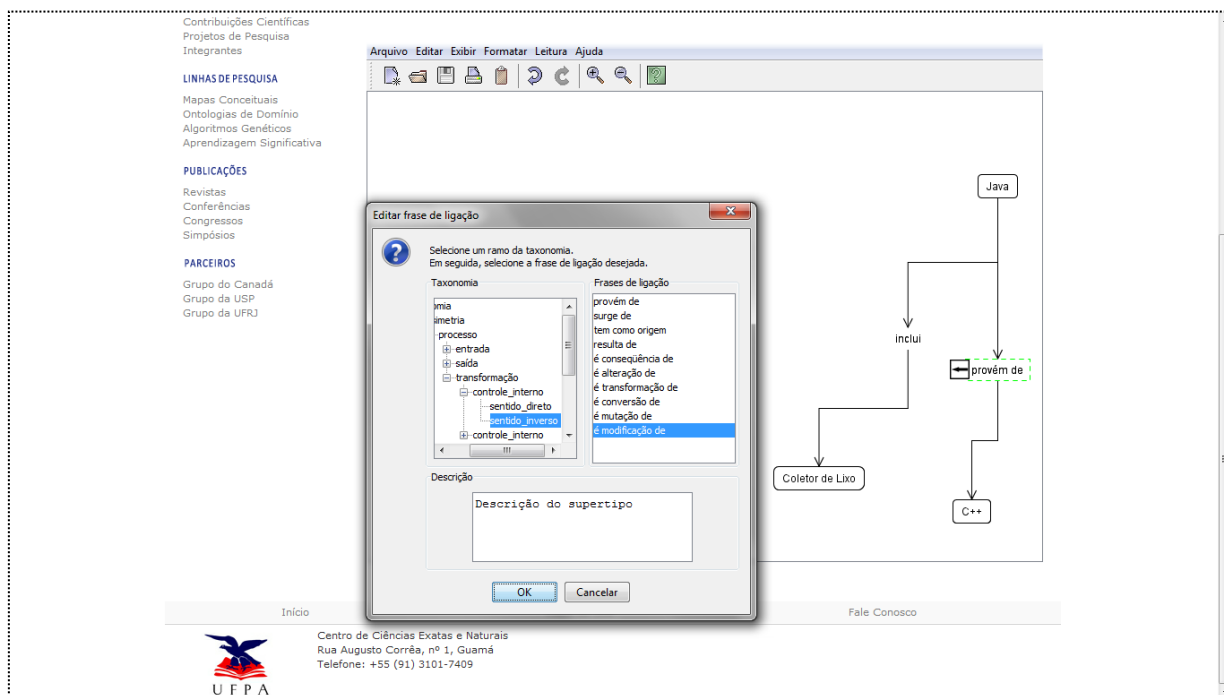
Módulos	Aulas	Sobre Java		
Módulo 1	1º Sábado (4h)	<ul style="list-style-type: none"> <li>▪ Introdução à programação;</li> <li>▪ Conceitos, características, vantagens e desvantagens de Java;</li> <li>▪ Instalação e configuração de ambiente</li> <li>▪ Análise estrutural do programa olá mundo</li> <li>▪ JVM e gerência de memória em Java</li> <li>▪ Declarações de variáveis e métodos</li> <li>▪ Tipos de retorno de métodos</li> <li>▪ API</li> <li>▪ Exercícios práticos</li> </ul>		
	2º Sábado (4h)	<ul style="list-style-type: none"> <li>▪ Identificadores</li> <li>▪ JavaBeans</li> <li>▪ Tipos de comentários</li> <li>▪ Declaração e definição de classes</li> <li>▪ Declaração e definição de interfaces</li> <li>▪ Declaração e definição de construtores</li> <li>▪ Modificadores não referentes a acesso e referentes a acesso</li> <li>▪ Atribuições e operadores: aritméticos, relacionais e lógicos</li> <li>▪ Exercícios práticos e orais</li> </ul>		
Módulo 2	3º Sábado (4h)	Sobre Java	Sobre o Ambiente	Exercício
		<ul style="list-style-type: none"> <li>▪ Var-args</li> <li>▪ Tipos de dados</li> <li>▪ Controles de seleção</li> <li>▪ Controles de repetição</li> <li>▪ Exercícios práticos</li> </ul>	<ul style="list-style-type: none"> <li>▪ Teoria da assimilação</li> <li>▪ Aprendizagem significativa</li> <li>▪ Mapas conceituais</li> <li>▪ Ontologias de domínio</li> </ul>	<ul style="list-style-type: none"> <li>▪ Exercício 1: minicurriculo</li> <li>▪ Exercício 2: a) tarefa 1: ontologia de características b) tarefa 2: ontologia de sintaxe</li> </ul>
Módulo 3	4º Sábado (4h)	Sobre Java	WebCMTTool e Exercícios	Questionário
		<ul style="list-style-type: none"> <li>▪ Classe Math</li> <li>▪ História da orientação a objetos</li> <li>▪ Conceitos e características de classes e objetos</li> <li>▪ Exercícios</li> </ul>	<ul style="list-style-type: none"> <li>▪ Apresentação do WebCMTTool</li> <li>▪ Características da avaliação</li> <li>▪ Testes de construção de MCs</li> <li>▪ Exercícios contextualizados</li> </ul>	<ul style="list-style-type: none"> <li>▪ Técnicas prospectivas para verificar a satisfação dos alunos que utilizaram o ambiente</li> </ul>

Fonte: (Elaborado pelo autor).

Ressalta-se que, durante os exercícios realizados no módulo 2, dúvidas surgiram e foram sanadas, principalmente pelo fato de muitos alunos não conhecerem MCs. No módulo 3, entretanto, já com um pouco mais de familiaridade com MCs, os alunos, mesmo com certa dificuldades na escolha de frases corretas, obtiveram um maior êxito nos exercícios, quando

comparado ao módulo 2. Em suma, a Figura 34 mostra o ambiente durante a construção de um MC contextualizado por um dos alunos durante o módulo 3, quando este tentava relacionar a frase correta para criar uma proposição no mapa.

**Figura 35. Construção do mapa conceitual contextualizado por um aluno no curso no *webcmtool***



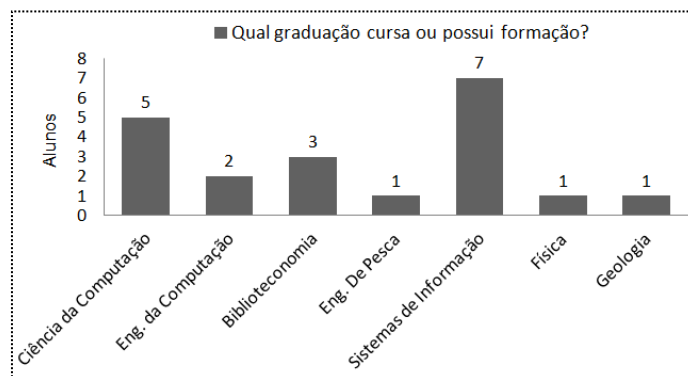
Fonte: (Elaborada pelo autor).

A seguir é apresentado o perfil dos alunos do curso e quais os impactos para a aprendizagem durante os testes realizados para a aplicação.

#### 4.1.1 Perfil dos Alunos

Todos os alunos que fizeram o curso de Java se registraram no ambiente. Assim, notou-se que os alunos ou eram graduados ou graduandos. O Gráfico 1 mostra a situação.

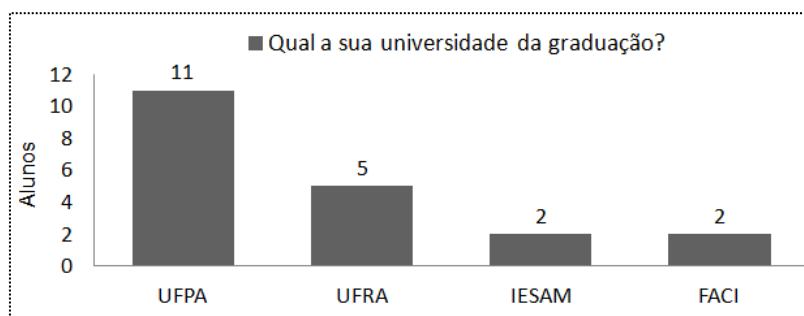
**Gráfico 1. Quantidade de alunos que utilizaram o ambiente por graduação**



Fonte: (Elaborado pelo autor).

O curso se constituiu, na maioria, por alunos que ainda cursavam uma graduação numa Instituição de Ensino Superior, destacando-se a UFPA com maior quantidade de alunos. Isto se deve, entre outras questões, pelo curso ser oferecido na própria Universidade seja pela divulgação de panfletos ou por indicação entre os alunos. O Gráfico 2 apresenta a Universidade dos alunos, tanto dos formados quanto dos que ainda cursavam-na.

**Gráfico 2. Quantidade de alunos distribuídos por universidades**



Fonte: (Elaborado pelo autor).

A divulgação do curso se deu por meio de redes sociais, panfletos, informações entre os alunos e pelo site, deixando livre a inscrição para diferentes perfis e alunos de diferentes instituições, seja graduado, graduando ou ainda no Ensino Médio.

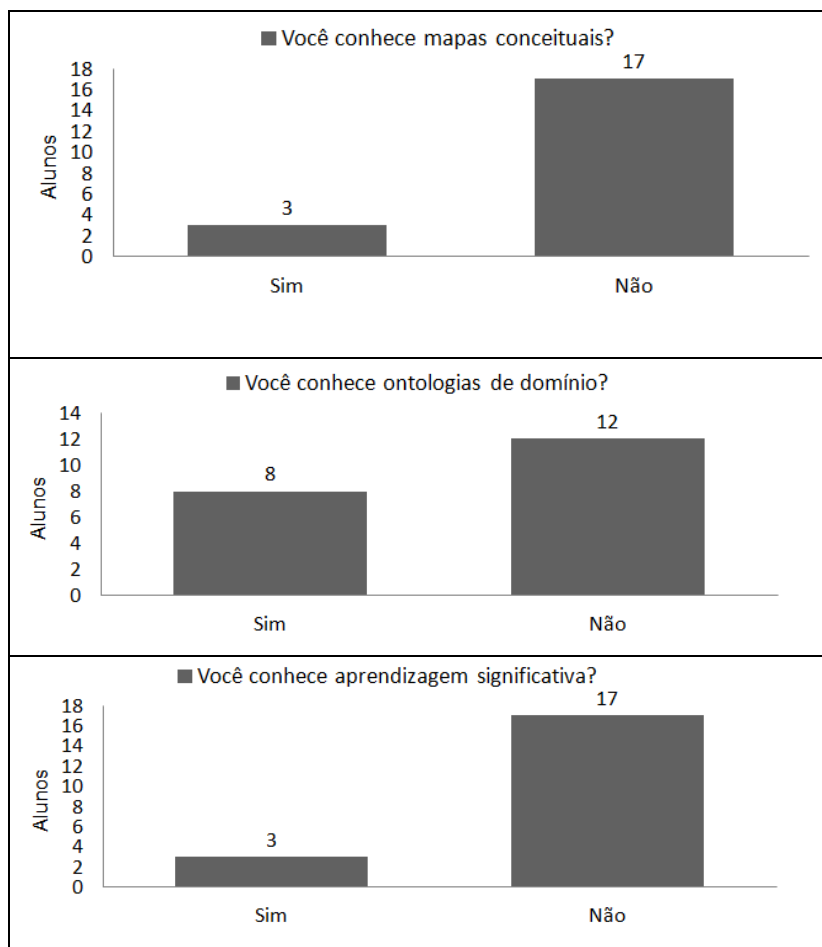
Os diferentes perfis foram interessantes na construção de MCs durante módulo 2 entre os grupos de alunos que discutiam no âmbito do domínio de programação qual frase utilizar para formar as proposições corretas nos MCs; levando em conta características semânticas predefinidas pela taxonomia de frases de ligação disponibilizada pelo professor, em conjunto com os conceitos da ontologia. Mais, principalmente interessantes, pelo conhecimento cognitivo que um aluno de uma graduação possuía em relação a outro aluno de outra graduação. Por exemplo, os três alunos que cursam Biblioteconomia obtiveram melhores resultados em seus grupos (como também nas tarefas realizadas individualmente) durante a formação das proposições dos MCs em comparação com os alunos das graduações de Sistemas de Informação, Engenharia da Computação e Ciências da Computação, ainda que o tema destacado fosse sobre programação.

Os melhores resultados desses três alunos, talvez possam estar alicerçados – de maneira não conclusiva - a ideia de que esses alunos possuem em suas estruturas cognitivas uma melhor compreensão do contexto semântico das frases utilizadas pela taxonomia, seja por

conhecimento mecânico ou significativo, aprendendo desta forma, satisfatoriamente conceitos novos.

Os alunos que utilizaram o ambiente foram também questionados se já conheciam MCs, ontologias e AS. O Quadro 7 resalta gráficos com a quantidade de alunos e suas respostas aos questionamentos.

**Quadro 7. Quantidade de alunos que já conheciam os conceitos estudados em sala**



Fonte: (Elaborada pelo autor).

Estudando os dados coletados, foi possível notar que três alunos conheciam MCs porque estudaram também AS anteriormente. Entre eles estão um aluno de Biblioteconomia, um aluno de Sistemas de Informação e um aluno de Física. A desenvoltura na construção dos MCs por esses alunos durante os primeiros exercícios realizados foram melhores que os demais que ainda não haviam trabalhado com esta ferramenta. Contudo, dois desses alunos (o de Física e o de Sistemas de Informação) tiveram as mesmas dificuldades que os demais ao escolher uma frase apropriada para a formação das proposições nos MCs.



A análise dos resultados obtidos sobre os perfis no curso aponta que todos os alunos declararam que frisaram o conhecimento sobre programação após os exercícios envolvendo mapas. Contudo, alguns alunos relataram que não aprenderam o domínio dos exercícios representados pelas ontologias, isto porque não compreenderam os conceitos de MCs e aprendizagem significativa de forma clara. Esta ação refletiu no desenvolvimento dos seus mapas qualificados como insatisfatórios, quando avaliados pelo *WebCMTTool*.

#### **4.1.2 Exercícios em grupo e individual sem o webcmttool**

O módulo 2 foi desenvolvido em quatro horas de aplicação. Neste tempo, duas horas foram destinadas à prática do exercício em grupo e ao exercício individual entre os alunos, sendo realizado sem a ajuda do ambiente *WebCMTTool*, apenas utilizando papel e caneta.

Destaca-se que os alunos primeiramente foram submetidos a um simples treinamento envolvendo a construção de minicurriculos, criando mapas livres de contextos. Este treinamento foi caracterizado pela familiarização dos alunos à prática de construção de MCs e pelas dúvidas que em geral foram sanadas.

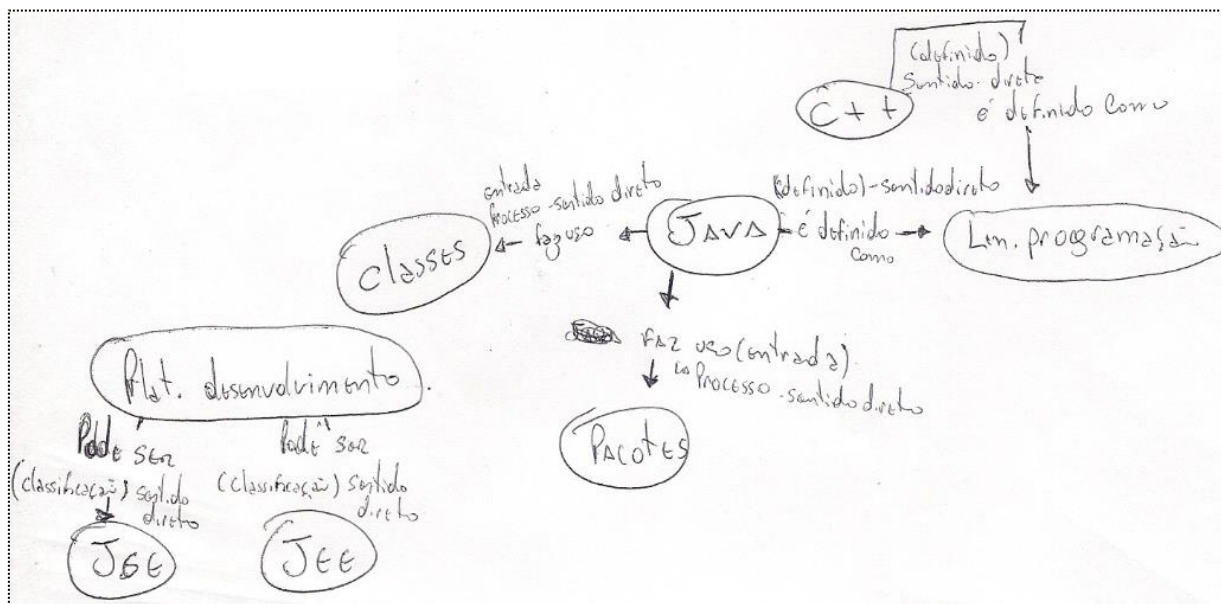
Os minicurriculos desenvolvidos em forma de MCs foram organizados em conceitos e frases utilizadas sem qualquer auxílio prévio, apenas com base na estrutura cognitiva pessoal. A ideia inicial para este mapa, não era submeter a um processo de avaliação, mas apenas auxiliar o aluno na construção de seu primeiro MC.

O próximo passo no processo de aprendizagem foi o exercício em grupo, em que os alunos foram divididos em quatro grupos de cinco alunos que se submeteram a construção de MCs, levando em conta a disponibilização de oito conceitos e 246 frases classificadas semanticamente, conforme definido por Junior et. al. (op. cit.). Os oito conceitos foram criados pelo professor com base numa ontologia sobre as características do Java. Neste cenário, os grupos tinham como meta definir quais conceitos formavam proposições e, principalmente, qual frase deveria ser usada para relacionar esses conceitos envolvidos.

Este exercício foi desenvolvido em uma hora e cinco minutos pelos grupos, que posteriormente ressaltaram dificuldades na construção. Como os alunos não puderam ver a estrutura da ontologia construída pelo professor, a escolha pelas frases corretas tornou-se uma tarefa difícil para os alunos que discutiram entre si, a fim de escolher qual seria a frase apropriada para uso. Após chegarem a um acordo, os MCs foram construídos e entregues ao professor que se sobrecarregou da avaliação desses mapas já que não possuía a ajuda de

nenhuma ferramenta automática. A Figura 35 apresenta a resposta manual de um dos grupos de alunos com base nos oito conceitos.

**Figura 36. Exemplo do mapa conceitual construído por um dos grupos**



Fonte: (Elaborada pelo autor).

Claramente nota-se, nessa figura, a dificuldade do grupo em criar um MC, cuja finalidade do exercício era compactar em um único MC as várias estruturas cognitivas dos alunos envolvidos. A escolha por uma frase apropriada dificultou a utilização dos oito conceitos e principalmente afetou na estrutura do MC.

O último exercício deste módulo foi realizado individualmente com base em seis conceitos sobre uma ontologia da sintaxe e elementos do Java estudados na sala de aula. O propósito do exercício era verificar: i) se os alunos compreenderam os assuntos ensinados na aula; ii) a estrutura cognitiva de cada aluno em relação à ontologia; iii) o comprometimento nos exercícios, verificando se os alunos se esforçavam na construção de mapas significativos. A finalização deste exercício foi acompanhada de críticas destacadas pela Seção 4.2.5.

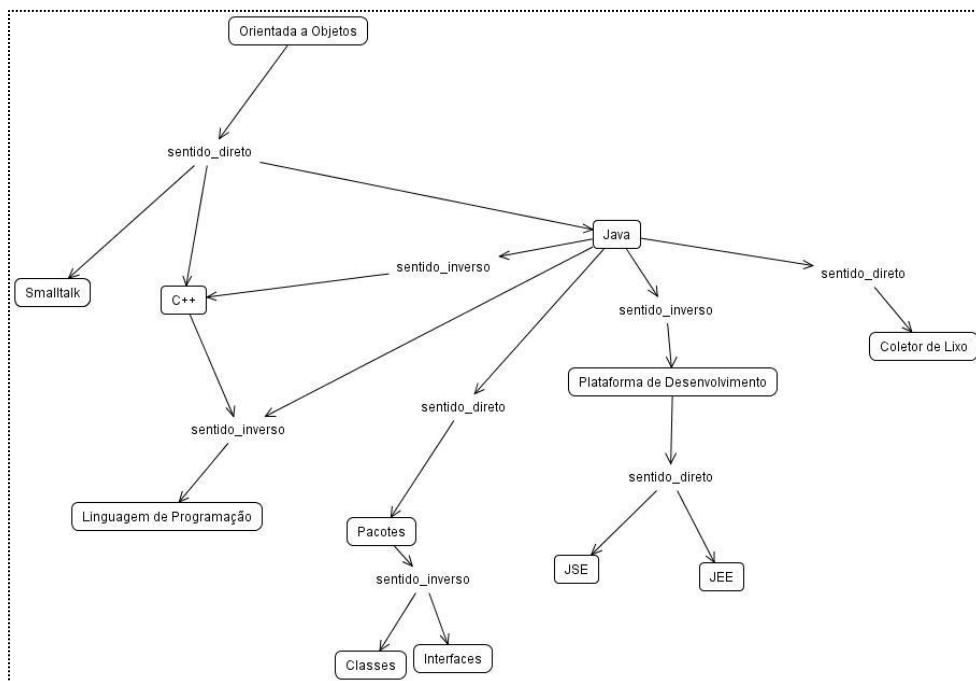
#### 4.1.3 Exercício com o webcmttool

O módulo 3 foi destacado pela utilização do ambiente *WebCMTTool* na construção de MCs. Para tanto, optou-se por fazer uma breve revisão de construção, utilizando agora um ambiente automatizado para realizar grande parte das tarefas que eram feitas à mão.

De acordo com o Capítulo 3, o ambiente é formado por módulos, onde se concentrou no módulo *WebCMEditor* para a formação dos MCs e no módulo *GAADT-CM*, para a avaliação transparente dos MCs construídos pelos alunos no editor. Basicamente, ocorria a construção e edição de mapas e sua submissão à avaliação pelos próprios alunos que analisavam seus erros e acertos acompanhando o relatório gerado após a avaliação.

Para o exercício, os alunos utilizaram a construção de MCs contextualizados, onde se entende por contextualizados, mapas que se baseiam em ontologias de domínio fornecidas pelo professor. A Figura 36 apresenta a ontologia utilizada no exercício dos alunos neste módulo, ela contém em sua estrutura 12 conceitos todos já estudados em sala.

**Figura 37. Ontologia utilizada para o exercício com o *webcmtool***



Fonte: (Elaborada pelo autor).

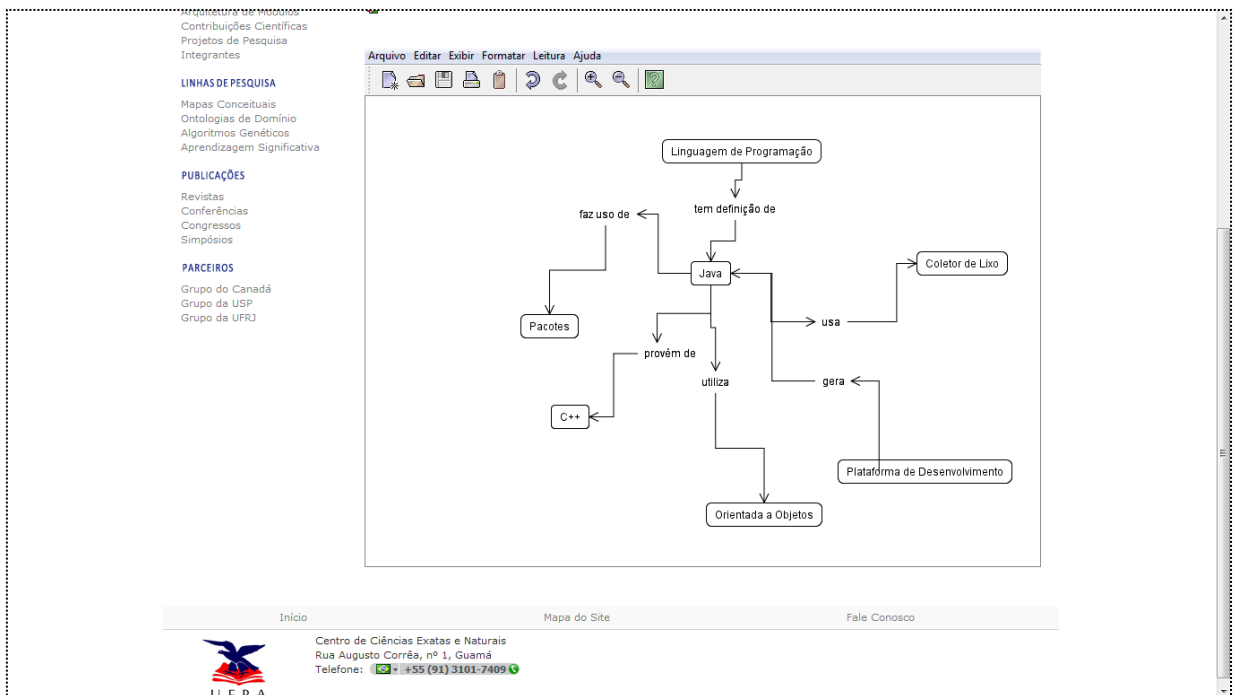
Foi observado, em sala, que quatro alunos não prestaram atenção na revisão de construção de mapas utilizando o ambiente, da mesma forma como não prestaram atenção em como se avaliava os MCs contextualizados. Tal fato gerou dificuldades entre os alunos, contudo a aprendizagem do ambiente foi conduzida pelos alunos por tentativa-e-erro através de experimentos que avaliaram a capacidade de encontrar soluções para problemas mediante várias tentativas até se chegar num resultado (THORNKIDE, 2010).

Quando os alunos submeteram seus MCs para avaliação com a ajuda do módulo *WebCMEditor* era acionado o módulo *GAADT-CM* que se preocupava em avaliar

qualitativamente cada mapa tentando provar os acertos dos alunos. Para a avaliação, se considerou que o tamanho da população inicial seria composto por 150 MCs. Assim, os MCs mais bem adaptados eram selecionados para cruzamento a fim de formar a descendência da espécie que estabeleceria a nova população de descendentes conforme retratado na Seção 2.5. O processo se repetia até que os MCs gerados se convergissem. Depois dessa etapa, os alunos recebiam um relatório com detalhes de sua avaliação, ressaltando, entre outras questões, as notas em percentuais atribuídas às proposições construídas e se o mapa foi considerado satisfatório ou insatisfatório com base em análise e cálculos de adaptação dos MCs.

Na Figura 37 se pode visualizar o MC em fase de construção, desenhado por um dos alunos do curso de Sistemas de Informação que utilizou o ambiente durante o exercício com base na ontologia descrita pela Figura 36. Para tanto, nota-se que foram utilizados os mesmos conceitos presentes na ontologia da figura e algumas frases especificando de maneira clara a compreensão dos conceitos segundo a estrutura cognitiva do aluno.

**Figura 38. Construção do mapa contextualizado de um aluno de sistemas de informação**



Fonte: (Elaborada pelo autor).

O MC da Figura 37, quando avaliado pelo aluno gerou um relatório específico que mostrou detalhes de sua estrutura; a Figura 38 retrata esse relatório, especificando, sobretudo, se o MC foi considerado satisfatório pela avaliação, além dos detalhes de um dos MCs gabaritos gerados após a avaliação com base no MC do aluno e na ontologia do professor.

**Figura 39. Parte do relatório da avaliação do mapa conceitual avaliado pelo aluno**

```

05. Resultado Final:
O desempenho do seu MC foi avaliado como = Satisfatório

(c)Mapas Conceituais Gabaritos Gerados na Avaliação:
01. Compare as Proposições de seu MC com as Proposições deste MCs:
Proposição 0:
{Java, assimetria.processo.entrada.sentido_direto.usa, Coletor de Lixo}
Tipo de aprendizagem: {d - diferenciação progressiva}
Conceitos usados: {Java, Coletor de Lixo}
Dimensão semântica: {assimetria.processo.entrada.sentido_direto}
Frase usada: {usa}

Proposição 1:
{Orientada a Objetos, assimetria.exemplificação.sentido_direto.          tem como exemplo, C++}
Tipo de aprendizagem: {d - diferenciação progressiva}
Conceitos usados: {Orientada a Objetos, C++}
Dimensão semântica: {assimetria.exemplificação.sentido_direto}
Frase usada: {          tem como exemplo}

Proposição 2:
{Orientada a Objetos, assimetria.exemplificação.sentido_direto.          tem como exemplo, Java}
Tipo de aprendizagem: {d - diferenciação progressiva}
Conceitos usados: {Orientada a Objetos, Java}
Dimensão semântica: {assimetria.exemplificação.sentido_direto}
Frase usada: {          tem como exemplo}

Proposição 3:
{Java, assimetria.partição.organização.sentido_direto.          é organizado em, Pacotes}
Tipo de aprendizagem: {d - diferenciação progressiva}
Conceitos usados: {Java, Pacotes}
Dimensão semântica: {assimetria.partição.organização.sentido_direto}
Frase usada: {          é organizado em}

```

Fonte: (Elaborada pelo autor).

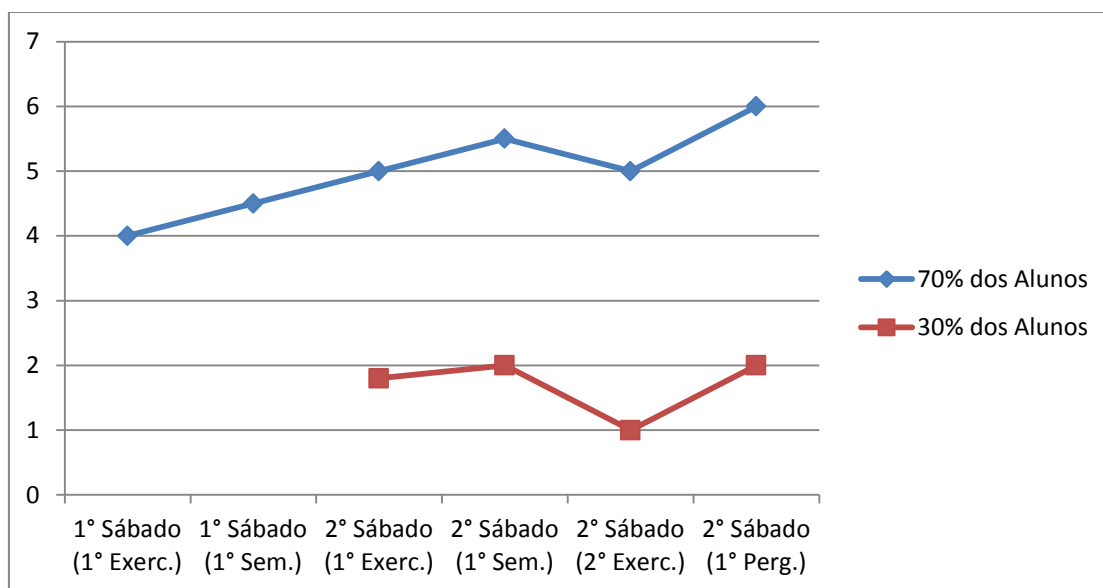
#### 4.1.4 Os resultados da aprendizagem dos alunos

Com o uso do ambiente *WebCMTTool* foi possível gerar e verificar quais mapas foram considerados satisfatórios e quais foram os alunos que tiveram um bom desempenho na construção das proposições. Isto se deve pelo fato de que, o AG consegue analisar cada proposição criada pelo aluno e comparar automaticamente aos MCs gerados com base nas quantidades de proposições construídas pelo aluno e com base na ontologia descritas pelo professor, conforme descrito pelo módulo de avaliação na Seção 3.2.3.

No módulo 1 - que se constituiu de dois sábados - foi ensinado a teoria sobre Java. Neste contexto, observou-se que no primeiro sábado os alunos não possuíam conhecimento algum sobre a linguagem. Enquanto que no segundo sábado, e apesar ainda de algumas visíveis dificuldades, demonstraram que tinham estudado durante a semana, ou seja, entre um sábado e outro, sendo notável a desenvoltura na resolução dos exercícios realizados em sala. O

Gráfico 3 demonstra a média da turma durante o módulo 1 em relação à evolução da aprendizagem com base nos exercícios que foram passadas. É importante ressaltar que a linha de cima representa o aprendizado dos 70% de alunos que frequentaram o curso desde o primeiro sábado, enquanto que a linha de baixo representa o aprendizado dos 30% dos alunos que começaram participando do curso a partir do segundo sábado.

**Gráfico 3. Evolução da aprendizagem em relação a média da turma no módulo 1**



Fonte: (Elaborada pelo autor).

Os alunos que começaram desde o início o curso apresentaram uma média de 4,0 pontos em relação ao 1º exercício realizado. Posteriormente, a aula seguiu-se e à medida que novos conceitos tentavam aderir-se na estrutura cognitiva individual, cada aluno respondia bem as tarefas melhorando sua aprendizagem. Dessa forma, quando se iniciou o 1º seminário o conceito dos alunos apresentaram melhoras, fazendo com que a média da turma subisse um pouco mais, neste caso, em 0,5 pontos em comparação ao 1º exercício.

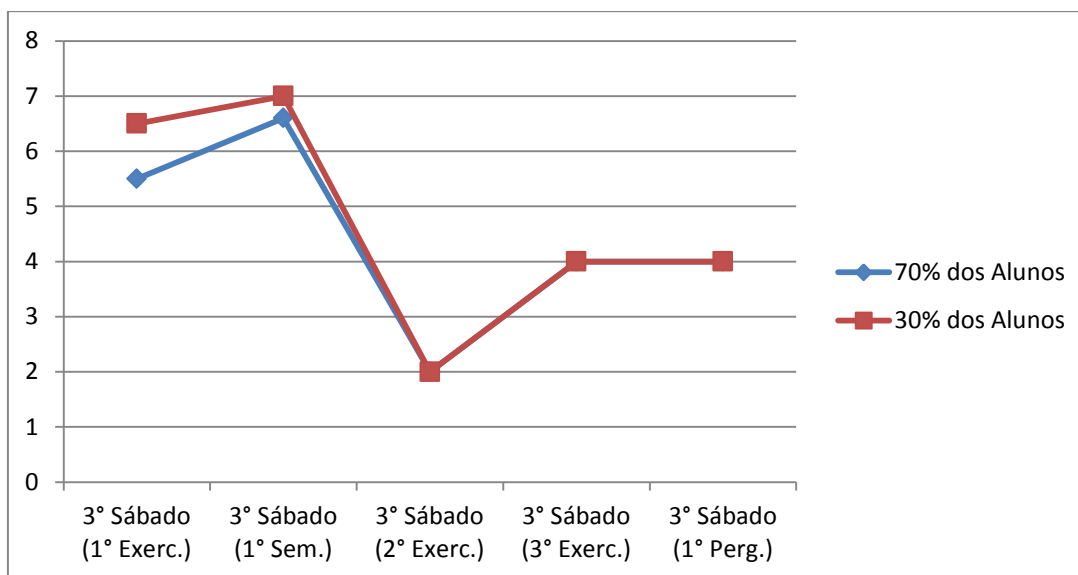
Durante o 1º exercício do 2º sábado, os alunos que iniciaram o curso desde o início apresentaram grandes melhoras, fazendo com que a média da turma tornasse a subir para 5,0 pontos, isto ocorreu porque os alunos estudaram significativamente durante a semana antes da aula o conteúdo apresentado no 1º sábado, o que foi significativo inclusive para uma excelente aula. Quando se iniciou o 1º seminário deste sábado a turma tornou a apresentar novas melhoras apesar de notar dúvidas entre os alunos quanto a alguns assuntos. Um decréscimo do aprendizado refletiu na média da turma no 2º exercício que envolveu novos conceitos do conteúdo apresentado em sala, ou seja, conteúdo novo aos alunos. Após isto, e sanada as

dúvidas durante esse exercício, foi realizado um seminário, que tornou a clarear as ideias da linguagem Java e fomentar o aprendizado, deixando a média em 6,0 pontos.

Os alunos que iniciaram as aulas no 2º sábado não acompanharam a evolução dos outros alunos, isto refletiu no aproveitamento, registrando uma média de 1,8 pontos no 1º exercício e no 1º seminário em 2,0 pontos. Entretanto, durante o 2º exercício a média desses alunos registraram uma queda de 1,0 ponto, devido aos conceitos novos abordados, mas principalmente devido à falta desses alunos no 1º sábado, perdendo portanto 4 horas de aulas. O amadurecimento no conteúdo e tirada as dúvidas garantindo-lhe uma maior atenção a esta cota, os alunos melhoraram seus aproveitamentos durante a sessão de perguntas. Tal fato só mostra a dificuldade que há quando se perde 4 horas de aula e uma gama de conteúdo que não serão mais vistos, a não ser mediante o estudo particular.

O módulo 2 se caracterizou pelas aulas de aprendizagem significativas, MCs, teoria da assimilação e ontologias; e pela introdução e exercícios com papel e caneta realizados envolvendo tais temas. O Gráfico 4 apresenta a aprendizagem em relação à média da turma durante este período.

**Gráfico 4. Média da turma no módulo 2 em relação aos exercícios realizados**



Fonte: (Elaborado pelo autor).

Antes de cada início de aula, ocorreu uma breve revisão sobre o que foi ministrado na aula anterior e foi passado um exercício referente ao conteúdo, contudo com um pouco mais de complexidade. De posse disto, durante o módulo 3, foram coletados dados interessantíssimos a respeito do nivelamento de aprendizagem e preocupação dos alunos. Por exemplo, os alunos

que começaram tarde e que não tiveram um bom aproveitamento no 2º sábado, foram os que se apresentaram e se destacaram durante o 1º exercício do 3º sábado, alcançando, uma média de 6,5 pontos contra 5,5 pontos dos alunos que estão com 100% de presença. O ponto a se destacar é que houve um aumento significativo no aprendizado e interesse, porque esses alunos recuperaram a aula perdida e estudaram os exercícios e conteúdos ensinados assumindo um comprometimento particular. Este 1º exercício tratou de implementações no computador de problemas do mundo real. O 2º e 3º exercício foram desenvolvidos em MCs somente com o uso de papel e caneta.

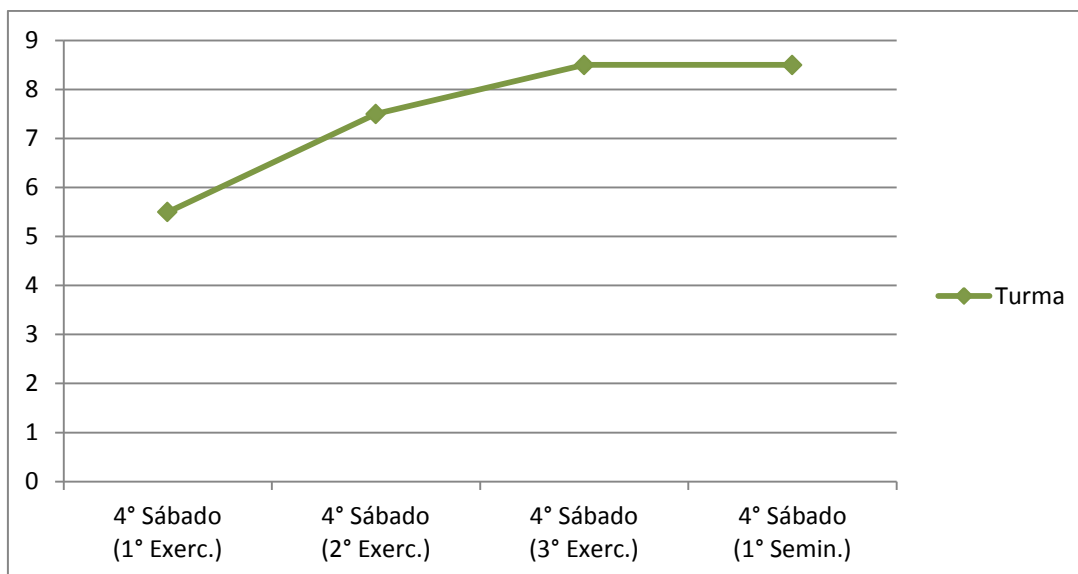
No 2º exercício foram criados MCs contextualizados, utilizando a taxonomia e oito conceitos (ver Seção 4.2.2). Este exercício foi realizado em grupo e discutiu principalmente a formação das proposições nos mapas. Durante a realização deste exercício, os alunos tiveram muitas dúvidas e principalmente dificuldade na construção dos MCs, fazendo com que sua aprendizagem referente ao conteúdo ensinado em sala diminuísse para 3,0 pontos em média. Em suma, conseguiram compreender, durante a prática, o objetivo dos MCs e sua finalidade na aprendizagem, este fato garantiu uma melhora no 3º exercício realizado pelos alunos individualmente com base em seis conceitos e nas frases apresentadas em um papel, fazendo com que a aprendizagem mantivesse em 4,0 pontos mesmo depois de perguntas sobre o tema.

O 4º sábado foi caracterizado pela apresentação do *WebCMTTool* em que neste dia, foi realizado um exercício mais complexo sobre tudo que já se havia estudado, onde se construiu um MC contextualizado com base numa ontologia de domínio em XML, que continha doze conceitos, apresentada pela Figura 4.5. Os alunos receberam aulas sobre a construção de MCs contextualizados e livres de contexto. Os livres foram exercitados na construção de um minicurriculo pessoal, enquanto que os contextualizados foram baseados na ontologia em questão. O Gráfico 4.4 mostra os detalhes da evolução da aprendizagem sob o ponto de vista da média da turma definida pela única linha do gráfico.

O 1º exercício foi sobre desenvolvimento e envolvia assuntos relacionados à aula passada e referente aos estudos sobre MCs, feitos manualmente. Como não houve tanto progresso no 2º e 3º exercícios do 3º sábado, a média da turma neste exercício durante o 4º sábado se efetivou em 5,7 pontos, ou seja, teve um aumento de 0,2 pontos em relação ao 1º exercício também de implementação realizado no 3º sábado, mostrando, com isto, que não houve progresso com a aplicação de MCs realizados sem o *WebCMTTool*.



**Gráfico 5. Evolução da aprendizagem em relação a média da turma no módulo 3**



Fonte: (Elaborado pelo autor).

No 2º exercício se utilizou o ambiente para edição de MCs na Web, a vantagem em relação aos MCs manuais construídos trata-se da facilidade atribuída ao processo, comparado a quem não o utiliza e na satisfação do aluno em construir e poder avaliar seu próprio mapa. Neste exercício, o aluno criou, avaliou e analisou seus erros e acertos com base no relatório gerado sobre sua avaliação. Vale ressaltar que a média da turma neste exercício foi de 7,5 pontos em relação ao exercício envolvendo também MCs, mas com menos conceitos realizados no 2º exercício do 3º sábado que aconteceu no módulo 2.

O 3º exercício foi de desenvolvimento, e no final os alunos demonstraram que estavam entendendo o conteúdo ensinado, assim como quando participaram do 1º seminário do 4º sábado que tratou de assuntos referentes à ontologia de domínio. Neste momento, a média da turma subiu para 8,0 pontos quando utilizado o ambiente *WebCMTool*.

Constatou-se que dezoito alunos aprenderam os conceitos explicados em sala de aula no curso de Java e comprovaram isto durante os exercícios de MCs, durante as implementações e seminários realizados que ocorreram em sala no módulo 3. Também se constatou que dois alunos não compreenderam totalmente o que foi explicado, isto foi verificado pela formação de seus mapas durante os exercícios construídos, seja pelos que utilizaram ou não o ambiente *WebCMTool*. Vale frisar ainda que, os dezoito alunos que aprenderam o conteúdo destacaram que o ambiente os ajudaram a observar melhor a formação de frases com que costumavam utilizar em tarefas e exercícios em outros domínios dentro ou fora da Universidade. A próxima seção mostra as principais dificuldades relatadas.

#### 4.1.5 Dificuldades relatadas pelos alunos

As dificuldades relatadas pelos alunos surgiram durante os exercícios que não envolveram a utilização do *WebCMTTool*, descrito na Seção 4.2.2. Neste cenário, os dezessete alunos que não conheciam MCs, conforme mostra o Quadro 4.2, tiveram mais dificuldades em formar as proposições de um mapa quando relacionado com as frases da taxonomia, do que os três alunos que já tinham um conhecimento prévio sobre o assunto. Contudo, todos os alunos relataram que era difícil escolher as frases de ligação, isto porque não sabiam distinguir entre os diferentes campos semânticos, por exemplo, a frase de ligação <é> pode assumir a semântica [assimetria.definição.definido] e [simetria.equivalência], ou seja, se indicar simetria, a leitura ocorre de maneira bidirecional diferente da assimetria que a leitura é unidirecional, além de que uma expressa o sentido de equivalência e a outra de definição.

Os alunos informaram, também, que a construção manual dos mapas era tediosa, notando inclusive que os mesmos rabiscaram várias vezes durante suas revisões, para verificar se seus mapas correspondiam ao conhecimento aprendido. Durante a aplicação dos exercícios no ambiente, os alunos foram auxiliados na construção dos mapas de maneira automática com a ajuda das ferramentas presentes no ambiente, garantindo uma avaliação positiva do ambiente pelos que usaram.

Posteriormente, os alunos puderam avaliar seus mapas e criteriosamente analisaram seus erros pelo relatório. Alguns alunos ressaltaram que esperavam ver no relatório uma nota avaliada de 0 a 10 de mapas e se surpreenderam, quando viram notas em percentuais baseado nas proposições de seus mapas e uma mensagem informando se o mapa era satisfatório.

#### 4.1.6 Entrevista e questionário de satisfação do ambiente

Para Cybis (2003), questionários de satisfação e entrevistas são técnicas prospectivas que servem para avaliar a satisfação do usuário. Elas são caracterizadas por serem baratas, simples, rápidas e fáceis, não gerando tantas dúvidas aos usuários.

Para a entrevista foram escolhidos aleatoriamente cinco alunos e para o questionário de satisfação foram selecionados catorze alunos que responderam vinte e duas perguntas as quais podem ser melhor visualizadas pelo Apêndice D.

As perguntas que sustentaram a entrevista entre os cinco alunos foram:

1. Você sentiu dificuldade ao trabalhar com MCs? Qual(is)?

2. Você sentiu dificuldade em avaliar um MC pelo WebCMTool? Qual(is)?
3. Você acha que o ambiente o(a) ajudou no aprendizado do curso? Por quê?
4. Se pudesse contribuir para o melhoramento, qual dica daria?

O questionário a qual os alunos se submeteram está dividido em três seções: o perfil, o conhecimento sobre os conceitos em sala, sobre o webcmttool. Questões relacionadas à usabilidade e interação, são definidas na próxima seção relacionadas aos testes por profissionais.

## 4.2 SEGUNDA ETAPA: ANÁLISE DE USABILIDADE

De acordo com Cybis (2003), para se avaliar uma usabilidade, deve-se considerar qual técnica será utilizada, quem serão os avaliadores e em que nível o projeto se encontra. Sobre este ponto de vista, utilizou-se a técnica preditiva também denominada técnica diagnóstica cujo objetivo é a prevenção de erros de projeto sem a participação direta do usuário. São classificadas em: a) avaliações analíticas, nas quais a técnica é aplicada nas primeiras etapas de desenvolvimento da interface; b) avaliação heurística, que consiste em realizar uma inspeção do sistema, e; c) avaliação por *checklist*, que é a técnica baseada em lista de verificação, através da qual se faz o diagnóstico dos problemas gerais e específicos da interface.

Esta etapa descreve a análise e os resultados obtidos segundo as técnicas preditivas: analítica, heurística e checklist, aplicadas para o ambiente e como esses resultados foram utilizados para melhorá-lo.

### 4.2.1 Usabilidade

A usabilidade é considerada como o fator que assegura que um determinado produto seja fácil de usar, eficiente e agradável (CHAVES, 2010). Em geral, a usabilidade se relaciona com regras específicas que visam melhorar a facilidade de uso de um produto ao longo de seu processo de criação, procurando a otimização das interações estabelecidas entre as pessoas e o produto (NIELSEN, 2010). De acordo com as ideias apresentadas por Preece (2005), a usabilidade pode ser dividida em seis componentes de qualidade:

- **Eficácia:** refere-se a quanto um sistema é bom em fazer o que se espera dele.

- **Eficiência:** refere-se à maneira como um sistema auxilia os usuários na realização de suas tarefas, mais especificamente à rapidez com que o usuário desempenha as atividades necessárias após aprender a utilizar o sistema.
- **Segurança:** relacionado à proteção do usuário de situações perigosas ou indesejáveis. Abrange questões relacionadas à ocorrência de erros por parte do usuário, como, por exemplo, a necessidade de prevenir o usuário de cometer erros graves e de fornecer ao usuário várias formas de recuperação ou retorno no caso de erros.
- **Utilidade:** se refere à verificação de um sistema propicia as funcionalidades que possibilitam a seus usuários realizar as tarefas de que necessitam ou desejam de forma a satisfazê-los.
- **Facilidade de aprendizado:** também conhecido por *Learnability*, refere-se à facilidade com que o usuário é capaz de realizar tarefas simples ao utilizar o sistema pela primeira vez, ou seja, verifica o quão fácil é para o usuário aprender a usar o sistema em questão.
- **Facilidade de memorização:** também conhecido por *Memorability*, trata de quão fácil é para o usuário lembrar-se de como utilizar o sistema após passar um período sem utilizá-lo.

A usabilidade ressalta a importância de se pensar no usuário e em sua reação ao utilizar um sistema (FERREIRA, 2002); por essa razão, possui relação direta com a satisfação ou sua frustração do usuário (ANANDHAN, 2006). Principalmente para aplicações *web*, como é o caso do *WebCMTool*, a usabilidade é uma questão fundamental; ao se deparar com *web sites* com falhas na usabilidade, como por exemplo, informações difíceis de serem encontradas, os usuários costumam abandonar o sistema ou apresentar baixa produtividade (NIELSEN, 2010).

Esta obra se baseou nos fundamentos de usabilidade estabelecido pela norma ISO 9241-11 (ISO, 2002) que considera mais o ponto de vista do usuário e seu contexto de uso do que as características ergonômicas do produto e nas regras definidas por Nielsen. Esta norma redefine usabilidade como “a capacidade de um produto ser usado por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso”.

Para que esses fundamentos e os componentes apresentados por Preece (op. cit.) ocorressem no *WebCMTool*, era fundamental que o usuário não sentisse preso a determinada

funcionalidade do ambiente. Para tanto, foram feitos testes de usabilidade na medida em que o ambiente era construído, conforme atesta Rocha e Baranauskas (2003):

[A Avaliação não deve ser vista como uma fase única dentro do processo de design e muito menos uma atividade a ser feita somente no final do processo e “se der tempo”. Idealmente, a avaliação deve ocorrer durante todo o ciclo de vida do design e seus resultados utilizados para a melhoria gradativa da interface].

Os testes realizados ocorreram durante a construção dos módulos descritos no Capítulo 3 e de acordo com o tempo estabelecido no cronograma do Quadro 4.2. Foram realizados três testes de usabilidade ao longo do projeto, organizados da seguinte forma:

1. Teste 1: usabilidade do módulo *webcmeditor*
2. Teste 2: usabilidade do módulo *webon\_tool*
3. Teste 3: usabilidade do *webcmtool*

Os Testes 1 e 2 consistem da técnica analítica e ocorreram após o desenvolvimento dos editores de mapas conceituais e ontologias para a *web*, respectivamente. Eles foram realizados pelas mesmas pessoas que participaram do desenvolvimento, ou seja, os analistas de sistemas. Por outro lado, o Teste 3 foi realizado por profissionais da área não ligados ao time de desenvolvimento do projeto, que avaliaram o ambiente heurísticamente e por *checklist* considerando as regras e fundamentos de usabilidades propostos por Nielsen (op. cit.). As Seções 4.3.2 e 4.3.3 descrevem os resultados encontrados por essas duas avaliações do Teste 3.

#### **4.2.2 Resultados encontrados na avaliação heurística**

A avaliação de interfaces reúne um conjunto de técnicas e metodologias usadas com o objetivo de conhecer as necessidades dos usuários e as dificuldades que eles enfrentam durante a experiência de uso de uma interface. Adicionalmente, para Rocha e Baranauskas (2003), a avaliação heurística é um método básico de inspeção da interface. Ele serve para controlar os problemas de usabilidade nas interfaces para que esses possam ser atendidos durante a etapa de projeto ou reprojeto.

Segundo Ongaro (2004), a avaliação heurística deve ser feita por mais de um avaliador, isto porque é difícil uma única pessoa conseguir identificar todos os problemas de usabilidade que o sistema apresenta. Cada avaliador examina a interface várias vezes, além de inspecionar os elementos de diálogo e fazer comparações com os princípios de usabilidade. Para ajudar a identificar os problemas, os avaliadores utilizaram as regras heurísticas definidas por Nielsen (1990; 1994):

1. **Visibilidade do status do sistema:** o usuário tem de estar informado sobre o que está acontecendo no sistema, através de *feedbacks*, dentro de um determinado tempo;
2. **Compatibilidade do sistema com o mundo real:** o sistema deve utilizar a linguagem que o usuário conhece e não a linguagem técnica;
3. **Controle e liberdade para o usuário:** quando o usuário escolher por engano uma função, o sistema deve mostrar, de maneira clara e rápida, a forma de voltar ao estado anterior;
4. **Consistência e padrões:** seguir as convenções de plataforma computacional utilizada;
5. **Prevenção de erros:** desenvolver um bom design para prevenir o erro antes de ele acontecer;
6. **Reconhecimento ao invés de lembrança:** tornar objetos, ações e opções visíveis para que o usuário não precise lembrar informações de uma parte para outra do diálogo;
7. **Flexibilidade e eficiência de uso:** o sistema tem que oferecer aos usuários mais experientes possibilidades de acelerar ações frequentes;
8. **Estética e projeto minimalista:** os diálogos devem apresentar apenas informações relevantes e a visibilidade deve ser favorecida;
9. **Ajuda aos usuários para reconhecer, diagnosticar e recuperar erros:** as mensagens de erro devem ser expressas na linguagem do usuário, haverá indicação do problema e, construtivamente, a sugestão para uma solução;
10. **Ajuda e documentação:** o sistema deve fornecer ajuda e documentação de forma acessível e não muito extensa.

A avaliação Heurística foi realizada por três avaliadores especialistas ligados a área de Computação com experiência sobre o tema. Ela consistiu em realizar inspeções sobre a

interface tendo como base as dez Heurísticas de Nielsen (op. cit.). Pela sua facilidade e simplicidade podem ser realizadas em algumas horas de duração, por exemplo, o tempo médio que os três avaliadores levaram para a avaliação do *WebCMTool* consistiu de duas horas de duração. Após os avaliadores aplicarem as regras heurísticas, deve-se formar uma lista de problemas (PRATES e BARBOSA, 2006), de modo que cada problema deva estar definido com um grau de severidade baseada na escala definida pelo Quadro 8.

Os problemas encontrados pelos alunos na aplicação foram os indícios de que seria necessário que testes de usabilidade fossem utilizados no ambiente. A partir deste ponto de vista, que se optou em avaliar a usabilidade do ambiente. Os avaliadores sentiram, inicialmente, dificuldades na construção dos mapas conceituais contextualizados pedindo ajuda nesta etapa da aplicação para que pudessem analisá-la.

**Quadro 8. Grau de severidade definido na avaliação**

Severidade	Tipo	Descrição
0	Sem importância	Não afeta a operação da interface.
1	Cosmético	Não há necessidade imediata de solução.
2	Simple	Problema de baixa prioridade ( <u>pode</u> ser reparado).
3	Grave	Problema de alta prioridade ( <u>deve</u> ser reparado).
4	Catastrófico	Muito grave, deve ser reparado de qualquer forma.

Fonte: (Elaborado pelo autor).

Considerando os fundamentos supracitados, o Quadro 9 apresenta a estrutura de como foi organizado cada problema anexado pelos avaliadores a respeito do *WebCMTool*, de acordo com as Heurísticas de Nielsen, o grau de severidade.

**Quadro 9. Estrutura dos resultados da avaliação realizada**

Heurística	
Questão:	Grau de Severidade
Problema:	( ) 0 / ( ) 1 / ( ) 2 / ( ) 3 / ( ) 4

Fonte: (Elaborado pelo autor).

Para a avaliação, os especialistas em Computação utilizaram como recursos materiais editores de textos automáticos, papel, caneta e ferramentas de desenhos como *Photoshop* para elaborar os *printscreen* dos problemas que foram encontrados e soluções para estes problemas.

O Quadro 10 mostra parte da avaliação Heurística já compactada em uma única lista. O Anexo A aborda mais detalhes a respeito dos problemas, qual a visão e sugestão que os avaliadores tiveram e sobre os nomes dos avaliadores, além das imagens por eles apresentadas como forma de retratar o problema.

**Quadro 10. Lista de parte dos problemas e os resultados dos avaliadores**

<b>1. Visibilidade do <i>status</i> do sistema</b>	
<b>Questão:</b> Os usuários são mantidos informados sobre o progresso do sistema com apropriado <i>feedback</i> em um tempo razoável?	<b>Grau de Severidade</b>
<b>Problema:</b> a) Houve uma pequena demora durante a edição das frases de ligação onde neste tempo, não houve um <i>feedback</i> ;	( ) 0 / (X) 1 / ( ) 2 / ( ) 3 / ( ) 4
<b>2. Compatibilidade do sistema com o mundo real</b>	
<b>Questão:</b> O sistema utiliza conceitos e linguagem familiar com o usuário em vez de termos orientados ao sistema? O sistema utiliza convenções do mundo real, exibindo informações com uma ordem lógica e natural?	<b>Grau de Severidade</b>
<b>Problema:</b> a) linguagem do programa excessivamente técnica, leigos podem ter muitos problemas com a utilização do mesmo.	( ) 0 / ( ) 1 / ( ) 2 / (X) 3 / ( ) 4
<b>3. Controle e liberdade para o usuário</b>	
<b>Questão:</b> Os usuários podem fazer o que querem e quando querem?	<b>Grau de Severidade</b>
<b>Problema:</b> a) Sim.	(X) 0 / ( ) 1 / ( ) 2 / ( ) 3 / ( ) 4

Fonte: (Elaborado pelo autor).

Os avaliadores também tiveram a preocupação em sugerir melhoras para o ambiente como forma de consertar o problema encontrado durante a avaliação. Estas melhorias foram diferentes entre os três avaliadores bem como os problemas encontrados pelos mesmos, enriquecendo ainda mais a análise de usabilidade, diferentemente se a avaliação tivesse, por



exemplo, ocorrida somente por um único avaliador. Assim, o Quadro 11 demonstra uma dessas sugestões tratadas segundo detalhes abordados pelo avaliador.

#### Quadro 11. Resultado da avaliação do princípio estética e projeto minimalista

**Problema:** O texto da página inicial contém informações válidas, porém com uma fonte de texto pequena e não dividido em seções pode fazer com que o usuário perca o interesse em ler todo o texto.



The screenshot shows the WebCMTool website. The main content area contains a red-bordered box with the following text:

**Conheça o WebCMTool -> Início**

O WebCMTool é um ambiente que visa facilitar a avaliação da Aprendizagem Significativa (AS) mediada por Mapas Conceituais (MCs) e Ontologias de Domínio. A facilidade da avaliação se caracteriza por diminuir a sobrecarga de trabalho do professor haja vista que ocorre de forma automática sobre a responsabilidade de um Algoritmo Genético (AG) incorporado ao ambiente. A tarefa de avaliação proposta no ambiente WebCMTool tende ao construtivismo, além de valorizar a idiossincrasia de cada aluno tentando provar quando este está correto, ressaltando acima de tudo, qual o resultado da avaliação sob o ponto de vista do próprio aluno que é submetido a uma análise detalhada de sua avaliação por meio de um relatório gerado pelo AG.

O propósito desta análise é fazer com que o aluno analise e entenda os erros e acertos em seu mapa conceitual avaliado, bem como entender porque sua avaliação foi considerada satisfatória ou não. Para que a tarefa de avaliação ocorra é necessário três insumos de entrada: o mapa do aluno, a ontologia do professor e uma taxonomia de frases de ligação que contém frases semanticamente definidas na língua portuguesa. Durante o processo de avaliação, são gerados mapas conceituais descendentes escolhidos com base em um valor de adaptação, quando este for maior ou igual à média da população. De posse desta estrutura, os MCs gerados por descendência resultante da avaliação, são utilizados para análise e compreensão de resultados pelo aluno, que verifica o valor atribuído às proposições de seus mapas, a nota em percentual que valida a quantidade de proposições consideradas corretas e o resultado de desempenho de seu mapa. Um protótipo do ambiente desenvolvido foi validado com base em duas ontologias de domínio: sobre o Ciclo da Água e sobre Paradigmas de Programação.

WebCMTool trata-se da extensão do CMTTool criado pelo pesquisador Francisco Edson Lopes da Rocha, cujo principal objetivo é a avaliação de MCs de forma automática. Para a criação destes MCs são disponibilizados conjuntos de conceitos por meio de uma ontologia de domínio mas sem as frases que une esses conceitos. A ontologia permite compartilhar conhecimento, neste caso, o conhecimento do professor ou especialista acerca de um domínio. Neste contexto, os mapas são desenvolvidos pelos alunos, no qual se baseiam em conceitos presentes em uma ontologia de domínio previamente criada por um professor.

Sobre o ponto de vista da pesquisa, a integração de mapas conceituais, ontologias de domínio e algoritmos genéticos possibilitam um avanço no estado da arte de avaliação e acompanhamento automático da aprendizagem. Quebra-se o paradigma das avaliações apenas quantitativas, apresentando uma nova abordagem de acompanhamento gradual e contínuo das atividades do estudante. Nesta abordagem pode-se fazer o acompanhamento individual, respeitando a forma idiossincrática de aprender, e/ou de grupo de estudantes, sendo possível agrupá-los por características cognitivas específicas ou por grau de desenvolvimento.

**Princípio de usabilidade infringido:** Estética e projeto minimalista

**Grau de severidade:** 2

Fonte: (Elaborado pelo autor).

#### 4.2.3 Resultados encontrados na avaliação por checklist pela ferramenta ergolist

As avaliações por *checklist* constituem uma técnica capaz de identificar vários problemas gerais e repetitivos da interface. Conforme Cybis (op. cit.), as vantagens da avaliação realizada por meio de *checklist* são: rapidez na aplicação, causando redução de custos da avaliação; facilidade de identificação de problema de usabilidade, devido à especificidade das questões do *checklist*; sistematização da avaliação, que garante resultados mais estáveis, mesmo quando aplicada separadamente por diferentes avaliadores e, para

realizar a avaliação, não há necessidade de profissionais especializados em ergonomia, pois o conhecimento ergonômico está embutido no próprio *checklist*.

Em geral, avaliações por *checklist* são desenvolvidas por meio de uma lista de questões e podem estar acompanhadas de glossários explicativos que servem para esclarecer dúvidas relacionadas às questões. Para a avaliação do *WebCMTTool* a ferramenta utilizada foi a *checklist* ergolist criada pelo LabiUtil (Laboratório de Utilizabilidade) da Universidade Federal de Santa Catarina (LABUTIL, 1998). Na Figura 39 pode-se visualizar a ergolist.

Figura 40. Visão geral da ferramenta ergolist

Ícone	Função
<input type="checkbox"/>	aciona Checklist
<input type="checkbox"/>	aciona Glossário
<input type="checkbox"/>	aciona Mais sobre...

**Instruções**

**Recomendação Geral**

Durante a aplicação dos Checklists tenha o cuidado de (a) não sair para outros sites e (b) de evitar o acionamento do link que leva para a HOMEPAGE do Ergolist. Essas ações poderão acarretar na perda das respostas já fornecidas.

No primeiro caso, volte ao ErgoList através da opção *Back* de seu navegador. No segundo, uma mensagem lhe será apresentada permitindo que você confirme ou não sua intenção de anular sua tarefa.

**Passos**

1 - Ao receber a página com o Menu de Checklists organizado por critérios ergonômicos, acione um dentro os checklists propostos. Para tanto, clique sobre o botão à esquerda do nome do critério. Obs: Você pode obter maiores detalhes sobre os critérios ergonômicos através dos recursos do glossário (G) e das informações adicionais (I) disponíveis.

2 - Realize a aplicação do checklist escolhido, valendo-se dos recursos de glossário (G) e de informações adicionais (I) relacionadas as questões e seus termos.

3 - Para realizar um novo checklist, volte ao Menu de Checklists, acionando o botão correspondente no final das páginas de checklists.

3.1 - Na página Menu de Checklists, escolha um outro checklist que lhe seja pertinente, repetindo as ações

**Glossário**

O glossário aqui apresentado não tem a intenção de ser uma obra de referência completa. O seu objetivo é esclarecer os significados de alguns termos usados nesse trabalho. Um termo pode ser incluído nesse glossário, se utilizado inconsistentemente na literatura em geral, ou se o seu significado nesse trabalho for mais específico do que o normal.

Fonte: (LABUTIL, 1998)

Esta ferramenta é baseada na norma internacional ISO 9241-11 e oferece várias recomendações, sendo cada uma delas especializada em um aspecto ou critério que determina a ergonomia das Interfaces Humano-Computador. A ferramenta é constituída por 18 recomendações ergonômicas que são:

1. **Presteza:** verifica a compatibilidade do sistema com as expectativas e necessidades do usuário em sua tarefa. É composta por 17 questões;
2. **Agrupamento por localização:** verifica se a distribuição espacial dos itens traduz as relações entre as informações. É composta por 11 questões;

3. **Agrupamento por formato:** verifica os formatos dos itens como meio de transmitir associações e diferenças. É composta por 17 questões;
4. **Feedback:** avalia a qualidade do *feedback* imediato às ações do usuário. É composta por 12 questões;
5. **Legibilidade:** verifica a legibilidade das informações apresentadas nas telas do sistema. É composta por 27 questões;
6. **Concisão:** verifica o tamanho dos códigos e termos apresentados e introduzidos no sistema. É composta por 14 questões;
7. **Ações Mínimas:** verifica a extensão dos diálogos estabelecidos para a realização dos objetivos do usuário. É composta por 5 questões;
8. **Densidade Informacional:** avalia a densidade informacional das telas apresentadas pelo sistema. É composta por 9 questões;
9. **Ações Explícitas:** verifica se é o usuário quem comanda, explicitamente, as ações do sistema. É composta por 4 questões;
10. **Controle do Usuário:** avalia as possibilidades de o usuário controlar o encadeamento e a realização das ações. É composta por 4 questões;
11. **Flexibilidade:** verifica se o sistema permite personalizar as apresentações e os diálogos. É composta por 3 questões;
12. **Experiência do Usuário:** avalia se usuários, com diferentes níveis de experiência, têm iguais possibilidades de obter sucesso em seus objetivos. É composta por 6 questões;
13. **Proteção contra erros:** verifica se o sistema oferece as oportunidades para o usuário prevenir eventuais erros. É composta por 7 questões;
14. **Mensagens de erro:** avalia a qualidade das mensagens de erro enviadas aos usuários em dificuldades. É composta por 9 questões;
15. **Correção de erros:** verifica as facilidades oferecidas para que o usuário possa corrigir os erros cometidos. É composta por 15 questões;
16. **Consistência:** avalia se é mantida uma coerência no projeto de códigos, telas e diálogos com o usuário. É composta por 11 questões;
17. **Significados:** avalia se os códigos e denominações são claros e significativos para os usuários do sistema. É composta por 12 questões;
18. **Compatibilidade:** verifica a compatibilidade do sistema com as expectativas e necessidades do usuário em sua tarefa. É composta por 21 questões.

A *ergolist* permite gerar um laudo técnico a respeito de toda a avaliação realizada sobre a usabilidade. Cada questão da ferramenta contém como resposta as opções: i) sim; ii) não; iii) não aplicável e; iv) adiar resposta. Além de um espaço para o avaliador inserir comentários sobre a questão que se está analisando. A Figura 40 mostra o aspecto de uma questão.

**Figura 41. Aspecto de uma questão na ferramenta *ergolist***

Questão 2 de 12 Significados

Os títulos das páginas de menu são explicativos, refletindo a natureza da escolha a ser feita? ▶

Sim    
  Não    
  Não aplicável    
  Adiar resposta

Comentários:

Fonte: (LABUTIL, 1998).

Na Figura 40, é possível notar um ícone “G” que se refere ao glossário da ferramenta, também é possível nota uma seta na lateral direita, esta seta se refere a significados e referências da questão tratada, além de um comentário mais significativo da questão. No Anexo B é possível notar mais detalhes abordados nesta avaliação e os resultados coletados.

Este capítulo, discutiu as análises dos resultados que foram obtidos no projeto *WebCMTool* desde a fase de concepção e elaboração até a fase de testes e avaliações por especialistas. As análises dos resultados foram divididas e organizadas em três etapas em conformidade com estudos de modelagem ágil *Scrum*.

A primeira etapa apresentou os estudos que validaram a continuação do projeto com base em resultados sobre o escopo, tempo, recursos, aspectos, objetivos, benefícios e riscos. A segunda etapa apresentou os dados da aplicação do ambiente numa Universidade. Neste cenário, foram realizados exercícios com e sem o uso do ambiente onde comparação foram feitas entre as duas situações de aplicação. Também foram discutidos sobre os perfis dos alunos e o impacto dos seus conhecimentos na aplicação, as dificuldades encontradas por estes, os resultados e melhoramentos da aprendizagem durante o curso de aplicação. A terceira etapa consistiu do estudo de usabilidade realizado por especialista que encontraram erros na

interação humano-computador e de ergonômias após terem sido relatados problemas pelos alunos durante a aplicação do ambiente.

Este capítulo refletiu a concretização do protótipo *WebCMTool* desenvolvido, demonstrando dados conclusivos que enriquecem ainda mais o valor agregado no contexto científico.

## 5 CONSIDERAÇÕES FINAIS

A modelagem e construção do ambiente *WebCMTTool* permitiu ganhos na área do ensino pelo apoio às práticas construtivistas com uma avaliação automática, utilizando a web como principal mediador e difusão do conhecimento.

Conforme mostrado nesta dissertação, encontrar maneiras viáveis de utilizar o modelo construtivista em sala de aula, para incluir um modelo adequado de avaliação, ainda representa um desafio a ser superado. O ambiente desenvolvido contribui para auxiliar nesta superação das seguintes maneiras:

- viabiliza um modelo de *aprender a aprender* (NOVAK e GOWIN, 1984), estabelecido na linguagem visual no *WebCMEditor* e *WebOn\_Tool*;
- viabiliza um modelo de avaliação da aprendizagem (ZABALA, 1998) que respeita o modo individual de aprender de cada aluno, sem carga adicional ao professor;
- viabiliza a aplicação em sala de aula e à distância de um modelo construtivista baseado na aprendizagem significativa, diminuindo a sobrecarga do professor na tarefa de avaliação, além de permitir que alunos construam conhecimento significativo na modalidade não presencial;
- viabiliza a inserção no contexto à distância, permitindo que cada vez mais alunos e professores construam conhecimento com as práticas da aprendizagem significativa;
- permite gerar relatório individualizados dos alunos, comparando seu progresso após cada avaliação.

A avaliação automática, mediada por mapas conceituais e ontologias de domínio na *Web*, resgata uma questão ainda em aberto, principalmente em Ambientes Virtuais de Aprendizagem (AVAs) (MOODLE, 2010): **Como avaliar automaticamente e qualitativamente a aprendizagem pela web, respeitando os processos cognitivos individuais dos alunos?**

Além disso, o ambiente proposto possui características do CMTTool que viabiliza a avaliação qualitativa da aprendizagem de grupos de alunos de qualquer tamanho, com o mínimo de esforço por parte do professor (ROCHA, 2007).

A ideia fundamental concretizada no ambiente é facilitar ao aluno o domínio da estrutura conceitual que envolve todo o tipo de conhecimento. Logo, o ambiente avalia

qualitativamente questões abertas e questões fechadas (por exemplo, questões do tipo **verdadeiro** ou **falso**). Uma questão aberta poderia ser representada por um exercício do tipo *fill in the map* – o estudante recebe um mapa parcialmente construído e preenche as lacunas. Este exercício aberto envolve a avaliação dos processos cognitivos *lembrar, entender e aplicar* descritos em (ROCHA, op. cit.).

Em relação à arquitetura do ambiente, a adoção de modelos de dados, baseados em XML para a taxonomia de frases de ligação, ontologia e mapa conceitual segue a tendência dominante para a transferência de informações na Web. Adicionalmente, a especificação JSF, na qual o ambiente foi modelado, permite uma melhor comunicação com tecnologias e frameworks que garantem maior flexibilidade para os dados no ambiente.

As ontologias são usadas no ambiente como compartilhamento de conhecimento, de tal modo que professor codifica o conhecimento estrutural associado a um sistema de classificação de dimensões semânticas que permite a ele aliviar a sua carga de trabalho.

Neste contexto, MCs contextualizados são construídos, transformando o processo de avaliação num processo de otimização combinatória. Para realizar este processo, é preciso um mecanismo que saiba combinar, com grande aleatoriedade, as estruturas de conhecimento que estão decompostas na ontologia. Esta necessidade levou ao projeto de um algoritmo genético adaptado à resolução do problema de avaliação.

O AG possui uma função de adaptação que calcula o valor cognitivo de cada mapa que ele gera, levando em conta o conhecimento que o professor incluiu na ontologia e aquele que o aluno apresentou no seu mapa conceitual, representando, assim, os insumos do algoritmo. O resultado desta tarefa pode ser um mapa ótimo, um mapa muito bom, um mapa bom ou simplesmente um mapa medíocre, relativo ao mapa submetido pelo aluno.

Além das contribuições científicas (seção 5.1), este trabalho favoreceu a combinação e exploração de tecnologias que, em geral, aparecem utilizadas separadamente na pesquisa científica: a combinação de ontologias e algoritmos genéticos na Web é uma importante contribuição desta dissertação que pode ser aplicada a qualquer contexto.

Os objetivos estabelecidos foram efetivamente alcançados. Entretanto, para que a avaliação qualitativa da aprendizagem na Web aconteça na forma prevista neste trabalho, é necessário que professores e estudantes desempenhem seus papéis de forma satisfatória. Aos professores cabe a tarefa de desenvolver ontologias para organizar o ensino em torno de estruturas que sejam apropriadas à construção do conhecimento nos domínios da(s) sua(s) área(s) de atuação e, eventualmente, a ampliação e/ou modificação das regras de proximidade

semântica inscritas na taxonomia de frases de ligação; aos estudantes cabe o processo de mapeamento conceitual do conhecimento contido nas ontologias, podendo para isso explorar as facilidades adicionais oferecidas pelo ambiente para atingir seus objetivos.

## 5.1 CONTRIBUIÇÕES CIENTÍFICAS

Como contribuições científicas geradas por esta pesquisa têm-se:

### ***Congressos internacionais - Qualis B2***

1. Rodrigues, R. C. R., Rocha, F. E. L. WebCMTool: *Um Ambiente Web para Facilitar a Avaliação Automática da Aprendizagem mediada por Mapas Conceituais e Ontologias de Domínio*. Conferência IADIS Ibero Americana WWW/Internet, pp. 219-226. 5 - 7 de Novembro. 2011.

### ***Trabalho de conclusão de curso***

1. Gusmão, A. L., Borges, T. B. *Melhoria de desempenho do algoritmo genético da ferramenta CMTool*. Trabalho de Conclusão de Curso para o título de Bacharel em Sistemas de Informação da Universidade Federal do Pará. 2011.

## 5.2 TRABALHOS FUTUROS

O WebCMTool é um ambiente desenvolvido com base em um dos trabalhos futuros descrito em (ROCHA, 2007): i) a disponibilização do CMTool na Internet. Sua versão atual é a 1.0 e possui uma arquitetura que atende a todos os objetivos previstos neste trabalho. Essa arquitetura pode evoluir, acrescentando novas funcionalidades ou ainda melhorar funcionalidades já existentes. Algumas vislumbradas são:

### **1. Em relação ao ambiente:**

- a. Colaboração entre professores e alunos, criando portfólios de mapas conceituais e ontologias de domínio, de acordo com os grupos de usuários registrados. Esses portfólios armazenariam os mapas na sessão dos alunos e as ontologias na sessão dos professores, permitindo, por exemplo, aos professores compartilhar suas ontologias entre as sessões diferentes sessões de grupos de alunos (turmas);



- b. Para ajudar o ensino/aprendizagem de conhecimento procedural, em paralelo com conhecimento conceitual, considera-se a adição de bases de conhecimento (KBs) ao repositório do *WebCMTool*. Estas KBs poderiam armazenar artefatos tais como ilustrações, esqueletos de código de programas de computador, jogos, informações sobre melhores práticas de ensino de disciplinas específicas, simulações, entre outras. Cada um destes artefatos seria indexado por um ontologia. O objetivo seria alavancar a prática de atividades colaborativas e discussões de grupos sobre tópicos específicos de diferentes disciplinas, guiado por um vocabulário comum. KBs conectadas a ontologias seriam também susceptíveis a buscas especializadas;
- c. O ambiente pode também se beneficiar da presença de algoritmos para extrair radicais de palavras (*stemming algorithms*). Isto evitaria problemas acidentais, que podem aparecer durante a pesquisa dos MCs mais adaptados, causados pela flexão do plural ou conjugações de verbos;
- d. Integração ao ambiente à distância Moodle, garantindo uma maior colaboração entre os envolvidos.

## **2. Em relação à ontologia, à gramática de inclusividade e ao GAADT-CM:**

- a. Implementação de uma função para criar sequenciamentos de materiais de cursos. O objetivo principal é delinear o conteúdo de cursos à distância;
- b. Implementação de uma função capaz de gerar automaticamente perguntas/respostas baseadas nas relações semânticas entre conceitos. O objetivo principal é enriquecer as formas de avaliar disponíveis no ambiente;
- c. Implementação de uma função capaz de pesquisar automaticamente caminhos de aprendizagem. O objetivo é facilitar ao usuário a delimitação de temas;
- d. Ampliar a capacidade de inferência, incluindo regras para inferir conhecimento descrito pela combinação de relações semânticas não triviais como *local*, *processo*, *temporal*, entre outras;
- e. Definir um mecanismo de auxílio ao professor para a escolha de frases sinônimas, quando da definição de ontologias. Este mecanismo pode ser: (i) um analisador sintático-semântico, ou (ii) a interligação com dicionários léxicos disponíveis baseado em sinonímia;

## REFERÊNCIAS BIBLIOGRÁFICAS

ANANDHAN, A.; DHANDAPANI, S.; REZA, H., et. al.. (2006) “Web usability testing – CARE methodology”. Proceedings of the Third International Conference on Information Technology: New Generations.

AIRASIAN, P. W; CRUIKSHANK, K. A; MAYER, R. E.; PINTRICH, P. R; RATHS, J. Wittrock, M. C. (2001) “A Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom’s Taxonomy of Educational Objectives”. Addison Wesley Longman, Inc., New York.

AGUIAR, J.; HERMOSILLA, L. (2006) “A Importância da Informática para a Educação”. Revista Científica Eletrônica de Psicologia (RCEP).

ALMEIDA, M. E. B. (2003) “Educação a distância na internet: abordagens e contribuições dos ambientes digitais de aprendizagem”. Educação e Pesquisa, São Paulo, v.29, n.2, p.327-340, jul./dez.

ARAÚJO, A. M. T.; MENEZES, C. S.; CURY, D. (2003) “Apoio automatizado à avaliação da aprendizagem utilizando mapas conceituais”. In Anais do XIV SBIE, p.306-315, Rio de Janeiro, RJ.

ARAÚJO, A. M. T.; MENEZES, C. S.; CURY, D. (2002) “Um ambiente integrado para apoiar a avaliação da aprendizagem baseado em mapas conceituais”. In Anais do XIII SBIE, p.45-59, UNISINOS, São Leopoldo, SBC.

AUSUBEL, D. P. (1962) “A subsumption theory of meaningful verbal learning and retention”. Journal of General Psychology, (66): p.213–224.

AUSUBEL, D. P. (1963) “The Psychology of Meaningful Verbal Learning”. Grune and Stratton, New York, NY.

AUSUBEL, D. P. (1968) “Educational Psychology: A Cognitive View”. Holt, Rinehart and Winston, New York, NY.

AUSUBEL, D. P. (2000) “The Acquisition and Retention of Knowledge”. Kluwer Academic Pub., New York.

AUSUBEL, D. P.; NOVAK, J. D.; HANESIAN, H. (1978) “Educational Psychology: A Cognitive View”. Holt, Rinehart and Winston, 2. Ed.

BARRICO, C. M. C. S. (2007) “Optimização Evolucionária Multi-Objectivo em Ambientes Incertos - Pesquisa de Soluções Robustas”. Tese de doutorado. Disponível em <<http://www.di.ubi.pt/~cbarrico/Doutoramento/Download/TeseDoutoramento.pdf>>. Acesso em 3 de setembro de 2010.

BELLONI, M. L. (2008) “Educação a Distância”. Editora Autores Associados. 5. Ed. São Paulo.

BREMBS, B. C. (1996) "Cheating and Cooperation: potential solutions to the Prisoner's Dilemma". *Oikos* 76: 14 – 24, Copenhagen. Disponível em <<http://www.brembs.net/ipd/ipd.html>>. Acesso em 03 Março 2011.

*BBC News - Internet access is 'a fundamental right'* (em inglês). [news.bbc.co.uk](http://news.bbc.co.uk). Página visitada em 09 de Março de 2011.

CALDAS, V. M.; FAVERO, E. L. (2009) "Uma Ferramenta de Avaliação Automática para Mapas Conceituais como Auxílio ao Ensino em Ambientes de Educação a Distância". XX Simpósio Brasileiro de Informática na Educação.

CRUZ, C.; ARAÚJO, I.; PEREIRA, L.; MARTINS, M. L. (2011) "Uma Abordagem de Avaliação Online no Ensino Superior: e-portfolios em Rede Social". *EDUSER: revista de educação*. Instituto Politécnico de Bragança.

CYBIS, W. A. (2003) "Engenharia de usabilidade: uma abordagem ergonômica". Disponível em <http://www.labiutil.inf.ufsc.br/hiperdocumento/conteudo.html>. Acesso em maio de 2011.

DARWIN, C. (1994) "Origem das Espécies". Tradução do original "*On The Origin of Species: by means of Natural Selection, or the preservation of favoured races in the struggle for life*". Feita por Eugênio Amado. Belo Horizonte: Villa Rica.

FARIAS, F. B.; RODRIGUES, R. C. R.; MORAES, M. A. F. (2009) "Modelo de Representação de *Topic Maps* Dirigido por Ontologia a Partir de uma Extensão do *OpenUp*". CISTI, Portugal.

FARIAS, F. B.; RODRIGUES, R. C. R.; MORAES, M. A. F. (2008) "Uma Extensão da Disciplina de Requisitos do OpenUP/Basic para a Construção de Ontologias Aplicadas à Web Semântica". 1º Seminário de Pesquisa em Ontologia no Brasil, Niterói.

FERLIN, W.; KEMCZINSKI, A.; GASPARINI, I.; CARMO, M.; FREITAS, D. (2008) "Uma ferramenta para avaliação de ambientes *e-learning* quanto aos aspectos da acessibilidade visual". SBIE. Fortaleza. CE.

FERREIRA, R. (2008) "Interatividade Educativa em Meios Digitais: Uma Visão Pedagógica". Tese (Doutorado em Educação). Faculdade de Computação, Universidade Estadual de Campinas - São Paulo.

FERREIRA, K. G. (2002) "Teste de Usabilidade". Monografia de Final de Curso (Especialização em Informática). Universidade Federal de Minas Gerais – Minas Gerais.

FREITAS, F.; SCHULZ, S.; MORAES, E. (2009) "Pesquisa de terminologias e ontologias atuais em biologia e medicina". *RECIIS (Revista Eletrônica de Comunicação Informação & Inovação em Saúde)*. Rio de Janeiro.

GOLDBERG, D. E.; BRIDGES, C. L. (1990) "*An Analysis of Reordering Operator on GAhard Problem*". *Biological Cybernetics*, 62:397-405.

GOLDBARG, M. C.; LUNA, H. P. L. (2000) “Otimização Combinatória e Programação Linear: Modelos e Algoritmos”. Rio de Janeiro: Campus.

GOMES, T. C. (2010) “Gerenciamento de Riscos Utilizando o PMBOK”. Monografia (Graduação em Ciência da Computação). Faculdade de Lourenço Filho - Fortaleza.

GONÇALVEZ, E. “Conhecendo o Comportamento de JavaServer Faces”. Disponível em <<http://www.edsongoncalves.com.br/tag/jsf-2-0/>>. Acesso em 03 de abril de 2011.

GRUNDSPEŃKIS, J.; STRAUTMANE, M. (2009) “*Usage of Graph Patterns for Knowledge Assessment Based on Concept Maps*”. Scientific Journal of RTU. 5. series., Datorzinātne. - 38. vol., pp. 60-71.

GONÇALVEZ, V. B.; CARRAPATOSO, E. M. (2009) “Web Semântica e Cérebro Global juntos por uma boa causa”. EDUSER: Revista de Educação. Instituto Politécnico de Bragança. Volume 1.

GONÇALVEZ, S. M.; OLIVEIRA, I. R. P. (2010) “Mudança na Práxis-Estudo de uma Inovação Pedagógica Apoiada nas Tecnologias de Informação e Comunicação”. EDUSER: revista de educação. Instituto Politécnico de Bragança.

GUARINO, N. (1998) “*Formal ontology and information systems*”. In: Proceeding of the First International Conference on Formal Ontology in Information Systems. Disponível em <<http://www.loa.istc.cnr.it/Papers/FOIS98.pdf>>. Acesso em 21 de janeiro de 2012.

GUARINO, N.; WELTY, C. (2000a). “*A Formal Ontology of Properties*”. LADSEB/CNR Technical Report. Available from <http://www.ladseb.pd.cnr.it/infor/ontology/Papers/OntologyPapers.html>.

GUIZZARD, G.; FALBO, R.; GUIZZARD, R. (2008) “A importância de ontologias de fundamentação para a engenharia de ontologias de domínio: o caso de domínio de processos de software”. IEEE Latino-americano.

GLOGER, B. “*The Zen of Scrum*”. Disponível em <<http://www.glogerconsulting.de>>. Acesso em 22 de outubro de 2009.

SCHWABER, K. (2004) “Agile Project Management with Scrum”. Microsoft.

HOLLAND, J. (1975) “*Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor.

IMPROVE IT. “*Scrum*”. Disponível em <<http://improveit.com.br/scrum>>. Acesso em 13 de abril de 2010.

ISO. (2002) “Requisitos Ergonômicos para Trabalho de Escritórios com Computadores”. NBR 9241-11. ABNT. Rio de Janeiro.

JÚNIOR, J. V. C.; ROCHA, F. E. L. da; FAVERO, E. F. (2004) “*Linking Phrases in Concept Maps: A Study on the Nature of Inclusivity*”. Pamplona, Spain.

KANE, D. (2003) “Finding a Place for Discount Usability Engineering in Agile Development: Throwing Down the Gauntlet”.

LABIUTIL. (1998) “ErgoList”. Disponível em <<http://www.labiutil.inf.ufsc.br/ergolist>>. Acesso em 13 de março de 2012.

LIMA, L.; MEIRINHOS, M. (2011) “A presença social e cognitiva em ambientes de aprendizagem virtual: estudo de caso em alunos do ensino Secundário”. EDUSER: Revista de Educação. Instituto Politécnico de Bragança.

LIMA, P. S. R. (2006) “Um Ambiente Colaborativo de Aprendizagem Interdisciplinar Apoiado por Interfaces Adaptativas”. Universidade Federal do Pará - Belém.

LINDEN, R. (2008) “Algoritmos genéticos: Uma importante ferramenta da inteligência Computacional”. Rio de Janeiro: Editora Brasport.

LOBO, E. L. M. (2005) “Uma solução do problema de horário escolar via algoritmo genético paralelo”. Dissertação (Mestrado). Disponível em: <<http://www.mmc.cefetmg.br/info/downloads/D006-EduardoLuizMirandaLobo2005.pdf>>. Acesso em 16 de fevereiro de 2011.

MACHADO, A. L.; ALMEIDA, F. A.; GERMANO, J. S. E. (2004) “*PHYSI-ASSESSMENT*- Uma Proposta de Ferramenta Automatizada para Avaliação a Distância com o Uso de Expressões Matemáticas”. In Anais do Simpósio Brasileiro de Informática na Educação, pp.132-141, Manaus.

MAFRA, S.; BARCELOS, R.; TRAVASSOS, G. H. (2006) “Aplicando uma Metodologia Baseada em Evidência na Definição de Novas Tecnologias de Software”. In: Anais do XX Simpósio Brasileiro de Engenharia de Software (SBES), v. 1, pp. 239-254.

MANCINELLI, C.; GENTILI, M.; PRIORI, G.; VALITUTTI, G. (2004) “*Concept maps in kindergarten*”. In Proceedings of 1st International Conference on Concept Mapping, Pamplona, Spain.

MARÇAL, A. S. C. (2009) “SCRUMMI: Um processo de gestão ágil baseado no SCRUM e aderente ao CMMI”. Dissertação (Mestrado em Informática). Universidade de Fortaleza.

UMA MENTE BRILHANTE. Direção de Ron Howard. Produção de Brian Grazer: Dreamworks Pictures, Universal Pictures e Imagine Entertainment. (2001). 2 DVD's: Disco 1: Filme, duração aproximada: 135 mim; Disco 2: Bônus.

MITCHELL, M. (1999) “An Introduction to Genetic Algorithms”. MIT Press.

MOURA, R. M. (1997) “O processo de aprendizagem autodirigida em adultos”. Dissertação de Mestrado - Universidade Católica Portuguesa, Faculdade de Ciências Humanas, Lisboa.

MOURA, D. G.; BARBOSA, E. F. (2011) “Trabalhando com Projetos: Planejamento e Gestão de Projetos Educacionais”. 6ª Edição. Editora Vozes. Petrópolis.

- MOREIRA, M. A. (1999) “Teorias da aprendizagem”. São Paulo: EPU.
- MOREIRA, M. A. (2006) “A teoria da aprendizagem significativa e sua implementação em sala de aula”. Brasília: Editora Universidade de Brasília.
- MOODLE. (2010) “*Course Management System*”. Disponível em <<http://moodle.org/>>. Acesso em Abril de 2011.
- MCGUINNESS, D.; FIKES, R.; HENDLER, J.; STEIN, L. (2002) “*DAML + OIL: an ontology language for the Web semantic*”. Journal IEEE Intelligent Systems. Volume 17. EUA, pp. 72-80.
- NIELSEN, J. (1994) “Usability Engineering”. San Francisco. Morgan Kaufman.
- NIELSEN, J.. “*Usability 101: Introduction to Usability*”. ISSN 1448-5552. Disponível em <<http://www.useit.com/alertbox/20030825.html>>. Acesso em 13 de agosto de 2011.
- NIELSEN, J.; MOLICH, R.. (1990) “*Heuristic evaluation of user interfaces*”. Proc. ACM CHI'90 Conf. (Seattle, WA, 1-5 April), pp. 249-256.
- NOVAK, J. D. (2010) “*Learning, Creating, and Using Knowledge: Concept Maps as Facilitative Tools in Schools and Corporations*”. Second Edition. New York, N.Y. Routledge.
- NOVAK, J. D. (1998a) “*Learning, Creating and Using Knowledge: Conceptual Maps as Facilitative Tools in Schools and Corporations*”. Lawrence Erlbaum Associates, Inc., New Jersey.
- NOVAK, J. D. (1998b) “*Teaching Science for Understanding*”. Chapter The Pursuit of a Dream: Education Can Be Improved, pp. 3–28. Academic Press, San Francisco, CA.
- NOVAK, J. D. “*The theory underlying concept maps and how to construct them*”. Disponível em <<http://cmap.coginst.uwf.edu/info/printer.html>>. Acesso em 13 junho 2009.
- NOVAK, J. D.; GOWIN, D. B. (1984) “*Learning how to Learn*”. Cambridge University Press, London.
- NOVAK, J. D.; GOWIN, D. B. (1999) “Aprender a Aprender”. 2. Ed. Plátano, Edições Técnicas, Lisboa.
- NOVAK, J. D.; GOWIN, D. B; JOHANSEN, G. T. (1983) “*The use of concept mapping and knowledge vee mapping with junior high school students*”. Science Education, 67(5): 625–645.
- PIAGET, J. (1977) “O desenvolvimento do pensamento: equilíbrio das estruturas cognitivas”. Dom Quixote, Lisboa.
- PRATES, R. O.; BARBOSA, S. D. J. “Avaliação de interfaces de usuário - conceitos e métodos”. Disponível em: <[http://homepages.dcc.ufmg.br/~rprates/ge\\_vis/cap6\\_vfinal.pdf](http://homepages.dcc.ufmg.br/~rprates/ge_vis/cap6_vfinal.pdf)>. Acesso em 16 de setembro de 2011.

PREECE, J; ROGERS, Y.; Sharp, H.. (2005) “Design de Interação: Além da interação homem-computador”. Tradução Possamai, Bookman. V.. Porto Alegre.

PROJECT MANAGEMENT INSTITUTE (PMI), (2008) “Um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos Quarta Edição (Guia PMBOK)”. Project Management Institute, inc.

RIBEIRO, R. L. O. (2011) “Gerenciando Projetos com PRINCE2”. Editora Brasport.

ROCHA, H. V. da; BARANAUSKAS, M. C. C. (2003) “Design e Avaliação de Interfaces Humano Computador”. São Paulo.

ROCHA, F. E. L. da.; JÚNIOR, J. V. C.; FAVERO, E. L. (2004) “*Linking Phrases in Concept Maps: A Study on the Nature of Inclusivity*”. First Int. Conference on Concept Mapping. Pamplona, Spain.

ROCHA, F. E. L. da.; Júnior, J. V. C.; Favero, E. L. (2005) “*CMTool: Facilitating Meaningful Learning Practice in the Classroom*”. Informática na Educação: teoria & prática. Volume 8 - Número 1. Porto Alegre.

ROCHA, F. E. L. da.; Júnior, J. V. C.; Favero, E. L. (2005) “Como Usar Ontologias na Avaliação da Aprendizagem Significativa Mediada por Mapas Conceituais”. Revista Brasileira de Informática na Educação. Volume 13 – Número 2.

ROCHA, F. E. L. da.; Júnior, J. V. C.; Favero, E. L. (2008) “*Na Approach to Computer-Aided Learning Assessment*”. Third Int. Conference on Concept Mapping. Tallinn, Estonia & Helsinki, Finland.

ROCHA, F. E. L. da.; Vieira, R. V.; Júnior, J. V. C.; Favero, E. L. (2004) “Especificação de um Algoritmo Genético para Auxiliar na Avaliação da Aprendizagem Significativa com Mapas Conceituais”. XV Simpósio Brasileiro de Informática na Educação, SBIE. UFAM.

ROCHA, F. E. L. (2007) “Avaliação da Aprendizagem: Uma abordagem Qualitativa Baseada em Mapas Conceituais, Ontologias de Domínio e Algoritmos Genéticos”. Tese de Doutorado em Engenharia Elétrica. Programa de Pós-Graduação em Engenharia Elétrica. Belém. Universidade Federal do Pará.

RODRIGUES, R. C. R.; ROCHA, F. E. L. da. (2011) “WebCMTTool: Um Ambiente Web para Facilitar a Avaliação da Aprendizagem Baseado em Mapas Conceituais e Ontologias de Domínio”. IADIS - Ibero-Americana. Rio de Janeiro.

RUIZ-PRIMO, M.; SHAVELSON, R. (1996) “Problems and issues in the use of concept maps in science assessment”. *Journal of Research in Science Teaching*, 33(6):569–600.

RUIZ-PRIMO, M. A. (2004) “*Examining concept maps as assessment tool*”. In Cañas, A. J., Novak, J. D., González, F. M., editor, *Proceedings of the First Int. Conference on Concept Mapping*, volume 1, pp.555–562, Pamplona, Spain, September. Universidade Pública de Navarra, Dirección de Publicaciones de la Universidade Pública de Navarra.

RUIZ-PRIMO, M. A.; SCHULTZ, E. S.; SHAVELSON, R. J. (1996) “*Concept map-based assessments in science: An exploratory study*”. Paper presented at the annual meeting of the American Educational Research Association, New York, NY.

RUIZ-PRIMO, M. A.; SCHULTZ, E. S.; LI, M.; SHAVELSON, R. J. (2001) “*Comparison of the reliability and validity of scores from two concept-mapping techniques*”. *Journal of Research in Science Teaching*, 38(2): pp. 260–278.

SILVEIRA, A. M. (2011) “Estudo de Viabilidade de Projetos sob a Ótica dos Skateholders”. Monografia (MBA Gerência de Projetos) - Fundação Getúlio Vargas, Passo Fundo, Rio Grande do Sul.

SIMÕES, R. (2011) “Elaborando o Backlog do Produto”. Disponível em <<http://scrumex.com.br/blog/?p=1091>>. Acessado em 22 de janeiro de 2012.

SOBRINHO, A. M.; MELO, G. M. (2007) “Projeto e Implementação da Máquina de Inferência para o Algoritmo Genético do ambiente CMTool”. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Departamento de Informática, Universidade Federal do Pará, Belém.

SOUZA, L. G.; LEITE, J. C.; SOUZA, C. F.; CASTRO, A. F.; AMARO V. E. (2005) “Uma Abordagem Baseada em Ontologias para Melhorar a Cooperação em Estudos Ambientais em Áreas Produtoras de Petróleo e Gás-Natural”. 3º Congresso Brasileiro de P&D em Petróleo e Gás.

TAVARES, D. B. (2008) “Procedimento de Análise para Validação de Diagrama de Classe de Domínio Baseado em Análise Ontológica”. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Viçosa, Viçosa.

TEXEIRA, O. N. (2005) “Proposta de um Novo Algoritmo Genético baseado na Teoria dos Jogos”. Dissertação (Mestrado em Engenharia Elétrica Aplicada a Computação) – Centro Tecnológico, Universidade Federal do Pará, Belém.

TEIXEIRA, O. N.; BRITO, F. H. de; TEIXEIRA, A. N.; OLIVEIRA, R. C. L. de. (2006) “*Genetic Algorithms with Social Interaction Phase as Phenotype Characterization*”. In: *Advances in Natural Computation and Datamining. ICNC / FSKD – II International Conference on Natural Computation and III International Conference on Fuzzy Systems and Knowledge Discovery*. Xi’an, China.

TEIXEIRA, O. N.; BRITO, F. H. de; TEIXEIRA, A. N.; OLIVEIRA, R. C. L. de. (2006) “*Game Theory as a New Paradigm for Phenotype Characterization of Genetic Algorithms*”. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO’06)*, pp. 1431-1432, Seattle, Washington, USA.

THORNDIKE, E. L. (2010) “Educational Psychology”. Reimpressão. BiblioBazar.



ZABALA, A. (1998) “A Prática Educativa: Como ensinar”. Artmed, Porto Alegre, RS.

WAKE, B. (2003) “*INVEST in Good Stories, and SMART Tasks*”. Disponível em <<http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>>. Acesso em 2 de junho de 2009.

WARD, J. L. (2000) “Project Management Terms: A Working Glossary”. 2. ed. ESI International.

## APÊNDICE A – DETALHES DA IMPLEMENTAÇÃO

O capítulo 3 descreveu de forma detalhada os módulos desenvolvidos no ambiente *WebCMTool*. O projeto bem como as implementações foram baseadas no estado da arte e melhores práticas da Engenharia de Software e da Interação Humano-Computador. Neste anexo são apresentados os requisitos gerais que guiaram os projetos de software e os detalhes de implementações que foram realizadas. Outras características herdadas do *CMTool* podem ser visualizados no Apêndice A de (Rocha, 2007).

**Requisitos Gerais.** Os protótipos desenvolvidos se enquadram numa metodologia geral de desenvolvimento que é sintetizada pela seguinte lista de requisitos:

- Maximizar a simplicidade do ambiente. O propósito do *WebCMTool* é facilitar a avaliação da aprendizagem na web como requisito parcial para melhorar a educação presencial e a distância servindo professores e alunos de todos os níveis de ensino e também ao público em geral no auxílio de outras atividades. Este requisito de simplicidade deverá se aplicar a manutenibilidade do ambiente;
- Disponibilizar a ação para avaliação da aprendizagem significativa pelo *WebCMEditor*: o ambiente deve permitir a avaliação de mapas gerando um relatório conclusivo do processo de avaliação do módulo *GAADT-CM* (descrito na seção 3.2.3 do capítulo 3) que tem como entrada um mapa submetido pelo estudante, uma taxonomia de frases de ligação verifica no Apêndice A em Rocha (2007) e uma ontologia submetida pelo professor;
- Disponibilizar ações para desenho de mapas conceituais: o ambiente deve permitir desenhar e editar mapas, oferecendo metáforas para conceitos e proposições, diferenciação progressiva e reconciliação integrativa e ações como cortar, colar, desfazer, salvar, arrastar, apagar e outras ações de mesma natureza notáveis em editores;
- O ambiente deverá oferecer as condições operacionais para a construção de mapas conceituais livres de contexto e mapas conceituais contextualizados. Mapas livres de contexto são mapas construídos sem qualquer restrição sintática ou semântica do ambiente e mapas contextualizados são aqueles construídos conforme as opções ofertadas pelos conceitos uma ontologia de domínio;
- Disponibilizar a ação para exportar mapas conceituais: o ambiente garante a interoperabilidade de sistemas com outras ferramentas, baseado em modelos comuns de representação de mapas conceituais, permitindo a exportação em arquivos XML dos mapas conceituais criados pelo editor *WebCMEditor*.

- Disponibilizar um relatório após a avaliação no protótipo *WebCMEditor*: o ambiente deverá ser capaz de gerar após cada avaliação um relatório específico sobre suas respectivas entradas analisadas. Este relatório conterá dados sobre o mapa conceitual do estudante, os níveis hierárquicos contidos no mapa do estudante, os mapas conceituais gerados considerados corretos pelo AG com níveis de adaptação mais próximos a ontologia submetida pelo professor, a dimensão semântica entre cada proposição do mapa do estudante com o mapa gerado pelo AG verificando quais proposições estão semanticamente corretas e quais estão incorretas e o resultado desta comparação com uma nota final representativa;
- O ambiente deverá incorporar o protótipo *WebOn\_Tool* para servir como construção de ontologias de domínio na web. Estas servirão para criar contextos que poderão ser associados a tarefas de aprendizagem além de ajudar na avaliação. Em particular, o protótipo deverá implementar a função de cálculo de distâncias semânticas entre proposições e mapas conceituais, além da gramática de inclusividade descrita na seção 5.2 em (Rocha, 2007);
- O ambiente deverá fornecer um módulo administrativo e um módulo repositório de dados. O módulo administrativo (descrito na seção 3.2.6) controlará o acesso ao ambiente (registro e identificação ao ambiente, por exemplo). O administrador poderá atribuir direito de acesso a funcionalidades específicas do ambiente ou ainda poderá atribuir outras permissões de acesso aos usuários. O módulo de repositório (ver seção 3.2.5) deverá armazenar MCs, resultados ou não de avaliações.

**Escolha do Processo de Desenvolvimento.** Durante as etapas de organização e estruturação, notou-se a necessidade de se utilizar um modelo de desenvolvimento ágil de software que fornecesse métodos para se definir um planejamento, os papéis das pessoas envolvidas e a forma de trabalho da equipe. Neste cenário, optou-se por utilizar o *Scrum* (ver seção 3.2) cuja ideia era definir papéis bem específicos para as pessoas envolvidas no projeto e como cada uma deveria trabalhar rumo ao desenvolvimento do software em questão: o *WebCMTool*.

Com base nesta visão, o *Scrum* adaptou-se a esta pesquisa considerando os seguintes pontos:

- No que rege a definição do ambiente, de acordo com o gerenciamento do *Product Owner*, verificou-se: quais são os requisitos principais do ambiente? E qual o interesse do cliente?
- O *Product Owner* verificou, também, quais as principais funcionalidades do ambiente e quais mais importam, definindo assim o que é denominado de *Product Backlog*.
- Após as prioridades previamente esclarecidas e organizadas, foi definido um coordenador para o projeto no papel de *ScrumMaster*. Neste contexto, o *Product Owner* junto com o *ScrumMaster* e a equipe de desenvolvimento definem o que chamamos de *Sprints*.

- Cada *Sprint* possui uma parte de todo o *Product Backlog* e devem ser tratadas de acordo com as prioridades definidas neste. Os *Sprints* devem ser preparados para durarem de 2 a 4 semanas, para que no final de cada período se possa ter um produto ou versão apresentável para o cliente.
- Os *Sprints* foram feitos até o *Product Backlog* acabar e o *Product Owner* definir que o projeto está pronto. Mas vale ressaltar que novas funcionalidades foram incluídas no *Product Backlog* após outras terem sido concluídas após cada *Sprint*.
- O *Product Owner* se preocupava com os interesses do cliente. Este representado pelo papel de professores e alunos da Universidade Federal do Pará.

Complementarmente, o *Scrum* possui três partes principais em seu modelo, conforme apresentado na Figura 3.3: Papéis (*Roles*), Cerimônias (*Cerimonies*) e Artefatos (*Artifacts*). Cada um destes três pilares foi utilizado no que chamamos de ciclo de desenvolvimento, ou seja, um *Sprint*, que contém outros sub-itens mais específicos. Cada *Sprint* possui suas próprias fases e utiliza destes papéis, cerimônias e artefatos para alcançar seu objetivo final.

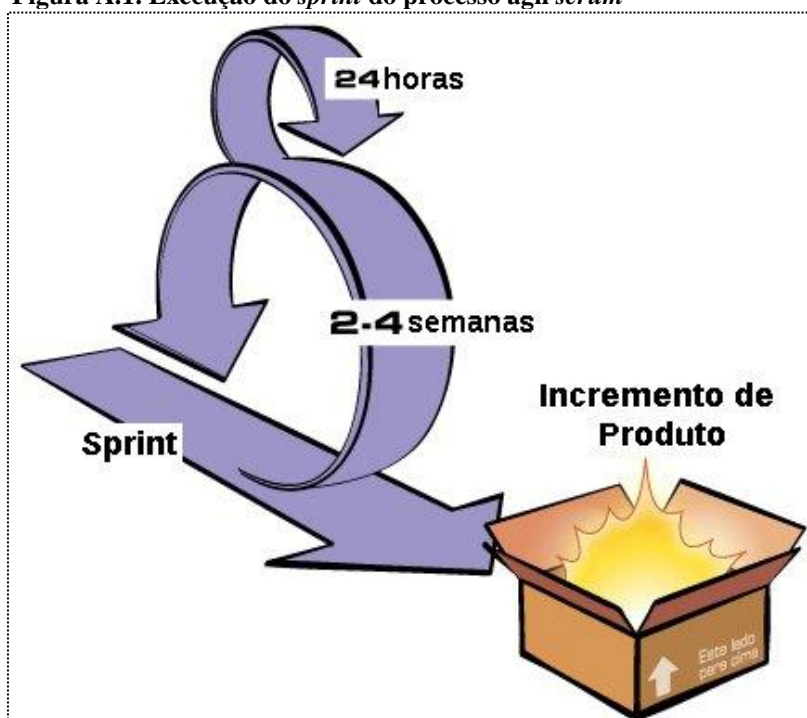
Os principais papéis definidos no projeto foram: i) um *Product Owner*, ii) um *ScrumMaster* e iii) três desenvolvedores no *Time*. O *Product Owner* representa o interesse do cliente, ou seja, a interface entre o cliente e a equipe de desenvolvedores, é exatamente quem sabe qual caminho o projeto seguirá. O *ScrumMaster* é o líder da equipe de desenvolvimento e durante o trabalho, fica mais em contato com o *Product Owner*, é responsável por gerenciar e repassar o trabalho que foi decidido durante o planejamento. A equipe de desenvolvimento é responsável pela construção e implementação do ambiente.

As cerimônias são eventos que acontecem dentro de um *Sprint*. Existem três tipos de cerimônias *Scrum*: a reunião de planejamento do *Sprint*, as reuniões diárias *Scrum* e a reunião de revisão do *Sprint*. Estes três tipos de evento caracterizam bem o ciclo de vida de cada *Sprint*: início, meio e fim.

Os artefatos são as ferramentas básicas para se trabalhar com este modelo de desenvolvimento. Estes artefatos servem como guias e indicadores durante o processo. São divididos em três. De acordo com a Figura 3.3: *Product Backlog*, *Sprint Backlog* e *Burndown Chart*.

Assim, o processo *Scrum* se dá por três etapas, de acordo com a Figura A.1: O início, marcado pela reunião de planejamento, o ciclo de desenvolvimento (*Sprint*) e a conclusão, marcada pela reunião de revisão do *Sprint* até a entrega do incremento do produto.

Figura A.1. Execução do *sprint* do processo ágil *scrum*



Fonte: Adaptado de Improveit, 2010

**Tecnologia Utilizada nos Protótipos.** Todos os protótipos foram desenvolvidos, utilizando a plataforma JSE (*Java Standard Edition*) e JEE (*Java Enterprise Edition*), tendo como fundamento a linguagem de programação Java e seu paradigma de orientação a objetos, além de tecnologias *Open Source*. A linguagem Java foi selecionada em virtude de suas características que se resume a ser: concisa e simples, orientada a objetos, provê acesso à Internet/WWW, robusta, portátil, segura, concorrente, interpretada e compilada, além de sua reusabilidade e manutenibilidade.

**Os Protótipos de Editores do WebCMTool.** Os dois editores incorporados ao ambiente *WebCMTool* são: o *WebOn\_Tool* e o *WebCMEditor*. Ambos foram desenvolvidos em *Applet* para garantir dois objetivos: i) manter a interface do usuário definida em pesquisas realizadas no *CMTool* entre os alunos dos cursos de Bacharelado em Ciência da Computação e Bacharelado em Sistemas de Informação da Universidade Federal do Pará e, ii) promover a eficiência máxima em tempo de execução do AG, diminuindo sua complexidade ao permitir que este seja executada no cliente e não no servidor. Desta forma, este método não sobrecarrega o servidor com as execuções já que são realizadas pelo próprio cliente.

As melhores técnicas de Engenharia de Usabilidade utilizadas para colocar em prática estas ideias são, principalmente, botões e técnicas de arrastar-e-soltar (Silva, 2006). Em ambos os editores em *Applet* foram utilizadas para desenvolvimento a classe *JApplet* da linguagem de programação Java que contém tanto peculiaridades herdada dos *Applets*, quanto características específicas do pacote *swing* com uma metodologia de organização de tarefas próprias da interface do usuário baseada no conceito de ação, como barra de menus, barra de ferramentas e menus flutuantes. As próximas seções

esclarecem detalhes dos editores levando em conta o diagrama de casos de uso geral do ambiente notáveis na Figura A.16.

**O Editor de Mapas Conceituais WebCMEditor:** Este editor está organizado estruturalmente em oitenta e oito classes distribuídas entre quinze pacotes que podem ser visualizados na Quadro A.1.

A Figura A.15 apresenta os pacotes utilizados e a relação entre os mesmos no editor. As classes que compõem estes pacotes, seus respectivos atributos e métodos, estão representadas nas Figuras A.2, A.3, A.4, A.5, A.6, A.7, A.8, A.9, A.10, A.11, A.12, A.13 e A.14. Didaticamente, a seguir, é feita uma breve discussão sobre o conteúdo de cada um desses pacotes.

**Pacote cmeditor:** A Figura A.2 mostra duas classes que compõem este pacote. A classe *Editor* representa o painel onde o *Applet* (por meio de um objeto) será alocado em tempo de execução. Nesta classe, são adicionadas propriedades do editor como barra de ferramentas, barra de menus e dimensões do mesmo. A classe *CMEditor* representa o *JApplet* em execução, que por herança contém o método *init* responsável por carregar os dados do editor em *cache* que posteriormente estaria alocado no painel.

**Pacote cmeditor.desenho:** A classe *AreaDesenho* serve para controlar os eventos da janela gráfica no decorrer de uma sessão de desenho de um MC, conforme apresentado na Figura A.3. Exemplos destes tipos de eventos podem ser controlar o uso dos botões “desfazer”, recolher informações textuais entradas pelo usuário, como o nome de arquivos, atualizar o tamanho da janela e outros de mesma natureza.

**Pacote cmeditor.desenho.elemento:** Na Figura A.4 ressaltam as classes deste pacote, onde foram projetadas para iniciar e finalizar o desenho de todos os elementos constituintes de um MC. São exemplos destes elementos os vértices em geral, que podem ser especializados em “conceito”, “exemplo” e “legenda”, as setas, as frases e os tipos de arco.

**Pacote cmeditor.desenho.metodo:** De acordo com a Figura A.6, nota-se que as classes deste pacote foram projetadas para realizar funções de apoio ao desenho e edição de mapas, como, por exemplo, auxiliar a conectar conceitos, inserir frases de ligação no ponto médio da distância entre dois conceitos, auxiliar na leitura semântica de mapas conceituais, ajudando a distinguir os construtores teóricos existentes nos mesmos, auxiliar no tratamento de strings, e outras funções utilitárias de mesma natureza.

**Pacote cmeditor.desenho.view:** A Figura A.5 apresenta as classes contidas neste pacote, que auxiliam na construção da visualização gráfica do *JApplet* de cada um dos elementos que pode estar contido num mapa conceitual.

**Pacote cmeditor.gui.** Este pacote contém as classes responsáveis pelo gerenciamento de ações envolvendo eventos originados na barra de menus, barra de ferramentas, barras de rolagem e menus *popup*, como mostra a Figura A.8. Exemplo destas ações: abrir (ou salvar) arquivo, consultar as páginas de ajuda e avaliar um determinado mapa conceitual.

Quadro A.1. Pacotes e classes do WebCMEditor

Pacotes	Classes
Cmeditor	CMEditor, Editor
cmeditor.desenho	AreaDesenho
cmeditor.desenho.elemento	ArcoContínuo, ArcoTracejado, Conceito, Exemplo, FraseContextualizada, FraseEnlace, FraseLivre, Legenda, Seta, SetaContínua, SetaTracejada, Vertice
cmeditor.desenho.metodo	AreaTransferencia, Conexoes, CriadorViews, EdicaoElementos, InsercaoElementos, LeituraMapa, TratamentoManuseio, TratamentoStrings
cmeditor.desenho.view	ViewConceito, ViewPortPadrao, ViewSetaContínua, ViewSetaTracejada, ViewVertice
cmeditor.gui	Acoes, BarraFerramentas, BarraMenuDoResultado, BarraMenus, BarraRolagem, MenuPopup
cmeditor.gui.dialogo	Dialogos, DialogoSobre, DialogoTaxonomia, PainelCor, PainelElementoContextualizado, PainelEstiloLinha, PainelFonte, PainelPreferencias, SeletorArquivo
cmeditor.io	ArvoreTaxonomia, EscritorXML, FiltroArquivo, LeitorXML, ManipuladorRecentes, ManipuladorXML, Mapa, ReadWriteFile, Supertipo
cmeditor.gaadt	Ambiente, AprendizagemRelacao, Cromossomo, GAADT, Gene, Inferencias, MapaAprendiz, Populacao, Relatorio
cmeditor.gaadt.gerador	GeradorMP
cmeditor.gaadt.ontologia	Cadeia, Conceito, Frase, Ontologia, Relacao, Taxonomia
cmeditor.gaadt.util	ConjuntoAciclico, Gaussiana, InsereCromossomosThread, Medidor, Ordenador, Util
cmeditor.gaadt.xmltools	Token, TokenDeArco, TokenDeConceito, TokenDeFrase, TokenValorRelacao, TradutorMapaAprendizXML, TradutorOntologiaXML, TradutorTaxonomiaXML, VectorToken, XMLParser
cmeditor.on_tool.io	LeitorXML2, ManipuladorXML2

**Pacote cmeditor.gui.dialogo:** A Figura A.9 apresenta as classes deste pacote, que são responsáveis pelo gerenciamento dos diálogos relacionados às funções existentes na barra de menus. Exemplos deste tipo de diálogo: diálogo para inserção de elemento, diálogo para a escolha da cor de contorno ou preenchimento de um elemento pré-existente num MC.

**Pacote *cmeditor.io*:** A Figura A.7 incorpora as classes deste pacote, que são responsáveis por: (i) transformar a representação XML de um MC na representação gráfica correspondente; (ii) transformar a representação XML de um MC em objetos Java e vice-versa; (iii) ler e gravar em mídia não volátil a representação XML de mapas conceituais. Vale ressaltar, também, que essas classes são auxiliadas pelas classes *ManipuladorXML2* e *LeitorXML2* presente ambas no pacote *cmeditor.on\_tool.io* apresentado pela Figura A.14.

**Pacote *cmeditor.gaadt*:** Este pacote contém as classes responsáveis por criar uma instância de execução para aplicação do GAADT-CM levando em conta atributos como quantidade de mapas conceituais (indivíduos) presentes em uma população, probabilidade de mutação destes mapas conceituais, técnicas de estado estacionário, número correlato de gerações, número de corridas, tamanho da população e fitness dos mapas, além das entradas do GAADT: a ontologia, o mapa conceitual e a taxonomia de frases de ligação. Este pacote também contém a máquina de inferências responsável por inferir genes baseado nas regras de proposições em lógicas matemáticas, ou seja, aumentando a possibilidade de um mapa conceitual do estudante ser mais próximo da ontologia e conceitos definidos pelas relações binárias desta ontologia em questão.

Este representa o mais importante pacote do algoritmo genético mostrado na Figura A.10. Ele possui as classes que colocam em prática a evolução dos cromossomos, bem como funções fundamentais para todo AG, como cruzamento e seleção.

A classe mais básica no pacote *GAADT* é a classe *AprendizagemRelacao*. Esta classe representa os tipos possíveis de aprendizagem que podem ser representados em um mapa conceitual, ou seja, diferenciação progressiva e reconciliação integrativa. A classe *Gene* representa as proposições que podem formar MCs. Estas proposições são formadas por conceitos origem e destino, uma frase de ligação, e o tipo de aprendizagem que se deseja representar. O método mais importante desta classe é o método *getAdaptacao()*, que retoma o grau de adaptação do gene.

A classe *Cromossomo* representa um mapa conceitual que, por sua vez, é um agrupamento de genes (proposições). Por isso, existe um relacionamento entre a classe *Cromossomo* e a classe *Gene*. Na classe *Cromossomo*, o método mais importante é o método *getAdaptacao()*, que retoma a adaptação do cromossomo (calculada a partir dos graus de adaptação de cada um de seus genes constituintes). A classe *Populacao* é um conjunto de cromossomos e representa as populações de indivíduos que são geradas pelo algoritmo genético. Alguns dos métodos desta classe incluem: *getMediaAdaptacao()*, que retoma a adaptação média da população; *getNr\_cromossomos()*, que retoma o tamanho da população; e *getMelhor()*, que retoma o cromossomo com maior adaptação da população.

*Ambiente* é a classe que possui os métodos fundamentais ao GAADT: *geraGenesAFG()*, que gera o alfabeto do GAADT; *geraCromossomoAFC()*, que gera a população inicial; *verificaCiclo()* e *verificaDesconexao()*, que indicam quando há mapas conceituais construídos incorretamente;



*SelecionaCromossomo()*, *Fecundação()*, *Cruzamento()*, *Mutacao()*, entre outros, que permitem a criação de novas populações a partir de uma população inicial.

A classe *GAADT*, por fim, é a responsável por chamar cada um dos métodos da classe *Ambiente* em seu devido momento, através do método *aplicaGAADT()*. Adicionalmente, é neste método que são definidos alguns parâmetros importantes do AG, como tamanho das populações, probabilidade de mutação, e assim por diante.

**Pacote *cmeditor.gaadt.gerador*.** Este pacote contém uma classe denominada *GeradorMP* que recebe os insumos do AG por meio do método *avaliar* definido na classe *Acoes* no pacote *cmeditor.gui*. Com estes insumos como entrada carregados, são inicializadas as análises léxicas e sintáticas dos arquivos em XML por um parser que posteriormente estes dados analisados são enviados para a classe *GAADT* do pacote *cmeditor.gaadt* para começar o início da execução do algoritmo. É nesta classe que dados como o número de corridas e o tamanho da população são previamente definidos.

**Pacote *cmeditor.gaadt.ontologia*.** Neste pacote a classe *Conceito* modela um conceito existente numa ontologia de domínio criada pelo professor, ou seja, para cada conceito inserido pelo professor na ontologia, um objeto da classe *Conceito* será criado. A classe *Frase* modela as diferentes frases de ligação que podem existir conectando dois conceitos em uma proposição. Objetos da classe *Frase* são criados tanto para as frases categorizadas na taxonomia, quanto para as frases selecionadas pelo professor na ontologia de domínio.

A Figura A.11 ilustra o diagrama de classes. Este pacote condensa as classes necessárias para o tratamento, na forma de objetos, tanto das ontologias de domínio, quanto da taxonomia de frases de ligação. Escolheu-se abordar estes dois aspectos do sistema em um único pacote em virtude das semelhanças que existem entre eles.

A classe *Relação* modela as relações entre conceitos estabelecidas pelo professor em sua ontologia de domínio. Essa classe possui dois objetos da classe *Conceito* (para modelar os conceitos origem e destino), e um objeto da classe *Cadeia*. Por fim, as classes *Ontologia* e *Taxonomia* modelam, respectivamente, as ontologias de domínio e a taxonomia de frases de ligação.

**Pacote *cmeditor.gaadt.util*.** Este pacote é formado pelas classes *Util*, *Ordenador* e *Gaussiana*, todas classes auxiliares conforme mostra a Figura A.13. A *Util* serve para simular probabilidades e geração de números aleatórias, a *Ordenador* implementa um método sofisticado de ordenação de números e a *Gaussiana* calcula valores de probabilidade aplicando a regra da distribuição normal.

**Pacote *cmeditor.gaadt.xmltools*:** Na Figura A.12, observa-se que todas as classes descendem de uma classe abstrata (*XMLParser*), que implementa um método (*parser()*) capaz de abrir um arquivo XML e “quebrá-lo” em seus elementos constituintes. As classes *TradutorMapaAprendizXML*, *TradutorOntologiaXML* e *TradutorTaxonomiaXML* são responsáveis, respectivamente, por fazer a análise dos três arquivos XML que sevem de entrada ao AG: o arquivo com o mapa do estudante, o arquivo com a ontologia e o arquivo com a taxonomia do ambiente. Nestas classes, métodos *get* são

responsáveis por identificar *tags* específicas dentro dos arquivos XML. Eles retornam vetores de *strings* com os elementos XML solicitados. Assim sendo, por exemplo, o método *getFrases()* da classe *TradutorTaxonomiaXML* é responsável por retornar um vetor de strings com as frases de ligação constantes da taxonomia.

**O Editor de Ontologias de Domínio WebOn\_Tool:** O projeto de software do *WebOn\_Tool* possui características de implementação semelhantes àsquelas do *WebCMEditor* descritas nas Figura de A.2 a A.9, com a diferença de que não há um módulo de avaliação presente no mesmo. Assim, ressalta-se que por questões de redundância estas classes serão omitidas, sendo considerado apenas que este editor está organizado estruturalmente em quarenta e quatro classes distribuídas entre oito pacotes. Estes pacotes e respectivas classes estão listados no Quadro A.2.

**Quadro A.2. Pacotes e classes do WebOn\_Tool**

Pacotes	Classes
on_tool	On_Tool, Editor
on_tool.desenho	AreaDesenho
on_tool.desenho.elemento	Vertice, Conceito, Legenda, Seta, SetaDireita, SetaEsquerda, RelacaoBinaria, Arco
on_tool.desenho.metodo	Conexoes, TratamentoStrings, TratamentoManuseio, InsercaoElementos, AreaTransferencia, EdicaoElementos, CriadorViews
on_tool.desenho.view	ViewSetaDireita, ViewVertice, ViewConceito, ViewPortPadrao, ViewSetaEsquerda
on_tool.gui	BarraMenus, Acoes, BarraFerramentas, BarrasRolagem, MenuPopup
on_tool.gui.dialogo	PainelPreferencias, PainelCor, PainelFonte, Dialogos, DialogoTaxonomia, DialogoSobre, SeletorArquivo, PainelEstiloLinha
on_tool.io	ManipuladorXML, Ontologia, ManipuladorRecentes, Supertipo, LeitorXML, EscritorXML, FiltroArquivo, ArvoreTaxonomia

**Pacotes com Recursos Utilizados pelos Editores:** Para finalizar, o Quadro A.3 frisa os pacotes organizados pelos editores de mapas conceituais e ontologias de domínio que foram utilizados durante a implementação de recursos como imagens, páginas, botões e outros de mesma natureza.

**Quadro A.3. Pacotes com recursos utilitários dos editores**

Pacotes	Classes
cmeditor.ajuda e on_tool.ajuda	bem_vindo.html, desfazer_refazer.html, exportar.html, formatacao.html, imprimir.html, index.html, mcs_contextualizados, mcs_livres_contexto, menu.html, preferencias.html, salvar_abrir.html, seleção_area_transferencia.html, zoom.html
cmeditor.ajuda.imagens	Imagens utilizadas pelo ajuda do <i>WebCMEditor</i>
on_tool.ajuda.imagens	Imagens utilizadas pelo ajuda do <i>WebOn_Tool</i>
on_tool.imagens	Imagens utilizadas pelo <i>Applet WebOn_Tool</i>
cmeditor.imagens	Imagens utilizadas pelo <i>Applet WebCMEditor</i>

Figura A.2. Classes do pacote cmeditor

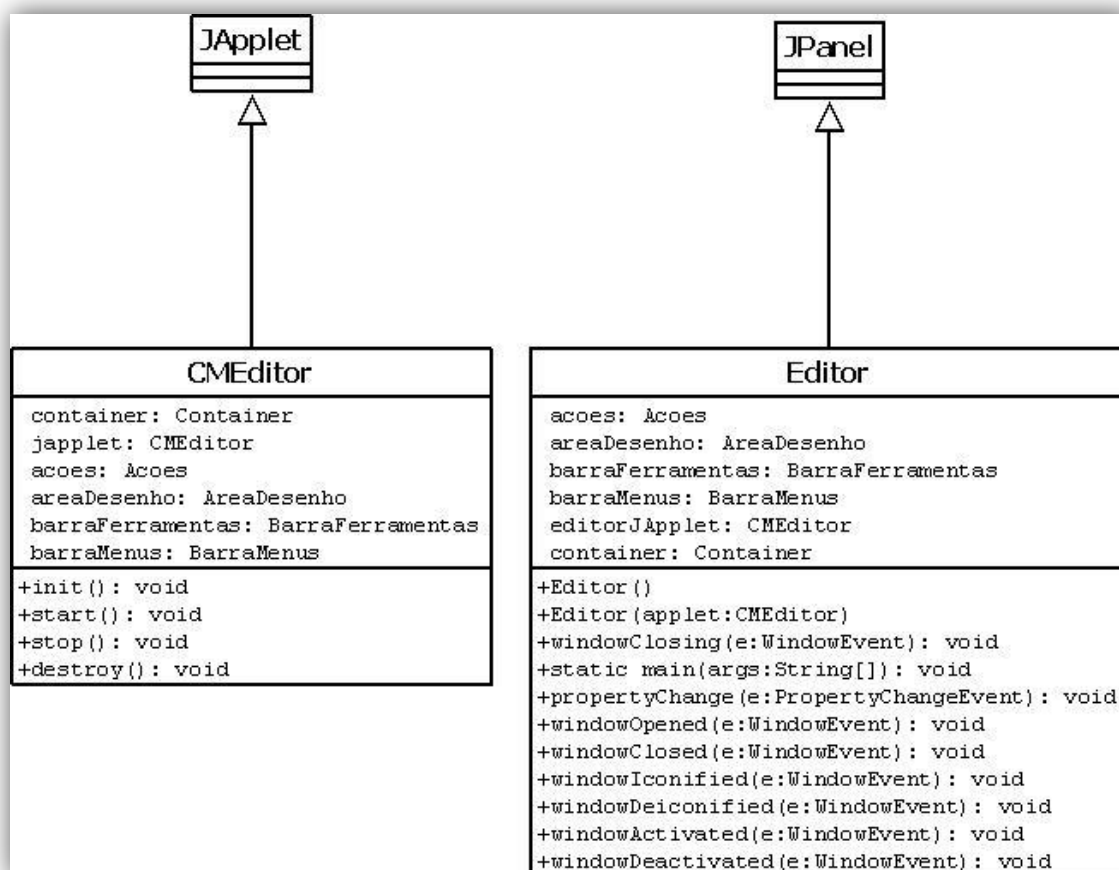


Figura A.3. Classes do pacote cmeditor.desenho

AreaDesenho
<pre> -acoas: Acoes #areaTransferencia: Vector -barrasRolagem: BarrasRolagem -tipoMap: int -graph: JGraph -nomeArquivo: JTextField -caminhoArquivo: String #conceitosOntologia: Vector #corLinha: Color #corBorda: Color #corPreenchimento: Color #corLetra: Color #fonte: Font #estiloLinha: int #gerenciaDesfazer: GraphUndoManager  +AreaDesenho() +atualizaBotoesDesfazer(): void +testaConexaoJaExistente(origem:Port,alvo:Port): boolean +mudaAcoes(acoas:Acoes): void +pegaAcoes(): Acoes +pegaBarrasRolagem(): BarrasRolagem +mudaNomeArquivo(s:String): void +pegaNomeArquivo(): String +mudaCaminhoArquivo(s:String): void +pegaCaminhoArquivo(): String +mudaTipoMapa(i:int): void +pegaTipoMapa(): int +botaConceitoOntologia(s:String): void +tiraConceitoOntologia(s:String): void +pegaConceitosOntologia(): Vector +limpaConceitosOntologia(): void +limpaAreaTransferencia(): void +mudaAreaTransferencia(v:Vector): void +pegaAreaTransferencia(): Vector +updateAutoSize(view:CellView): void +valueChanged(e:GraphSelectionEvent): void +getToolTipText(e:MouseEvent): String </pre>

Figura A.4. Classes do pacote cmeditor.desenho.elemento

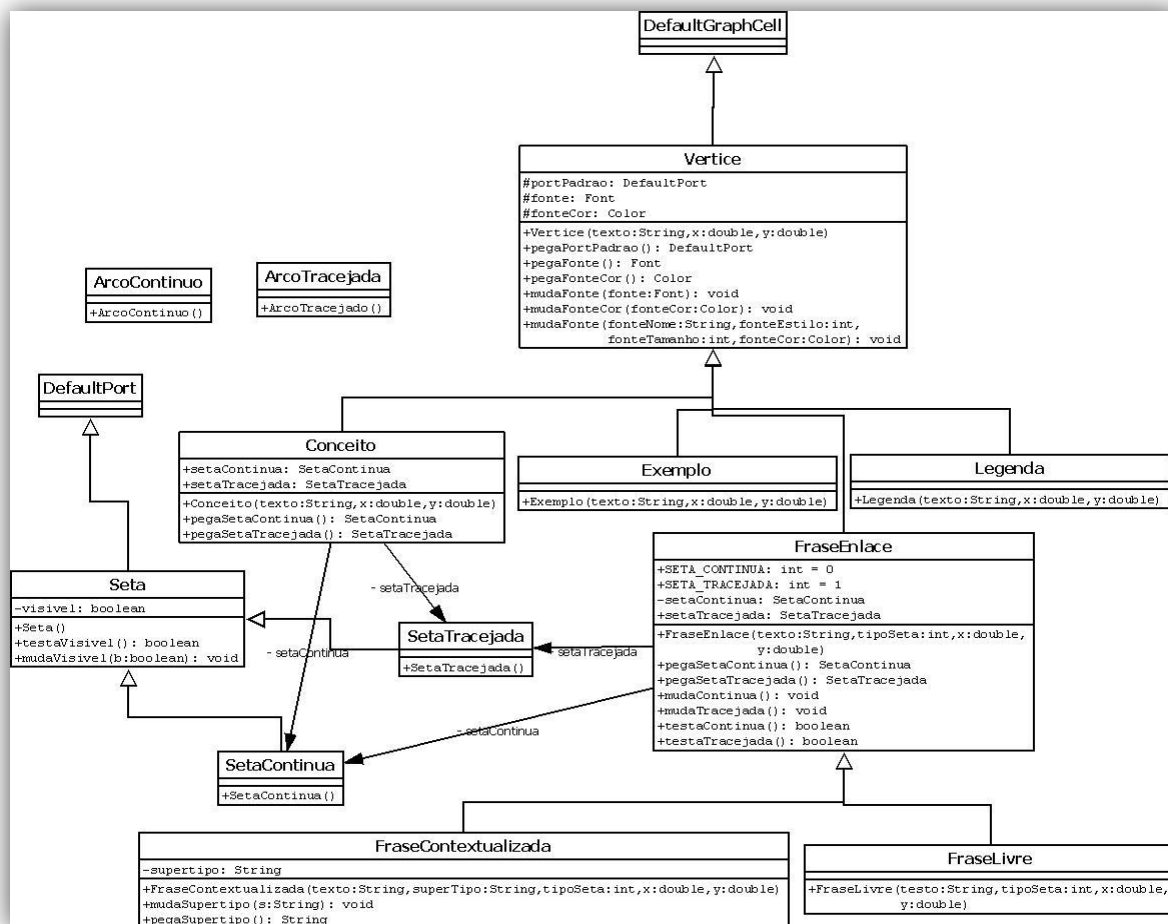


Figura A.5. Classes do pacote cmeditor.desenho.view

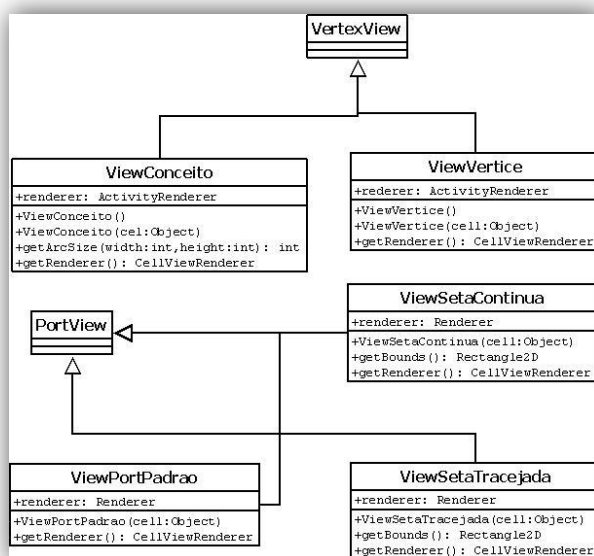


Figura A.6. Classes do pacote cmeditor.desenho.metodo

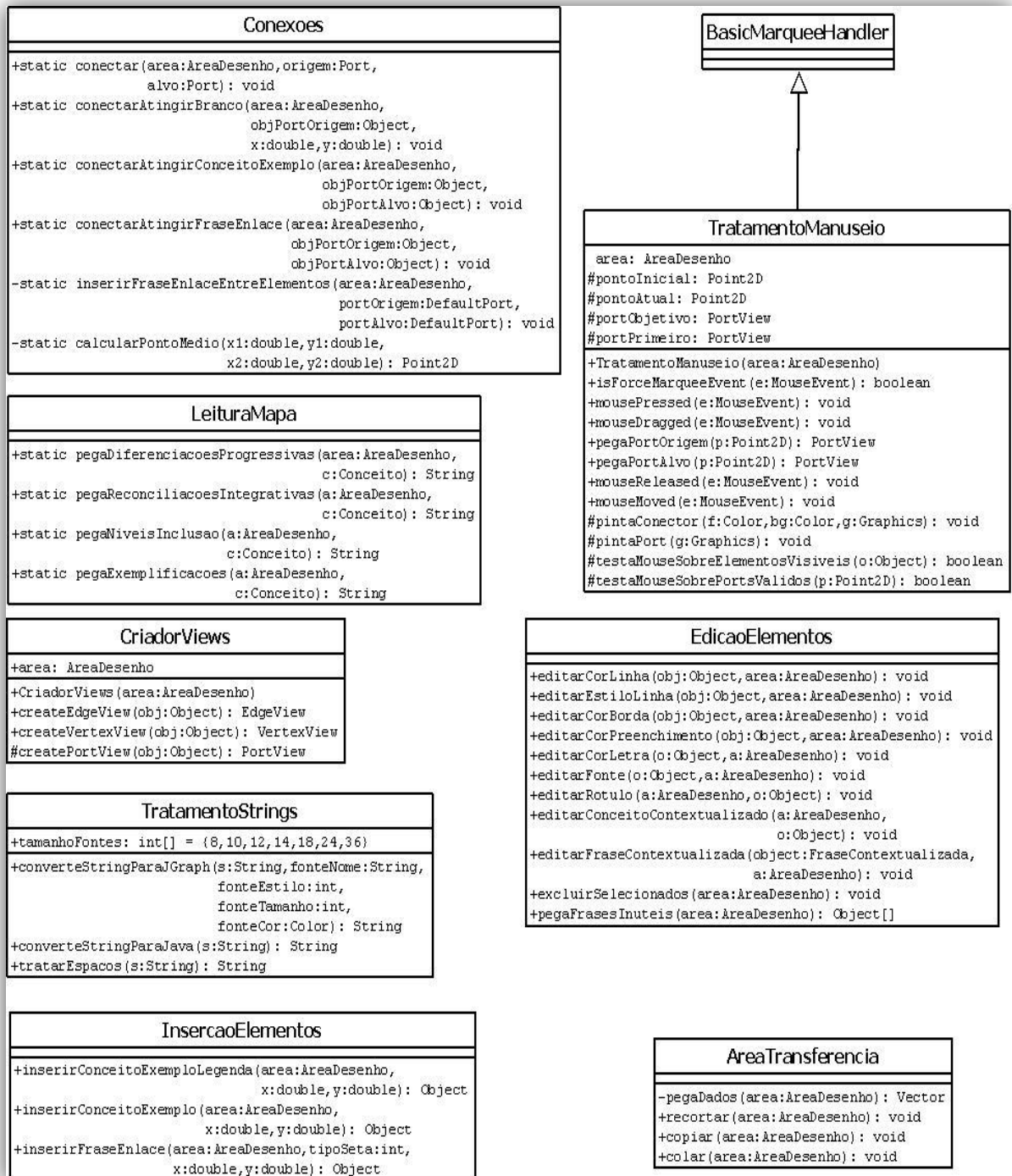


Figura A.7. Classes do pacote cmeditor.io

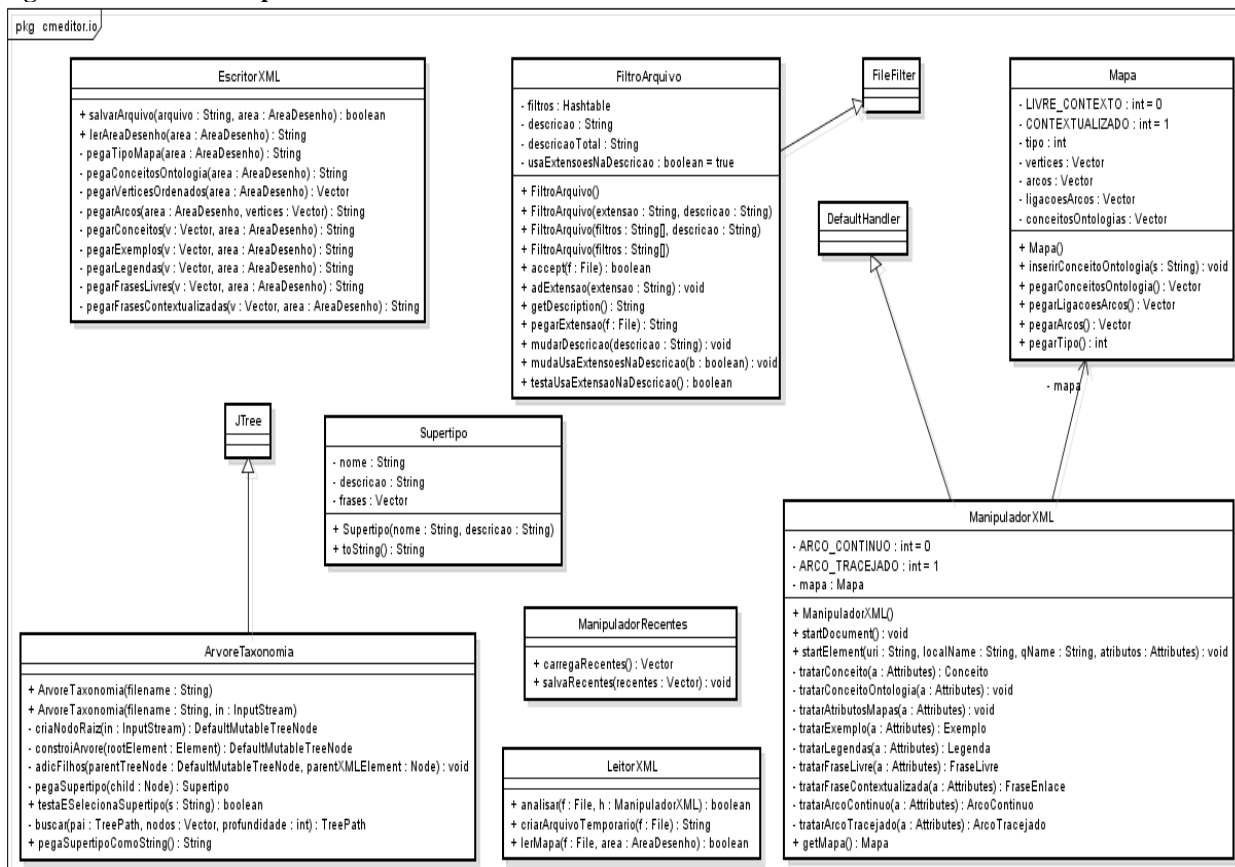


Figura A.8. Classes do pacote cmeditor.gui

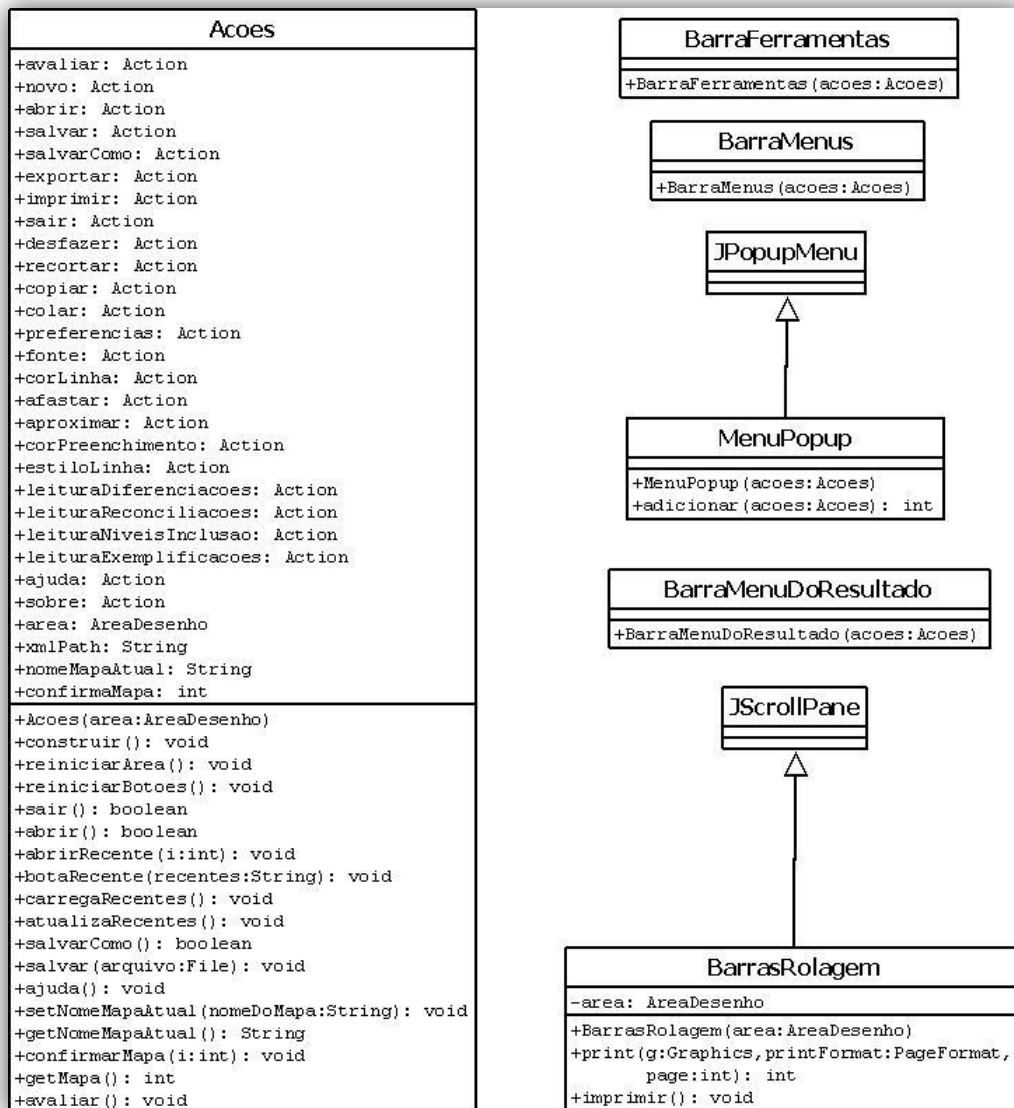




Figura A.9. Classes do pacote cmeditor.gui.dialogo

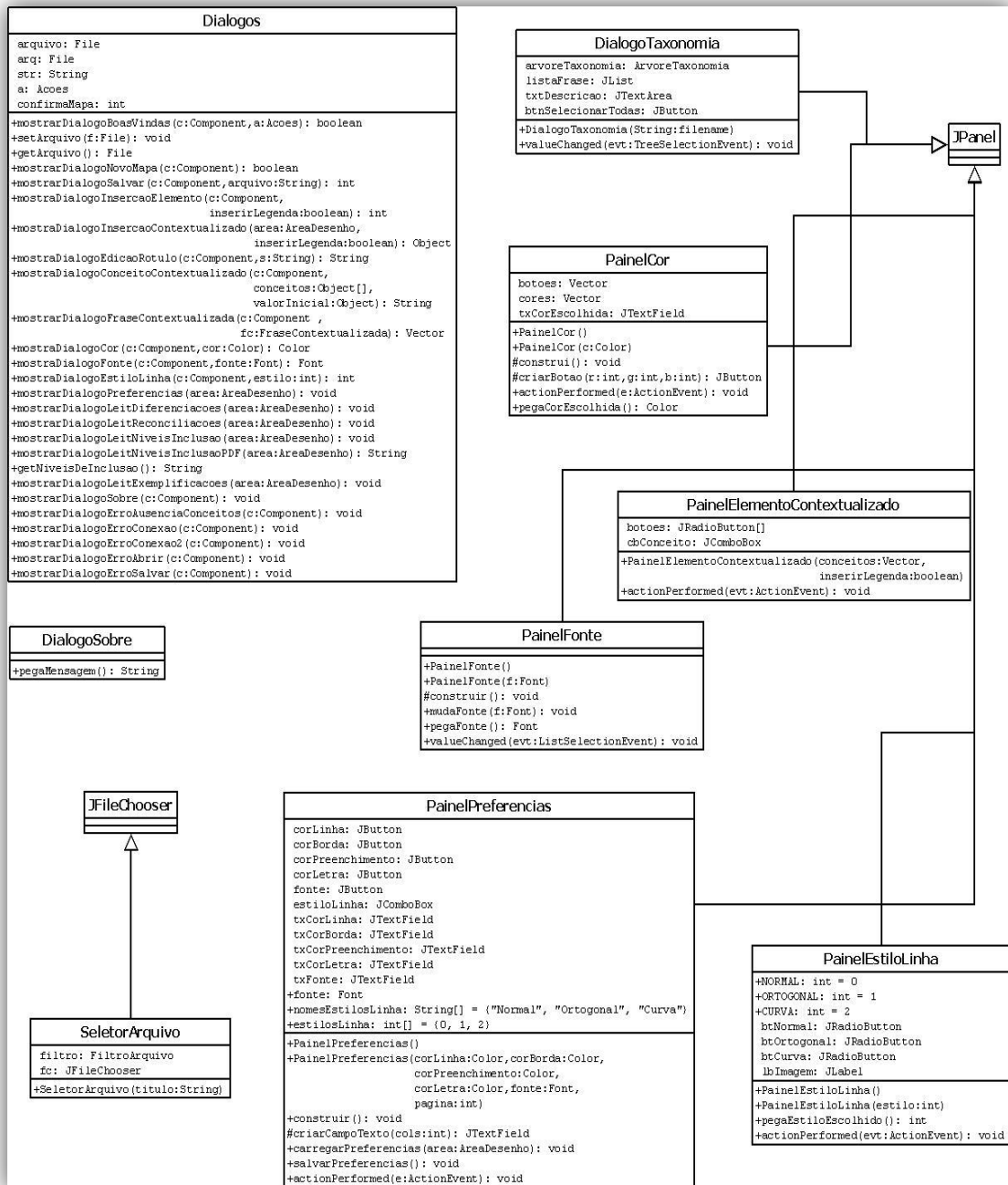


Figura A.10. Classes do pacote cmeditor.gaadt

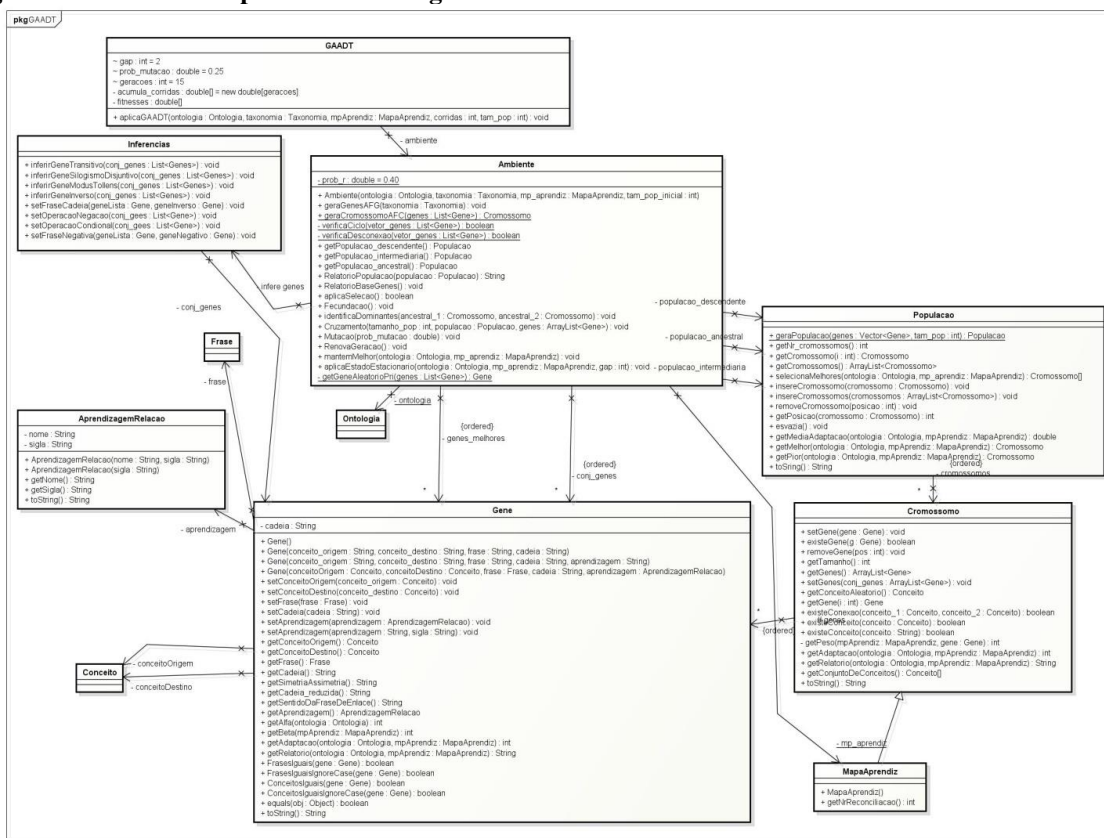


Figura A.11. Classes do pacote cmeditor.gaadt.ontologia

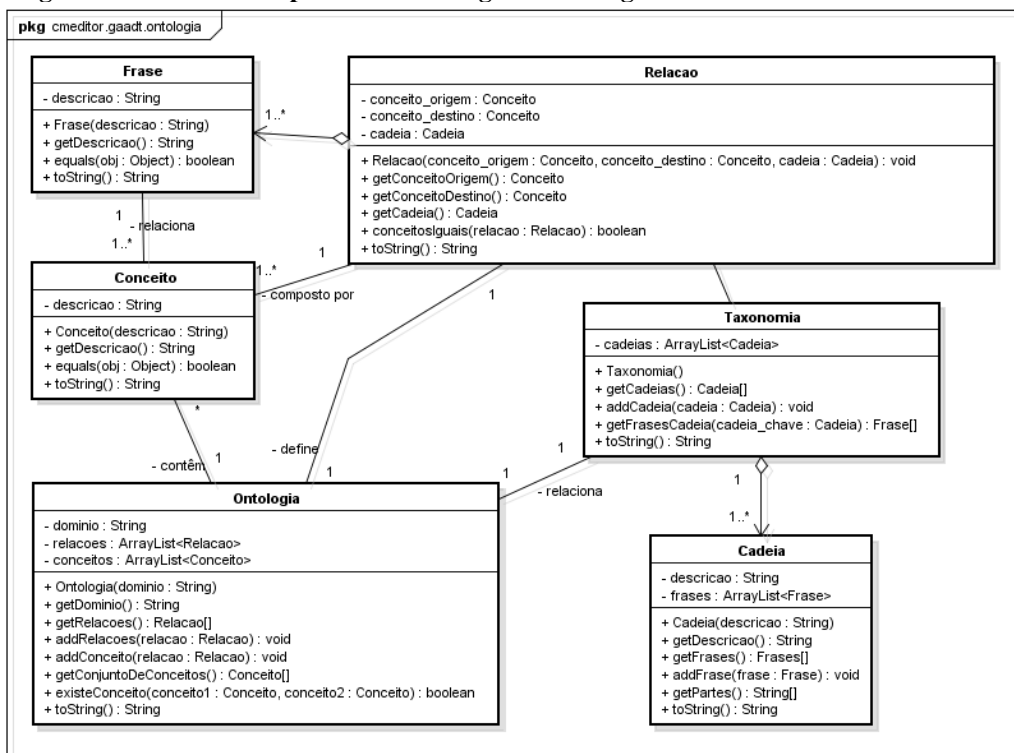


Figura A.12. Classes do pacote cmeditor.gaadt.xmltools

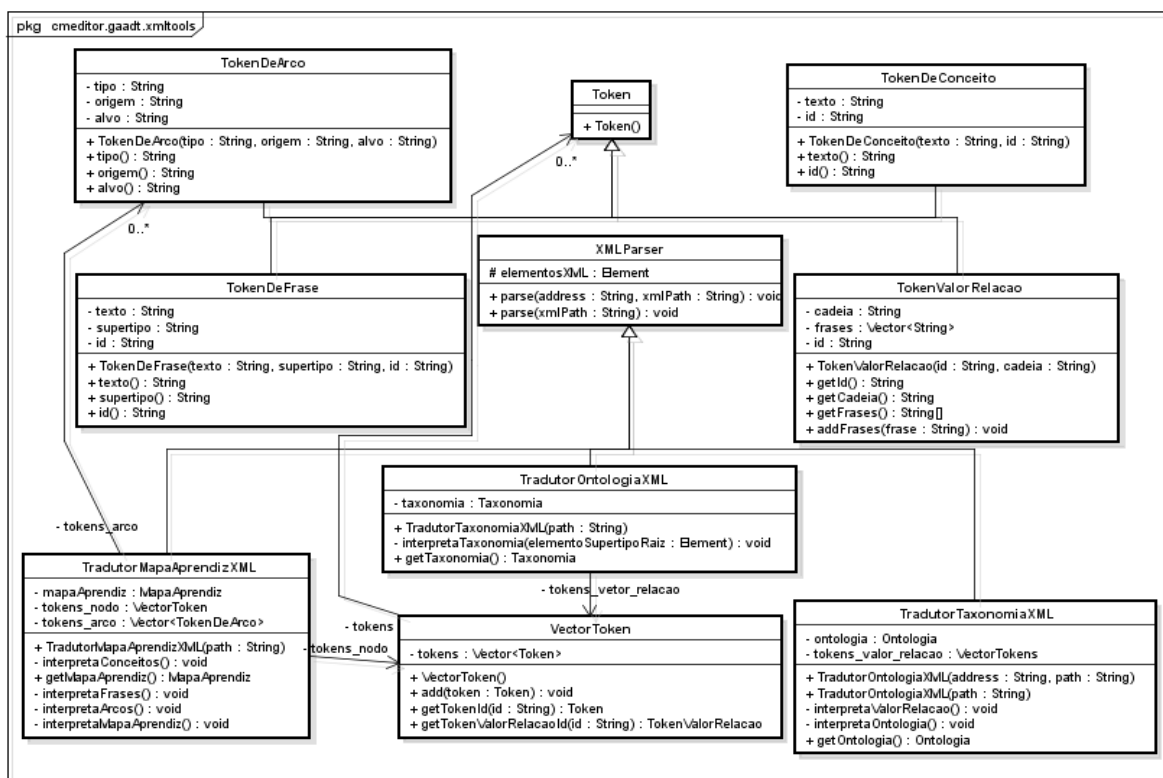


Figura A.13. Classes utilitárias do pacote gaadt.util

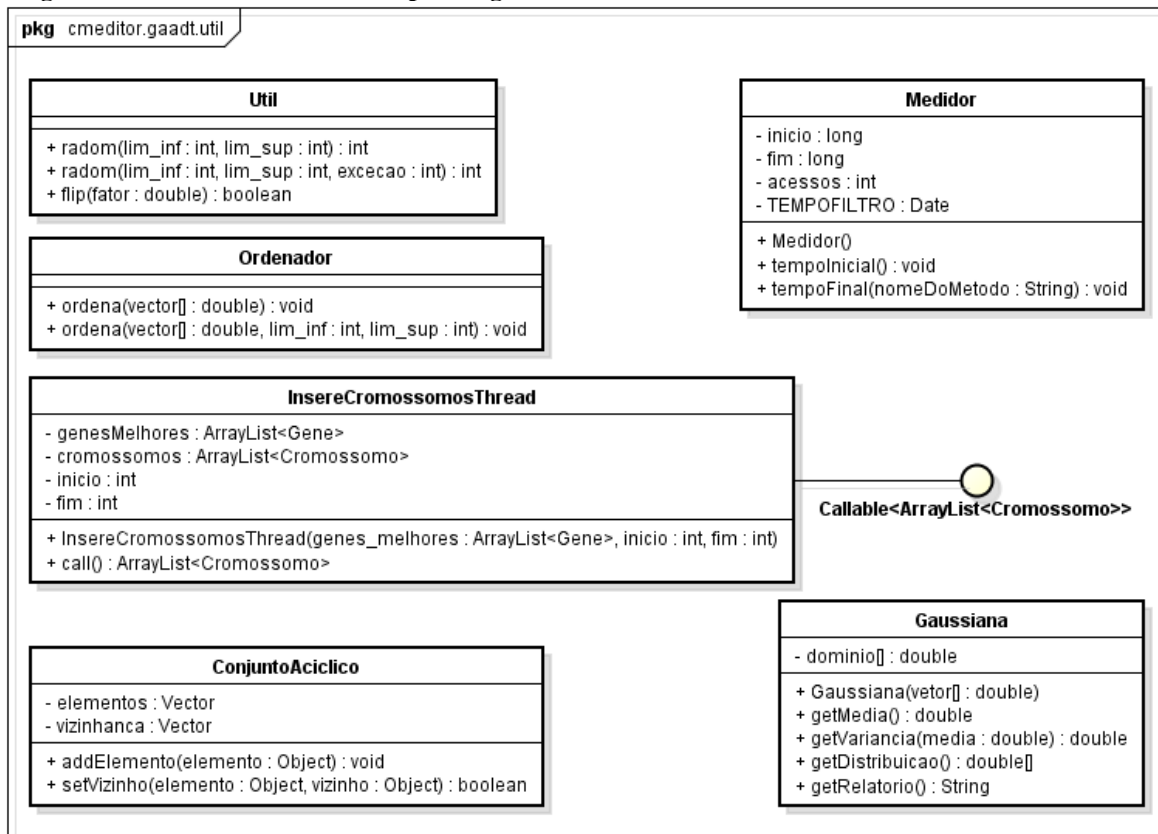


Figura A.14. Classes do pacote cmeditor.on\_tool.io

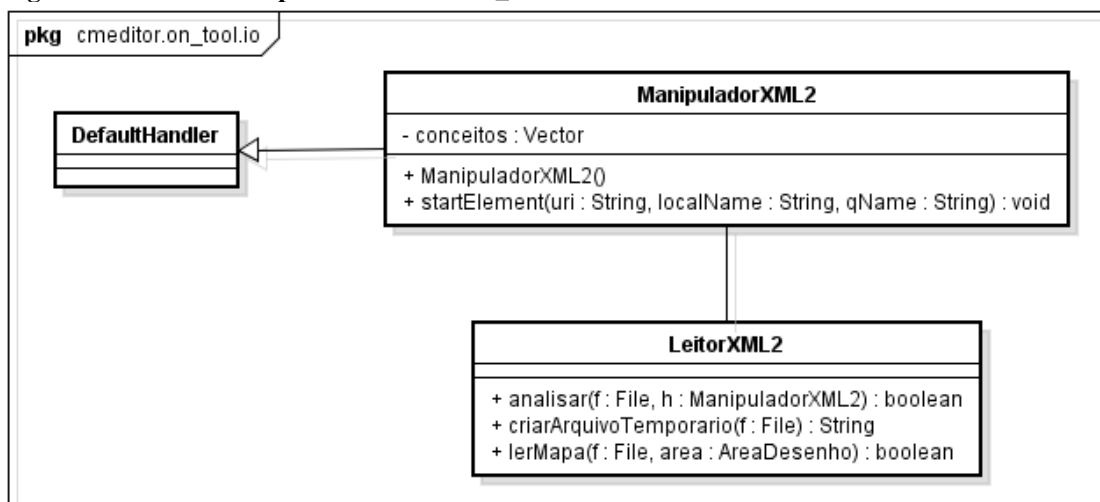


Figura A.15. Pacotes do webcmeditor

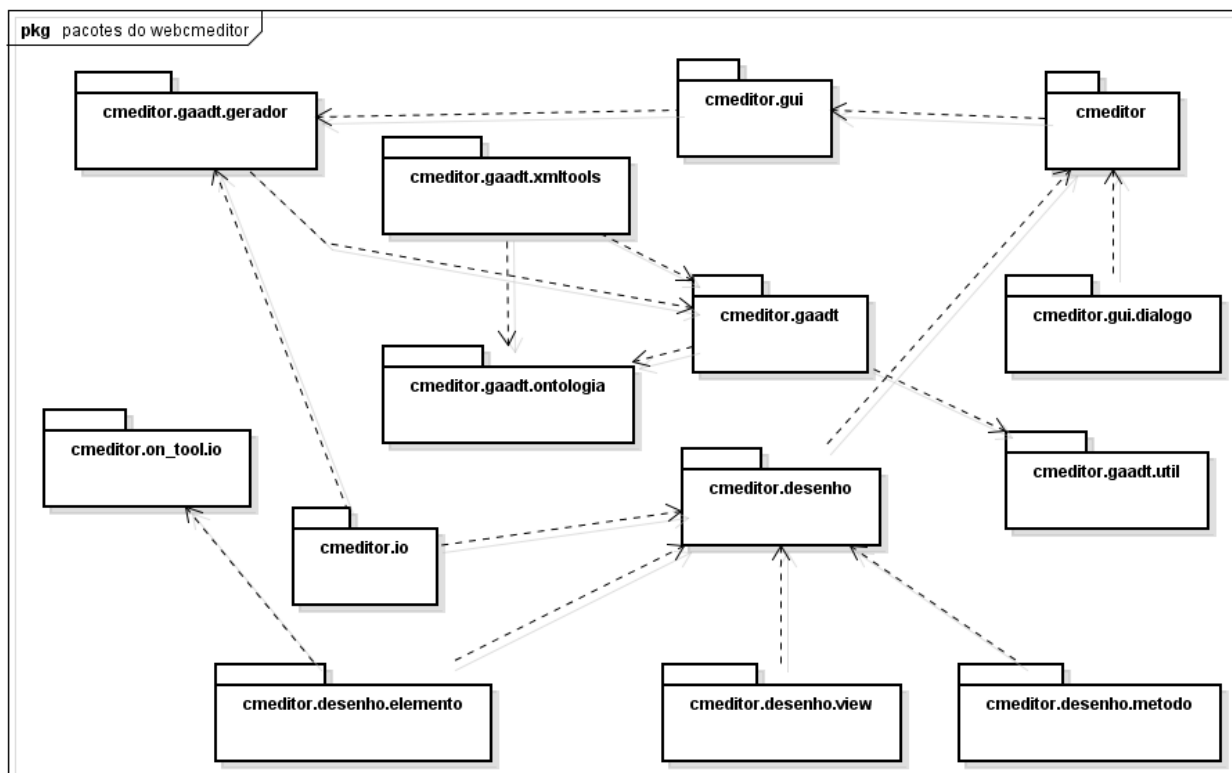
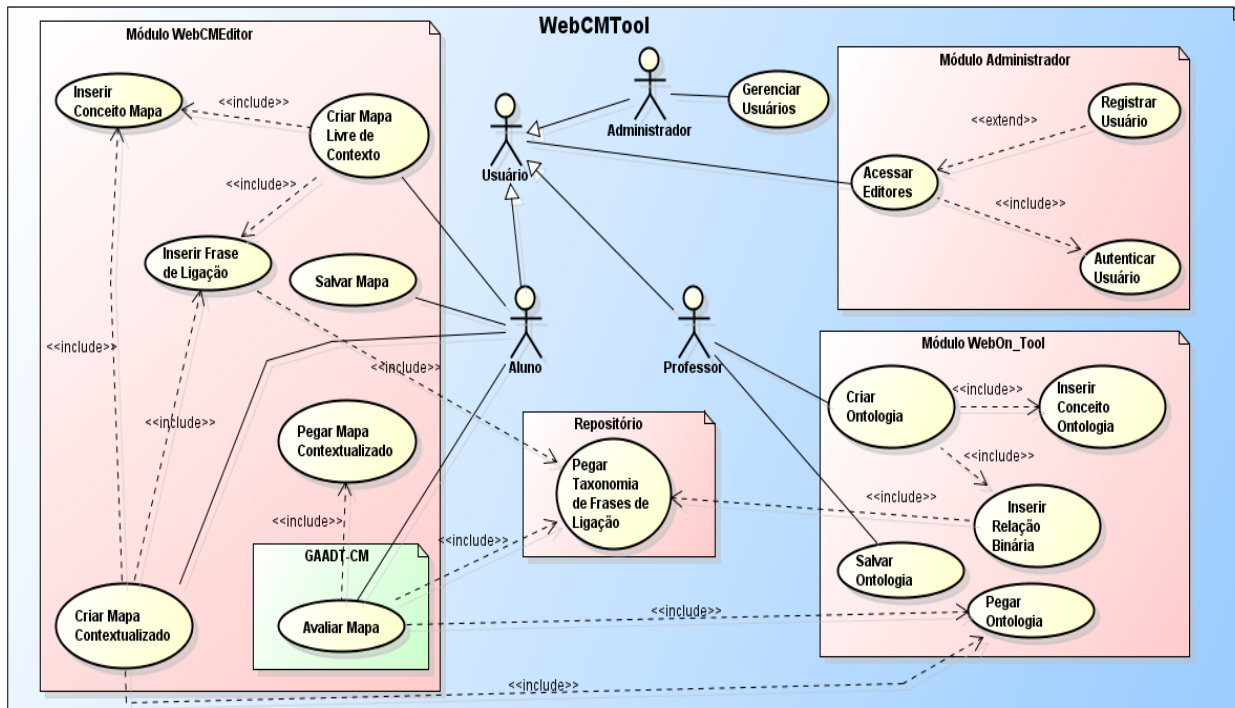


Figura A.16. Diagramas de casos de uso geral webcmttool



## APÊNDICE B – RELATÓRIO DA AVALIAÇÃO DO MÓDULO AVALIADOR DO WEBCMTOOL

Este Apêndice contém uma descrição do relatório no módulo avaliador que é um dos elementos fundamentais do *WebCMTool*, descrito na seção 3.2.4. O relatório é impresso em um arquivo no formato PDF e se destaca por possuir detalhes sobre a avaliação, como: 1) os conceitos, a cadeia semântica e a frase de ligação utilizada pelo MCEst; 2) os níveis de hierarquia dos conceitos utilizados no MCEst; 3) a relação semântica das proposições avaliadas por notas de 0 a 2 onde: a nota 0 refere-se a proposição do MCEst formada por um par de conceitos que não possuem nenhum tipo de relação semântica com as proposição dos MCs gerados pela a avaliação do *GAADT-CM*, frisando que aquela proposição formada pelo aluno possui uma má formação, sendo considerada totalmente errada; a nota 1 refere-se a proposição do MCEst formada por um par de conceitos que contém uma proximidade semântica mas que a frase de ligação utilizada está errada, mostrando que o aluno apesar da frase está errada, acertou quando tentou criar uma proposição com os dois conceitos escolhidos; e, a nota 2, que refere-se a proposição do MCEst formada por um par de conceitos que contém não só uma proximidade semântica entre os conceitos escolhidos, mas também a mesma cadeia semântica e frase de ligação utilizada por alguma das proposições dos MCs formados pelo *GAADT-CM*, indicando com isto que a proposição está correta e que o aluno acertou a formação da proposição e que entende o domínio de aprendizado; 3) notas em percentual com: i) todas as proposições feitas pelo MCEst em relação à ontologia de domínio submetida para avaliação; ii) quantidade de proposições do MCEst que foram considerados corretas levando em conta o percentual das proposições construídas no MCEst; iii) quantidade de proposições do MCEst que foram considerados incorretas levando em conta o percentual das proposições construídas no MCEst. As Figuras B1 a B5 apresentam todos os dados supracitados, contendo um como exemplo um MCEst avaliado durante a aplicação.

Figura B.1. Relatório inicial do mapa do aluno com todos os dados do MC criado

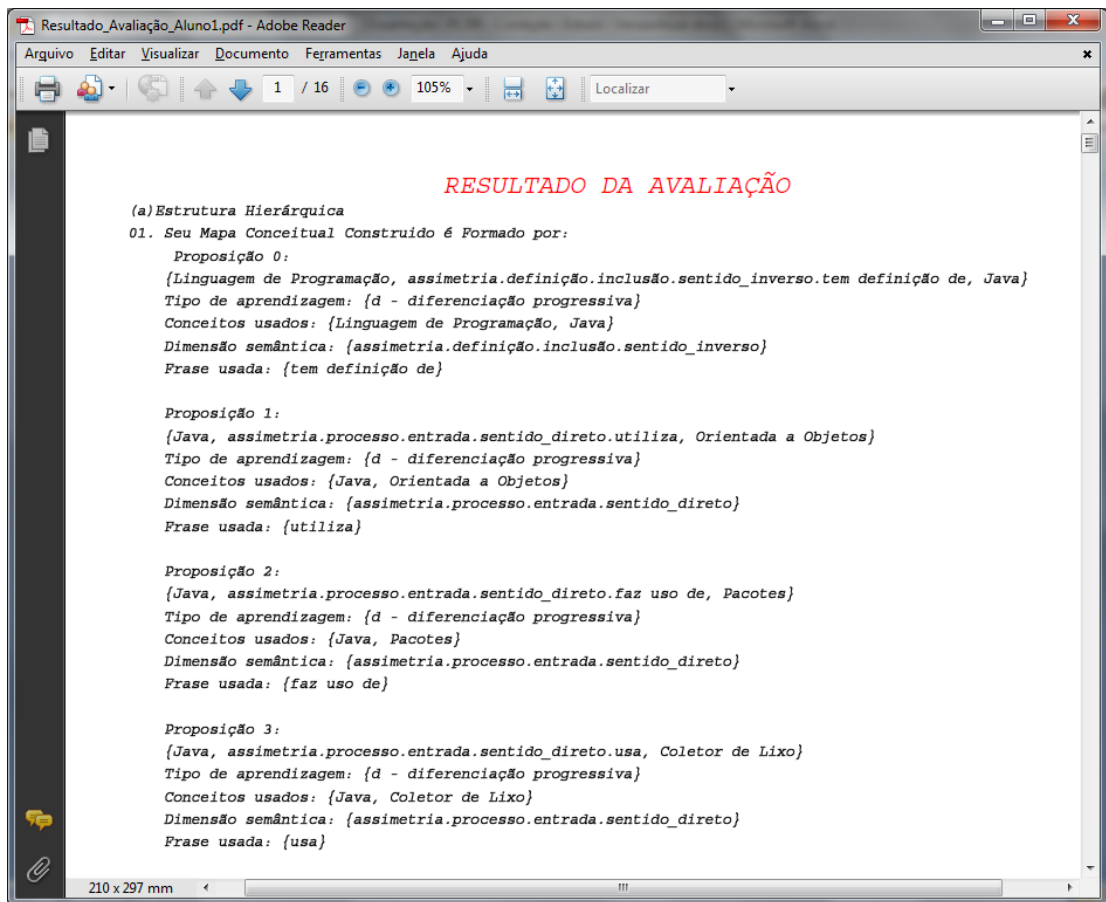
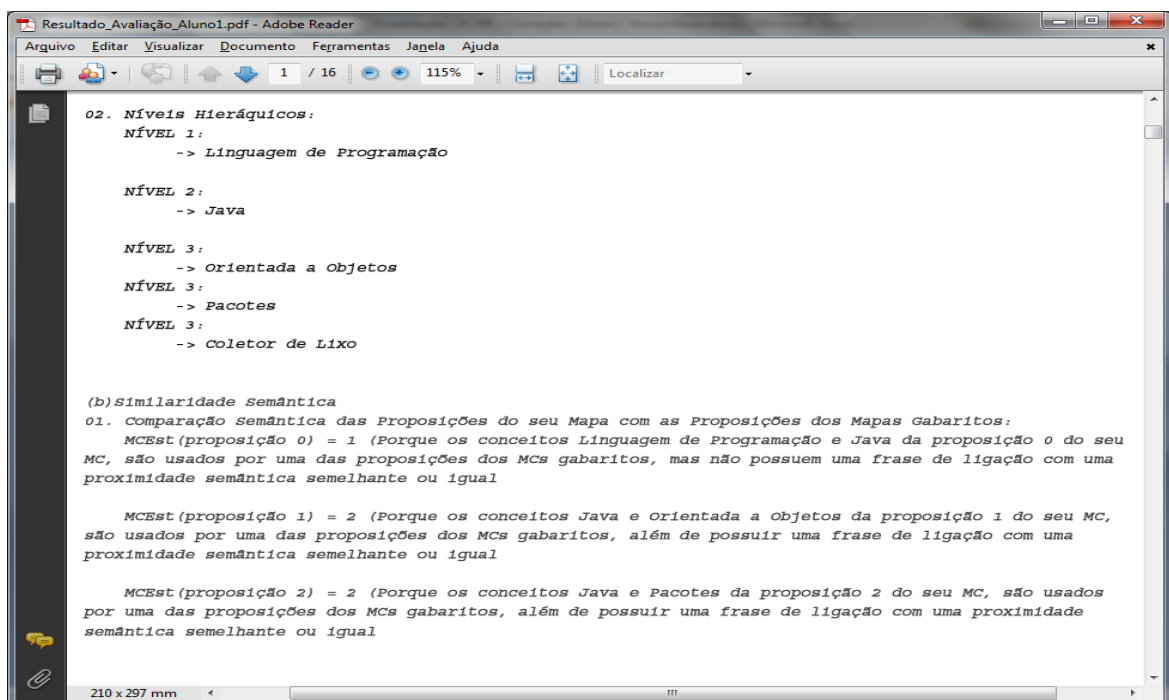


Figura B.2. Níveis hierárquicos e similaridade semântica das proposições do MC



**Figura B.3. Resultado em percentual das proposições do mcest em relação ao mcs gerados**

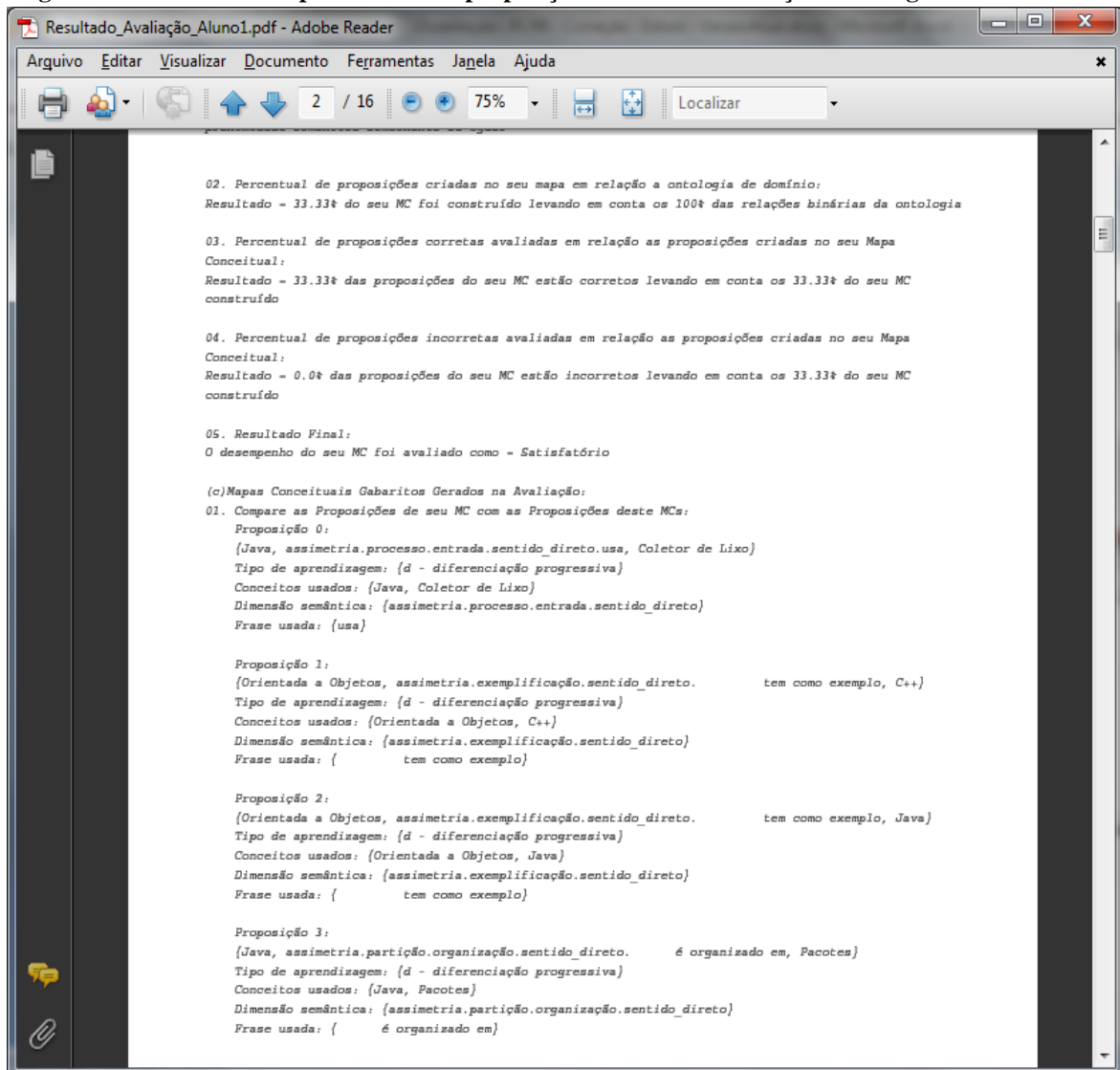




Figura B.4. Exemplo do mc gerado pelo AG após a avaliação do aluno

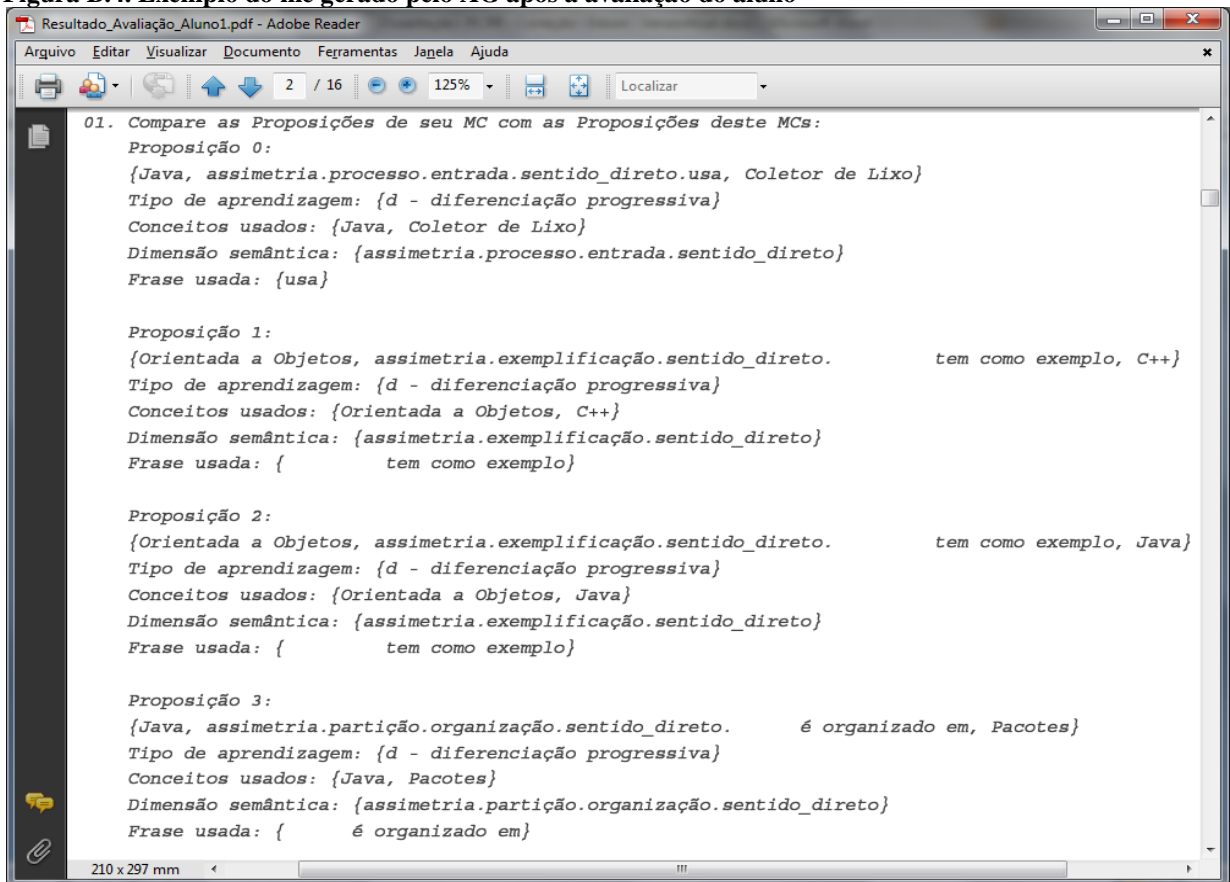


Figura B.5. Visão geral de todos os dados gerados pelo relatório

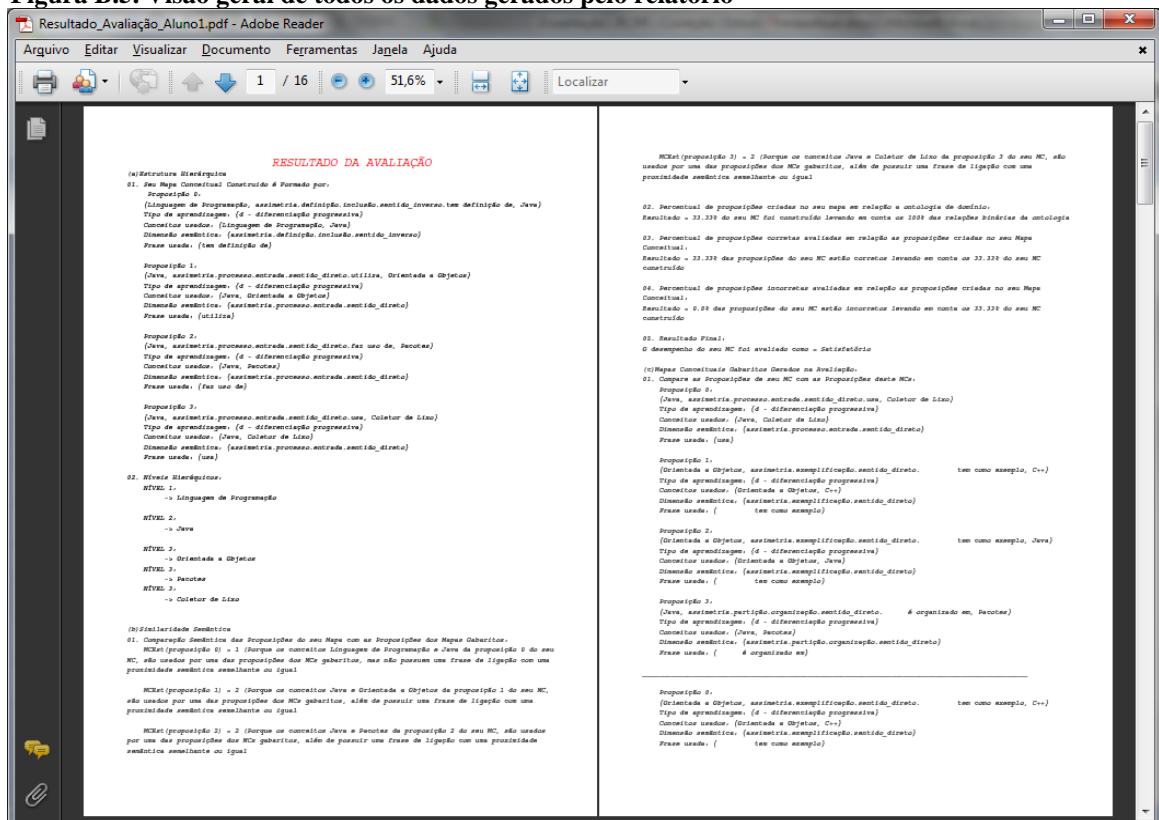


Figura B.6. Visão geral de 4 mapas gabaritos gerados na avaliação

The image shows a screenshot of an Adobe Reader window displaying a document titled 'Resultado\_Avaliação\_Aluno1.pdf'. The document is viewed at 51.6% zoom and shows two pages, each containing 16 items (Proposições) arranged in a grid of 4 groups and 4 items per group. The items are technical specifications for programming concepts, including Java, C++, and C#. Each item lists its title, type of question, user context, dimensions, and the specific question text.

**Page 1 (Left):**

- Proposição 1:** (Java, assimetria.partição.organização.sentido\_direto. # organizado em, Pacotes) Tipo de aprendizagem: (d - diferenciação progressiva) Conhecimentos usados: (Java, Pacotes) Dimensão avaliativa: (assimetria.partição.organização.sentido\_direto) Frase usada: ( # organizado em)
- Proposição 2:** (Orientada a Objetos, assimetria.exemplificação.sentido\_direto. tem como exemplo, Java) Tipo de aprendizagem: (d - diferenciação progressiva) Conhecimentos usados: (Orientada a Objetos, Java) Dimensão avaliativa: (assimetria.exemplificação.sentido\_direto) Frase usada: ( tem como exemplo)
- Proposição 3:** (Java, assimetria.processo.entrada.sentido\_direto.usa, Coletor de Lixo) Tipo de aprendizagem: (d - diferenciação progressiva) Conhecimentos usados: (Java, Coletor de Lixo) Dimensão avaliativa: (assimetria.processo.entrada.sentido\_direto) Frase usada: (usa)
- Proposição 4:** (Java, assimetria.processo.entrada.sentido\_direto.usa, Coletor de Lixo) Tipo de aprendizagem: (d - diferenciação progressiva) Conhecimentos usados: (Java, Coletor de Lixo) Dimensão avaliativa: (assimetria.processo.entrada.sentido\_direto) Frase usada: (usa)
- Proposição 1:** (Orientada a Objetos, assimetria.exemplificação.sentido\_direto. tem como exemplo, C++) Tipo de aprendizagem: (d - diferenciação progressiva) Conhecimentos usados: (Orientada a Objetos, C++) Dimensão avaliativa: (assimetria.exemplificação.sentido\_direto) Frase usada: ( tem como exemplo)
- Proposição 2:** (Java, assimetria.partição.organização.sentido\_direto. # organizado em, Pacotes) Tipo de aprendizagem: (d - diferenciação progressiva) Conhecimentos usados: (Java, Pacotes) Dimensão avaliativa: (assimetria.partição.organização.sentido\_direto) Frase usada: ( # organizado em)
- Proposição 3:** (Orientada a Objetos, assimetria.exemplificação.sentido\_direto. tem como exemplo, Java) Tipo de aprendizagem: (d - diferenciação progressiva) Conhecimentos usados: (Orientada a Objetos, Java) Dimensão avaliativa: (assimetria.exemplificação.sentido\_direto) Frase usada: ( tem como exemplo)
- Proposição 4:** (Orientada a Objetos, assimetria.exemplificação.sentido\_direto. tem como exemplo, C++) Tipo de aprendizagem: (d - diferenciação progressiva) Conhecimentos usados: (Orientada a Objetos, C++) Dimensão avaliativa: (assimetria.exemplificação.sentido\_direto) Frase usada: ( tem como exemplo)

**Page 2 (Right):**

- Proposição 1:** (Java, assimetria.partição.organização.sentido\_direto. # organizado em, Pacotes) Tipo de aprendizagem: (d - diferenciação progressiva) Conhecimentos usados: (Java, Pacotes) Dimensão avaliativa: (assimetria.partição.organização.sentido\_direto) Frase usada: ( # organizado em)
- Proposição 2:** (Java, assimetria.processo.entrada.sentido\_direto.usa, Coletor de Lixo) Tipo de aprendizagem: (d - diferenciação progressiva) Conhecimentos usados: (Java, Coletor de Lixo) Dimensão avaliativa: (assimetria.processo.entrada.sentido\_direto) Frase usada: (usa)
- Proposição 3:** (Orientada a Objetos, assimetria.exemplificação.sentido\_direto. tem como exemplo, Java) Tipo de aprendizagem: (d - diferenciação progressiva) Conhecimentos usados: (Orientada a Objetos, Java) Dimensão avaliativa: (assimetria.exemplificação.sentido\_direto) Frase usada: ( tem como exemplo)
- Proposição 4:** (C++, assimetria.classificação.sentido\_inverso. # tipo de, Linguagem de Programação) Tipo de aprendizagem: (d - diferenciação progressiva) Conhecimentos usados: (C++, Linguagem de Programação) Dimensão avaliativa: (assimetria.classificação.sentido\_inverso) Frase usada: ( # tipo de)
- Proposição 1:** (C++, assimetria.classificação.sentido\_inverso. # tipo de, Linguagem de Programação) Tipo de aprendizagem: (d - diferenciação progressiva) Conhecimentos usados: (C++, Linguagem de Programação) Dimensão avaliativa: (assimetria.classificação.sentido\_inverso) Frase usada: ( # tipo de)
- Proposição 2:** (Java, assimetria.partição.organização.sentido\_direto. # organizado em, Pacotes) Tipo de aprendizagem: (d - diferenciação progressiva) Conhecimentos usados: (Java, Pacotes) Dimensão avaliativa: (assimetria.partição.organização.sentido\_direto) Frase usada: ( # organizado em)
- Proposição 3:** (Java, assimetria.processo.entrada.sentido\_direto.usa, Coletor de Lixo) Tipo de aprendizagem: (d - diferenciação progressiva) Conhecimentos usados: (Java, Coletor de Lixo) Dimensão avaliativa: (assimetria.processo.entrada.sentido\_direto) Frase usada: (usa)
- Proposição 4:** (Java, assimetria.partição.organização.sentido\_direto. # organizado em, Pacotes) Tipo de aprendizagem: (d - diferenciação progressiva) Conhecimentos usados: (Java, Pacotes) Dimensão avaliativa: (assimetria.partição.organização.sentido\_direto) Frase usada: ( # organizado em)

## APÊNDICE C – AJUDA NO WEBCMTOOL

A ajuda do ambiente representa o manual para a compreensão de cada ferramenta de edição incorporada, seja ela o *WebCMEditor* ou o *WebOn\_Tool*. Para tanto, são apresentados alguns *screenshot* do *WebCMTool* destacando este manual presentes em ambos os editores. As Figuras de C.1 a C.5 mostram alguns *screenshot* sobre a ajuda no *WebCMEditor* e as Figuras de C.6 a C.8 ilustram alguns *screenshot* sobre a ajuda no *WebOn\_Tool*.

**Figura C.1 - Ajuda do *webcmeditor* ensinando a criar mapas conceituais livres de contextos**



**Figura C.2. Ajuda do *webcmeditor* ensinando a criar mapas conceituais contextualizados**



As operações comuns a editores como salvar, desfazer, exportar, imprimir, preferências e formatação incluindo a atribuição de cores além de outras como dar zoom a uma ontologia ou a um mapa estão presentes em ambos os editores e, conseqüentemente, no manual de ajuda online de ambas

as ferramentas de edição. Portanto, e como exemplo, as Figuras C.3 a C.6 mostram algumas dessas operações básicas apenas no *WebOn\_Tool* (também presentes no *WebCMEditor*).

**Figura C.3. Início do manual de ajuda da ferramenta *webon\_tool***



**Figura C.4. Operação de salvamento**

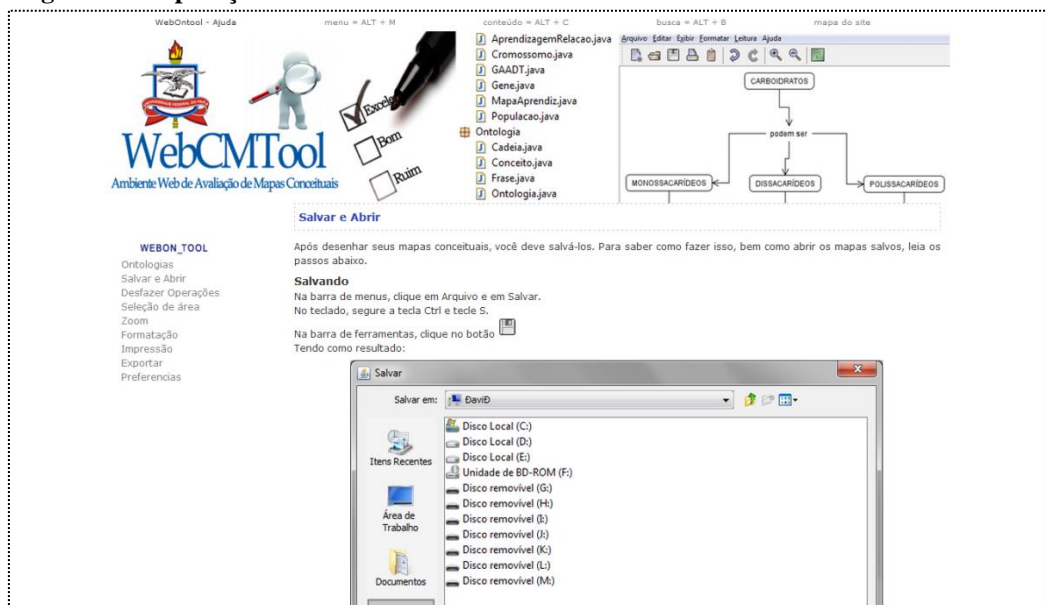


Figura C.5. Operação de preferências

The screenshot displays the WebCMTool web application interface. At the top, there are navigation menus: 'WebCMTool - Ajuda', 'menu = ALT + M', 'conteúdo = ALT + C', 'busca = ALT + B', and 'mapa do site'. The main content area features the WebCMTool logo and a list of Java-based content items such as 'AprendizagemRelacao.java', 'Cromossomo.java', 'GAADT.java', 'Gene.java', 'MapaAprendiz.java', 'Populacao.java', and 'Ontologia.java'. A conceptual map titled 'CARBOIDRATOS' is shown, with 'CARBOIDRATOS' at the top, branching into 'MONOSSACARIDEOS', 'DISSACARIDEOS', and 'POLISSACARIDEOS', with the relationship 'podem ser' (can be) indicated between the parent and child nodes.

**Preferências**

Você pode querer fazer com que os elementos de seus mapas conceituais já tenham uma cor pré-definida, ou um tipo de letra pré-definida. Ou então definir um tipo de papel de acordo com sua impressora, para que as impressões saiam corretamente. Para isso foi feito o diálogo de preferências, onde você pode fazer tais operações.

Há duas maneiras de abrir o diálogo de preferências:  
 Na barra de menus, clique em Editar e, em seguida, "Preferências";  
 No teclado, segure a tecla Alt e tecle P.

O diálogo de preferências abaixo. Nele, você verá 6 opções. Basta clicar em "Alterar", do lado de cada opção. Exceto, é claro, na última opção, onde você seleciona em uma lista a opção desejada.

**WEBON\_TOOL**  
 Ontologias  
 Salvar e Abrir  
 Desfazer Operações  
 Seleção de área  
 Zoom  
 Formatação  
 Impressão  
 Exportar  
 Preferencias

**Preferências**

Cor padrão de linhas:  Alterar...

Estilo padrão de linhas: Normal

Cor padrão de bordas:  Alterar...

Cor padrão de preenchimento:  Alterar...

Cor padrão de letras:  Alterar...

Fonte padrão: Dialog, normal, 12 Alterar...

OK Cancelar

## APÊNCIDE D – QUESTIONÁRIO DE SATISFAÇÃO

### UNIVERSIDADE FEDERAL DO PARÁ INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Caro(a) estudante,

Sou aluno do Programa de Pós-Graduação em Ciência da Computação da UFPA e atualmente estou desenvolvendo a minha dissertação, sob a orientação do prof. Dr. Francisco Edson Lopes da Rocha, intitulado “**WEBCMTOOL: Ambiente Web para Facilitar a Avaliação da Aprendizagem Significativa Baseado em Mapas Conceituais e Ontologias de Domínio**”. O objetivo da pesquisa é facilitar a avaliação da aprendizagem significativa ajudando na construção do conhecimento individual de cada aluno por meio da criação de mapas conceituais baseados em conjunto de conceitos disponibilizados por meio de ontologias de domínios.

Sendo assim, gostaria de convidá-lo(a) para participar desta pesquisa, destacando que a sua colaboração é de grande importância neste estudo, para que possamos melhorar o aprendizado do aluno sobre o conteúdo de uma disciplina, matéria ou assunto específico que vier ser estudado.

Cabe ressaltar que as opiniões apresentadas são fundamentais para o desenvolvimento desta pesquisa, ficando garantido o anonimato de suas contribuições.

Desde já agradeço a sua colaboração e me coloco à disposição para quaisquer esclarecimentos.

Atenciosamente,

---

**Renan Rodrigues**

---

### QUESTIONÁRIO

#### 1. PERFIL DO(A) ALUNO(A)

**1.1 Curso:** ( ) CBCC ( ) CBSI ( ) OUTRO **Especifique:** \_\_\_\_\_

**1.2 Semestre que está cursando:** \_\_\_\_\_

**1.3 Idade:** \_\_\_\_\_

**1.4 Frequência com que usa a internet:**

- ( ) Menos de 1 hora diária.
- ( ) De 1 a 3 horas diárias.
- ( ) De 3 a 5 horas diárias.
- ( ) De 5 a 7 horas diárias.
- ( ) Mais de 7 horas diárias.

## **2. CONHECIMENTO SOBRE APRENDIZAGEM SIGNIFICATIVA, ONTOLOGIAS E MAPAS CONCEITUAIS**

**2.1 Antes de acessar o ambiente WebCMTTool você já tinha algum conhecimento prévio sobre Aprendizagem Significativa?**

- Sim.
- Não.
- Um pouco.

**2.2 Antes de ouvir sobre ontologias de domínio e começar a criar seu mapa conceitual contextualizado sobre programação você já tinha algum conhecimento prévio sobre ontologias de domínio?**

- Sim.
- Não.
- Um pouco.

**2.3 Antes de começar a criar seu mapa conceitual contextualizado sobre programação você já tinha algum conhecimento prévio sobre mapa conceitual?**

- Sim.
- Não.
- Um pouco.

## **3. SOBRE O AMBIENTE WEBCMTOOL E SUA FERRAMENTA WEBCMEDITOR**

**3.1 Você considera o ambiente WebCMTTool interessante para o aprendizado na educação?**

- Sim.
- Não.
- Um pouco.

**3.2 Você sentiu dificuldade para escolher qual frase correta utilizar para formar as proposições de seu mapa conceitual?**

- Sim.
- Não.
- Um pouco.

**3.3 Você encontrou dificuldade no uso da ferramenta WebCMEditor?**

- Sim.
- Não.
- Um pouco: Qual dificuldade? \_\_\_\_\_

**3.4 Você consideraria ter aprendido os conceitos de programação ensinados no curso após analisar o resultado da avaliação gerado no documento em PDF referente a avaliação do seu mapa conceitual avaliado e após visualizar os gabaritos de mapas gerados no mesmo?**

- Sim.
- Não.
- Um pouco: Por que? \_\_\_\_\_

**3.5 Você considera a ferramenta WebCMEditor útil para ajudar o aluno a aprender o significado correto de cada conceito estudado no curso com base na análise dos resultados?**

- Sim.
- Não.
- Um pouco.

**3.6 Você considera importante para o aprendizado do aluno no processo educacional a pesquisa pelo conhecimento, o estudo individual e, a análise e compreensão de erros cometidos em tarefas realizadas, como critérios fundamentais para a construção do conhecimento individual?**

- Sim.
- Não.
- Um pouco.

**3.7 Você acha que o ambiente apresenta algum problema específico?**

- Sim.
- Não.
- Um pouco.

**3.8 Você acha que o ambiente está bom, teria alguma dica de melhoria para que possamos melhorar ainda mais o WebCMTTool?**

- Sim
- Não
- Um pouco

Melhoria(s):

---

---

---

#### **4. QUANTO AO ENSAIO DE INTERAÇÃO**

**4.1 A interface de apresentação do ambiente WebCMTTool é agradável?**

- Sim
- Não

**4.2 A navegação pelo ambiente WebCMTTool é fácil?**

- Sim.
- Não.

**4.3 A ferramenta de construção de mapa conceitual WebCMEditor é fácil de ser utilizada?**

- Sim.
- Não.

**4.4 Os comandos de uso da ferramenta de mapa conceitual WebCMEditor são claros?**

- Sim.
- Não.



**4.5 Você encontrou dificuldade para escolher uma frase apropriada para ligar dois conceitos no seu mapa conceitual quando utilizou papel e caneta?**

Sim.

Não.

Caso sim, qual? \_\_\_\_\_

---

---

**4.6 Você encontrou dificuldade para escolher uma frase apropriada para ligar dois conceitos no seu mapa conceitual quando utilizou a ferramenta automática WebCMEditor?**

Sim.

Não.

Caso sim, qual? \_\_\_\_\_

---

---

**4.7 Atribua uma nota de 0 a 10 ao ambiente WebCMTTool utilizado?**

Nota: \_\_\_\_\_

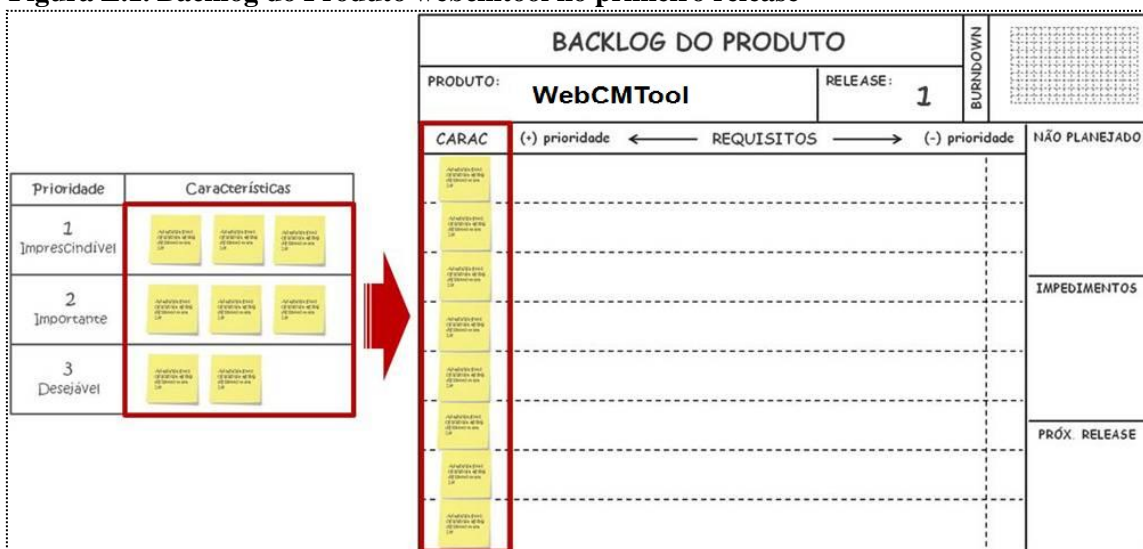
Grato pela atenção.

## APÊNCIDE E – ESTUDO DE VIABILIDADE

O estudo de viabilidade (*Feasibility Study*), segundo Silveira (2011), avalia a proposta de um novo projeto, considerando os pontos de aceitação e rejeição, quais metas são esperadas, recursos necessários e os riscos associados. De posse disto, o estudo de viabilidade realizado para o *WebCMTTool* fez parte do *Backlog* do Produto da metodologia ágil *Scrum*, que ocorreu na fase inicial, cuja finalidade consistiu em escrever as características-chave do projeto, a fim de verificar se a sua construção iria satisfazer ao escopo definido.

O *Backlog* do Produto (Seção 3.3) contém estórias, que são funções do produto que podem ser descritas, priorizadas e estimadas, em que descreve uma necessidade ou situação futura que o usuário pretende alcançar através da utilização do produto a ser desenvolvido. A Figura E.1 apresenta como o *Backlog* do Produto foi organizado em conjunto com estórias definidas por níveis de prioridades de acordo com as características do *WebCMTTool*.

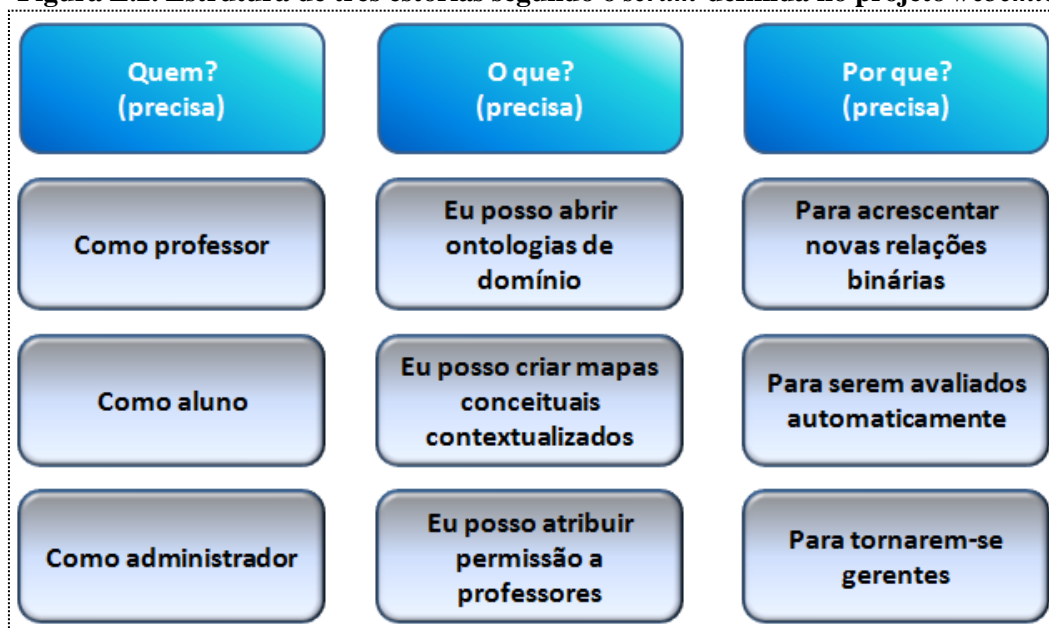
**Figura E.1. Backlog do Produto webcmttool no primeiro release**



Fonte: Adaptado de Simões, 2011

As estórias correspondem aos requisitos funcionais e não funcionais, segundo a modelagem ágil *Scrum*. Na Figura E.2, é apresentado um exemplo da estrutura de uma estória utilizada no ambiente em que é possível visualizar as necessidades ou situações futuras, a partir de três pontos de vista diferentes: i) quem - que representa o usuário que utilizará o ambiente; ii) o que - que representa a necessidade ou situação futura desejada e; iii) para que - que representa a finalidade. Para que seja considerada completa, a estória precisa estar formada por essas três partes: **quem, o que e por que**.

**Figura E.2. Estrutura de três estórias segundo o *scrum* definida no projeto *webcmtool***



Segundo o modelo *INVEST*, criado por Bill Wake (2003), para um requisito (estória) fazer parte do Backlog do Produto, deve respeitar os seguintes critérios:

- ◆ **ser independente:** o requisito deve possuir capacidade para atender a necessidade ou situação futura, informada pelo cliente, sem depender de outro requisito. Essa condição é fundamental para garantir flexibilidade durante o ciclo de desenvolvimento do produto;
- ◆ **ser negociável:** enquanto não for convertido em produto, o requisito deve permitir alterações, como mudança de prioridade, aumento/redução da sua abrangência e desdobramento em outros requisitos, podendo inclusive ser reescrito ou mesmo descartado, desde que mantido o valor esperado pelo cliente final na entrega do produto;
- ◆ **ser priorizado:** o requisito deve, obrigatoriamente, assegurar a entrega de valor para o cliente, caso contrário, seu desenvolvimento poderá representar desperdício de esforço para o Time de Projeto. Por este motivo, o requisito é priorizado pelo Dono do Produto, de acordo com o valor agregado ao negócio/cliente final;
- ◆ **ser estimado:** o requisito deve apresentar condições para que seu prazo de desenvolvimento/entrega possa ser estimado. Caso o requisito não ofereça essas

condições, em virtude, por exemplo, do seu tamanho, deverá ser desdobrado em requisitos menores, até que possa ser adequadamente estimado;

- ◆ **ser pequeno:** este critério está diretamente relacionado ao anterior. O requisito deve estar descrito de uma forma que permita sua estimativa com certo nível de certeza (quanto menor, melhor). Outro aspecto a ser observado é que a duração prevista do requisito não deve ultrapassar o prazo de execução dos ciclos de trabalho estabelecidos para o projeto (Sprints), e;
- ◆ **ser inspecionado:** o requisito propriamente dito ou sua descrição deve prover as informações necessárias para que possa ser inspecionado/testado pelo cliente final. Requisitos que não podem ser validados elevam os níveis de risco do projeto e podem ocasionar problemas graves em etapas mais avançadas do projeto.

Assim, e para a construção destas estórias, optou-se por relacioná-las com base em quatro aspectos definidos no *PRINCE2 (PRoject IN Controlled Environment 2)* (Ribeiro, 2011). Esses aspectos controlam o bom desenvolvimento de um projeto, onde frisam se o ambiente proposto é viável para a construção. São eles:

1. Tempo;
2. Escopo;
3. Riscos;
4. Benefícios.

A seguir, são relatados os aspectos que validaram a viabilidade e continuação do projeto na construção das estórias para o *Backlog* do Produto sob o ponto de vista dos aspectos do PRINCE2.

### **E.1 Tempo e escopo**

Inicialmente, o tempo do projeto foi definido segundo um cronograma em meses, ampliado, posteriormente, para semestres, desenhado desde os estudos bibliográficos conceituais e técnicos até a modelagem e construção sob o ponto de vista dos interessados. Para tanto, foram ressaltados o escopo do projeto, quais recursos seriam utilizados e quais eram necessários para a execução.

Segundo Moura et. al. (2011, p. 122) o escopo expressa a extensão ou amplitude do projeto (em termos do que se pretende realizar, abarcar ou abranger), estabelece o seu raio de ação ou cobertura, definindo, portanto, seus limites. De posse disto, os elementos que constituíram e definiram o escopo no âmbito do projeto estão descritos na Figura E.3:

**Figura E.3. Elementos que basearam a determinação do escopo do *webcmtool***

<b>ELEMENTOS DO ESCOPO</b>	<ul style="list-style-type: none"> <li><input type="checkbox"/> <b>Definição do problema ou situação geradora do projeto</b> (problema, necessidade, desafio, oportunidades);</li> <li><input type="checkbox"/> <b>Justificativa (o porquê) do projeto</b> (que pode conter um diagnóstico da situação inicial, também denominado de <i>baseline</i> ou "linha de base");</li> <li><input type="checkbox"/> <b>Objetivos geral e específicos do projeto</b> (a razão de ser e o para quê);</li> <li><input type="checkbox"/> <b>Resultados esperados com a realização do projeto</b> (diretamente relacionados com os objetivos específicos do mesmo);</li> <li><input type="checkbox"/> <b>Abrangência do projeto</b> (público alvo e caracterização da extensão e área de atuação do projeto);</li> </ul>
----------------------------	---

Fonte: Moura e Barbosa, 2011, p. 122

Portanto, o escopo é um componente que respondem às seguintes questões:

- Do que se trata o projeto? Qual a situação, problema ou necessidade que deu origem ao projeto?
- Por que vale a pena investir recursos no desenvolvimento do projeto?
- Quais resultados podemos esperar com a realização do projeto? O que se pretende realizar com seu desenvolvimento? Que benefícios são esperados? Quais serão os beneficiados com sua realização?
- Qual a área de atuação do projeto? Qual sua dimensão em termos de público alvo? Que volume de recursos deverá ser investido?

Estas questões permitiram levantar os recursos que deveriam ser utilizados no ambiente para que as atividades fossem realizadas; desta forma, destacaram-se dois tipos de recursos: i) pessoas e; ii) materiais. O Quadro E.1 apresenta os recursos que foram utilizados.

**Quadro E.1. Recursos utilizados no *webcmtool***

Recursos	Exemplos
Pessoas	Professores
	Alunos
	Analistas de sistemas
Materiais	Computador
	Ferramentas de desenvolvimento
	Livros, artigos, e outras leituras bibliográficas
	Energia
	Tempo e estrutura (laboratório)

Após definir o escopo, foi analisado o tempo para o desenvolvimento do webcmtool com base num cronograma gerado em semestres. O Quadro E.2 demonstra este cronograma.

**Quadro E.2. Exemplo do cronograma geral utilizado no projeto webcmtool**

Nome do projeto: WebCMTTool							
Etapa	Semestres						
	1° (2009)	2° (2009)	1° (2010)	2° (2010)	1° (2011)	2° (2011)	1° (2012)
Estudos bibliográficos do tema							
Análise arquitetura do CMTTool							
Adaptação do Scrum							
Construção do WebOn_Tool							
Construção do WebCMEditor							
Elaboração do repositório							
Remodelagem do AG							
Integração AG e WebCMEditor							
Construção do WebCMTTool							

## E.2 Riscos e benefícios

A metodologia ágil Scrum que compunha as métricas deste desenvolvimento não possui uma gestão de risco. Portanto, para auxiliar nesta tarefa, a fim de verificar os riscos aderidos ao projeto, foi utilizado o modelo de risco que constitui o *PMBOK* (PMI, 2008).

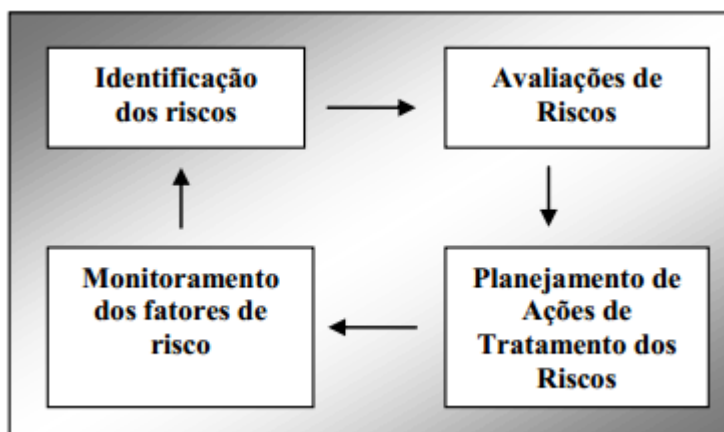
Para Ward (2000, p. 138), o risco é definido como:

“o efeito acumulativo da probabilidade de incerteza que pode afetar positivamente (oportunidade) ou negativamente (ameaça) o projeto.”

Este conceito apenas demonstra que gerenciar riscos de forma criteriosa é fundamental para o sucesso do projeto. Nesse contexto, durante o desenvolvimento a preocupação com o risco não diminui, pois segundo Gomes (2010, p. 66), os riscos encontrados no início do projeto devem ser monitorados até sua conclusão. O monitoramento e controle do risco é um processo contínuo que deve ocorrer em cada iteração. A Figura E.4 apresenta o processo cíclico de gerência de risco, que acompanhou o desenvolvimento, em que quando identificado um risco ele era avaliado e com isto se realizava um

planejamento de ações para tratamento daquele risco. Realizada esta etapa, um monitoramento dos impactos era estabelecido e feito um levantamento de novos riscos no projeto.

**Figura E.4. Processo cíclico de gerência de risco**



Fonte: Gomes, 2010, p. 66

A identificação dos riscos do projeto foi feita com base no modelo proposto por Carvalho (2004, p. 4252), apresentado pelo Quadro E.3. Neste quadro, os riscos foram anotados, detalhados e classificados quanto ao seu aspecto crítico e os impactos e magnitudes foram divididas em nove subclasses, conforme se destaca.

**Quadro E.3. Modelo de análise de riscos**

Atividade	Impacto do Risco			Magnitude do Impacto			Ação Planejada
	Baixo (1-3)	Médio (4-6)	Alto (7-9)	Baixo (1-3)	Médio (4-6)	Alto (7-9)	

Fonte: Carvalho, 2004, p. 4253

A ideia de se analisar os riscos é para resolver os problemas que ameaçam as funcionalidades e o sucesso. Por outro lado, também há os benefícios que se caracterizam pelos pontos positivos que realçam os objetos do projeto. Alguns dos benefícios que abordaram o webcmttool durante sua análise de viabilidade foram:

1. Diminuição da sobrecarga do professor na tarefa de avaliação;
2. Construção do conhecimento significativo pelo aluno por meio da análise dos resultados da avaliação;
3. Disponibilização de um sistema de avaliação automático na Web;
4. Integração dos módulos já desenvolvidos no CMTTool;
5. Facilidade de construção de MCs e ontologias de domínio.

## ANEXO A – RELATÓRIO DA AVALIAÇÃO HEURÍSTICA DO WEBCMTOOL

AVALIADOR(ES):

GABRIEL SOUZA, GLAUBER MONTEIRO E RODRIGO BARBALHO

### 1. Software avaliado

Ambiente *WebCMTool* para avaliação de MCs.

### 2. Público

Usuários professores e alunos em geral

### 3. Quantidade de Heurísticas violadas

Na avaliação heurística realizada foram encontradas 26 violações heurísticas segundo o esquema abaixo:

**Quadro A.1. Relação dos totais das avaliações realizadas pelo webcmtool**

Heurística	Explicação	Qtd de violações
1	Visibilidade do status do sistema	1
2	Compatibilidade do sistema com o mundo real	1
3	Controle e liberdade para o usuário	2
4	Consistência e padrões	9
5	Prevenção de erros	1
6	Reconhecimento ao invés de relembração	1
7	Flexibilidade e eficiência de uso	6
8	Estética e projeto minimalista	5
9	Ajuda aos usuários a reconhecer, diagnosticar e corrigir erros	0
10	Ajuda e documentação	0

### 4. Detalhes da avaliação:

Os detalhes analisados e avaliado pelos especialistas estão especificados pelos quadros a seguir, que demonstram o problema encontrado, o princípio de usabilidade infringido, o grau de severidade, sugestão e exemplo de correção.

### Quadro A.2. Violação de usabilidade na índex do ambiente



**Problema:** O texto da página inicial contém informações válidas, porém com uma fonte de texto pequena e não dividido em seções pode fazer com que o usuário perca o interesse em ler todo o texto.

**Conheça o WebCMTTool -> Início**

O WebCMTTool é um ambiente que visa facilitar a avaliação da Aprendizagem Significativa (AS) mediada por Mapas Conceituais (MCs) e Ontologias de Domínio. A facilidade da avaliação se caracteriza por diminuir a sobrecarga de trabalho do professor haja vista que ocorre de forma automática sobre a responsabilidade de um Algoritmo Genético (AG) incorporado ao ambiente. A tarefa de avaliação proposta no ambiente WebCMTTool tende ao construtivismo, além de valorizar a idiossincrasia de cada aluno tentando provar quando este está correto, ressaltando acima de tudo, qual o resultado da avaliação sob o ponto de vista do próprio aluno que é submetido a uma análise detalhada de sua avaliação por meio de um relatório gerado pelo AG.

O propósito desta análise é fazer com que o aluno analise e entenda os erros e acertos em seu mapa conceitual avaliado, bem como entender porque sua avaliação foi considerada satisfatória ou não. Para que a tarefa de avaliação ocorra é necessário três insumos de entrada: o mapa do aluno, a ontologia do professor e uma taxonomia de frases de ligação que contém frases semanticamente definidas na língua portuguesa. Durante o processo de avaliação, são gerados mapas conceituais descendentes escolhidos com base em um valor de adaptação, quando este for maior ou igual à média da população. De posse desta estrutura, os MCs gerados por descendência resultante da avaliação, são utilizados para análise e compreensão de resultados pelo aluno, que verifica o valor atribuído às proposições de seus mapas, a nota em percentual que valida a quantidade de proposições consideradas corretas e o resultado de desempenho de seu mapa. Um protótipo do ambiente desenvolvido foi validado com base em duas ontologias de domínio: sobre o Ciclo da Água e sobre Paradigmas de Programação.

WebCMTTool trata-se de uma extensão do CMTTool criado pelo pesquisador Francisco Edson Lopes da Rocha, cujo principal objetivo é a avaliação de MCs de forma automática. Para a criação destes MCs são disponibilizados conjuntos de conceitos por meio de uma ontologia de domínio, mas sem as frases que une esses conceitos. A ontologia permite compartilhar conhecimento, neste caso, o conhecimento do professor ou especialista acerca de um domínio. Neste contexto, os mapas são desenvolvidos pelos alunos, no qual se baseiam em conceitos presentes em uma ontologia de domínio previamente criada por um professor.

Sobre o ponto de vista da pesquisa, a integração de mapas conceituais, ontologias de domínio e algoritmos genéticos possibilitam um avanço no estado da arte de avaliação e acompanhamento automático da aprendizagem. Quebra-se o paradigma das avaliações apenas quantitativas, apresentando uma nova abordagem de acompanhamento gradual e contínuo das atividades do estudante. Nesta abordagem pode-se fazer o acompanhamento individual, respeitando a forma idiossincrática de aprender, e/ou de grupo de estudantes, sendo possível agrupá-los por características cognitivas específicas ou por grau de desenvolvimento.

**Princípio de usabilidade infringido:** Estética e projeto minimalista

**Grau de severidade:** 2

**Sugestão:** Aumentar o tamanho da fonte de texto e o espaçamento. Destacar trechos do texto e dividir o texto em seções.

**Exemplo:**

**THE POMI APPROACH**

We will address the barriers described above by creating **platforms for innovations** and making them available to others to use and build on in the context of our proposed **Shoka** three-tier architecture comprising handheld devices, generic desktop computers and displays, and the cloud with lots of computing and storage.

**Our proposed platforms include:**

1. a virtual data system called PRPL that enables users to take back ownership of their data,
2. a computing substrate with networks of VMs and mobile VMs to ease the entry of new web services,
3. an open network to promote network innovation and make wireless capacity available across heterogeneous and abundant radio networks in a vicinity, and
4. open-source software to promote existing efforts in opening the cell phone including energy efficient secure OS, secure and extensible mobile browser, and customizable UI system.

**As a demonstration of our research, our team will deploy a prototype system of infrastructure, devices and applications of scale across the Stanford campus.** We believe it will be the most comprehensive, experimental deployment of mobile technology ever performed by a university. We will also seek to have a profound impact by providing mobile wireless infrastructure and content for K-12 schools, especially in underserved communities.

In short, our team aims to open up and secure the devices, software development and the network infrastructure enabling lots of innovations leading to Mobile Internet of 2020.

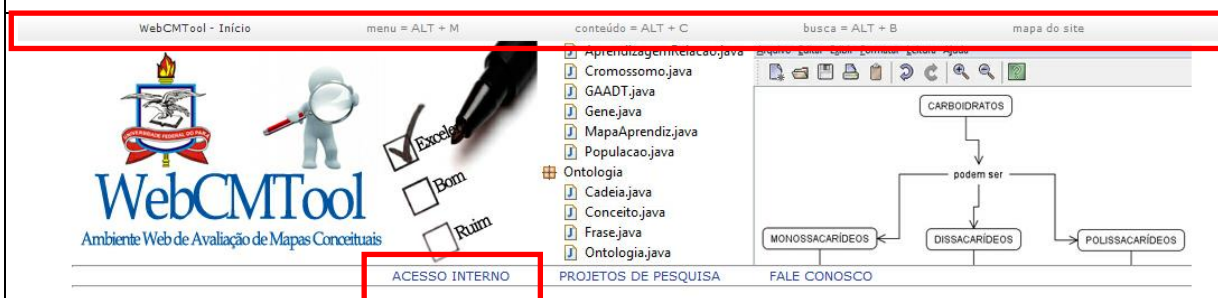
**FOR MORE INFORMATION**

We invite you to find out more by visiting the following links and by watching this space as the research progresses.

- [Overview presentation](#)
- [Press release](#)

**Quadro A.3. Violação de usabilidade na barra de menu do ambiente**

**Problema:** a) A barra do menu possui os itens muito afastados entre si, o que não corresponde a um padrão de menu. Também não existe um destaque em cada item ao movimentar o mouse. Apesar de ser um sistema voltado para os usuários inicialmente da área de computação, não é interessante visualizar informações a respeito de atalhos no label do menu. Por fim, o menu “Início” é um recurso em desuso, uma vez que os usuários podem acessar a página inicial do *WebCMTool* clicando na figura da logo; b) A opção “ACESSO INTERNO” não se destaca e contém um nome pouco sugestivo.

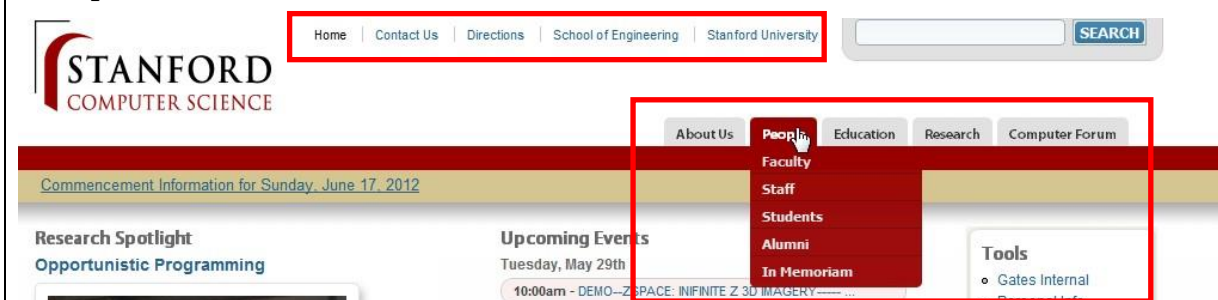


**Princípio de usabilidade infringido:** Flexibilidade e eficiência de uso, compatibilidade do sistema com o mundo real, estética e projeto minimalista, consistência e padrões.

**Grau de severidade:** 3

**Sugestão:** Utilizar um padrão de menu, como os exemplos de Stanford e MIT

**Exemplo 1:**



**Exemplo 2:**



**Quadro A.4. Violação de usabilidade na barra lateral do ambiente**

**Problema:** Pesquisas apontam que ao acessar um site, o usuário possui a frequência de acessar as informações da esquerda para a direita e do topo ao rodapé da página. O menu lateral poderia ser integrado a barra de menu. Além disso, existem informações que poderiam estar em outras seções (Por exemplo: o conteúdo “Contribuições científicas” poderia estar em “Publicações”), desta forma pode ser necessária a reorganização das informações do menu.

#### CONHEÇA O WEBCMTOOL

Início  
 História do Ambiente  
 Identificação do Problema  
 Arquitetura de Módulos  
 Contribuições Científicas  
 Projetos de Pesquisa  
 Integrantes

#### LINHAS DE PESQUISA

Mapas Conceituais  
 Ontologias de Domínio  
 Algoritmos Genéticos  
 Aprendizagem Significativa

#### PUBLICAÇÕES

Revistas  
 Conferências  
 Congressos  
 Simpósios

#### PARCEIROS

Grupo do Canadá  
 Grupo da USP  
 Grupo da UFRJ

**Princípio(s) de usabilidade(s) infringido(s):** Consistência e padrões, Estética e projeto minimalista.

**Grau de severidade:** 2

**Sugestões:**

- Projetos de pesquisa está relacionado com as linhas de pesquisa. Poderia haver um menu “Pesquisa” em que mostram as duas subseções.
- “Integrantes” poderia ser um menu a parte, com o nome “Equipe”. Nesta seção há a palavra Membros antigos que pode ser considerada uma palavra forte – trocar para “Ex-membros”.

**Quadro A.5. Violação de usabilidade no rodapé do ambiente**

**Problema:** O rodapé da página poderia conter outros links que não fossem apenas o mapa do site. Desta forma, os usuários poderiam acessar itens do menu principal pelo rodapé, ao invés de retornar ao topo da página.



**Princípio(s) de usabilidade(s) infringido(s):** Consistência e Padrões, Flexibilidade e eficiência de uso.

**Grau de severidade:** 2

**Sugestões:** Seguir padrões de outros sites, como o do site do greenpeace.

**Exemplo:**



**Quadro A.6.**

**Problema:** É recomendável publicar o link somente quando estiver pronto.

CONHEÇA O WEBCMTOOL

Início

História do Ambiente

Identificação do Problema

Arquitetura de Módulos

Contribuições Científicas

Conheça o WebCMTool -> História

Em construção...

**Princípio(s) de usabilidade(s) infringido(s):** Estética e projeto minimalista, Prevenção de erros.

**Grau de severidade:** 2

**Sugestões:** Utilizar o mesmo nome do menu e publicar a página somente quando estiver pronta para publicação.

### Quadro A.7. Violação de usabilidade no acesso interno do ambiente

**Problema:** A seção da área reservada não se encontra no canto superior direito da página. Além disso, o botão possui um label grande, fugindo do padrão existente em outros sites e de outros botões. Também existe a função de recuperação da senha está localizada antes do login, o que também não está de acordo com os padrões. Por fim, os chamados breadcrumbs não foram inclusos nesta página.

The screenshot shows a login interface with several usability issues highlighted by a red box:

- A large, prominent button at the top left says "Informe seu usuário e sua senha!".
- Below it are two smaller buttons: "Registre-se caso ainda não possua um usuário e senha" and "Esqueci minha senha".
- The main login form is titled "Acesso Interno" and contains:
  - Fields for "Usuário: \*" and "Senha: \*".
  - A checkbox for "Entrar automaticamente".
  - An "Acessar" button.

**Princípio(s) de usabilidade(s) infringido(s):** Consistência e Padrões, Flexibilidade e eficiência de uso.

**Grau de severidade:** 3

**Sugestões:** Seguir padrões de outros sites, como o do site do greenpeace ou do gmail. O gmail possui um botão com label menor para indicar a criação de uma nova conta. Enquanto o Greenpeace possui a localização do acesso à área restrita no canto superior direito.

**Exemplo:**

Two examples of better login page designs are shown:

- Example 1 (Gmail):** Shows a clean layout with "New to Gmail?" and a "CREATE AN ACCOUNT" button in the top right corner.
- Example 2 (Greenpeace):** Shows a search bar at the top right with "Press Publications" above it, and a "Sign in | New here? Sign up!" button in the bottom right corner.

### Quadro A.8. Violação de usabilidade no registro do ambiente

**Problema:** A área de registro de usuário possui alguns pontos que não seguem o padrão de formulários. O campo de idade possui um limite maior que 3 algarismos. Alguns estados possuem a grafia errada, conforme o destaque. Por fim, os campos possuem um espaçamento pequeno o que pode induzir o usuário a confundir qual campo está sendo acessado.

**Registro de Usuário**

Informe seu nome e sobrenome:

Qual a sua idade?

E-mail para contato:

Onde estuda/trabalha?

Informe qual curso/disciplina aplicará:

Qual o seu perfil?  Aluno  Professor

Informe a sua titulação:

Qual o seu estado?

Idioma:

Conhece mapas conceituais?

Conhece ontologias de domínio?

Como soube do ambiente?

Login:

Senha:

Confirmar Senha:

**Princípio(s) de usabilidade(s) infringido(s):** Consistência e padrões, flexibilidade e eficiência de uso, estética e design minimalista.

**Grau de severidade:** 2

**Sugestões:** Seguir padrões de formulários de outros sites com maior espaçamento (como o exemplo do gmail) e realizar as correções dos campos de “Idade” e “Estado” (mais precisamente o estado do Amazonas e também Roraima).

**Exemplo:**

**Name**

**Choose your username**

@gmail.com

**Create a password**

**Confirm your password**

**Birthday**

### Quadro A.9. Violação de usabilidade na saída do ambiente

**Problema:** a) Ao realizar o login no ambiente, o usuário não é informado que será necessária a instalação do plugin do Java. Além disso o ícone sair (conforme o destaque) não possui nenhum label para indicar a função textualmente; b) Quando aberto a ontologia para se criar um MC contextualizado no *WebCMEditor* a tela ficou branca não apresentando nenhuma mensagem de erro ou aviso sobre a situação ou status de processamento.

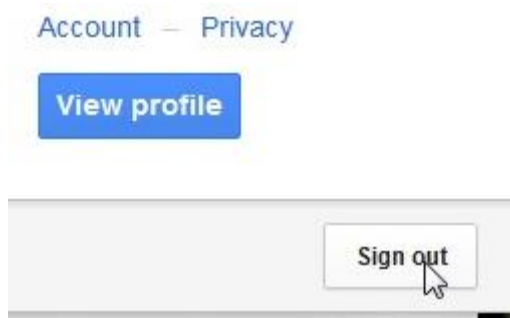


**Princípio(s) de usabilidade(s) infringido(s):** Consistência e padrões, visibilidade do status, controle e liberdade do usuário, flexibilidade e eficiência de uso.

**Grau de severidade:** a - 2 / b - 4

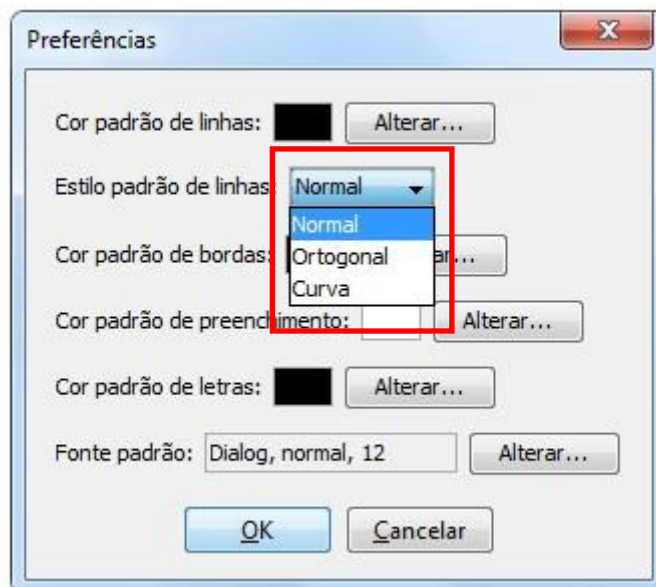
**Sugestões:** 1) Ao invés de utilizar um ícone, poderia ser um botão com a indicação textual “sair”; 2) Informar o usuário com mensagens de erros que possam vir a ocorrer.

**Exemplo:**



**Quadro A.10. Violação de usabilidade no estilo de linha do ambiente**

**Problema:** O usuário não pode visualizar qual o estilo de linha para ser utilizado no mapa conceitual.

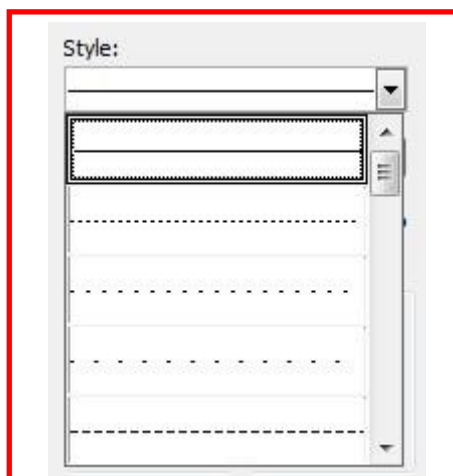


**Princípio(s) de usabilidade(s) infringido(s):** Consistência e padrões, reconhecimento ao invés de relembração.

**Grau de severidade:** 3

**Sugestões:** Colocar o estilo das linhas antes de seleccionar os itens. Ex.: Corel Draw 12.

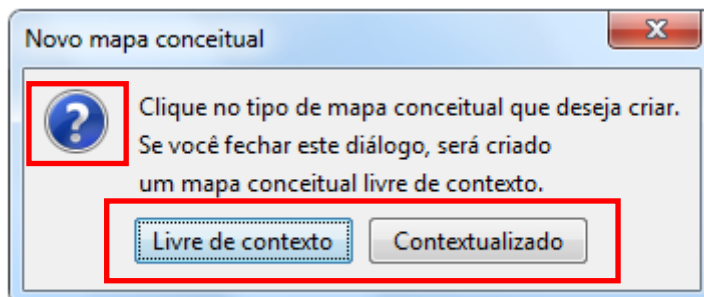
**Exemplo:**





### Quadro A.11. Violação de usabilidade no diálogo de novo mapa do ambiente

**Problema:** a) O usuário não pode cancelar a opção de criar um novo mapa conceitual. É imprescindível acrescentar uma opção para a saída do usuário; b) Além disso, o ícone da caixa se refere a caixas de diálogo de ajuda e não de algum aviso.



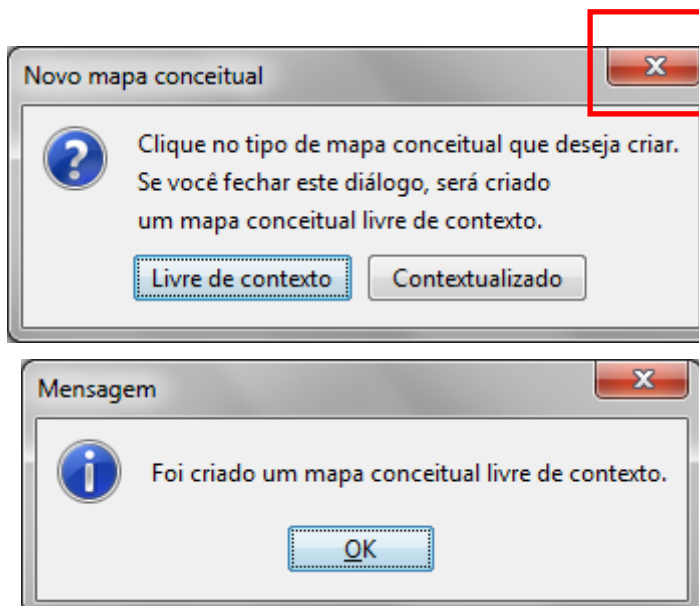
**Princípio(s) de usabilidade(s) infringido(s):** Consistência e padrões, controle e liberdade do usuário, flexibilidade e eficiência de uso.

**Grau de severidade:** 4

**Sugestões:** Criar um botão para cancelar. Alterar o ícone da caixa de diálogo. Talvez se possa integrar a opção na própria página do sistema.

### Quadro A.12. Violação de usabilidade no diálogo do ambiente

**Problema:** a) Ao se clicar no ícone “fechar” da janela de dialogo entre um mapa “Contextualizado” ou “Livre de contexto” ele apresentou a mensagem de que criou um mapa “Livre de contexto”, o que não era a intenção.



**Princípio(s) de usabilidade(s) infringido(s):** Reconhecimento ao invés de lembrança.

**Grau de severidade:** 1

**Sugestões:** O “fechar” não crie nada, apenas feche o dialogo cancelando a operação.



## ANEXO B – RELATÓRIO DA AVALIAÇÃO POR CHECKLIST DO WEBCMTOOL

**AVALIADOR(ES):**

**GABRIEL SOUZA, GLAUBER MONTEIRO E RODRIGO BARBALHO**

### 1. Software avaliado

Ambiente *WebCMTool* para avaliação de mapas conceituais

### 2. Público

Professores e alunos em geral

### 3. Laudo final pela ferramenta *ergolist*

O Quadro B.1 informa o total de questões disponíveis pela ferramenta, o total de questões que foram respondidas, o total que não foram respondidas, e a quantidade de respostas que obtiveram sim, não, não foram aplicáveis e eram adiadas.

**Quadro B.1. Informações do laudo final segundo a *ergolist***

<b>Total de questões: 194</b>			
<b>Total de questões respondidas: 190</b>			
<b>Total de questões não respondidas: 4</b>			
<b>Sim</b>	<b>Não</b>	<b>Não aplicáveis</b>	<b>Adiadas</b>
50	30	50	60

A Figura B.1 apresenta exemplos de questões que obtiveram respostas sim pelos três avaliadores no *ergolist*. Vale ressaltar segundo os avaliadores, que havia algumas questões na ferramenta que não se aplicava ao *WebCMTool* e não foram respondidas neste caso foram no total 50 questões, outras porém não foram respondidas sendo consideradas adiadas, seja por exemplo por questões de dúvida entre os avaliadores.

Por se tratar de várias relações de perguntas são apresentados somente alguns exemplos para observação, omitindo-se, portanto as demais avaliadas, no mais utilizadas para efeito de correção.

**Figura B.1. Questionários com repostas positivas sobre o critério significados**

**CheckList** 17 de 18  
**Critério: Significados**  
 Responda as questões abaixo valendo-se das definições do glossário e das informações adicionais.  
 Para evitar a perda de suas respostas, tenha o cuidado de completar um checklist antes de começar outro ou de voltar para a homepage.

---

Questão 1 de 12 Significados

As denominações dos títulos estão de acordo com o que eles representam? ▶

Sim  Não  Não aplicável  Adiar resposta

Comentários:

---

Questão 2 de 12 Significados

Os títulos das páginas de menu  são explicativos, refletindo a natureza da escolha a ser feita? ▶

Sim  Não  Não aplicável  Adiar resposta

Comentários:

Fonte: Ergolist, 2012

Algumas perguntas relacionadas ao critério Significados estão definidas na Figura B.2:

**Figura B.2. Perguntas segundo o critério significados**

<input type="checkbox"/> Questão 1 de 12	Significados
As denominações dos títulos estão de acordo com o que eles representam?	
<input type="checkbox"/> Questão 2 de 12	Significados
Os títulos das páginas de menu <input type="checkbox"/> são explicativos, refletindo a natureza da escolha a ser feita?	
<input type="checkbox"/> Questão 3 de 12	Significados
Os títulos das páginas de menus são distintos entre si?	
<input type="checkbox"/> Questão 4 de 12	Significados
Os títulos das páginas de menus são combináveis ou componíveis?	
<input type="checkbox"/> Questão 5 de 12	Significados
As denominações das opções de menu <input type="checkbox"/> são familiares ao usuário?	
<input type="checkbox"/> Questão 6 de 12	Significados
O vocabulário utilizado nos rótulos <input type="checkbox"/> , convites e mensagens de orientação são familiares ao usuário, evitando palavras difíceis?	
<input type="checkbox"/> Questão 7 de 12	Significados
O vocabulário utilizado em rótulos, convites e mensagens de orientação é orientado à tarefa, utilizando termos e jargão técnico normalmente empregados na tarefa <input type="checkbox"/> ?	
<input type="checkbox"/> Questão 8 de 12	Significados
Os cabeçalhos <input type="checkbox"/> de colunas de dados <input type="checkbox"/> são significativos e distintos?	
<input type="checkbox"/> Questão 9 de 12	Significados
O sistema <input type="checkbox"/> adota códigos significativos ou familiares aos usuários?	

A Figura B.3 apresenta exemplos de questões no critério de *feedback* que foram respondidas com não, uma com um comentário particular.

**Figura B.3. Questionários com repostas negativas sobre o critério *feedback***

**CheckList** 4 de 18  
**Critério: *Feedback***  
 Responda as questões abaixo valendo-se das definições do glossário e das informações adicionais.  
 Para evitar a perda de suas respostas, tenha o cuidado de completar um checklist antes de começar outro ou de voltar para a homepage.

---

Questão 1 de 12 *Feedback*

O sistema  fornece *feedback*  para todas as ações do usuário? ▶

Sim  Não  Não aplicável  Adiar resposta

Comentários:

1) Existem algumas interações que não há feedback, principalmente quando demorada uma ação como na edição de uma frase de ligação;  
 2) Na avaliação mapas grandes demoram mais que 12 segundos para serem executados, o que torna-se um problema na web.

---

Questão 2 de 12 *Feedback*

Quando, durante a entrada de dados, o sistema torna-se indisponível ao usuário, devido a algum processamento longo, o usuário é avisado desse estado do sistema e do tempo dessa indisponibilidade? ▶

Sim  Não  Não aplicável  Adiar resposta

Comentários:

Fonte: Ergolist, 2012

A seguir, as Figuras B.4 a B10 apresentam as perguntas e seus respectivos critérios.

**Figura B.4. Questionários com perguntas sobre o critério compatibilidade**

**Questões** 18 de 18  
**Critério: Compatibilidade**

---

Questão 1 de 21 Compatibilidade

As telas  são compatíveis com o padrão do ambiente?

---

Questão 2 de 21 Compatibilidade

A imagem do formulário  na tela do terminal assemelha-se com o formulário de entrada em papel?

---

Questão 3 de 21 Compatibilidade

O sistema propõe uma caixa de diálogo  modal , quando a aplicação deve ter todos os dados antes de prosseguir ou quando o usuário tenha de responder a uma questão urgente?

---

Questão 4 de 21 Compatibilidade

As caixas de diálogo do sistema  apresentam um botão  de validação, um botão de anulação e, se possível, um botão de ajuda?

---

Questão 5 de 21 Compatibilidade

Os significados usuais das cores são respeitados nos códigos de cores definidos?

---

Questão 6 de 21 Compatibilidade

As opções de codificação por cores são limitadas em número?

---

Questão 7 de 21 Compatibilidade

As informações  codificadas através das cores apresentam uma codificação adicional redundante?

---

Questão 8 de 21 Compatibilidade

A taxa de intermitência para elementos piscantes está entre 2 e 5 Hz (2 a 5 piscadas por segundo)?

**Figura B.5. Questionários com perguntas sobre o critério consistência**

**Questões** 16 de 18  
**Critério: Consistência**

---

Questão 1 de 11 Consistência  
A identificação das caixas , telas  ou janelas  são únicas?

---

Questão 2 de 11 Consistência  
A organização em termos da localização das várias características das janelas é mantida consistente de uma tela para outra?

---

Questão 3 de 11 Consistência  
A posição inicial do cursor  é mantida consistente ao longo de todas as apresentações de formulários .

---

Questão 4 de 11 Consistência  
Uma mesma tecla de função  aciona a mesma opção de uma tela para outra?

---

Questão 5 de 11 Consistência  
Os ícones  são distintos uns dos outros e possuem sempre o mesmo significado de uma tela para outra?

---

Questão 6 de 11 Consistência  
A localização dos dados  é mantida consistente de uma tela para outra?

---

Questão 7 de 11 Consistência  
Os formatos de apresentação dos dados são mantidos consistentes de uma tela para outra?

---

Questão 8 de 11 Consistência  
Os rótulos  estão na mesma posição em relação aos campos  associados?


**Figura B.6. Questionários com perguntas sobre o critério correção de erros**

**Questões** 15 de 18  
**Critério: Correção de erros**


---

Questão 1 de 5 Correção de erros  
Qualquer ação do usuário pode ser revertida através da opção DESFAZER?

---

Questão 2 de 5 Correção de erros  
Através da opção REFAZER, a regressão do diálogo , também pode ser desfeita?


---

Questão 3 de 5 Correção de erros  
Os comandos  para DESFAZER e REFAZER o diálogo estão diferenciados?

---

Questão 4 de 5 Correção de erros  
O sistema reconhece e através de uma confirmação do usuário, executa os comandos mais frequentes mesmo com erros de ortografia?

---

Questão 5 de 5 Correção de erros  
Depois de um erro de digitação de um comando ou de dados , o usuário tem a possibilidade de corrigir somente a parte dos dados ou o comando que está errada?

**Figura B.7. Questionários com perguntas sobre o critério mensagens de erro**

**Questões** 14 de 18  
**Critério: Mensagens de erro**

---

Questão 1 de 9 Mensagens de erro  
 As mensagens de erro ajudam a resolver o problema do usuário, fornecendo com precisão o local e a causa específica ou provável do erro, bem como as ações que o usuário poderia realizar para corrigi-lo?

---

Questão 2 de 9 Mensagens de erro  
 As mensagens de erro são neutras e polidas?

---

Questão 3 de 9 Mensagens de erro  
 As frases das mensagens de erro são curtas e construídas a partir de palavras curtas, significativas e de uso comum?

---

Questão 4 de 9 Mensagens de erro  
 As mensagens de erro estão isentas de abreviaturas e/ou códigos gerados pelo sistema operacional?

---

Questão 5 de 9 Mensagens de erro  
 O usuário pode escolher o nível de detalhe das mensagens de erro em função de seu nível de conhecimento?

---

Questão 6 de 9 Mensagens de erro  
 A informação principal de uma mensagem de erro encontra-se logo no início da mensagem?

---

Questão 7 de 9 Mensagens de erro  
 Quando necessário, as informações que o usuário deve memorizar encontram-se localizadas na parte final da mensagem de erro?

---

Questão 8 de 9 Mensagens de erro  
 Em situações normais as mensagens de erro são escritas em tipografia mista?

**Figura B.8. Questionários com perguntas sobre o critério proteção contra erros**

**Questões** 13 de 18  
**Critério: Proteção contra erros**

---

Questão 1 de 7 Proteção contra erros  
 O sistema apresenta uma separação adequada entre áreas selecionáveis de um painel de menu de modo a minimizar as ativações acidentais?

---

Questão 2 de 7 Proteção contra erros  
 Em toda ação destrutiva, os botões selecionados por *default* realizam a anulação dessa ação?

---

Questão 3 de 7 Proteção contra erros  
 Os campos numéricos para entrada de dados longos estão subdivididos em grupos menores e pontuados com espaços, vírgulas, hífens ou barras?

---

Questão 4 de 7 Proteção contra erros  
 Ao final de uma sessão de trabalho o sistema informa sobre o risco de perda os dados?

---

Questão 5 de 7 Proteção contra erros  
 O sistema emite sinais sonoros quando ocorrem problemas na entrada de dados?

---

Questão 6 de 7 Proteção contra erros  
 As teclas de funções perigosas encontram-se agrupadas e/ou separadas das demais no teclado?

---

Questão 7 de 7 Proteção contra erros  
 O sistema solicita confirmação (dupla) de ações que podem gerar perdas de dados e/ou resultados catastróficos?

**Figura B.9. Questionários com perguntas sobre o critério experiência do usuário**

**Questões** 12 de 18  
**Critério: Experiência do usuário**

---

Questão 1 de 6 Experiência do usuário

Caso se trate de um sistema de grande público, ele oferece formas variadas de apresentar as mesmas informações aos diferentes tipos de usuário?

---

Questão 2 de 6 Experiência do usuário

Os estilos de diálogo são compatíveis com as habilidades do usuário, permitindo ações passo-a-passo para iniciantes e a entrada de comandos mais complexos por usuários experimentados?

---

Questão 3 de 6 Experiência do usuário

O usuário pode se deslocar de uma parte da estrutura de menu para outra rapidamente?

---

Questão 4 de 6 Experiência do usuário

O sistema oferece equivalentes de teclado para a seleção e execução das opções de menu, além do dispositivo de apontamento (mouse,...)?

---

Questão 5 de 6 Experiência do usuário

O sistema é capaz de reconhecer um conjunto de sinônimos para os termos básicos definidos na linguagem de comando, isto para se adaptar aos usuários novatos ou ocasionais?

---

Questão 6 de 6 Experiência do usuário

O usuário experiente pode efetuar a digitação de vários comandos antes de uma confirmação?

**Figura B.10. Questionários com perguntas sobre o critério experiência do usuário**

**Questões** 10 de 18  
**Critério: Controle do usuário**

---

Questão 1 de 4 Controle do usuário

O usuário pode terminar um diálogo seqüencial repetitivo a qualquer instante?

---

Questão 2 de 4 Controle do usuário

O usuário pode interromper e retomar um diálogo seqüencial a qualquer instante?

---

Questão 3 de 4 Controle do usuário

O usuário pode reiniciar um diálogo seqüencial a qualquer instante?

---

Questão 4 de 4 Controle do usuário

Durante os períodos de bloqueio dos dispositivos de entrada, o sistema fornece ao usuário uma opção para interromper o processo que causou o bloqueio?