

SAW - Sistema de Aprendizagem pela Web: Motivações e Desenvolvimento

Janine Gomes Moura

DISSERTAÇÃO APRESENTADA AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DO TÍTULO DE MESTRE
EM
CIÊNCIAS

Área de Concentração: Ciência da Computação

Orientador: Prof. Dr. Leônidas de Oliveira Brandão

São Paulo, Janeiro de 2007.

Aos meus pais
Eugênio e Rita
e ao meu noivo
Wellington,
com amor.

Agradecimentos

Em primeiro lugar a *Deus*, que permitiu que eu chegasse até aqui.

Ao professor *Leônidas de Oliveira Brandão*, pela paciência, confiança e dedicação ao longo do programa de mestrado.

Ao professor *Alfredo Goldman* pela confiança e incentivo desde a graduação.

A professora *Cristina Cerri* pelas contribuições e sugestões para o aperfeiçoamento deste trabalho.

A *Anarosa*, pela amizade, ajuda e sugestões sempre que precisei.

Ao *Seiji*, pela amizade e ajuda neste trabalho e em todo o programa de mestrado.

Aos meus pais, *Eugenio* e *Rita*, e a minha irmã *Larissa*, por todo o esforço, carinho, dedicação e confiança que depositaram em mim.

Ao meu noivo *Wellington*, pelo apoio, paciência, incentivo e compreensão.

A *Escola Lourenço Castanho* e aos professores *Júlio* e *Paulo*, que colaboram utilizando o *SAW* em suas disciplinas.

As minha amigas, *Janice*, *Carol*, *Aline* e *Roberta* pela excelente companhia que me proporcionam desde o início da graduação até os dias de hoje.

A todos que direta ou indiretamente me ajudaram a chegar até aqui.

Resumo

Neste trabalho, apresentamos um sistema gerenciador de cursos pela Web, o Sistema de Aprendizagem pela Web (*SAW*), que começamos a desenvolver no final de 2003, e algumas experiências didáticas importantes em seu desenvolvimento.

O desenvolvimento do *SAW* teve como objetivos ajudar no aprendizado do aluno e auxiliar o professor na produção de conteúdos e no acompanhamento dos alunos. Esse sistema é gratuito (e aberto) e sua principal característica é permitir a incorporação de módulos de aprendizagem específicos para determinados conteúdos. O *SAW* permite a inserção de recursos e pode ser utilizado em diferentes domínios, com diferentes módulos de aprendizagem (applet Java).

Esse sistema dispõe atualmente de um módulo para aprendizagem de geometria e um outro para ensino de programação.

A navegação nos cursos cadastrados no *SAW*, é baseada na resolução de exercícios e na autoria de textos por parte do aluno.

Abstract

In this study we present a learning management system, the *SAW* - Sistema de Aprendizagem pela Web (Web Learning System) and some important experiences in its development.

The *SAW* development was initiated by us at the end of 2003, aiming to help the student's learn, the teacher to produce contents and to follow the students progress. The *SAW* is an open source software, distributed under GNU licence. The main characteristic of this system is to allow the incorporation of learning modules specific for certain contents. It permits resources to be inserted and can be used in different domains with different learning modules (*applet* Java).

At present, this system has been used with a module for geometry learning and another used to teach programming.

The navigation on the courses registered at *SAW* is based on the resolution of exercises and on texts written by students.

Sumário

Resumo	v
Abstract	vii
Lista de Figuras	xii
1 Introdução	1
1.1 Caracterização do problema	1
1.2 Justificativa	2
1.3 Objetivos	3
1.4 Conteúdo da dissertação	4
2 Sistemas de Gerenciamento de Cursos na Web	5
2.1 Funcionalidades	8
2.1.1 Fórum de discussão	8
2.1.2 Bate-papo (chat)	9
2.1.3 Disponibilização de conteúdo	9
2.1.4 Avaliação online	10
2.2 Sistemas Gerenciados de Cursos	12
2.2.1 TelEduc	12
2.2.2 Moodle	13

2.2.3	AulaNet	15
2.2.4	BlackBoard/WebCT	16
3	SAW - Sistema de Aprendizagem pela Web	19
3.1	A arquitetura	22
3.1.1	Camada de dados	22
3.1.2	Camada de Tarefas	24
3.1.3	Camada de Comunicação	26
3.1.4	Camada de Interação	27
3.1.5	Módulos de aprendizagem	27
3.2	Estrutura	31
3.3	Funcionalidades	33
3.3.1	Editor de textos MaRTE	34
3.3.2	Dicionário Webmídia	36
3.3.3	Fórum de discussão	39
3.3.4	Painel de Avisos e Recados	41
3.3.5	Trabalhos	42
3.3.6	Área para download	43
3.3.7	Relatórios	43
3.3.8	Lista de presença	44
3.3.9	Provas Presenciais Online	45
3.3.10	Notas	45
3.4	Os usuários	46
3.4.1	Administrador	46
3.4.2	Professor	47
3.4.3	Monitor	49

3.4.4	Aluno	49
4	Estudos de caso	53
4.1	Escola Lourenço Castanho	54
4.2	Cursos do LEM - Laboratório de Ensino de Matemática	57
4.3	Construção de Curso de Geometria à Distância	57
4.4	MAC110	58
4.5	MAC118	58
4.5.1	Análise do SAW	59
5	Conclusões	65
5.1	Trabalhos Futuros	67
A	Arquitetura	69
	Apêndice	68
B	Manual do Professor	73
B.1	Primeiro passo	73
B.1.1	Cadastro de disciplina	73
B.2	Módulo Inserir	74
B.2.1	Usuário	74
B.2.2	Componentes	76
B.2.3	Texto - Editor MaRTE	77
B.2.4	Exercício	79
B.2.5	Exemplo	81
B.2.6	Tópico	81
B.2.7	Aula	82

B.2.8	Nota	84
B.2.9	Arquivo	85
B.2.10	Dicionário Webmídia	85
B.2.11	Painel de Avisos e Recados	86
B.2.12	Trabalhos	87
B.2.13	Menu do aluno	87
B.2.14	Liberar componentes	88
C	Tabelas do Banco de Dados	93
D	Questionário para o estudo de caso em MAC118 - PRÉ-TESTE	117
E	Questionário para o estudo de caso em MAC118 - PÓS-TESTE	119
	Bibliografia	121
	Índice Remissivo	131

Lista de Figuras

2.1	TelEduc.	12
2.2	Moodle	14
2.3	AulaNet.	15
2.4	WebCT.	17
3.1	Tentativa de demonstração de um aluno	21
3.2	Arquitetura do SAW.	23
3.3	Construção no programa de Geometria Dinâmica, iGeom	29
3.4	Tela do programa emulador/compilador iCG	30
3.5	Tela do programa de visualização gráfica iGraf	31
3.6	Relação entre os componentes	32
3.7	Possíveis cenários	32
3.8	Exemplo de componentes em um curso	33
3.9	Primeira versão do editor de hipertexto MaTE	35
3.10	Inserção de imagens no MaRTE	37
3.11	Versão atual do editor de textos - MaRTE	38
3.12	Estrutura do dicionário webmídia	39
3.13	Exemplo de criação de um dicionário	40
3.14	Exemplo de um vocábulo	41
3.15	Exemplo de uma publicação do dicionário.	42

3.16	Exemplo do fórum phpBB integrado ao <i>SAW</i>	43
3.17	Exemplo da funcionalidade Painel de Avisos e Recados	44
3.18	Exemplo da funcionalidade Trabalhos	45
3.19	Exemplo da funcionalidade envio de arquivo	46
3.20	Exemplo de um relatório por turma	47
3.21	Exemplo de um relatório por aluno	48
3.22	Resolução de um exercício no SAW utilizando o <i>MA iGeom</i>	50
3.23	Tela inicial do SAW para o usuário aluno.	51
3.24	Possível resultado da avaliação de um exercício com o <i>MA iGeom</i>	52
4.1	Exemplo de resolução do exercício 2 no iGeom	56
4.2	Análise de uso: inserção de applets e fórmulas	62
4.3	Comparação: médias de exercícios e provas	63
4.4	Utilização relativa de ferramentas na edição da atividade final	64
B.1	Interface inicial do <i>SAW</i>	73
B.2	Interface com o menu do professor	74
B.3	Interface de inserção de usuários	75
B.4	Interface de inserção de alunos por grupo	76
B.5	Interface de inserção de componentes	76
B.6	Interface de inserção de textos	77
B.7	Editor MaRTE, recursos de alinhamento	77
B.8	Editor MaRTE, recursos de formatação	78
B.9	Editor MaRTE, inserir imagem	78
B.10	Editor MaRTE, inserir imagem interativa	79
B.11	Editor MaRTE, inserir bibliografia	80

B.12 Editor MaRTE, inserir citação	81
B.13 Editor MaRTE, inserir expressão matemática	82
B.14 Interface de inserção de exercícios	83
B.15 Inserção de exercício por arquivo	84
B.16 Inserção de exercício online	85
B.17 Construção do gabarito do exercício	86
B.18 Interface de inserção de exemplo.	86
B.19 Interface de inserção de tópico	87
B.20 Interface de inserção de tópico com pré-requisitos	88
B.21 Interface de inserção de aula	89
B.22 Interface de inserção de nota por aula	89
B.23 Interface de inserção de arquivos	90
B.24 Interface de inserção de dicionário webmídia	90
B.25 Exemplo da funcionalidade Painel de Avisos e Recados	90
B.26 Exemplo da funcionalidade Painel de Avisos e Recados	91
B.27 Exemplo da funcionalidade Painel de Avisos e Recados	91
B.28 Interface de liberação de funcionalidades para o aluno	92

Lista de Tabelas

2.1	Alguns <i>SGC</i> gratuitos	7
2.2	Recursos de alguns dos programas de GD	17
3.1	Recursos dos <i>SGC</i> apresentados	34
A.1	Relação de páginas com a arquitetura na <i>Camada de Tarefas</i>	69
A.2	Relação de páginas com a arquitetura na <i>Camada de Comunicação</i>	70
A.3	Relação de páginas com a arquitetura na <i>Camada de Interação</i>	71
C.1	Estrutura da Tabela saw_admin_admin	93
C.2	Estrutura da Tabela saw_admin_applet	93
C.3	Estrutura da Tabela saw_admin_curso	93
C.3	Estrutura da Tabela saw_admin_curso (continued)	94
C.4	Estrutura da Tabela saw_admin_saw	94
C.5	Estrutura da Tabela saw_curso_aluno_aula	94
C.6	Estrutura da Tabela saw_curso_aluno_curso	94
C.7	Estrutura da Tabela saw_curso_aluno_exercicio	95
C.8	Estrutura da Tabela saw_curso_aluno_nota	95
C.9	Estrutura da Tabela saw_curso_aluno_presenca	95
C.10	Estrutura da Tabela saw_curso_aluno_presenca_prof	95
C.11	Estrutura da Tabela saw_curso_aluno_topico	96

C.12	Estrutura da Tabela saw_curso_aluno_trabalho	96
C.13	Estrutura da Tabela saw_curso_arquivo	96
C.14	Estrutura da Tabela saw_curso_aula_turma	96
C.14	Estrutura da Tabela saw_curso_aula_turma (continued)	97
C.15	Estrutura da Tabela saw_curso_aula_turma_aluno	97
C.16	Estrutura da Tabela saw_curso_data_aula	97
C.17	Estrutura da Tabela saw_curso_data_prof	97
C.18	Estrutura da Tabela saw_curso_dic_comentario	97
C.18	Estrutura da Tabela saw_curso_dic_comentario (continued)	98
C.19	Estrutura da Tabela saw_curso_dic_dicionario	98
C.20	Estrutura da Tabela saw_curso_dic_publicacao	98
C.21	Estrutura da Tabela saw_curso_dic_topico	98
C.22	Estrutura da Tabela saw_curso_dic_vocabulo	99
C.23	Estrutura da Tabela saw_curso_menu_aluno	99
C.24	Estrutura da Tabela saw_curso_notas	99
C.25	Estrutura da Tabela saw_curso_notas_curso	99
C.25	Estrutura da Tabela saw_curso_notas_curso (continued)	100
C.26	Estrutura da Tabela saw_curso_recado_curso	100
C.27	Estrutura da Tabela saw_curso_recado_turma	100
C.28	Estrutura da Tabela saw_curso_recado_usuario	100
C.29	Estrutura da Tabela saw_curso_trabalho	101
C.30	Estrutura da Tabela saw_curso_trabalho_turma	101
C.31	Estrutura da Tabela saw_curso_turma	101
C.32	Estrutura da Tabela saw_user_applets	101
C.32	Estrutura da Tabela saw_user_applets (continued)	102

C.33 Estrutura da Tabela saw_user_aula	102
C.34 Estrutura da Tabela saw_user_aula_componente	102
C.35 Estrutura da Tabela saw_user_aula_pre_requisito	102
C.36 Estrutura da Tabela saw_user_bibliografia	102
C.36 Estrutura da Tabela saw_user_bibliografia (continued)	103
C.37 Estrutura da Tabela saw_user_exemplo	103
C.38 Estrutura da Tabela saw_user_exercicio	103
C.39 Estrutura da Tabela saw_user_formulas	104
C.40 Estrutura da Tabela saw_user_menu_aluno	104
C.41 Estrutura da Tabela saw_user_texto	104
C.42 Estrutura da Tabela saw_user_topico	104
C.43 Estrutura da Tabela saw_user_topico_componente	105
C.44 Estrutura da Tabela saw_user_usuario	105
C.45 Estrutura da Tabela saw_user_usuario_criptografado	105
C.46 Estrutura da Tabela saw_user_usuario_curso	105
C.46 Estrutura da Tabela saw_user_usuario_curso (continued)	106
C.47 Estrutura da tabela saw_phpbb_auth_access	106
C.48 Estrutura da tabela saw_phpbb_banlist	106
C.49 Estrutura da tabela saw_phpbb_categories	106
C.50 Estrutura da tabela saw_phpbb_config	107
C.51 Estrutura da tabela saw_phpbb_confirm	107
C.52 Estrutura da tabela saw_phpbb_disallow	107
C.53 Estrutura da tabela saw_phpbb_formula_forum	107
C.54 Estrutura da tabela saw_phpbb_forum_prune	107
C.55 Estrutura da tabela saw_phpbb_forums	107

C.55 Estrutura da tabela saw_phpbb_forums (continued)	108
C.56 Estrutura da tabela saw_phpbb_groups	108
C.57 Estrutura da tabela saw_phpbb_posts	108
C.57 Estrutura da tabela saw_phpbb_posts (continued)	109
C.58 Estrutura da tabela saw_phpbb_posts_text	109
C.59 Estrutura da tabela saw_phpbb_privmsgs	109
C.60 Estrutura da tabela saw_phpbb_privmsgs_text	110
C.61 Estrutura da tabela saw_phpbb_ranks	110
C.62 Estrutura da tabela saw_phpbb_search_results	110
C.63 Estrutura da tabela saw_phpbb_search_wordlist	110
C.64 Estrutura da tabela saw_phpbb_search_wordmatch	110
C.65 Estrutura da tabela saw_phpbb_sessions	111
C.66 Estrutura da tabela saw_phpbb_smilies	111
C.67 Estrutura da tabela saw_phpbb_themes	111
C.67 Estrutura da tabela saw_phpbb_themes (continued)	112
C.68 Estrutura da tabela saw_phpbb_themes_name	112
C.68 Estrutura da tabela saw_phpbb_themes_name (continued)	113
C.69 Estrutura da tabela saw_phpbb_topics	113
C.70 Estrutura da tabela saw_phpbb_topics_watch	113
C.70 Estrutura da tabela saw_phpbb_topics_watch (continued)	114
C.71 Estrutura da tabela saw_phpbb_user_group	114
C.72 Estrutura da tabela saw_phpbb_users	114
C.72 Estrutura da tabela saw_phpbb_users (continued)	115
C.73 Estrutura da tabela saw_phpbb_vote_desc	115
C.74 Estrutura da tabela saw_phpbb_vote_results	115

C.75 Estrutura da tabela saw_phpbb_vote_voters	116
C.76 Estrutura da tabela saw_phpbb_words	116

Capítulo 1

Introdução

1.1 Caracterização do problema

Atualmente, tanto no Brasil quanto em outras partes do mundo, ocorre uma grande expansão do uso da *Web* no ensino/aprendizagem e o número de pesquisas nesta área têm aumentado consideravelmente, como demonstra (Litto et al., 2004). Esta forma de utilização possibilita a disponibilização de um grande volume de informações a partir de material *online*. Além disso, permite que o aprendiz siga seu próprio ritmo de aprendizado, acessando o conteúdo disponível a hora que for conveniente pelo tempo que for necessário.

A utilização da *Web* como suporte a cursos pode ser separada em duas formas de uso: *aberto*, na qual qualquer pessoa com acesso à *Web* consegue visualizar o conteúdo; *fechado*, no qual um sistema faz o controle de acesso dos usuários e do conteúdo.

Sistemas que controlam o acesso podem permitir adaptações do mesmo em resposta as ações de cada aluno, resultando numa individualização que não é possível com a mera disponibilização de conteúdos de forma aberta. Uma característica essencial para possibilitar a individualização é a interatividade do ambiente com o aluno.

Muitos sistemas com estas características surgiram até o presente momento como apresenta Brusilovsky (2000), porém a maioria deles não possuem recursos especializados para o aprendizado de conteúdos específicos, como a geometria. Por exemplo, para incorporar uma construção de *Geometria Dinâmica* (GD) (Gravina, 1996) é necessário utilizar um programa de GD, externo ao sistema, e depois inserir no sistema o arquivo com a construção (*up-load*). Entretanto, muitos destes sistemas adaptativos restringem a incorporação de novos recursos, dificultando o uso de novas ferramentas educacionais.

Outra deficiência relacionada é a dificuldade de incorporar ao sistema a autoria e a avaliação automática de exercícios abertos (que não sejam do tipo múltipla-escolha).

Estas deficiências nos motivaram a desenvolver um novo sistema para gerenciamento de cursos via Web, no qual fosse possível trabalhar com conteúdos específicos e incorporar de forma simples novos recursos. Denominamos este novo sistema por *Sistema de Aprendizagem pela Web (SAW)*, e sua atual versão está disponível a partir do endereço Web

<http://milanesa.ime.usp.br/saw2/>.

1.2 Justificativa

Existem várias denominações para descrever sistemas que controlam acesso a conteúdos de cursos disponíveis na Web, como *Learning Management System (LMS)*, *Content Management System (CMS)* e *Learning Content Management System (LCMS)*. Adotaremos aqui a denominação *Sistema de Gerenciamento de Cursos (SGC)* para designar qualquer sistema que possa ser utilizado para publicação de material na Web e que disponha de autenticação de usuários.

Um requisito interessante para tais *SGC* é que eles sejam flexíveis, no sentido de permitir incorporação de novos recursos, e um outro requisito interessante é que sejam adaptáveis em resposta às interações com o usuário. Existem vários sistemas com características de adaptabilidade (Brusilovsky, 1996) e outros mais genéricos como *Teleduc* (Rocha, 2002), *Moodle* (Moodle, 2004) e *AulaNet* (Fuks et al., 2004).

Grande parte dos *SGC* dispõem de vários recursos genéricos, que podem ser empregados em quaisquer cursos via *Web*, como *fórum de discussão*, *área para bate-papo (chat)* e ferramentas para avaliações do tipo “*múltipla-escolha*”. Entretanto, nestes sistemas a inserção de módulos específicos para aprendizado de conteúdos, que denotaremos aqui por *Módulos de Aprendizagem (MA)*, pode ser uma tarefa complicada, dependendo do tipo de módulo e do sistema considerado é até mesmo inviável. Este é o caso dos sistemas comerciais de código “fechado”, que não permitem alterações de programação.

Estes módulos de aprendizagem podem incrementar a interatividade do sistema, ajudando os alunos na aprendizagem e facilitando algumas tarefas dos professores.

No caso específico da aprendizagem relativa à geometria, as avaliações do tipo “*múltipla-escolha*” não são tão interessante quanto permitir que o aluno realize uma construção. Para isso seria conveniente incorporar ao *SGC* um módulo de aprendizagem para geometria que

possibilitasse construções.

Além disso, como as aplicações de cursos pela Web podem atender um grande número de alunos, seria muito útil que o MA tivesse um recurso de avaliação automática de exercícios (e também um recurso para autoria de exercícios para o professor).

O *Sistema de Aprendizagem pela Web (SAW)* começou a ser desenvolvido no final de 2003, visando suprir as citadas características de flexibilidade e adaptabilidade. O SAW é um SGC gratuito e tem como principal característica a possibilidade de incorporação de recursos interativos específicos para aprendizagem, os Módulos de Aprendizagem (*MAs*). No SAW, um *MA* é um *applet Java* (Thomas et al., 1997), com características que permitam seu uso na aprendizagem de determinados conteúdos, e que disponha de recurso para comunicação com um servidor Web.

O *MA* usado no primeiro estudo de caso realizado com o SAW foi a versão *applet* do sistema *iGeom - Geometria Interativa na Internet* (Isotani and de Oliveira Brandão, 2003). O *iGeom* é um sistema de GD, como o *GSP* (Jackiw, 1995) e o *Cabri* (Laborde and Bellemain, 1997), implementado em *Java* e gratuitamente disponível no endereço <http://www.matematica.br/igeom>.

A possibilidade de incorporação de *MAs* ao SAW, especialmente aqueles destinados ao ensino/aprendizagem de conteúdo matemático, ainda estimulou-nos a implementar recursos de edição e de construção de conteúdo mais sofisticados que os comumente encontrados em sistemas de gerenciamento de cursos, dadas as especificidades inerentes ao texto matemático.

1.3 Objetivos

O objetivo deste trabalho é elaborar e testar um sistema de aprendizagem que contribua para o aprendizado do aluno e auxilie o professor no desenvolvimento de conteúdo e no acompanhamento dos alunos. Para tanto estabelecemos os seguintes requisitos:

- o sistema deve permitir fácil acoplamento de *módulos de aprendizagem*, por exemplo, para o ensino de geometria acoplar um programa que permita construções geométricas.
- o sistema deve permitir compartilhamento de componentes, isto é, permitir que os professores possam reutilizar aulas e exercícios presentes no sistema;
- o sistema deve possibilitar a inclusão de textos matemáticos, tanto para o aluno quanto para o professor;

- o sistema deve permitir que o professor tenha acesso rápido às atividades e trabalhos realizados pelos alunos.

1.4 Conteúdo da dissertação

Esta dissertação é composta de 5 capítulos, incluindo esta introdução.

No capítulo 2 mostramos as principais funcionalidades encontradas na maioria dos *SGC* e uma introdução a alguns sistemas disponíveis atualmente.

No capítulo 3 apresentamos o *SAW* - Sistema de Aprendizagem pela Web, descrevendo como foi seu desenvolvimento e suas principais funcionalidades.

No capítulo 4 descrevemos as aplicações do *SAW* em sala de aula e sua avaliação através de questionário em alguns casos de uso.

No capítulo 5 apresentamos as conclusões, as principais contribuições e recomendações para trabalhos futuros.

Capítulo 2

Sistemas de Gerenciamento de Cursos na Web

Os computadores e as tecnologias relativas à *Web* têm sido bastante utilizados no ensino, existindo iniciativas de várias naturezas no emprego destes recursos. Uma iniciativa é o desenvolvimento de sistemas para gerenciamento de cursos, que possuem controle de usuários e de disponibilização de conteúdos, alguns destes sistemas são: *AulaNet* (EduWeb, 2007), *Manhattan* (Manhattan, 2007), *Blackboard/WebCT* (bla, 2007; WebCT, 2007), *Moodle* (Moodle, 2007), *ClassWeb* (ClassWeb, 2007) e *TelEduc* (TelEduc, 2007).

Encontramos na literatura relatos de cursos presenciais que utilizam a Web para disponibilizar conteúdos de forma *aberta*, sem um sistema de gerenciamento (Basso et al., 1999). Com esta forma de publicação não é possível saber quando e quais alunos acessaram o conteúdo e além disso este fica disponível para qualquer usuário na Web. Com a introdução de um sistema gerenciador cada usuário cadastrado recebe uma senha para acesso, sendo possível controlar o conteúdo que cada um visualiza. Deste modo pode-se fazer um acompanhamento de cada usuário, personalizando a disponibilização dos conteúdos.

Existem várias denominações para descrever sistemas gerenciadores, geralmente distinguindo-os pelo tipo de suporte que oferecem ao processo de ensino/aprendizagem. De acordo com Nichani (2001); Kaplan-Leiserson (2006); Itmazi et al. (2005) os mais comuns são *LMS*, *CMS* e *LCMS*, explicados a seguir:

- *LMS - Learning Management System* (Sistema Gerenciador de Aprendizagem) são sistemas que auxiliam o gerenciamento de alunos. O gerenciamento é feito através do armazenamento, rastreamento e manipulação de dados dos alunos.

- *CMS - Content Management System* (Sistema Gerenciador de Conteúdo) são sistemas que provêm recursos a criação, gerenciamento e publicação de conteúdos de um curso.
- *LCMS - Learning Content Management System* (Sistema Gerenciador de Conteúdo e Aprendizagem) são sistemas que combinam a capacidade de gestão de cursos dos *LMS* com a capacidade de armazenamento e criação de conteúdos dos *CMS*.

Entretanto, ocorre uma certa confusão na diferenciação entre *LCMS*, *LMS* e *CMS*, como cita Ellis (2001). Isto pode dificultar a classificação dos sistemas gerenciadores existentes, pois muitos deles apresentam parte das funcionalidades para serem classificados como *LMS* ou *CMS*, mas possuem funcionalidades que não são encontrados em sistemas do tipo *LCMS*. Um exemplo desta confusão é o sistema Moodle, que é classificado como *LMS* por alguns autores (Depow, 2003; Kennedy, 2005) e também por *CMS* por outros autores (Cole, 2005).

Aqui adotamos o termo ***Sistema de Gerenciamento de Cursos - SGC*** para designar qualquer sistema que faça algum controle de usuários e de conteúdos específicos para cursos. Deste modo, os sistemas citados no início deste capítulo podem ser denominados *SGC*.

Independente da denominação, a busca por sistemas que ofereçam a disponibilização de conteúdo e o gerenciamento de usuário vem aumentando. Instituições e empresas estão adotando *SGC* para promover cursos a distância e presenciais devido às muitas vantagens que estes sistemas oferecem (Pereira and Bax, 2002).

Atualmente existe um grande número de *SGC*, sendo vários deles gratuitos e de código aberto (EdTechPost, 2007), como os citados na tabela 2.1. Entretanto, a disputa pelo mercado comercial de *SGC* está muito acirrada, como indica a compra do WebCT pela empresa dona do sistema BlackBoard e mais ainda pela tentativa desta última de conseguir um monopólio na área com o registro de uma patente de métodos e sistemas para *e-learning** (U.S. patent 6,988,138) (Inc., 2006; Patent and Office, 2006). A atitude da BlackBoard tem gerado várias manifestações contrárias (Pope, 2007; Balbino, 2006; boy, 2007; education patents, 2006), prometendo uma grande batalha jurídica.

A seguir descrevemos as principais funcionalidades disponíveis na maioria dos *SGC* e fazemos uma breve descrição daqueles que apareceram com mais frequência em artigos de congressos brasileiros.

*Internet-based education support system and methods - <http://patft.uspto.gov>

Tabela 2.1: Alguns SGC gratuitos

SGC	URL	Desenvolvedor	País
Adept	http://adept.sourceforge.net/		Russia
AulaNet	http://www.eduweb.com.br/	PUCSP	Brasil
ATutor	http://www.atutor.ca/	University of Toronto	Canada
Boddington	http://boddington.org/index.php	University of Leeds	U.K.
CHEF	http://chefproject.org/	University of Michigan	US
Claroline	http://www.claroline.net/	Université catholique de Louvain	France
ClassWeb	http://classweb.ucla.edu/	University of California Los Angeles	US
Coursework	http://aboutcoursework.stanford.edu/	Stanford University	US
eConf	http://econif.sourceforge.net/	University of Namur	Belgium
eLecture Online Lecturing System	http://physik.uni-graz.at/cbl/lecture/	Karl-Franzens-Universität Graz	Austria
Eledge	http://eledge.sourceforge.net/	University of Utah	US
e-Tutor	http://etutor.sourceforge.net/	University of Ottawa	Canada
File	http://fle3.uiah.fi/	University of Art and Design Helsinki	Finland
H2O Project	http://h2oproject.law.harvard.edu	Harvard Law School	US
ILE	http://virtualschool.edu/ile/	Brad Cox	US
Jones e-education	http://www.jonesknowledge.com	Jones Advisory Group	US
KEWL	http://kewl.uwc.ac.za/	University of Western Cape	South Africa
LearnLoop	http://learnloop.sourceforge.net/	ITUniversity in Gothenburg	Sweden
LON-CAPA	http://www.lon-capa.org/	Michigan State University	US
OLAT	http://www.olat.org	University of Zuerich	Switzerland
Manhattan Virtual Classroom	http://manhattan.sourceforge.net/	Western New England College	US
MnITS Internet Teaching System	http://sourceforge.net/projects/mnits	Daniel Bartholomew	US
Moodle	http://moodle.org/	Martin Dougiamas	Australia
OpenUSS	http://openuss.sourceforge.net/openuss	University of Münster	Germany
sTeam	http://www.open-steam.org/	University of Paderborn	Germany
TeLEduc	http://teleduc.nied.unicamp.br/teleduc/	UNICAMP	Brasil
TextWeaver	http://www.textweaver.org/	San Diego State University	US
WBT-Master	http://coronet.iicm.edu/	Graz University of Technology	Austria
WebWork	http://webwork.math.rochester.edu/	University of Rochester	US
Whiteboard	http://whiteboard.sourceforge.net/	Todd Templeton	US

2.1 Funcionalidades

Em geral, sistemas de apoio ao gerenciamento de cursos na Web são desenvolvidos de maneira que possam ser utilizados por usuários que não tenham domínio de computação. Tais sistemas possuem, pelo menos, três tipos de usuários básicos que desempenham os papéis de administrador, professor e aluno.

As funcionalidades mais comuns disponíveis na maioria dos *SGC* são: fórum de discussão, ferramenta de bate-papo (chat), ferramenta para disponibilização de conteúdo e ferramentas para avaliações *online*. A seguir faremos uma descrição de cada uma.

2.1.1 Fórum de discussão

No fórum um usuário pode a qualquer momento enviar uma mensagem (por exemplo, contendo uma dúvida) e outros usuários, incluindo o professor, poderão ler esta mensagem e respondê-la ou comentá-la a qualquer tempo. Como cita Cabeda (2005), com o fórum o usuário tem maior liberdade para escolher o horário de sua participação, bem como a profundidade de suas reflexões. A maioria dos fóruns existentes nos *SGC* possuem organização por assunto/tópico, o que facilita o acompanhamento das discussões. Esta funcionalidade pode auxiliar no ensino/aprendizado fornecendo dados sobre o desempenho do aluno, como observa Cabeda (2005) “*as mensagens de um fórum podem fornecer dados interessantes para a avaliação das práticas adotadas, do ambiente computacional utilizado e do comportamento de alunos, professores e outros agentes envolvidos em um curso*”.

Entretanto, como cita Cavaroli and Coello (2004) muitas vezes os usuários criam tópicos/assuntos desnecessários e, apesar de muitos fóruns possuírem sistema de busca, a maioria dos usuários não o utilizam para verificar se mensagem semelhante já foi enviada, ele simplesmente envia outra mensagem. Outro fator que se pode observar é que o número de mensagens publicadas ao longo do curso pode se tornar um problema, pois a procura pela informação necessária pode se tornar tediosa e desmotivar o uso da funcionalidade (Fussell et al., 1998). Segundo (Fuks et al., 2002) uma maneira de evitar isso é: “*a redução da confusão em relação às mensagens pode-se dar através da organização e estruturação da discussão, e do fornecimento de informações simples e representativas que ajudem os participantes a identificar a relevância e o contexto das mensagens*”.

2.1.2 Bate-papo (chat)

Permite que alunos e professores conduzam uma discussão eletrônica de modo síncrono, ou seja, dois ou mais usuários que estejam conectados simultaneamente no sistema podem conversar entre si. O uso desta funcionalidade em cursos é geralmente feito com horários agendados. Isto possibilita, principalmente para cursos totalmente a distância, que alunos esclareçam suas dúvidas com o professor, monitor ou com outros alunos do curso. Além disso, para Cabeda (2005), esta funcionalidade proporciona proximidade entre os usuários, o que contribui para que ocorra a colaboração entre alunos do curso.

Porém esta funcionalidade pode gerar confusão quando o número de debatedores é grande, como apontam alguns estudos (Cabeda, 2005; Pimentel et al., 2003; Otsuka et al., 2002). Pimentel et al. (2003) propõe a inclusão de um responsável para tentar evitar esta “confusão” em sessões de bate-papo, cujo papel principal é “*conduzir a sessão propondo questões para discussão, manter o foco na questão proposta evitando que a discussão se disperse ou tome rumos inadequados, cuidar para que o debate não ocorra num ritmo muito exagerado ou monótono, coordenar os outros aprendizes estimulando a participação de todos mantendo a ordem*”.

Outro problema apontado na literatura é a dificuldade de avaliar a participação efetiva do aluno em uma discussão via bate-papo (Otsuka et al., 2002). Apesar destas desvantagens, como cita Pimentel et al. (2003), a utilização desta funcionalidade pode ser uma alternativa na tentativa de repensar o modelo tradicional de ensino.

2.1.3 Disponibilização de conteúdo

Muitos *SGC* possuem funcionalidades específicas para a disponibilização de conteúdo para o aluno tais como: textos, imagens, gráficos, áudios e vídeo. Essas funcionalidades podem ser classificadas em dois tipos:

- Sem personalização: é a forma mais comum de disponibilização de arquivos, onde o aluno pode descarregar ou visualizar o arquivo no computador (eventualmente via *plug-in*).
- Com personalização: de acordo com o histórico do aluno no sistema é possível fazer a disponibilização de conteúdos de maneira personalizada para cada aluno.

2.1.4 Avaliação online

No processo de ensino/aprendizagem é essencial que se tenha meios de avaliar o aluno. A maioria dos *SGC* existentes possuem funcionalidades de avaliação *online*, podendo a avaliação ser automática ou não. Os tipos mais comuns são:

1. **Múltipla escolha**

As questões de múltipla escolha podem ter um número variável de alternativas. O professor descreve a questão e oferece alternativas, das quais apenas uma é a correta.

Segundo Tancredi (2002), esse tipo de questão é útil para verificar o conhecimento de fatos e a capacidade de compreensão e de aplicação do conteúdo que está sendo proposto. Porém questões *Múltipla escolha* podem ser difíceis de elaborar, pois demanda maior tempo por parte do professor.

2. **Verdadeiro ou Falso**

Neste tipo de questão o professor desenvolve uma pergunta e elabora duas alternativas, sendo uma verdadeira e a outra falsa. É uma restrição sobre o tipo descrito no item acima.

3. **Preencher lacunas**

São questões que contêm uma ou mais frases com algumas partes omitidas, que devem ser preenchidas por uma única palavra, número ou símbolo. Questões deste tipo são de fácil elaboração, podem ser usadas para testar vários assuntos, dificultam a adivinhação por parte do aluno e permitem alguma liberdade de expressão (Tancredi, 2002). Entretanto, a tendência é deixar lacunas demais pois a construção de itens puramente objetivos pode ser difícil (Haydt, 2000).

4. **Descritiva ou Dissertativas**

Neste tipo de questão o professor elabora a questão e o aluno tem que redigir sua resposta. Questões *dissertativas* demandam maior tempo e empenho do professor na correção, porém é possível observar o raciocínio e grau de compreensão do aluno sobre o assunto.

Vários *SGC* disponibilizam alguns destes tipos de avaliações *online*, sendo que em alguns sistemas é possível elaborar avaliações com mais do que um tipo de questão.

Questões do tipo *Múltipla escolha*, *Verdadeiro ou Falso* e *Preencher lacunas*, são denominados avaliações automáticas e são muito utilizadas em *SGC*, devido a facilidade de aplicação e resposta imediata, já que é possível a correção automática pelo sistema, como mostra Nelson (1998).

Este tipo de avaliação é baseado em gabarito: o professor, na elaboração da questão, envia a alternativa correta que fica armazenada no banco de dados. Quando o aluno for resolver a questão, o sistema compara sua resposta com o gabarito fornecido pelo professor e consegue mostrar se o aluno acertou ou não, além de armazenar a resposta do aluno. Com os resultados das avaliações do aluno, é possível gerar relatórios para auxiliar o professor no acompanhamento do aluno.

Em provas com questões descritivas não é possível a correção *online*, neste tipo de avaliação, em *SGC*, o aluno resolve a avaliação, envia para o sistema, o professor recebe, corrige e só depois o aluno recebe um retorno.

A maioria dos *SGC* não oferecem funcionalidades para incrementar a interatividade em cursos específicos. Por exemplo, no ensino de geometria provas com questões do tipo “múltipla-escolha” não são tão interessantes quanto a disponibilização de um exercício em que o aluno precise realizar uma construção. Para isso seria conveniente incorporar ao sistema um módulo de conteúdo específico, no caso da geometria um programa de Geometria Dinâmica - *GD* (Laborde and Bellemain, 1997; Jackiw, 1995), onde o aluno pudesse realizar construções geométricas. Este tipo de funcionalidade pode ajudar o aluno em sua aprendizagem, além de simplificar algumas das tarefas do professor.

Na literatura existem vários relatos de experiências de cursos via Web que utilizavam programas para trabalhar com conteúdos específicos, mas os programas utilizados não eram integrados com o *SGC* utilizado (Miranda Júnior, 2005; Togni, 2005; Moraes and Teixeira, 2005; Melo et al., 2000; Jones, 2003). Os professores tinham que utilizar o *SGC* e, em paralelo, outros programas computacionais locais para trabalhar com conteúdos específicos. Assim, quando era proposto um exercício através do *SGC* o aluno deveria sair do sistema, abrir o programa que estava sendo utilizado para resolver o exercício e depois retornar ao *SGC* para enviar a resposta do exercício através de um arquivo. O professor, por sua vez, para analisar a resolução do aluno deveria, dependendo do tipo de exercício, entrar no *SGC*, acessar o arquivo contendo a resolução, gravar no seu computador, abrir o programa, corrigir o exercício e se o *SGC* permitir, inserir uma nota ou comentário para dar um retorno para o aluno. Um exemplo na literatura é Costa et al. (2004) que relata uma experiência com uma turma de licenciatura em matemática da UFRJ onde trabalhou com o *SGC* AulaNet e em paralelo com vários *softwares* de ensino de matemática como Modellus (mod, 2007), Graphmatica (Ksoft, 2007), Winplot (win, 2007b) e Wingeom (win, 2007a).

2.2 Sistemas Gerenciados de Cursos

Nesta seção descrevemos alguns *SGC* destacando seus desenvolvedores, tecnologias utilizadas no desenvolvimento, funcionalidades relacionadas ao processo de ensino/aprendizagem e as principais instituições que utilizam o sistema.

2.2.1 TelEduc

O *TelEduc* (Ambiente de Ensino-Aprendizagem a Distância) (Rocha, 2002) está sendo desenvolvido no NIED (Núcleo de Informática Aplicada à Educação) da Unicamp (Unicamp, 2007) desde 1997 e tem como meta a formação de professores para a Informática na Educação (?). Sua primeira versão foi disponibilizada em 2001 como um sistema gratuito e desde então vem sendo desenvolvidas novas funcionalidades de acordo com seu uso em vários contextos diferentes, como disciplinas de graduação da Unicamp, cursos de aperfeiçoamento para professores, entre outros. A figura 2.1 mostra a tela inicial do sistema TelEduc.



Figura 2.1: TelEduc.

O TelEduc foi desenvolvido usando várias tecnologias: linguagem de programação *PHP* (php, 2007) para as páginas geradas no sistema; *Perl* (per, 2007) na ferramenta de bate-papo para a exibição de mensagens; *Java* (Java, 2007) utilizada na funcionalidade *Intermap* (Romani, 2000) que auxilia no acompanhamento do aluno no curso, e o banco de dados *MySQL* (mys, 2007).

O TelEduc tem como elemento principal a funcionalidade *Atividade* (Rocha, 2002), partindo

do conceito de que o aprendizado é desenvolvido através da resolução de problemas. As demais funcionalidades foram desenvolvidas para auxiliar o aluno a desenvolver os problemas propostos. Entre estas funcionalidades podemos destacar: *Portifólio*, onde são armazenados os materiais enviados pelos alunos, assim como seus comentários e respostas de exercícios e o *Diário de Bordo*, onde os alunos podem fazer qualquer anotação, mantendo-a privada ou não (Oeiras, 2005). No *Diário de Bordo* o professor pode visualizar a anotação do aluno e anexar comentário.

A avaliação no TelEduc está estruturada na análise da participação do aluno. Para isso o TelEduc disponibiliza funcionalidades que auxiliam na análise quantitativa dos registros das interações. Está em desenvolvimento uma funcionalidade para apoio à elaboração e resolução de exercícios *online* envolvendo questões dissertativas, de múltipla-escolha, verdadeiro-falso, associativa e preencher lacunas.

Dentre as instituições que são citadas como usuárias do TelEduc estão grupos (departamentos ou unidades) de grandes universidades como: Universidade Estadual de Campinas (Unicamp, 2007), Universidade de Brasília (UnB, 2007), Universidade Federal do Rio Grande do Sul (UFRGS, 2007), Universidade São Francisco (USF, 2007), Pontifícia Universidade Católica de São Paulo (PUCSP, 2007), Universidade Federal de Campina Grande (UFCG, 2007), entre outros.

2.2.2 Moodle

O Moodle (Modular Object-Oriented Dynamic Learning Environment) começou a ser desenvolvido por Martin Dougiamasé em 1999 baseado em uma pedagogia construtivista da educação (Dougiamasé, 2006) e foi desenvolvido como um sistema para gestão de aprendizagem, com trabalho colaborativo, permitindo a criação de cursos e páginas de disciplinas *online*. Sua primeira versão foi liberada em 2002, como um sistema livre, e desde então se tornou um projeto de desenvolvimento contínuo, contando com profissionais de vários países. Uma de suas principais características é a possibilidade de incorporar módulos com novas funcionalidades. A figura 2.2 mostra a tela inicial do *SGC Moodle*.

O sistema *Moodle* (Moodle, 2004) é desenvolvido em *PHP*, podendo trabalhar com os bancos de dados *MySQL*, *PostgreSQL* (pos, 2007), *Oracle* (ora, 2007), *Access* (Microsoft, 2007), *Interbase* (bor, 2007) e *ODBC*.

O conceito fundamental do Moodle, (Filho, 2005) é ser um sistema, onde professores disponibilizam recursos e propõe atividades para os alunos. Entre as funcionalidades do Moodle



Figura 2.2: Moodle

podemos destacar: *Wiki*, onde é possível a construção de um texto de forma coletiva usando um navegador Web e *Expressão algébrica*, que permite a edição de fórmulas matemáticas utilizando filtros que suportam $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.

A avaliação no Moodle consiste de questionários com questões descritivas ou de “múltipla-escolha”, desenvolvimento de texto colaborativo, além de atividades que são desenvolvidas pelo alunos e enviadas em formato digital ao servidor do curso utilizando o sistema (Filho, 2005).

O Moodle reúne uma enorme quantidade de usuários, com mais de 75.000 usuários registrados, falando 70 idiomas em 138 países. No Brasil é utilizado em importantes instituições, entre elas temos: Universidade de Brasília (UnB, 2007), Universidade Federal de São Paulo (UNIFESP, 2007), Centro Federal de Educação Tecnológica de Goiás (GO, 2007), Universidade Federal de Uberlândia (GO, 2007), Pontifícia Universidade Católica de São Paulo (PUCSP, 2007), Universidade Federal de São Carlos (UFSCar, 2007), Universidade Federal da Paraíba (UFP, 2007), entre outros.

2.2.3 AulaNet

O *AulaNet* começou a ser desenvolvido em 1997 pelo Laboratório de Engenharia de Software da Universidade Católica do Rio de Janeiro (LES, 2007). Segundo Fuks et al. (2003), este sistema tem como objetivo auxiliar o docente na tarefa de disponibilizar o conteúdo de um curso na internet. É gratuito e foi desenvolvido utilizando as linguagens *Java* e *Javascript* e um banco de dados relacional compatível com *ODBC*. Atualmente, o *AulaNet* está disponível em português, inglês e espanhol. A *EduWeb* (edu, 2007) é distribuidora e representante exclusiva do sistema. A figura 2.3 mostra a tela inicial do *SGC AulaNet*.

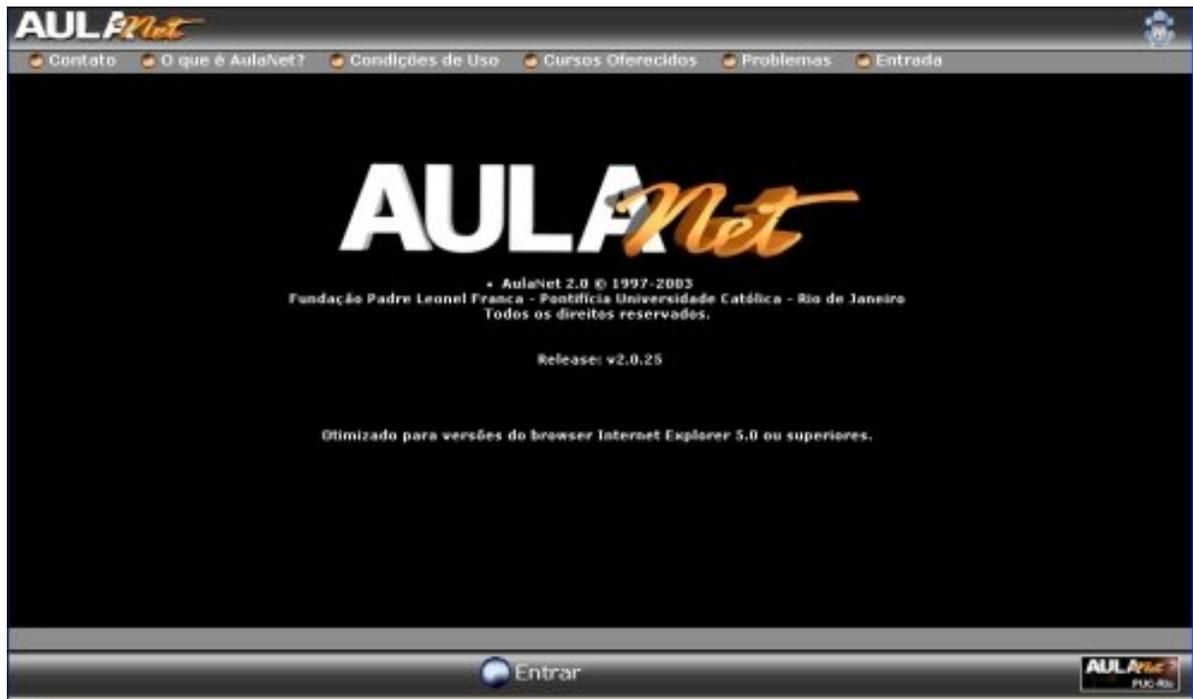


Figura 2.3: AulaNet.

O conteúdo do sistema AulaNet segundo Fuks et al. (2004), é baseado nos conceitos de cooperação, coordenação e comunicação onde as interações entre os usuários são mais valorizadas que o estudo individual dos conteúdos do curso. Assim a proposta do AulaNet é oferecer suporte à cooperação que se realiza nas interações do aluno com o professor, com os outros alunos e com os conteúdos didáticos colocados na Web pelo professor (Fuks et al., 2003).

No AulaNet é possível avaliar o aluno através de provas e testes *online*, lista de exercícios (enviada pelos alunos através de *upload*) e através da interação dos alunos com o sistema, onde é possível acompanhar caminhos percorridos, contribuições nas atividades em grupo (como fórum

e chats), utilização dos conteúdos inseridos pelo professor, entre outros. O professor obtém o retorno destas avaliações através de relatórios e gráficos individuais (Menezes et al., 1998).

2.2.4 BlackBoard/WebCT

O *WebCT* (Web Course Tools) (Goldberg and Salari, 1997) foi inicialmente desenvolvido pelo Departamento de Ciência da Computação da Universidade British Columbia no Canadá, utilizando as linguagens de programação *Java* e *Perl* e os bancos de dados *Microsoft SQL* (mic, 2007) e *Oracle*, e atualmente está disponível em 14 idiomas. Este sistema é comercial e em 2006 foi comprada pela empresa Blackboard por U\$178 milhões, passando a ser denominado *BlackBoard/WebCT*.

Entre suas funcionalidades podemos destacar o *Whiteboard* (Guerra, 2000) uma ferramenta de edição colaborativa, que permite a criação de conteúdo de forma síncrona, por usuários dispersos geograficamente; *Work Offline/Synchronize* (Tools, 2007) ferramenta que permite que professor e aluno façam *download* de todo o conteúdo do curso em um formato que possa ser impresso ou armazenado localmente. A figura 2.4 mostra a tela inicial do *SGC WebCT* †.

Assim como outros *SGC* o *WebCT* permite acompanhar e avaliar os alunos. Na funcionalidade de acompanhamento do aluno é possível visualizar o número de páginas acessadas pelo aluno em relação ao número total de páginas que deverá acessar, histórico das páginas visitadas e o tempo gasto em cada acesso. Como avaliação o *WebCT* oferece as funcionalidades: *auto-avaliação*, onde o aluno faz uma avaliação de “múltipla-escolha” para verificar seus conhecimentos; *avaliações*, onde o professor envia um arquivo com a prova em formato texto, onde as perguntas podem conter imagens e vídeos e o aluno resolve a prova e reenvia através de *upload*; *teste*, onde o professor elabora provas de “múltipla-escolha” e “verdadeiro/falso” ou utiliza questões do banco de dados (que armazena as questões elaborada pelos professores), que são corrigidas automaticamente pelo sistema.

Na tabela 2.2 fazemos um resumo das características dos *SGC* citados anteriormente. A coluna “URL” refere-se ao endereço da Web em que o sistema está disponível. A coluna “Início” refere-se a ano de início de desenvolvimento do *SGC*. A coluna “Linguagem” refere-se a linguagem de programação em que o sistema foi desenvolvido. Na coluna “BD” encontram-se os banco de dados que os sistemas citados suportam. E finalmente, a coluna “Licença” refere-se ao tipo de licença que cada um dos programas possui.

† São desconhecidas informações de interface sobre o *WebCT* após a união com o *BlackBoard*, assim vamos mostrar as referências encontradas sobre o *WebCT*

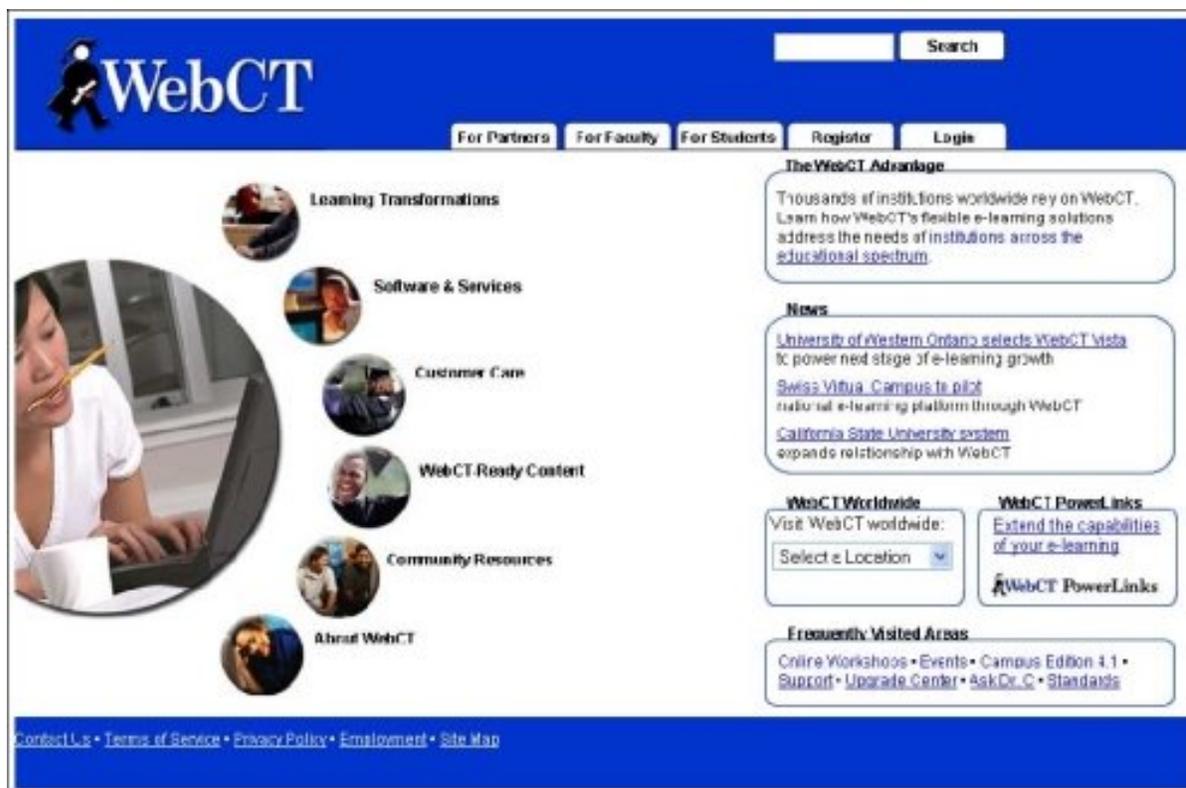


Figura 2.4: WebCT.

Tabela 2.2: Recursos de alguns dos programas de GD

SGC	URL	Início	Linguagem	BD	Licença
TelEduc	teleduc.nied.unicamp.br/teleduc/	2001	PHP, Perl e Java	MySQL	Gratuito
AulaNet	www.eduweb.com.br/portugues/home.asp	1997	Java e Javascript	ODBC	Gratuito
Moodle	moodle.org/	2002	PHP	MySQL	GNU [‡]
WebCT	www.webct.com/	1996	Java e Perl	Microsoft SQL e Oracle	Comercial

[‡]GNU General Public License - permite livre distribuição e modificação do programa e seu código fonte. Para maiores informações consulte o endereço: <http://www.gnu.org/copyleft/gpl.html>.

Capítulo 3

SAW - Sistema de Aprendizagem pela Web

A maioria dos *SGC* citados na tabela 2.2 oferecem funcionalidades para disponibilização de conteúdos (na forma de hipertextos), para comunicação e cooperação (através de fóruns e bate-papos) e para disponibilização de exercícios e avaliações na forma de testes. Porém a maioria destes gerenciadores não possuem recursos especializados para o ensino/aprendizado de conteúdos específicos.

O desenvolvimento do **SAW - Sistema de Aprendizagem pela Web**, iniciado no final de 2003, visava servir de ferramenta motivadora e facilitadora para alunos e professores da disciplina **Noções de Ensino de Matemática usando o computador (MAC118)**, obrigatória para licenciandos em matemáticas do Instituto de Matemática e Estatística da Universidade de São Paulo (IME-USP, 2007). O *SAW* teve como requisito central possibilitar o acoplamento de programas computacionais para ensino/aprendizagem de conteúdos específicos, sem a necessidade de programação.

O *SAW* é baseado na arquitetura cliente/servidor e está sendo desenvolvido utilizando as linguagens de programação *PHP* (php, 2007) e *Javascript* (Javascript, 2007) e o banco de dados *MySQL* (mys, 2007) e seu código fonte é disponibilizado na forma de “software livre” (License, 2007). A arquitetura do *SAW* consiste na extensão dos tradicionais *SGC* existentes. Uma parte importante do *SAW* é estruturada a partir de aulas, com cada aula possuindo um ou mais tópicos ou exercícios, sendo que os exercícios utilizam programas computacionais que podem incrementar a interatividade. Estes programas computacionais são incorporados ao *SAW* como *Módulos de Aprendizagem (MA)*. Cada *MA* é um aplicativo *Java* na forma de *applet* que deve ficar responsável pela análise/avaliação de cada interação ou resolução de exercícios pelo usuário. Cabe ao sistema, verificar os resultados da comunicação com os *MAs* e fazer

a adaptação adequada do conteúdo e da navegação para cada usuário. Os *MA*s funcionam independentemente do sistema e se comunicam apenas no início e no fim de cada interação com o usuário. Por exemplo, na resolução de um exercício, o *SAW* libera o mesmo para o usuário, que ao final deve enviar sua solução. Deste modo, não há sobrecarga no servidor, pois o *MA* faz localmente toda a computação necessária. Atualmente existem dois *MA*s acoplados ao *SAW*, o *iGeom* que é um programa de geometria dinâmica e o *iCG*, um protótipo que simula um computador simplicado, contendo também um pequeno compilador para uma linguagem de programação semelhante a C (Kernighan and Ritchie, 1988). Está em desenvolvimento um outro *MA*, o *iGraf*, que é um programa de visualização de gráficos de funções.

Uma condição essencial para o acoplamento do *MA* é que este disponha de recurso para comunicação (métodos HTTP e POST). Além disso, para que o *MA* possa ser plenamente utilizados no *SAW*, ele deve dispor de recursos para avaliação automática. Com a avaliação automática é possível adaptar a disponibilização de páginas de acordo com o que o aluno desenvolve no *MA*. Se o *MA* não dispõe destes recursos, só poderá ser acoplado como uma funcionalidade de conteúdo.

Outra funcionalidade importante do *SAW* é o “Dicionário Webmídia” que permite autoria de hipertextos, contendo fórmulas matemáticas, imagens, *links* e construções interativas (via *MA*) e pode ser utilizado em trabalhos colaborativos. Esta funcionalidade foi desenvolvida após o estudo de caso conduzido na segunda versão do *SAW*, quando percebemos que os alunos, em geral, apresentavam dificuldade em elaborar demonstrações matemáticas, e que esta dificuldade podia ser reforçada pelo uso intensivo da geometria dinâmica. Para dimensionar este problema, em 2005, apresentamos aos alunos uma atividade em que eram desafiados a reconhecer um objeto geométrico descrito por uma construção dinâmica. A atividade proposta aos alunos consistia em descobrir propriedades do locus gerado, no caso uma elipse (vide figura 3.1), e demonstrar este fato. A atividade foi realizada com 15 alunos da disciplina *MAC118*. O aluno deveria seguir os seguintes passos de construção:

1. construa dois pontos A e B , livres na área de desenho;
2. construa a reta r definida por A e B ;
3. construa a reta s perpendicular a r , contendo o ponto A ;
4. construa um ponto C , solto sobre a reta r ;
5. construa a circunferência $c0$, de centro A e contendo o ponto B ;
6. construa a circunferência $c1$, de centro A e contendo o ponto C ;
7. construa um ponto D , solto sobre a circunferência $c0$;

8. construa a semi-reta t definida por A e D ;
9. construa a reta u paralela a reta r , contendo o ponto D ;
10. construa o ponto E , interseção entre t e $c1$;
11. construa a reta v , perpendicular a r e contendo o ponto E ;
12. construa o ponto F , interseção entre u e v ;
13. construa o *locus* gerado pelo ponto F ao mover-se o ponto D sobre $c0$.

Dos alunos que realizaram a atividade, apenas um encaminhou uma resposta que poderia ser classificada como demonstração, sendo que os demais encaminharam argumentos baseados no exame de exemplos utilizando-se “medições”. Assim, aproveitando-se da característica dinâmica do *iGeom*, os alunos obtinham algumas medidas dinâmicas e observavam que uma propriedade de elipse era sempre obtida em todas as configurações testadas, como na figura 3.1, que foi a resposta encaminhada por um dos alunos.

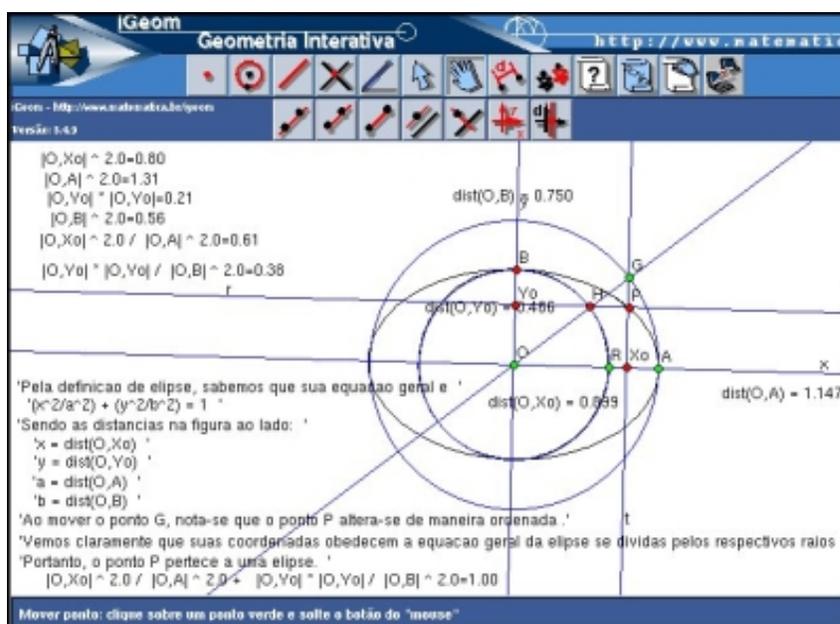


Figura 3.1: Tentativa de demonstração de um aluno

Este engano permitiu ao professor esclarecer que para demonstrar que uma conjectura está errada basta apresentar um contra-exemplo, porém para mostrar que a conjectura está correta, não é suficiente a realização de um número, mesmo que grande, de testes. Por outro lado, também serviu para mostrar a necessidade de explorar mais atividades que exijam a realização de demonstrações, uma vez que o aluno de *MAC118* será um futuro professor de matemática

no ensino fundamental II e médio. Esta necessidade motivou-nos a desenvolver um editor de hipertexto com fórmulas matemáticas e, a partir dele, um construtor de conteúdo em que os alunos pudessem compor, por exemplo, um dicionário de demonstrações de teoremas fundamentais de geometria. A estes recursos demos os nomes, respectivamente, **MaRTE** e **Dicionário WebMídia**.

A seguir apresentamos a arquitetura do *SAW*, sua estrutura e as funcionalidades desenvolvidas desde sua primeira versão em 2003.

3.1 A arquitetura

Muitos dos *SGC* utilizados pela internet são baseados na arquitetura cliente/servidor, como os citados no capítulo 2. A principal característica destes sistemas é a parte cliente requisitar uma operação e o servidor cuidar de sua análise e execução enviando ao final uma resposta ao cliente. Para tornar o sistema mais flexível e deixar o conteúdo mais interativo, desenvolvemos uma arquitetura cliente/servidor sendo que a parte cliente é equipada com *MA*, aplicativos *Java* (*applets*), que faz parte do processamento associado com o domínio de aplicação (o conteúdo do curso). Assim, parte das operações computacionais são executadas no computador cliente, diminuindo as chances de sobrecarga do servidor. Além disso, como cada *MA* tem seu próprio analisador de interação, os diferentes domínios podem dispor de diferentes formas de avaliação (trocando o *MA*), tornando o sistema independente do domínio e utilizável em diversos contextos.

A parte servidor possui uma arquitetura em quatro camadas (Camada de dados, Camada de Tarefas, Camada de Comunicação e Camada de Interação) (Dijkstra, 1968; Buschmann et al., 1996), onde cada camada é um subsistema que oferece serviços à camada superior e é composta por componentes (páginas com programação *PHP* e *Javascript*) que tenham o mesmo nível de abstração.

Na figura 3.2 apresentamos um diagrama da arquitetura do *SAW*. No anexo A apresentamos a relação entre as camadas e seus respectivos códigos de programação.

3.1.1 Camada de dados

A *camada de dados* é onde ficam armazenados os dados referentes aos conteúdos e aos usuários do *SAW*. Atualmente estão separados em quatro tipos: dados gerais, dados de conteúdo, dados do curso e modelo do aluno. A relação das tabelas da *camada de dados* pode ser vista em anexo C.

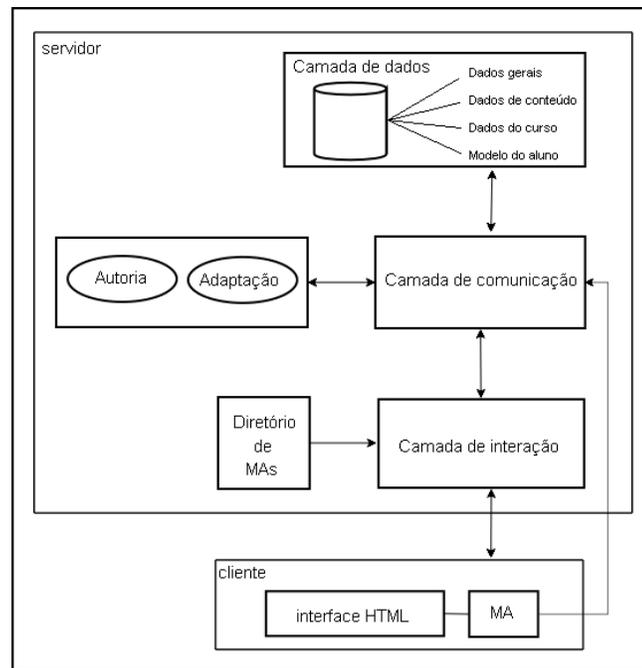


Figura 3.2: Arquitetura do SAW.

- **Dados gerais:** são armazenados dados pessoais relativos aos usuários do sistema, configurações dos *MAs* e configurações do *SAW* para cada curso.
- **Dados de conteúdo:** os conteúdos no *SAW* são armazenados em seis diferentes tipos de **componentes**: cursos, aulas, tópicos, exercícios, textos e exemplos. Aqui são armazenados os conteúdos de cada componente.
- **Dados da turma:** são armazenadas as relações dos componentes que pertencem a turma, ou seja, armazena quais os componentes (criados anteriormente) pertencem a uma determinada turma. No *SAW* um curso pode ter uma ou mais turmas, onde turmas de um mesmo curso podem possuir os mesmos componentes ou componentes distintos.
- **Modelo do aluno:** O modelo do aluno representa o conhecimento e as características do aluno num determinado instante. Para isso o sistema armazena informações que incluem os componentes que o aluno já visitou, dados sobre a resolução de exercícios, caminhos percorridos e preferências, entre outros.

3.1.2 Camada de Tarefas

A *Camada de Tarefas* está associada com os componentes exercícios, textos e exemplos. Esta camada é responsável por criar e adaptar o conteúdo a ser repassado para o usuário. Nesta camada é feita a conexão com a *Camada de Dados* (através da *Camada de Comunicação*), que retorna a organização e apresentação do conteúdo, levando em conta as preferências do sistema e o *modelo do aluno*. Esta camada define o caminho do usuário em uma turma, sendo que o caminho depende do estado de conhecimento de cada usuário, que é definido através de suas últimas interações com o sistema. Esta camada está dividida em dois módulos: *Adaptação* e *Autoria*.

1. Autoria

A autoria de conteúdos no *SAW* é feita por usuários do tipo professor. As permissões para este tipo de usuário estão apresentadas na seção 3.4.2. Os conteúdos podem ser armazenados em até seis diferentes tipos de componentes:

- **textos**: utilizando o editor de hipertextos MaRTE (o editor MaRTE é descrito na sessão 3.3.1), o professor pode inserir conteúdos com textos, imagens, *links*, fórmulas matemáticas e construções interativas utilizando os *MAs*.
- **exercícios e exemplos**: utilizando algum *MA* acoplado ao *SAW*, o professor elabora o exercício (e seu gabarito) ou exemplo e o insere no *SAW*.
- **tópicos**: este componente pode ser composto por textos, exercícios ou exemplos.
- **aulas**: este componente é dividido em duas partes, aulas de tópicos, compostos por tópicos anteriormente criados e aulas de exercícios, composto por textos, exercícios ou exemplos.
- **courses**: componente composto por aulas.

A descrição de como o professor faz a inserção dos componentes no *SAW* pode ser vista em anexo B.

Como uma parte da arquitetura do sistema é baseada em *MAs*, a autoria de alguns componentes (exercícios, textos e exemplos) necessita que parte deste processo seja realizada pelos *MAs* e que estes possuam o recurso de comunicação e, no caso de um exercício, o recurso de avaliação automática. Cada componente pode ser editado pelo professor de forma *online*, e caso o *MA* permita, também é possível a criação *offline* de componentes.

Os componentes criados são enviados para o servidor onde ficarão armazenados. Dessa forma, um aluno conectado ao servidor pode, por exemplo, resolver um exercício e solicitar sua avaliação. Ao fazer esta requisição o *MA* avaliará a solução do aluno e a enviará o resultado ao servidor para que este responda de forma apropriada.

2. Adaptação

Os métodos de adaptação, conforme a classificação dada por Brusilovsky (1996), são a *adaptação do conteúdo* e a *navegação adaptativa*. A *adaptação do conteúdo* diz respeito à forma como o conteúdo será apresentado, enquanto a *navegação adaptativa* considera os caminhos possíveis dentro do conteúdo do sistema. O enfoque da arquitetura do *SAW* é na *navegação adaptativa*, pois permite direcionar o usuário dentro do espaço hipermídia. Como mostra Brusilovsky (1998), existem diversas formas de fazer uso da navegação adaptativa: *guia direto*, *ordenação adaptativa*, *ocultação* e *anotação*.

- O *guia direto* oferece um *link* para a próxima página cujo o sistema considera mais adequada para o usuário de acordo com as preferências e as informações presente no modelo do aluno.
- A *ordenação adaptativa* faz um arranjo de *links* colocando os mais relevantes no topo de acordo com um critério de avaliação baseado no modelo do aluno.
- A *ocultação* esconde o *link* para uma página tornando-a inacessível ao usuário quando o sistema acreditar que o conteúdo desta página é inapropriado, considerando as informações do modelo do aluno e características do sistema.
- A *anotação* modifica a forma de apresentar os *links* para algumas páginas, por exemplo, modificar a cor dos *links* de páginas já visitadas ou colocar um ícone ao lado de *links* para páginas onde as atividades estão incompletas.

Existem benefícios e problemas na utilização de algumas destas formas de *navegação adaptativa*. Por exemplo, utilizando a *ordenação adaptativa* podemos apresentar *links* ao aluno cuja informação seja inadequada, usando a *ocultação* podemos restringir demais o espaço de navegação do usuário, dando pouca liberdade para que este explore o conteúdo.

No *SAW* optamos por utilizar a *navegação adaptativa* através da *anotação*. Neste sistema os conteúdos são apresentados através de componentes que podem ter quatro estados: liberado, não liberado, pré-requisito e completo, sendo:

- Liberado

Quando está liberado o componente fica disponível, através de um *link* na **cor preta**, para o aluno visualizar o conteúdo e, no caso do componente ser um exercício resolvê-lo;

- Não liberado

Não liberado significa que o componente pertence à turma mas está indisponível. Este estado é indicado através de um *link* inacessível na **cor vermelha** e é muito utilizado quando um professor faz o reuso de um curso já existente, pois neste caso o curso já está com todas as aulas no seu início e o professor vai liberando-as de acordo com o andamento da turma;

- Pré-requisito

Para auxiliar na adaptação, cada componente pode possuir pré-requisitos, ou seja, um determinado componente será disponibilizado ao usuário caso o sistema acredite que o mesmo possua os conhecimentos mínimos necessários ou a configuração do sistema permita a visibilidade deste componente. Um componente pode ter como pré-requisito, nenhum ou vários outros componentes do mesmo nível, sendo que quem definirá essas ligações será o professor no momento que elaborar o conteúdo para o curso. Este estado é indicado com um *link* inativo na **cor amarela**;

- Completo

Quando o aluno visita um componente e completa as tarefas propostas no componente, por exemplo, no caso de um exercício, quando o aluno consegue resolvê-lo de maneira correta, o sistema interpreta que o aluno já adquiriu o conhecimento relativo ao que foi proposto e considera o componente completo. Este estado é definido com um *link* na **cor azul**.

Quem determina o estado liberado e os pré-requisitos de cada componente em uma determinada turma é o professor responsável pela turma. O professor tem ainda a opção de liberar componentes para toda a turma ou apenas para alguns alunos.

3.1.3 Camada de Comunicação

A *camada de comunicação* é responsável por fazer a conexão entre as camadas do sistema, sendo que suas principais funções são:

- Enviar os dados da *camada de tarefas* para a *camada de interação*. Estas informações

podem ser dados sobre o usuário, preferências do sistema, adaptações de interface, dentre outras.

- Receber os dados da *camada de interação* e atualizar as informações presentes na *camada de dados*. Assim, a cada interação de um aluno é possível atualizar o *modelo do aluno* e, no caso do usuário do tipo professor, inserir um componente criado em *dados de conteúdo*.
- Notificar a *camada de tarefas* sobre o recebimento de dados da *camada de interação* e a modificação da informação presente na *camada de dados*.

3.1.4 Camada de Interação

A *camada de interação* tem como principal função acoplar os *MA*s e gerar o código *HTML* que deverá ser interpretado pelo navegador do usuário. O código *HTML* é gerado a partir das informações recebidas da *camada de tarefas* (intermediada pela *camada de comunicação*), que indica quais os *MA*s pertencentes a esta página *HTML*, quais são os componentes a serem apresentados e como os *links* devem ser apresentados. Ao utilizar a adaptação, em conjunto com os *MA*s, podemos explorar dados em diferentes formas de mídia * (sons, vídeos, textos e animações) aumentando a capacidade de percepção e memorização dos alunos. Um exemplo da interação do usuário com os *MA*s que estamos particularmente interessados são as resoluções de exercícios. O *MA* deve dispor de avaliadores automáticos, que quando executados analisam a resposta do aluno e devolvem o resultado para o servidor modificando as informações pertencentes à *camada de dados* e possibilitando o efeito de adaptação.

3.1.5 Módulos de aprendizagem

Os *MA*s são fundamentais nesta arquitetura, pois estão relacionados com o domínio de aplicação, sendo responsáveis pela avaliação das interações do usuário e pela maior interatividade com o sistema. A comunicação entre o *SAW* e o *MA* ocorre através da passagem de parâmetros para o *MA* via “tags” *HTML* e via método *POST*, para receber informações enviadas do *MA* ao *SAW*. Cada *MA* é um programa completo e independente do sistema, podendo ser acoplado a qualquer momento pelo administrador do sistema.

Entretanto como parte do processamento das interações com o usuário é efetuada na própria máquina cliente pelos *MA*s, estes devem se comportar da mesma maneira independentemente da

*Os módulos para inserção e vídeos, animações e sons está em desenvolvimento.

plataforma onde o usuário se encontra. Essa é a razão pela qual os *MA*s devem ser desenvolvidos utilizando a linguagem *Java* e codificados na forma de *applet*. Para que o *MA* possa ser utilizável no *SAW* é essencial que disponha recursos para:

- **Autoria:** O uso dos *MA* é viável graças ao seu sistema de autoria que permite criar/editar exercícios, textos e outras formas de apresentação do conteúdo. O sistema de autoria deve permitir a criação/edição *online*, ou seja, via *Web*, mas também pode possuir uma forma *offline* que permita a um professor criar um componente em sua máquina local para depois enviá-lo ao servidor.
- **Comunicação:** Todo *MA* deve ter um módulo de comunicação que receberá as informações do sistema repassando-a para os outros módulos e enviará para o sistema o resultado da avaliação da interação do usuário.

E para que os *MA* possam ser plenamente utilizável no *SAW* deve possuir recursos para:

- **Avaliador:** Ao final da interação do usuário com o *MA* o avaliador irá analisar o resultado desta interação e devolverá o resultado para o *SAW*. Por exemplo, o *avaliador* deve corrigir o exercício que um aluno fez e retornar o resultado para o sistema.
- **Núcleo:** O núcleo deve ser o responsável por gerenciar o *MA*. Será ele que tomará as decisões pertinentes a cada tipo de interação com o usuário, incluindo a apresentação do conteúdo, interação do sistema com o usuário, animações, avaliações, comunicação entre módulos, entre outras.
- **Adaptação:** Os *MA*s também precisam se adaptar ao receberem informações do servidor. Esta adaptação pode modificar a forma de correção de um exercício, atualizar a interface mostrando/escondendo algumas das funcionalidades do *MA* ou apresentar o conteúdo de forma diferenciada.
- **Interface:** Para desenvolver a interface dos *MA*s devemos oferecer um design “amigável” que interaja com a interface do *SAW*. Os recursos do *MA* devem ser apresentados de forma a facilitar a identificação de suas funcionalidades, levando a conta a usabilidade, consistência, visibilidade e o mais importante à flexibilidade que será utilizada para adaptar a interface do *MA* quando necessário.

A seguir apresentamos dois *MA*s que atualmente estão acoplados ao *SAW* e um *MA*, o *iGraf*, que está em desenvolvimento.

1. iGeom

O iGeom: Geometria Interativa na Internet (Isotani and de Oliveira Brandão, 2003), é um programa gratuito de Geometria Dinâmica (GD) (Laborde and Bellemain, 1997; Jackiw, 1995), que proporciona recursos facilitadores para o ensino e aprendizagem de Geometria. Além disso, o iGeom oferece ferramentas que auxiliam o professor na produção de material didático e no acompanhamento de alunos. Este programa foi desenvolvido em *Java* com o intuito que o mesmo pudesse ser utilizado nas formas aplicativo e *applet*. A versão atual deste programa permite realizar todas as operações básicas de GD, como por exemplo: criar objetos geométricos (como pontos, retas, semi-retas, segmentos e circunferências); opções de edição (esconder/mostrar, remover/rastrear objetos); e opções de gravação/recuperação de arquivos em diferentes formatos (incluindo imagens em PostScript e GIF). A figura 3.3 mostra a interface principal do *iGeom*, com um exemplo de construção geométrica.

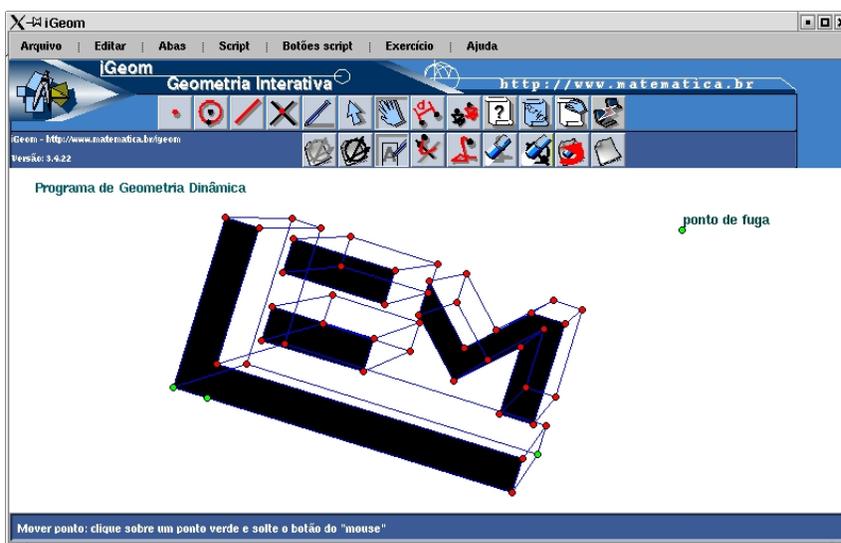


Figura 3.3: Construção no programa de Geometria Dinâmica, iGeom

Atualmente, desconsiderando o *iGeom*, conhecemos apenas dois programas de GD que possuem ferramentas de produção e avaliação de conteúdo para *Web*: o *Cinderella* (Kortenkamp, 1999) e o *C.a.R.* (Grothman, 2007). Porém, nenhum deles possuem recursos de comunicação. Sem este recurso, os programas citados não permitem a troca de informação com um servidor e, portanto, a sua utilização em sistemas *SGC* torna-se restrita,

principalmente quando queremos verificar ou analisar as interações de um aluno com o programa.

2. iCG

O iCG, é um programa gratuito para introdução à computação a que foi desenvolvido pelo Prof. Dr. Leônidas de Oliveira Brandão e um grupo de alunos do IME-USP, com fins didáticos. Ele está dividido em duas partes principais, o emulador e o compilador.

Emulador: é um modelo simplificado de computador, com apenas 100 posições de memória, 10 instruções, um acumulador e uma calculadora.

Compilador: utiliza uma linguagem de alto-nível bem simples e que pode ser facilmente alterada, dispendo de um comando de seleção (*if*), um comando de repetição (*while*) e comando para entrada (leia) e para saída (escreva) de dados.

Este programa possui o recurso de avaliação automática e comunicação e já foi utilizado como *módulo de aprendizagem* no SAW em 2005 no curso MAC0110 - Introdução à programação, oferecido para as turmas de licenciatura em matemática e bacharelado em matemática do IME-USP. A figura 3.4 mostra a interface do emulador do iCG.

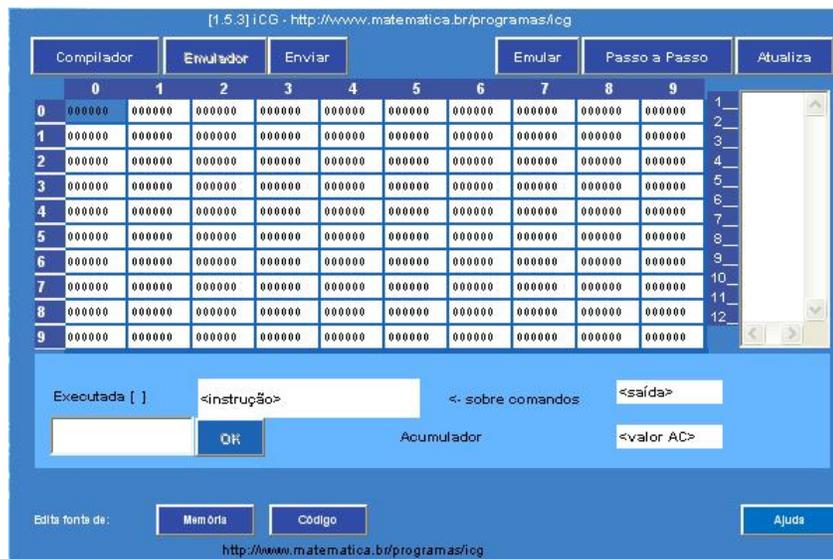


Figura 3.4: Tela do programa emulador/compilador iCG

3. iGraf

O iGraf (Prado and de Oliveira Brandão, 2006) é um programa de visualização de gráficos de funções que está sendo desenvolvido pelo Prof. Dr. Leônidas de Oliveira Brandão e pelo aluno de mestrado Reginaldo do Prado.

Na versão atual do iGraf é possível o usuário: digitar uma função e ver sua visualização gráfica; fazer animações - através da variação de um parâmetro da função; fazer o cálculo da integral definida de uma função, entre outros. Está em desenvolvimento o módulo de correção automática que fará com que o iGraf possa ser integrado ao *SAW* como um *módulo de aprendizagem*. A figura 3.5 mostra a interface do iGraf com o gráfico da função $f(x) = \text{sen}x$.

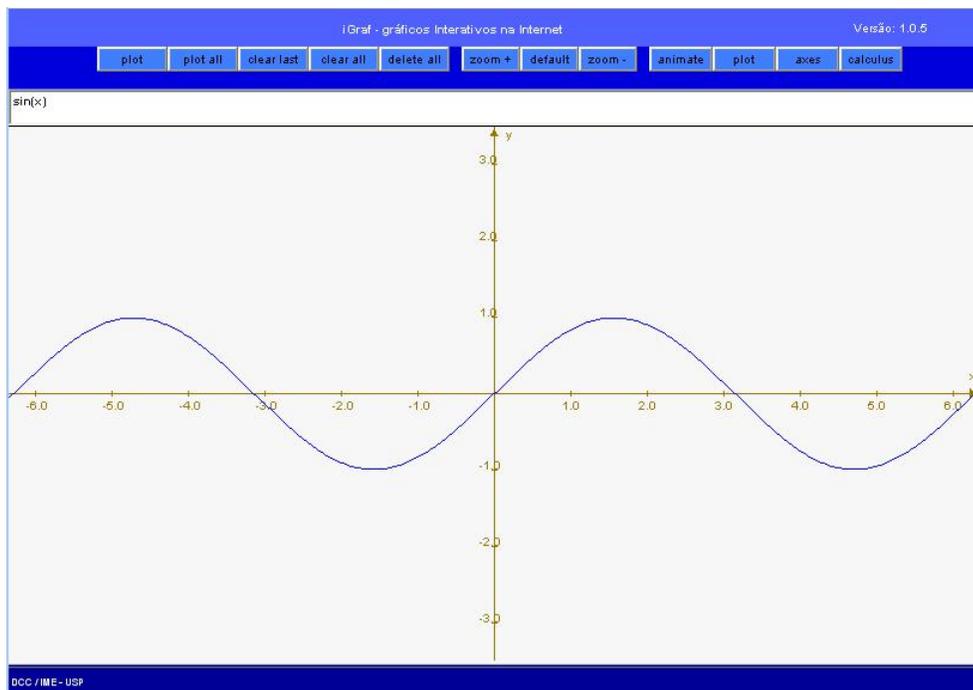


Figura 3.5: Tela do programa de visualização gráfica iGraf

3.2 Estrutura

Os conteúdos no *SAW* são armazenados em seis diferentes tipos de componentes: cursos, aulas, tópicos, exercícios, textos e exemplos. Estes componentes estão divididos em 4 níveis: curso está no nível 1, aula no nível 2, tópico no nível 3 e no nível 4 estão exercícios, exemplos e textos. Um componente do nível $i < 4$ é composto por um conjunto de componentes do nível $i + 1$, podendo haver dependência entre os componentes do mesmo nível, ou seja, para que o aluno possa fazer um exercício e_i , que depende de um texto t_j e de um exercício e_k , precisará antes ler o texto t_j e resolver o exercício e_k . A figura 3.6 mostra a relação entre os componentes. Sendo possível disponibilizá-los em dois cenários, figura 3.7:

- Cenário 1: curso composto por aula e aulas compostas por exercícios, exemplos e textos.
- Cenário 2: curso composto por aula, aulas compostas por tópicos e tópicos compostos por exercícios, exemplos e textos.

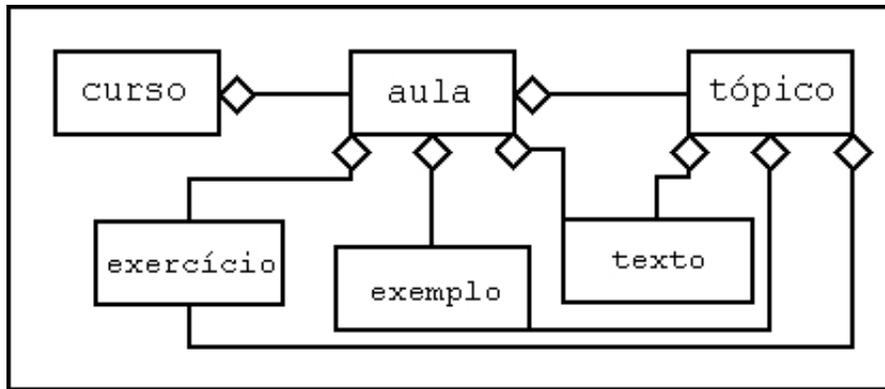


Figura 3.6: Relação entre os componentes

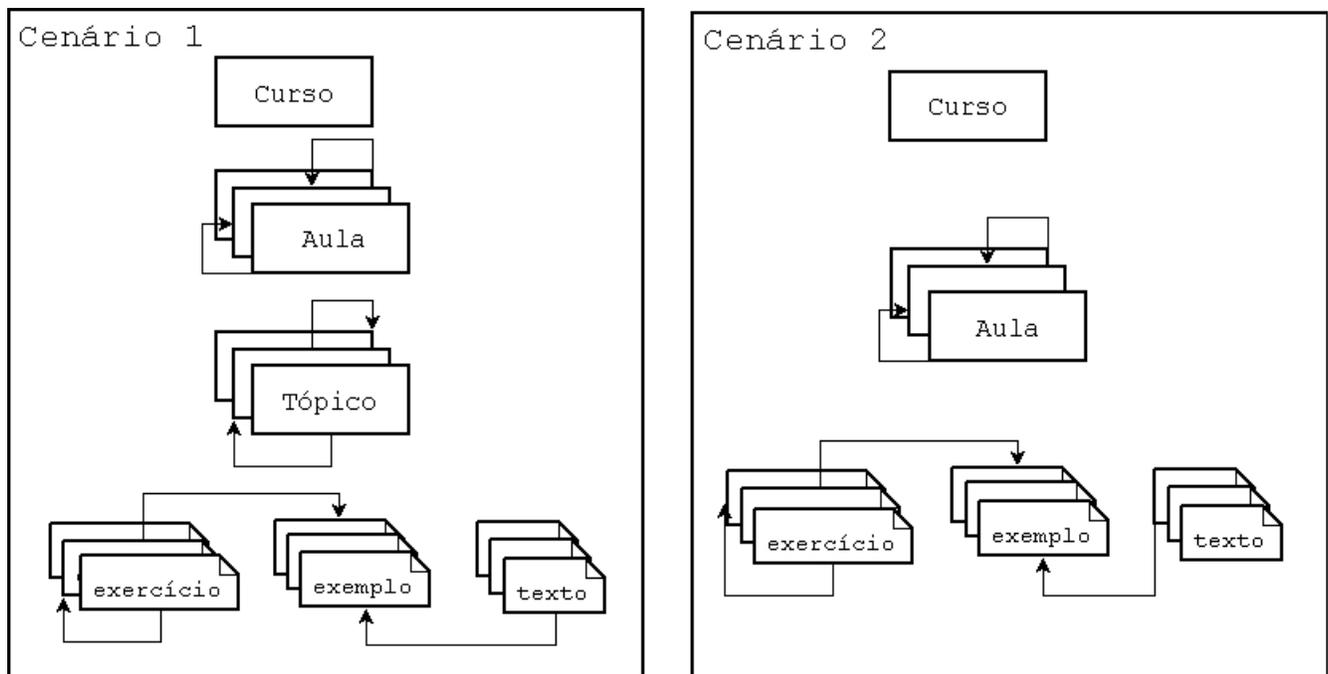


Figura 3.7: Possíveis cenários

A figura 3.8 mostra um exemplo de relação de dependência entre componentes do *SAW*. O armazenamento em componente visa facilitar o reuso de conteúdo. Por exemplo, uma determinada aula cadastrada no sistema pode fazer parte de diversos cursos.

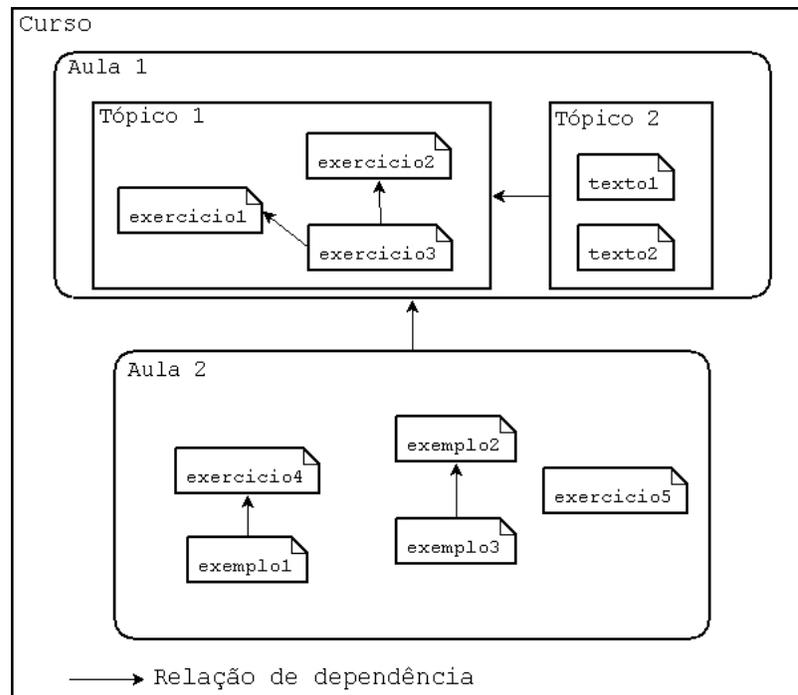


Figura 3.8: Exemplo de componentes em um curso

Todo componente criado por um professor passa por um período de teste[†] onde somente o autor pode utilizá-lo, alterá-lo e removê-lo, após o período de teste o componente fica liberado para todos os professores e não pode mais ser alterado ou removido. Isso evita que um professor altere ou apague um componente que está sendo utilizado em algum outro curso. Dessa forma, um componente c de nível $i + 1$ pode ser inserido em qualquer outro de nível i sem a necessidade de recriar um novo componente.

Todo exercício realizado pelo aluno é armazenado em banco de dados para que o professor possa verificar posteriormente a construção do aluno. Caso o professor perceba alguma inconsistência entre a solução do aluno e o que ele esperava com resposta, ele pode mudar a avaliação final do exercício do aluno no banco de dados.

3.3 Funcionalidades

A possibilidade de incorporação de *MA* no *SAW*, especialmente aqueles destinados ao ensino/aprendizagem de conteúdo matemático, estimulou-nos a implementar funcionalidades mais sofisticadas que os comumente encontrados em *SGC*. Na tabela 3.1 destacamos algumas das funcionalidades exis-

[†]O período de teste é determinado pelo autor do componente

tentes no *SAW* em comparação aos sistemas *SGC* apresentados no capítulo 2, indicando com * quando a funcionalidade está presente. A linha “Rastreamento” refere-se ao acompanhamento do aluno do curso. A linha “Edição Colaborativa” refere-se às ferramentas que permitam uma colaboração entre os alunos e professores. A linha “Registro de informações” refere-se aos sistemas que fazem o controle das mensagens trocados no fórum ou bate-papo. Na linha “Testes” encontram-se aqueles que permitem a elaboração de avaliações automáticas. A linha “Fórmulas matemáticas” refere-se aos sistemas que permitem o uso de fórmulas matemáticas nas ferramentas de edição de conteúdo. Na linha “Inserção de MA” estão aqueles que possuem recursos para acoplar módulos de conteúdos específicos. A linha “Conteúdo” se referem aos *SGC* que possuem controle do conteúdo que será passado ao aluno. E finalmente, a linha “Licença” refere-se ao tipo de licença que cada um dos programas possui.

Tabela 3.1: Recursos dos SGC apresentados

Sistemas	SAW	Teleduc	AulaNet	WebCT	Moodle
Rastreamento	*	*	*	*	*
Edição Colaborativa	*			*	*
Envio de Trabalho Via Web	*	*	*	*	*
Registro de informações	*	*	*	*	
Testes			*	*	*
Fórmulas matemáticas	*				*
Inserção de MA	*				
Conteúdo	*	*		*	
Licença	GPL	Gratuito	Gratuito	Comercial	GPL

A seguir são apresentadas as funcionalidades desenvolvidas no *SAW*.

3.3.1 Editor de textos MaRTE

Com o uso do *SAW* em sala de aula, notamos a necessidade de um editor de hipertexto que permitisse o uso de fórmulas matemáticas (Moura and de Oliveira Brandão, 2005a). Desta necessidade construímos o editor *MaTE* (Math Text Editor), um editor de textos *HTML* que permite a inserção de fórmulas matemáticas. A segunda versão do *SAW* com o *MA iGeom* e o *MaTE* foi utilizada durante os anos de 2005 e 2006 com as turmas de *MAC118* do IME-USP (Moura and de Oliveira Brandão, 2005a)

No *MaTE* a edição de hipertextos não exigia conhecimentos da linguagem *HTML*, pois o mesmo apresentava em sua interface botões que geravam automaticamente as “tags” *HTML*. Porém, a visualização apresentada ao usuário durante a edição era o código *HTML* propriamente dito (figura 3.9). Desta forma, o usuário só visualizava o documento em sua forma final após a submissão do código do texto e seu processamento (e resposta) no servidor.

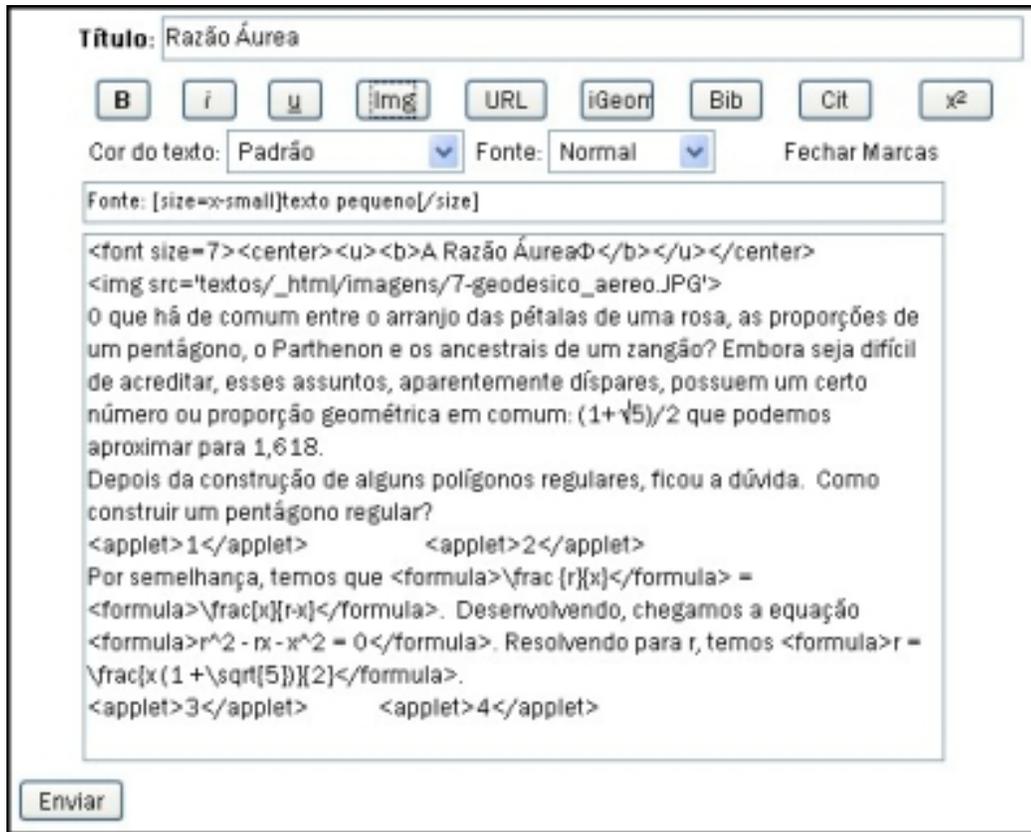


Figura 3.9: Primeira versão do editor de hipertexto MaTE

Depois do uso deste editor em algumas situações reais, especialmente na disciplina *MAC118*, ficou evidenciado que seria mais confortável ao usuário se ele tivesse a pronta visualização do formato final do hipertexto. Editores que apresentam prontamente ao usuário o formato final do texto são tipicamente conhecidos por editores *WYSIWYG* - “*What You See Is What You Get*” (wys, 2007).

A partir da análise de editores *WYSIWYG*, buscamos dentre aqueles que são gratuitos (HTMLArea (htm, 2007), FCKEditor (fck, 2007), TinyMce (tin, 2007), RTE (rte, 2007)), um que apresentasse código simples e fosse facilmente extensível, a fim de acrescentarmos às funcionalidades existentes àquelas presentes no *MaTE*. O editor *WYSIWYG* escolhido foi o **Rich Text Editor** (RTE) (rte, 2007). As funcionalidades presentes no RTE são: edição de hipertextos com fontes que podem apresentar diversos tipos, cores e tamanhos; formatos em negrito, itálico e sublinhado; edição e inserção de tabelas; alinhamento e inserção de figuras. O RTE é baseado em *JavaScript* (Javascript, 2007) e funciona em navegadores a partir das versões: Netscape 7.1+; Mozilla Firefox 0.6.1+; Mozilla 1.3+ e Netscape 7.1+ e Internet Explorer 5.5+. Em navegado-

res mais antigos os recursos visuais não funcionam e os textos aparecem com o formato *HTML* (campo *textarea*).

O RTE foi estendido com a adição das funcionalidades do *MaTE*, e passamos a denotá-lo por *MaRTE* - *Math Rich Text Editor*. Além das funcionalidades existentes no RTE o *MaRTE* conta com a inserção de:

- Módulos de aprendizagem: O usuário pode inserir no *MaRTE* construções do programa de geometria dinâmica *iGeom* (na forma de *applet*). As construções inseridas podem ser manipuladas dinamicamente pelos usuários, pois o *MaRTE* utiliza o próprio *applet* para apresentar a construção. A idéia é que todo *MA* acoplado no *SAW* possa ser inserido no hipertexto.
- Fórmulas matemáticas: O usuário digita a fórmula no formato \LaTeX (lat, 2007), que é transformada em imagem equivalente pelo servidor e inserida no hipertexto. Como esta funcionalidade é baseada em \LaTeX , é necessário que o usuário tenha acesso à informações básicas de sua sintaxe.
- Referências bibliográficas: O usuário pode inserir uma referência bibliográfica e fazer citações no hipertexto. Uma vez inserida pelo usuário, uma referência bibliográfica pode ser usada em qualquer hipertexto editado no *MaRTE*.
- Imagens: O RTE só permite a inserção de imagens a partir de endereço Web, fornecendo sua *URL*. No *MaRTE* é também possível enviar ao servidor um arquivo de imagem disponível na máquina do usuário (figura 3.10) para que esta imagem seja inserida no hipertexto.

Todas as funcionalidades adicionadas ao RTE demandam o uso de um servidor que contenha o *MA* (como o *iGeom*) e o compilador \LaTeX . Na figura 3.11 apresentamos a interface da atual versão do *MaRTE*, com um exemplo de produção de hipertexto.

O editor *MaTE*, e posteriormente o *MaRTE*, mostrou-se uma ferramenta bastante interessante para a produção de conteúdo matemático webmídia e serviu como motivação para o desenvolvimento do *dicionário webmídia*.

3.3.2 Dicionário Webmídia

Segundo o Aurélio (Ferreira, 2004), o significado da palavra *dicionário* é: “Conjunto de vocábulos duma língua ou de termos próprios duma ciência ou arte, dispostos, em geral, alfabeticamente, e com o respectivo significado, ou a sua versão em outra língua”.

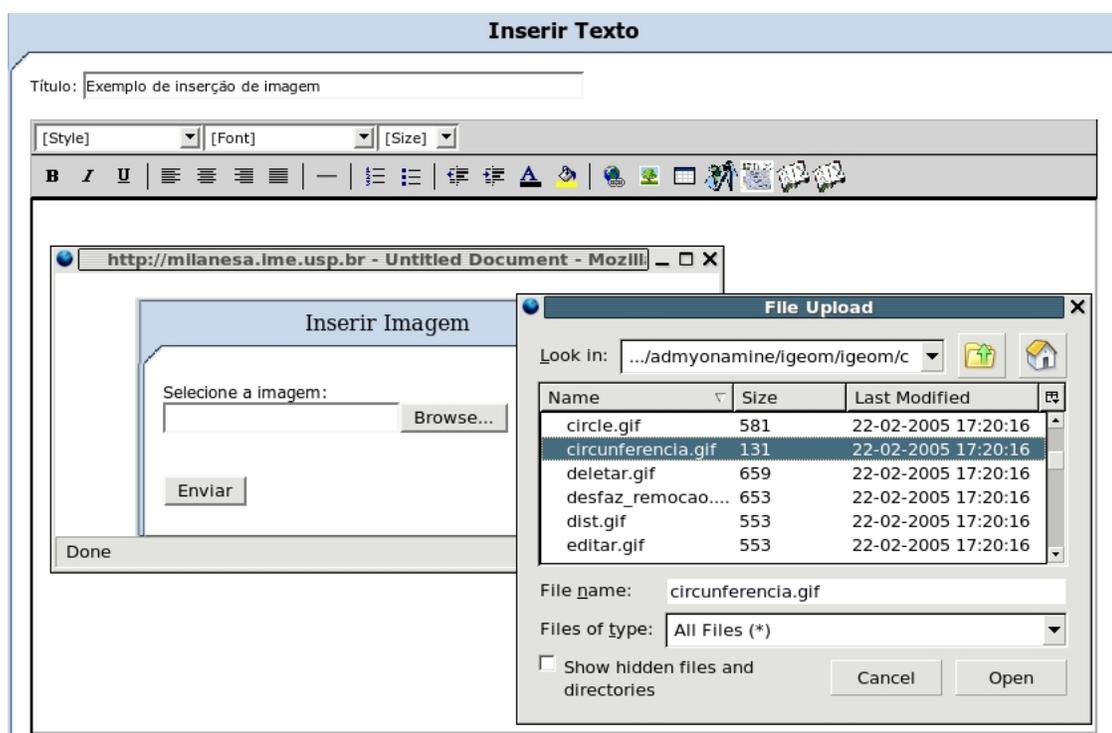


Figura 3.10: Inserção de imagens no MaRTE

Como forma de explorar atividades que exigissem dos alunos uma descrição mais rigorosa dos conceitos matemáticos estudados na disciplina *MAC118*, iniciamos o desenvolvimento do recurso “Dicionário” do *SAW*, cuja primeira descrição foi feita em Moura and de Oliveira Brandão (2005b). Este recurso, agora denominado *Dicionário WebMídia*, permite o desenvolvimento de conteúdo webmídia a partir da criação de hipertextos, onde cada hipertexto é considerado um *vocábulo*. Cada vocábulo é de responsabilidade de seu autor (um aluno ou professor), mas todos os demais participantes no desenvolvimento do conteúdo (demais alunos e professor) podem contribuir para o aperfeiçoamento de um vocábulo através da postagem de comentários. Estas contribuições para o refinamento de um vocábulo caracterizam o *Dicionário WebMídia* como uma ferramenta de construção colaborativa de conteúdo webmídia, onde usamos o termo colaboração como “trabalho conjunto”.

Estrutura do Dicionário Webmídia

O *Dicionário Webmídia* é estruturado para desenvolver conteúdos relacionados a uma determinada área de conhecimento, a qual pode ser, *a priori*, subdividida em subáreas, como é comumente feito em sistemas de classificação. Assim, para cada área de conhecimento pode ser

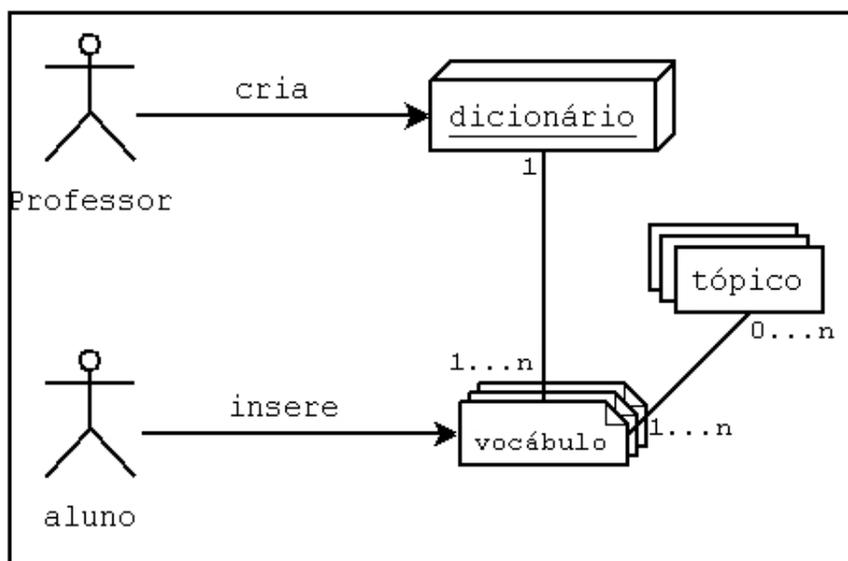


Figura 3.12: Estrutura do dicionário webmídia

fica disponível apenas ao professor, que pode liberá-lo para que os demais alunos postem seus comentários. A liberação de um vocábulo permite que todos os outros alunos cadastrados na mesma turma tenham acesso a esse vocábulo e possam visualizá-lo e inserir comentários para seu autor (figura 3.14). A idéia é que assim o professor estimule o trabalho colaborativo.

A publicação do dicionário é feita após o refinamento dos vocábulos. A publicação pode ser feita seguindo a ordem lexicográfica dos vocábulos ou a ordem lexicográfica dos tópicos. A figura 3.15 apresenta um exemplo de publicação produzido na disciplina *MAC118*.

O *Dicionário Webmídia* foi desenvolvido de modo que o professor possa acompanhar o processo de desenvolvimento de cada vocábulo, pois a edição de vocábulos é feita diretamente no editor do *SAW*, ficando armazenada no banco de dados do sistema mesmo antes de estar finalizada.

Além de outras possibilidades, este recurso pode ser utilizado de modo parecido com uma *Wikipedia* (Aronsson, 2002).

3.3.3 Fórum de discussão

O fórum implementado no *SAW* é um projeto internacional de código aberto, independente do *SAW*, o *phpBB* (phpBB, phpBB). O *phpBB* foi desenvolvido na linguagem de programação *PHP* e distribuído sob a licença GPL. Possui um painel de controle onde é possível administrar todo

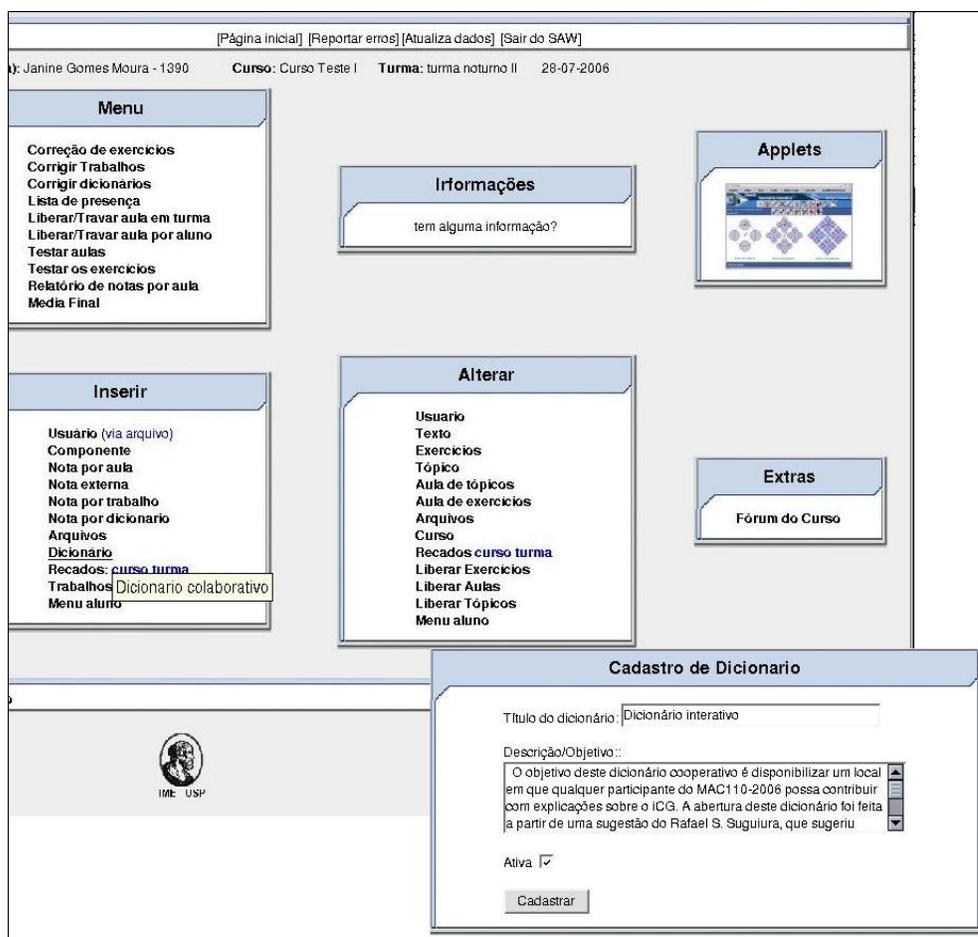


Figura 3.13: Exemplo de criação de um dicionário

o fórum, inclusive sua parte gráfica de forma simples.

Foram feitas algumas alterações no código fonte do *phpBB* para que este pudesse ser incorporado ao *SAW*. Os principais recursos do fórum *phpBB* integrado ao *SAW* são:

- Somente usuários cadastrados no *SAW* possuem acesso ao fórum;
- Quanto uma turma é cadastrada no *SAW*, automaticamente cria-se um fórum associado a esta turma, assim alunos de turmas diferentes possuem cadastrados em fóruns diferentes;
- Toda participação do usuário no fórum é armazenado no banco de dados, assim, caso o professor necessite, é possível desenvolver relatórios com as participações dos alunos;
- Acrescentamos ao editor *phpBB* o recurso de edição de fórmulas matemáticas, baseado no $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.

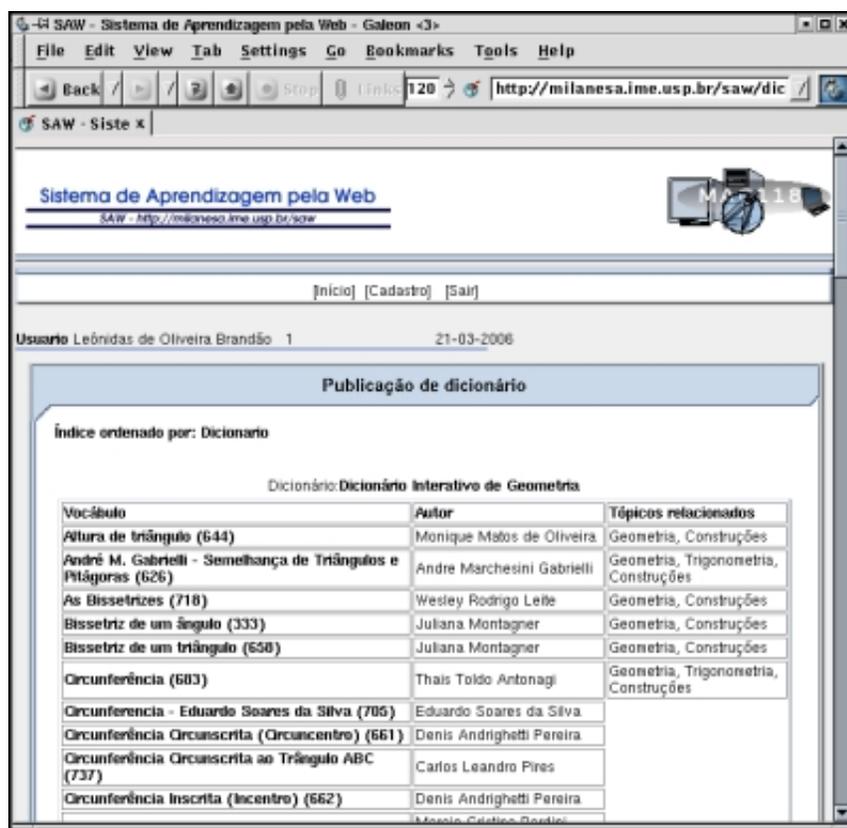


Figura 3.15: Exemplo de uma publicação do dicionário.

Assim que o usuário entra no *SAW*, o sistema verifica se há avisos/recados não lidos e em caso positivo o *link* “Recados” aparece em azul no menu do usuário.

3.3.5 Trabalhos

No *SAW* o professor pode propor trabalhos para serem feitos fora do sistema. Esta funcionalidade possui uma área onde o professor pode digitar o enunciado do trabalho (usando o editor MaRTE) e inserir uma data limite para a entrega. O aluno faz o trabalho fora do *SAW* e pode encaminhar um arquivo com seu trabalho (*upload*) até a data limite. O professor tem ainda a possibilidade de corrigir os trabalhos e inserir uma nota no sistema. A figura B.27 mostra a tela do professor para produzir um trabalho e comentário.

The screenshot shows the phpBB forum interface integrated into the SAW system. The page title is "milanesa.ime.usp.br/saw" and "Fórum do Sistema de Aprendizagem pela Web". The interface includes navigation links like "FAQ", "Pesquisar", "Membros", and "Grupos". It also displays user statistics, such as "1154 Usuários registrados" and "463 mensagens". The footer mentions "Powered by phpBB 2.0.11".

Figura 3.16: Exemplo do fórum phpBB integrado ao SAW

3.3.6 Área para download

Nesta área podem ser disponibilizadas informações e arquivos em duas formas: na forma de *cronograma de aula* ou na forma de área comum para *troca de arquivos* entre os alunos. Na primeira forma só o professor pode disponibilizar informações (como data e descrição da aula) e arquivos. Na segunda forma o aluno envia o arquivo que deseja compartilhar e, após a liberação do professor, o arquivo é disponibilizado para toda a turma. A figura 3.19 mostra a tela para envio de arquivo.

3.3.7 Relatórios

O SAW conta com relatórios que auxiliam o professor no acompanhamento do aluno, sendo:

- Relatório por turma: fornece dados dos alunos referente a uma determinada aula. O professor seleciona um aula e é gerado um relatório, no formato de tabela, com todos os componentes pertencentes a aula e o modelo do aluno para cada componente. Isto é, mostra para cada aluno (pertencente a turma), quais componentes acertou, errou ou se o componente não foi acessado. A figura 3.20 mostra um exemplo de relatório por turma.
- Relatório por turma resumido: É uma versão simplificada do relatório anterior, mostra apenas quantos componentes pertencentes a aula, o aluno errou ou acertou.

Recados	
.. Legenda .. <small>convenções de cores para recados</small>	
não lidos	recados ainda não lidos pelo usuário (destinatário)
lidos	recados já lidos pelo usuário (destinatário)
.. Turma .. <small>recados do professor/monitor para a turma toda</small>	
Leônidas de Oliveira Brandão	Alunos das listas 4 e 5: exercícios
Leônidas de Oliveira Brandão	Dica sobre navegadores (use o Firefox)
Leônidas de Oliveira Brandão	Texto sobre programação: introdução
Leônidas de Oliveira Brandão	Nota final no SAW e JúpiterWeb
.. Enviados para .. <small>recados ENVIADOS pelo usuário (Leônidas de Oliveira Brandão)</small>	
Louise Franco de Oliveira	Sobre "Avaliação"
Pedro Leonardo Silva de Almeida	Re: Deixei uma ajuda ai...
usuario 1	mais um teste, com fórmula
usuario 1	mais um teste, com fórmula
Andre de Camargo Fernandes	Sobre: Por favor,avise os professores
Anelise Eichinger	EP2
Thays Tagliaterra de Grazia	Re: Notas das Provas
Abigail Rodrigues Cizzoto	Você NÃO está matriculado(a) em MAC110 segundo o JúpiterWeb
Bruno César Spósito	Você NÃO está matriculado(a) em MAC110 segundo o JúpiterWeb
Bruno César Spósito	Re: Você NÃO está matriculado(a) em MAC110 segundo o Júpiter
Abigail Rodrigues Cizzoto	Re: Você NÃO está matriculado(a) em MAC110 segundo o Júpiter
.. Recebidas de .. <small>recados específicos RECEBIDOS pelo usuário (Leônidas de Oliveira Brandão)</small>	
usuario 1	Apenas um teste...
Louise Franco de Oliveira	Avaliação
Pedro Leonardo Silva de Almeida	Deixei uma ajuda ai...
Louise Franco de Oliveira	Avaliação

Figura 3.17: Exemplo da funcionalidade Painel de Avisos e Recados

- Relatório por aluno: fornece dados referente a um determinado aluno. O professor seleciona o aluno e é gerado um relatório com todas as aulas cadastradas no curso. Para cada componente pertencente a aula o relatório mostra se o aluno acertou, errou ou se não acessou. A figura 3.21 mostra um exemplo de relatório por aluno.

Nos relatórios gerados, o professor tem a opção de visualizar a resposta do aluno e se essa apresentar alguma inconsistência ele pode mudar a avaliação final do exercício do aluno no banco de dados.

3.3.8 Lista de presença

No SAW é possível manter uma lista de presença dos alunos pertencentes a uma determinada turma. O professor ou monitor insere os dias que quer anotar e presença e o sistema gera um relatório com os alunos que acessaram o SAW nos dias solicitados. O professor pode ainda fazer a lista de presença manualmente, indicando quais alunos estiveram presentes em uma determinada data. Cada aluno têm acesso somente a sua presença.

Cronograma de aula: descrições de aulas ou de arquivos disponíveis por aula

Aula	Data	Arquivo	bytes	Descrição
------	------	---------	-------	-----------

Pasta pessoal na turma: área para arquivos na turma Turma 42

Arquivo: Procurar...

Aula: Data da aula (dd/mm/aaaa):

Descrição do arquivo/aula:

Libera este item para a turma toda? (clique para resposta afirmativa)

Enviar

Figura 3.19: Exemplo da funcionalidade envio de arquivo

acertou é nota 10 e se o aluno errou é nota 0) e calculará a média aritmética.

- **Nota de dicionário:** O professor pode atribuir uma nota para cada vocábulo criado pelos alunos em cada *dicionário webmídia*.
- **Nota de trabalho:** O professor pode atribuir uma nota para cada trabalho enviado pelo aluno através da funcionalidade *Trabalhos*.
- **Nota externa:** O professor tem a opção de inserir no *SAW* notas referentes a atividades que tenha sido realizada fora do *SAW*.
- **Média Final:** O professor pode gerar uma média final, escolhendo quais das notas criadas anteriormente deseja inserir e atribuindo peso para cada uma.

Em todos os casos, cada aluno só tem acesso a sua nota.

3.4 Os usuários

O *SAW* atualmente possui quatro tipos de usuários: administrador, professor, monitor e aluno.

3.4.1 Administrador

Um usuário no papel de administrador tem à sua disposição as seguintes funcionalidades:

1. Inserir/remover *módulos de aprendizagem (applets)* no sistema;

Relatório de aula por turma			
Curso :: Aula			
MAC118 :: Paulo1			
Legenda: ✓ é acerto, ✗ significa que o módulo de aprendizagem avaliou o exercício com incorreto e ∅ significa que o exercício ainda não foi feito!			
Se já foi encaminhada solução para o exercício, entre parênteses ficarão o número de tentativas no exercício e a data do último enviado			
Aluno	Angulo de 60 graus ∅	Círculo com locus	quadrado paralelo
Adilson Marques Almeida (5478315)	✓ (1: 2006-05-12 11:07:00)	✓ (1: 2006-05-12 11:11:00)	✓ (1: 2006-05-12 11:18:00)
Alcides Ferreira Filho (5383824)	✗ (1: 2006-05-12 10:10:00)	✓ (1: 2006-05-12 10:24:00)	✓ (1: 2006-05-12 10:14:00)
Aluno de Teste 2 de Paulo Silva (1112)	∅	∅	∅
Aluno de Teste de Paulo Silva (1111)	✓ (1: 2006-05-12 08:58:00)	∅	∅
Arthur Domingues Tamborino Pinto (5384172)	✓ (1: 2006-05-12 11:09:00)	✓ (1: 2006-05-12 11:11:00)	✓ (1: 2006-05-12 11:19:00)
Caio Moura Quina (5383682)	✓ (1: 2006-05-12 10:09:00)	✓ (1: 2006-05-12 10:46:00)	✓ (1: 2006-05-12 10:16:00)
Carla de Jesus Lima (5383629)	✗ (2: 2006-05-12 10:24:00)	✓ (1: 2006-05-12 10:27:00)	✓ (2: 2006-05-19 11:48:00)
Cleiton dos Santos Matias (5384061)	✓ (3: 2006-05-12 11:16:00)	✓ (1: 2006-05-12 10:48:00)	✗ (1: 2006-05-12 10:24:00)
Cristiane Litoldo Marques (5384099)	✓ (1: 2006-05-12 10:09:00)	✓ (1: 2006-05-12 10:51:00)	✓ (1: 2006-05-12 10:34:00)
Cynthia Militao Domingos (5383293)	✓ (1: 2006-05-12 11:14:00)	✗ (1: 2006-05-12 11:26:00)	✓ (1: 2006-05-12 11:30:00)
Daniel Alejandro Torrez Martins (4896348)	✗ (2: 2006-05-12 11:11:00)	✗ (2: 2006-05-19 11:34:00)	✗ (1: 2006-05-12 11:24:00)
Daniel Gomes da Lapa Junior (5603335)	✓ (1: 2006-05-12 10:09:00)	∅	✓ (1: 2006-05-12 10:21:00)

Figura 3.20: Exemplo de um relatório por turma

2. Inserir ou alterar dados de qualquer outro usuário;
3. Cadastrar cursos e turmas, além de definir os professores responsáveis pelas turmas;
4. É o único que possui permissão para excluir de fato algo da base de dados.

3.4.2 Professor

Algumas das funcionalidades à disposição de usuário no papel de professor são:

1. Criar ou modificar os componentes de conteúdo (cursos, aulas, tópicos, exercícios, textos e exemplos), podendo definir eventuais pré-requisitos e provas *online*. O professor tem papel fundamental na geração do conteúdo, pois os sucessos das atividades propostos e do sistema também dependem de suas habilidades em propor exercícios e montar as aulas, sendo eles os responsáveis pelas escolhas de adaptação, criação e apresentação dos componentes para o aluno.
2. Definir o intervalo de tempo que cada componente fica disponível;

.: Exercício .:	aval.	data	enunciado	resposta do aluno	descarregar arquivo
1.1. Construção de circunferência definida por centro e ponto	correto	2006-03-08 08:55:00	Ver exercício	Ver resposta do aluno	Clique aqui
1.2. Construção de reta passando por A e por B	correto	2006-03-08 09:10:00	Ver exercício	Ver resposta do aluno	Clique aqui
1.3. Construir a semi-reta entre A e B	correto	2006-03-08 09:11:00	Ver exercício	Ver resposta do aluno	Clique aqui
1.4. Construir o segmento entre A e B	correto	2006-03-08 09:12:00	Ver exercício	Ver resposta do aluno	Clique aqui
1.5. Construir ponto interseção entre c0 e r	correto	2006-03-08 09:13:00	Ver exercício	Ver resposta do aluno	Clique aqui
1.6. Reta perpendicular ao segmento AB	correto	2006-03-08 09:15:00	Ver exercício	Ver resposta do aluno	Clique aqui
1.7. Ângulo de 60° em circunferência	correto	2006-03-08 09:18:00	Ver exercício	Ver resposta do aluno	Clique aqui
1.8. Arco de $\frac{\pi}{3}$ em circunferência	correto	2006-03-08 09:20:00	Ver exercício	Ver resposta do aluno	Clique aqui
1.9. Reta com inclinação de 60°	correto	2006-03-08 09:23:00	Ver exercício	Ver resposta do aluno	Clique aqui
1.10. Ponto médio entre A e B	correto	2006-03-08 09:24:00	Ver exercício	Ver resposta do aluno	Clique aqui
2.1. Ângulo em radianos (razão arco/raio)	correto	2006-03-10 11:13:00	Ver exercício	Ver resposta do aluno	Clique aqui
2.2. Ponto médio entre AB	correto	2006-03-10 10:25:00	Ver exercício	Ver resposta do aluno	Clique aqui
2.3. Mediatriz dos pontos A e B	correto	2006-03-10 10:51:00	Ver exercício	Ver resposta do aluno	Clique aqui
2.4. Bissetriz (semi-reta)	correto	2006-03-10 10:54:00	Ver exercício	Ver resposta do aluno	Clique aqui
2.5. Triângulo equilátero	correto	2006-03-10 10:56:00	Ver exercício	Ver resposta do aluno	Clique aqui
2.6. Quadrado sobre AB	correto	2006-03-10 11:06:00	Ver exercício	Ver resposta do aluno	Clique aqui
3.1. Distância mínima 1	correto	2006-03-15 08:22:00	Ver exercício	Ver resposta do aluno	Clique aqui
3.2. Distância mínima 2	correto	2006-03-15 08:26:00	Ver exercício	Ver resposta do aluno	Clique aqui
4.1 Distância mínima entre 3 pontos	erro	2006-03-17 10:29:00	Ver exercício	Ver resposta do aluno	Clique aqui
5.1 Baricentro	correto	2006-03-22 09:35:00	Ver exercício	Ver resposta do aluno	Clique aqui
5.2 Circuncentro	correto	2006-03-22 08:37:00	Ver exercício	Ver resposta do aluno	Clique aqui
5.3 Incentro	correto	2006-03-22 08:45:00	Ver exercício	Ver resposta do aluno	Clique aqui
5.4 Ortocentro	correto	2006-03-22 08:53:00	Ver exercício	Ver resposta do aluno	Clique aqui
6.1 Transladar segmento	correto	2006-03-23 21:02:00	Ver exercício	Ver resposta do aluno	Clique aqui
6.2 Transladar uma circunferência tangenciando reta	correto	2006-03-24 10:20:00	Ver exercício	Ver resposta do aluno	Clique aqui
6.3 Transladar circunferência, para centro em reta	correto	2006-03-24 09:44:00	Ver exercício	Ver resposta do aluno	Clique aqui
6.4 Escada deslizante	correto	2006-03-24 10:07:00	Ver exercício	Ver resposta do aluno	Clique aqui
7.1 Construir reta perpendicular ao eixo x	erro	2006-03-29 09:50:00	Ver exercício	Ver resposta do aluno	Clique aqui
8.1 Tangente	correto	2006-03-31 10:08:00	Ver exercício	Ver resposta do aluno	Clique aqui
8.2 Cosseno	correto	2006-03-31 11:11:00	Ver exercício	Ver resposta do aluno	Clique aqui
13.1 Curva de Nicomedes	correto	2006-04-28 10:39:00	Ver exercício	Ver resposta do aluno	Clique aqui
14.1 Razão especial ($\sqrt{5}/2$)	correto	2006-04-28 10:43:00	Ver exercício	Ver resposta do aluno	Clique aqui

Figura 3.21: Exemplo de um relatório por aluno

3. Escolher algumas características do sistema como: funcionalidades que serão disponibilizadas para o aluno, forma de apresentação e acesso do aluno ao resultado da avaliação de exercícios;
4. Definir e agendar tarefas para os alunos, onde tarefas podem ser trabalhos, textos ou exercícios;
5. Acessar relatórios sobre atividades desenvolvidas pelos alunos.
6. Disponibilizar materiais de apoio para *download*.
7. Criar dicionários, liberar vocábulos dos alunos e publicar o dicionário.
8. Inserir e responder mensagens do fórum.
9. Atualizar a lista de presença dos alunos

3.4.3 Monitor

As funcionalidades disponíveis para um usuário no papel de monitor são definidas pelo professor, sendo limitadas àquelas que próprio professor tem a sua disposição.

3.4.4 Aluno

O usuário aluno é o que dispõe de menos funcionalidades, entretanto ele é o personagem central no sentido dos outros existirem para propiciar-lhe um melhor aproveitamento no curso.

As funcionalidades disponíveis para o aluno são:

- A principal funcionalidade para o aluno é a possibilidade de resolução de exercícios (dentro de aulas ou tópicos). A figura 3.22 apresenta a interface do *SAW* com um exercício no *MA iGeom*.
- Criar vocábulos para o dicionários existentes.
- Inserir e responder mensagens do fórum.
- Visualizar notas e cronograma de aulas inseridos pelo professor.
- Inserir arquivos para compartilhar entre os alunos de uma mesma turma.

Como o *SAW* foi inicialmente projetado a partir da avaliação de exercícios de geometria (utilizando o *MA iGeom*), a participação do aluno é definida principalmente por resolução de exercícios, ou seja, qual exercício fez e qual o resultado da avaliação. O *modelo do aluno* representa o conhecimento e as características do aluno num determinado instante. Para isso é armazenado o caminho percorrido pelo aluno, incluindo os componentes concluídos, número de acessos, tempo que demorou em cada exercício e trabalhos enviados. Dessa maneira o professor pode obter informações detalhadas sobre a avaliação do aluno e o sistema pode indicar os possíveis caminhos que o aluno pode seguir (através de recursos de navegação). A mudança do *modelo do aluno* ocorre de dois modos:

1. Dependências componentes. Cada componente criado pelo professor pode ser pré-requisito de outros componentes do mesmo nível. Por exemplo, o aluno só estará apto para fazer determinada aula se concluir todos os pré-requisitos para esta aula. Internamente são criados grafos de relacionamento, nos quais os nós representam os conceitos e os arcos

Sistema de Aprendizagem pela Web
SAW - <http://milanesa.lme.usp.br/saw>

[Início] [Cadastro] [Sair]

Exercício Multiplicação geométrica entre OA e OB: construa P sobre r tq $\text{dist}(O,P) = (\text{dist}(O,A) * \text{dist}(O,B)) / \text{dist}(O,\text{Unid})$
Saída Resposta: marque P, $\text{dist}(P,O)$ e expressão $(\text{dist}(O,A) * \text{dist}(O,B)) / \text{dist}(O,\text{Unid})$ 1

iGeom Geometria Interativa <http://www.matematica.br>
iGeom - <http://www.matematica.br/iGeom>
Version: 3.6.4

$|B,O| * |A,O| / |U,O| = 1.87$

"Multiplicação geométrica" entre OA e OB
Construa P sobre r tq $\text{dist}(O,P) = (\text{dist}(O,U) * \text{dist}(O,A)) / \text{dist}(O,B)$
Resposta: marque P, $\text{dist}(P,O)$ e expressão $(\text{dist}(O,U) * \text{dist}(O,A)) / \text{dist}(O,B)$

$\text{dist}(O,A) = 1.872$
 $\text{dist}(B,O) = 0.825$
 $\text{dist}(U,O) = 0.850$
 $\text{dist}(A,O) = 1.928$

Draw line

Figura 3.22: Resolução de um exercício no SAW utilizando o MA iGeom.

o tipo de relacionamento (dependência) entre os conceitos, montando assim o *modelo do aluno*.

2. Número mínimo de componentes necessários. Por exemplo, numa aula de exercício, pode-se determinar o número mínimo de exercícios que o aluno deve fazer (e acertar) para que tal conteúdo seja classificado pelo sistema como concluído pelo aluno.

Ao entrar no sistema, figura 3.23, o aluno terá acesso aos cursos em que está cadastrado (um mesmo aluno pode estar cadastrado em vários cursos). Em cada curso estarão acessíveis as aulas disponíveis (definido pelo *modelo do aluno* no sistema, e pela orientação do professor), além de outras informações, como nota, trabalhos entregues e lista de presença, fórum, produção de textos, etc. Dentro de cada aula o aluno terá acesso aos tópicos ou exercícios disponíveis para esta aula, sendo que cada tópico poderá conter um conjunto de exercícios ou textos explicativos. Para que um tópico seja considerado concluído o aluno necessita fazer/examinar: os exercícios/textos obrigatórios e o número mínimo de exercícios/textos. O mesmo princípio se aplica às aulas,

neste caso precisa concluir os tópicos ou exercícios correspondentes.

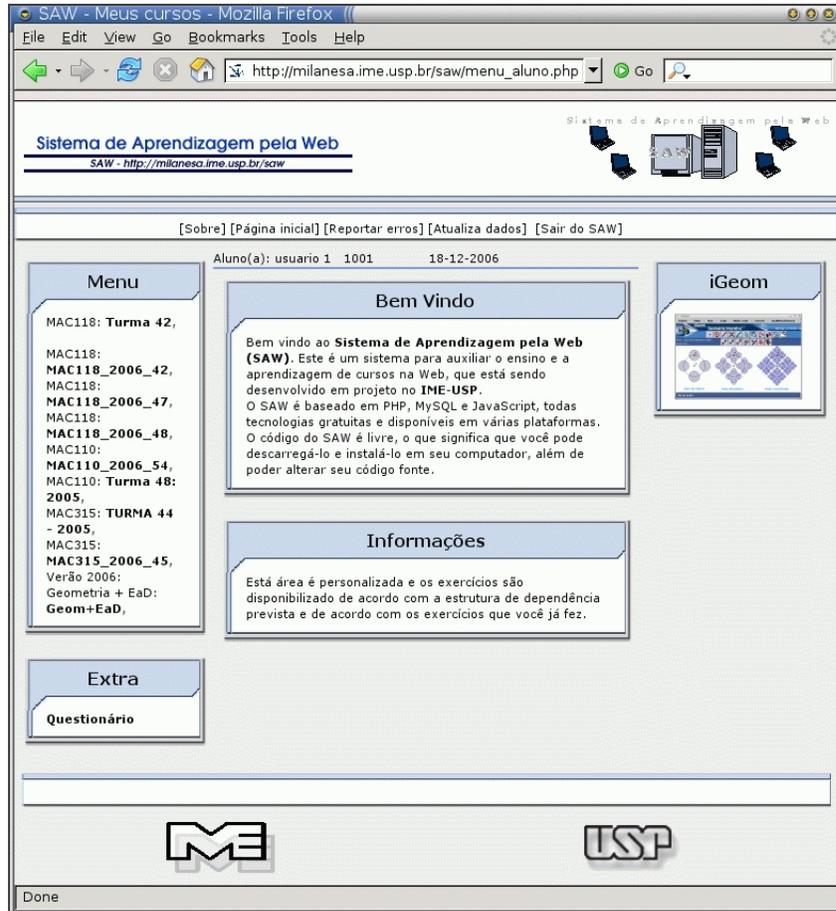


Figura 3.23: Tela inicial do SAW para o usuário aluno.

A apresentação do exercício, figura 3.22, consta de duas partes, uma gerada pelo *SAW* (texto e imagens no HTML) e a outra dentro do próprio *MA*. O aluno pode enviar sua solução ao sistema e eventualmente (se o professor permitir), obter a avaliação do exercício (figura 3.24). Após receber estes dados o sistema indica quais serão as páginas mais apropriadas para o aluno, podendo ser dicas de ajuda ou novos conteúdos.

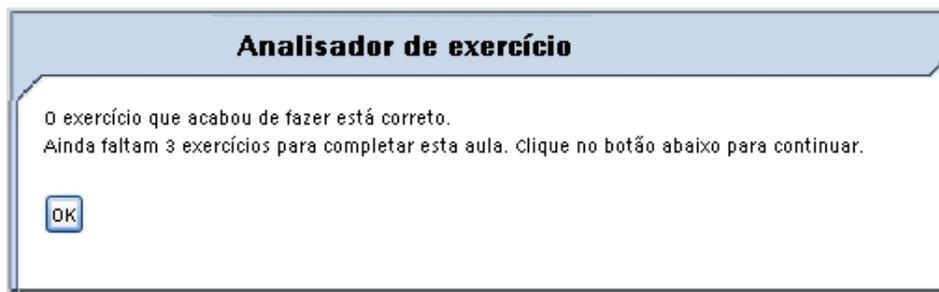


Figura 3.24: Possível resultado da avaliação de um exercício com o *MA iGeom*

Capítulo 4

Estudos de caso

O desenvolvimento do *SAW* teve como motivação inicial a ampliação dos recursos computacionais utilizados na disciplina “Noções de Ensino de Matemática usando o Computador (*MAC118*)” do curso de licenciatura do IME-USP. As aplicações em *MAC118* têm sido a principal motivação para o desenvolvimento do *SAW*.

Nesta disciplina o *iGeom* já era utilizado em sua versão aplicativo e, conjuntamente com o desenvolvimento da primeira versão do *SAW*, foi iniciado, pelo aluno Seiji Isotani e pelo Prof. Dr. Leônidas de Oliveira Brandão, o desenvolvimento de novas funcionalidades do *iGeom*: autoria e validação automática de exercícios, além de comunicação Web para enviar e receber exercícios e exemplos. Assim, a versão *applet* do *iGeom* foi incorporada à primeira versão do *SAW*, para incrementar o processo de ensino/aprendizagem em *MAC118*. A primeira versão do *SAW* dispunha de recursos para a criação de cursos, os quais são formados por componentes. Um componente de curso pode ser uma aula, um tópico, um texto, um exemplo ou um exercício. Além disso, já era possível estabelecer relações de dependência entre os componentes de um curso, durante a definição de sua composição. A partir dos estudos de caso conduzidos com a primeira versão do *SAW*, notamos a necessidade de um editor de hipertexto que permitisse o uso de fórmulas matemáticas. Desta necessidade construímos o editor *MaTE* (Math Text Editor), um editor de textos *HTML* que permite a inserção de fórmulas matemáticas.

A segunda versão do *SAW* foi utilizada em disciplinas durante os anos de 2005 e 2006 considerando três públicos distintos:

- Ferramenta de auxílio à alunos e professores

Em 2005 o *SAW* com o *MA iGeom* foi utilizado nas aulas de geometria para o primeiro ano do ensino médio da escola Lourenço Castanho, da rede particular de ensino. Também em

2005, o *SAW* com o *MA iCG* foi utilizado na disciplina *MAC110* para alunos de graduação do bacharelado em matemática do IME-USP.

- Ferramenta de apoio para preparação a docência e para ensino continuado de professores
Em 2005 e em 2006 o *SAW* com o *MA iGeom* foi utilizado por alunos de graduação em licenciatura em matemática do IME-USP na disciplina *MAC118*. Em 2006 o *SAW* com o *MA iGeom* foi utilizado para auxiliar os professores de matemática no curso de verão do LEM - Laboratório de Ensino Matemática.
- Ferramenta de apoio a curso à distância
Em 2005 e em 2006 o *SAW* foi utilizado na disciplina “Construção de Curso de Geometria à Distância” oferecida nos cursos de verão do IME-USP, para professores de matemática do ensino fundamental II e médio.

Nas próximas seções faremos uma descrição da utilização do *SAW* nas disciplinas citadas e apresentaremos a avaliação do *SAW* a partir da descrição dos resultados de um estudo de caso conduzido ao longo do primeiro semestre de 2006 com alunos da disciplina *MAC118*.

4.1 Escola Lourenço Castanho

A escola Lourenço Castanho é uma escola da rede particular de ensino da cidade de São Paulo. Nesta escola o uso de programas computacionais como ferramentas de apoio ao ensino é muito incentivado e adotado em quase todas as disciplinas, principalmente no ensino fundamental e médio. Nesta escola o *SAW*, com o *MA iGeom*, foi utilizado com os alunos do primeiro ano do ensino médio.

O primeiro ano do ensino médio possui 4 turmas com aproximadamente 30 alunos cada. Nesta série o conteúdo de matemática é dividido em duas disciplinas, Matemática I, onde são ensinados os conteúdos de álgebra e Matemática II, em que são trabalhados os conteúdos de geometria.

O programa de Matemática II inclui técnicas de construções tais como, polígonos regulares e irregulares, trigonometria, entre outros. Entretanto o número de aulas, uma aula semanal de 45 minutos, mostra-se insuficiente para examinar todas as construções propostas, ficando a cargo do aluno fazer muitos exercícios fora do horário de aula. Os exercícios não precisavam ser entregues e a correção dos mesmos era feita geralmente de maneira coletiva, durante a aula, o que não permitia uma avaliação individual do trabalho dos alunos e, conseqüentemente, impossibilitava

o professor de identificar a dificuldade de cada aluno. Em 2004 esta disciplina resultou em quase 80% dos alunos em recuperação no final do primeiro semestre.

No primeiro semestre de 2005 a professora da disciplina Matemática II e a professora de informática desenvolveram um projeto piloto utilizando o *SAW* para as atividades fora do horário de aula. Todos os alunos das 4 turmas foram cadastrados e incentivados a realizar as atividades propostas usando o *SAW* com o *iGeom*, mas o uso do *SAW* não era obrigatório.

O *SAW* com o *iGeom* foi utilizado de duas maneiras:

- nas duas primeiras semanas foi realizada uma introdução ao *SAW* e ao *MA iGeom*, utilizando o laboratório de informática da escola;
- no restante do semestre os alunos trabalharam no *SAW* fora do horário de aula, contando com plantões de dúvidas dos professores em alguns horários.

As aulas no *SAW* foram elaboradas pelos dois professores responsáveis, quase sempre com os mesmo exercícios utilizados nos anos anteriores, e os alunos tinham o período de até 14 dias para resolvê-los. Neste projeto procurou-se observar o desempenho dos alunos que usaram o *SAW* em relação àqueles que não usaram e a relação entre as construções que os alunos fizeram no *iGeom* com as construções feitas em atividades com régua e compasso.

No final do semestre observou-se que em torno de 40% dos alunos usaram o *SAW*. Dentre os que o usaram, aproximadamente 30% ficaram de recuperação, enquanto dentre os que não usaram o *SAW*, aproximadamente 70% ficaram de recuperação.

Uma das aulas elaboradas continha os seguintes exercícios e enunciados:

1. Quadrado inscrito na circunferência

Dada uma circunferência de centro A e raio AB , construir um quadrado inscrito na circunferência, sendo B um dos vértices. Calcule também a apótema do quadrado formado. Selecione como resposta os lados do quadrado e a medida da apótema.

2. Hexágono regular inscrito na circunferência

Dada uma circunferência de centro A e raio AB , construir um hexágono regular inscrito na circunferência, sendo B um dos vértices. Calcule também a apótema do hexágono formado. Selecione como resposta os lados do hexágono e a medida da apótema.

3. Triângulo inscrito na circunferência

Dada uma circunferência de centro A e raio AB , construir um triângulo equilátero inscrito

na circunferência, sendo B um dos vértices. Calcule também a apótema do triângulo formado. Selecione como resposta os lados do triângulo e a medida da apótema.

Um fato interessante ocorrido nesta aula foi que alguns alunos conseguiam fazer as construções no *iGeom* mas não conseguiam fazer o mesmo exercício usando régua e compasso. Este problema foi observado em construções com um grande número de passos, o que nos faz supor que a causa é a facilidade que o aluno encontra no *iGeom*, podendo esconder os objetos intermediários, o que é difícil de ser feito em construções com régua e compasso. A figura 4.1 mostra uma possível solução para o exercício 2, feito no *iGeom*, à esquerda com todos os objetos visíveis e a direita com os objetos intermediários escondidos. Este é um fato que precisa ser analisado em experimentos futuros.

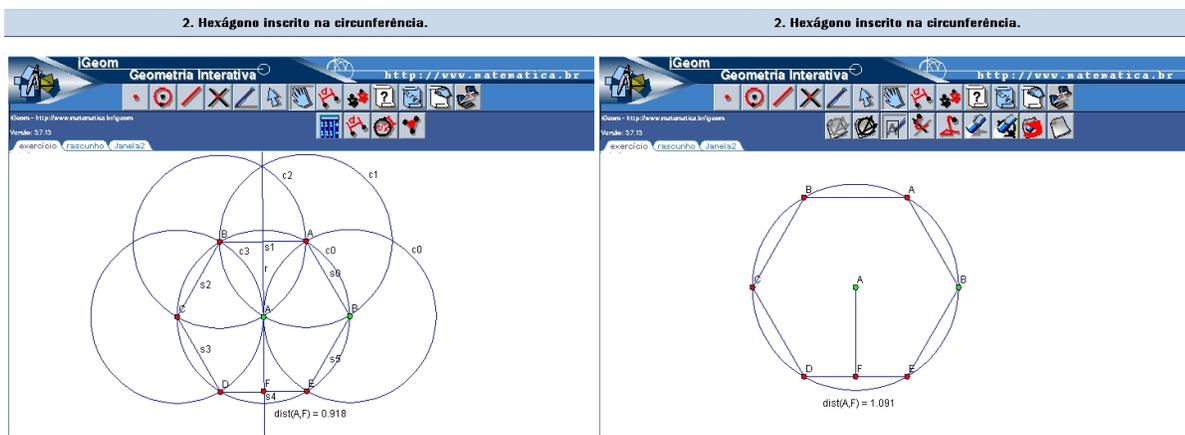


Figura 4.1: Exemplo de resolução do exercício 2 no *iGeom*

Nesta aula também ficou evidenciado que parte significativa dos alunos não dominavam os conceitos de propriedades geométricas. No primeiro exercício, por exemplo, muitos alunos tentaram fazer a construção do quadrado inscrito na circunferência através de “medições”, ou seja, construíam 4 segmentos com pontos pertencentes à circunferência, calculavam a medida dos segmentos e, através da característica dinâmica do *iGeom*, moviam os pontos sobre a circunferência até que os 4 segmentos tivessem a mesma medida.

Entretanto, após as explicações do professor em sala de aula e nos plantões de dúvida, os alunos conseguiram resolver os demais exercícios sem maiores dificuldades. Vale ressaltar que só foi possível verificar esta e outras dificuldades dos alunos através da funcionalidade do *SAW* que gera um relatório com todos os exercícios resolvidos pelos alunos, que permite ainda, que o professor tenha acesso a solução de cada aluno e ao resultado da avaliação da mesma.

Em 2006 não foi possível continuar o projeto, mas em 2007 pretendemos retomar o projeto

com um número maior de exercícios e com obrigatoriedade de acesso ao *SAW*. O fórum será o principal veículo para esclarecer eventuais dúvidas. Pretendemos ainda fazer um experimento com a ferramenta Dicionário Webmídia.

4.2 Cursos do LEM - Laboratório de Ensino de Matemática

Esta é uma disciplina oferecida no programa de cursos de verão do IME-USP e é destinada a professores que atuam nos ensinos fundamental II e médio e tem como objetivo auxiliá-los em como utilizar programas computacionais em suas aulas de matemática. A carga horária da disciplina é de 40 horas e o programa é dividido em 5 módulos:

1. Introdução ao uso do computador e à geometria dinâmica;
2. Funções usando o Winplot;
3. Trigonometria e polinômios através da geometria dinâmica;
4. Construção de mosaicos através da geometria dinâmica;
5. Algoritmos e fractais através da geometria dinâmica.

Esta disciplina é oferecida nos cursos de verão do IME-USP desde 1997 e em 2006 o *SAW*, com o *MA iGeom*, foi introduzido para facilitar o acompanhamento dos alunos.

4.3 Construção de Curso de Geometria à Distância

A disciplina tem sido oferecida no programa de cursos de verão do IME-USP desde 2005, destinada a professores de matemática que atuam nos ensinos fundamental II e médio. O objetivo da disciplina é mostrar algumas possibilidades para o gerenciamento de cursos envolvendo geometria na modalidade educação a distância (EaD). Nas duas primeiras edições da disciplina foi utilizado o *SAW* com o *iGeom*. A carga horária da disciplina é de 20 horas. Em 2006 a disciplina contou com 15 professores, sendo que 93% deles lecionam em escola pública.

Apesar da ementa da disciplina estabelecer como pré-requisito o conhecimento prévio de programas de GD, alguns dos professores não tinham este conhecimento.

Esta disciplina teve 20% da carga horária em atividades à distância, dividida em dois blocos. No primeiro foi examinado o ambiente para usuário do tipo aluno e no segundo examinou-se o

ambiente para usuário do tipo professor. No primeiro bloco os professores de matemática fizeram atividades de alunos para entender as possibilidades do sistema e no segundo bloco eles deviam desenvolver atividades como responsável por uma turma fictícia de um curso de geometria.

As aulas à distância foram feitas com hora marcada, onde os professores inscritos no curso podiam se comunicar com os professores responsáveis através do fórum do *SAW*. Nas aulas à distância eles tinham que desenvolver o conteúdo para o curso e outro participante atuava como aluno para testar na prática os cursos elaborados.

Com os professores desenvolvendo conteúdos para os cursos criados, notamos que algumas funcionalidades do *SAW* não eram intuitivas para o professor. A partir destas constatações desenvolvemos algumas melhorias no *layout* e nas funcionalidades do *SAW* (como a forma de inserção de conteúdo), além de desenvolver um manual para o professor, anexo B.

4.4 MAC110

A disciplina “Introdução à Computação” (MAC110) é obrigatória para os alunos de bacharelado em matemática do Instituto de Matemática e Estatística da USP. O objetivo desta disciplina é introduzir a programação de computadores através do estudo de uma linguagem algorítmica e de exercícios práticos.

O *SAW*, com o *MA iCG*, foi utilizado no início do curso para introduzir um modelo de computador e o conceito de programação. No decorrer da disciplina foram utilizadas as funcionalidades fórum, cronograma, recados e trabalhos.

4.5 MAC118

Noções de Ensino de Matemática usando o Computador (MAC118) é uma disciplina obrigatória para os alunos de licenciatura em matemática do Instituto de Matemática e Estatística da USP e, atualmente, é totalmente ministrada com uso o *SAW*. O objetivo desta disciplina é apresentar programas computacionais que possam ser usados no ensino/aprendizagem de matemática. A metodologia da disciplina consiste em mostrar ao licenciando como tais programas podem ajudá-lo nas atividades de ensino, tornando suas aulas mais dinâmicas, e como poderiam ajudar um aluno em seu aprendizado. Assim, ao longo do semestre são propostas várias atividades que poderão ser aplicadas em sala de aula pelos futuros professores. Do ponto de vista didático é esperado que as atividades propostas, após reflexão dos licenciandos, sejam discutidas com toda

a turma. Geralmente, soluções encaminhadas por licenciandos ao professor são apresentadas a todos para estimular as discussões. Tal fato pode levar a uma sobrecarga de trabalho para o professor, devido ao tamanho da turma (entre 60 e 70 alunos).

Considerando a metodologia da disciplina, todas as aulas de *MAC118* são ministradas em laboratório de informática, com dois alunos por micro e, em anos anteriores a 2004, eram propostas atividades que deveriam ser resolvidas com a ajuda de programas de *Geometria Dinâmica (GD)* (Gravina, 1996) e com menor destaque, outros como *Scilab* e *MuPad* (Scilab, 2007; MuPAD, 2007). Estas atividades demandavam tempo razoável, pois era necessário aguardar a conclusão da atividade pela maioria dos alunos, a fim de apresentar sua solução.

Observou-se que o tempo entre proposição e solução das atividades não só impedia que o professor acompanhasse o rendimento da turma, como também que os alunos seguissem, individualmente, o seu ritmo de aprendizagem. Como forma de diminuir o tempo entre a proposição da atividade e a apresentação da sua solução e dar ao aluno a oportunidade de executá-la em seu ritmo, o uso do *SAW* foi introduzido na disciplina em 2004.

Como já citado, em 2004, o *iGeom* ganhou uma nova funcionalidade para ser melhor empregada com o *SAW*, a autoria e avaliação automática de exercício Isotani (2005). Com a uso do *SAW* e *iGeom* passou a ser possível: a imediata apresentação ao aluno da avaliação de sua solução; a definição de sequências de atividades proposta pelo professor através de pré-requisitos, permitindo que o aluno siga o seu ritmo de aprendizagem; a execução de um número maior de atividades por aula; o acompanhamento do rendimento da turma através de relatórios individuais fornecidos pelo sistema; o reuso de atividades propostas em edições anteriores da disciplina.

Vale observar que a experiência obtida com o uso do *SAW* em *MAC118* propiciou o levantamento de outros requisitos funcionais desejáveis para o sistema como, por exemplo, a produção de textos com conteúdo matemático (Moura and de Oliveira Brandão, 2005a).

4.5.1 Análise do SAW

A fim de avaliar o desempenho e a utilidade do *SAW*, vários estudos de caso foram conduzidos desde 2004 na disciplina *MAC118*, todos de forma *ad hoc*. A partir do primeiro semestre de 2006 adotamos a metodologia descrita em Yin (2005) como embasamento para conduzir o estudo de caso. Assim, os procedimentos adotados para a coleta de dados foram:

- Documentação: Analisamos as atividades produzidas pelos alunos durante o semestre;

- Entrevistas: Elaboramos questionários direcionados. Em vários itens onde a razão da escolha era relevante, havia um quadro onde o entrevistado poderia comentar seus argumentos.
- Observação direta e participante: Através do acompanhamento pessoal do curso no papel de monitor, realizamos observações qualitativas diretas como por exemplo sobre o comportamento dos alunos, sobre o tempo que os alunos levaram para realizar as atividades propostas, entre outras.

Nas subseções seguintes apresentamos separadamente as avaliações qualitativas e quantitativas para as 3 turmas de 2006.

1. Resultados da avaliação qualitativa

Para a avaliação qualitativa foram realizadas entrevistas com os alunos da disciplina *MAC118* através da aplicação de questionários. O primeiro questionário (pré-teste) foi aplicado no primeiro dia de aula, imediatamente após o primeiro contato dos alunos com o *SAW*. O segundo questionário (pós-teste) foi aplicado no final do semestre.

O questionário pré-teste foi respondido por 60 alunos e teve por objetivo constatar qual o grau de familiaridade do aluno com o computador, qual o nível de conhecimento em construções geométricas, qual o conhecimento sobre sistemas de gerenciamento e sobre educação à distância e fazer uma avaliação da primeira aula utilizando o *SAW*.

A faixa etária dos alunos variava entre 18 e 40 anos, sendo que 78% dos alunos tinham entre 18 e 23 anos e 80% faziam o uso do computador pelo menos 5 vezes na semana. Dos que responderam ao questionário, 30% disseram já ter lecionado ou feito alguma disciplina utilizando o computador e 48% disseram já ter feito algum curso à distância. Porém quando perguntado “*Qual ambiente virtual utilizou*”, nenhum aluno soube responder adequadamente, o que pode significar que os alunos não entenderam a questão ou que não lembraram qual foi o ambiente usado.

Na primeira aula os alunos tiveram que realizar 10 exercícios simples com o *iGeom* e submeter a resposta ao sistema. Como o *iGeom* é um *MA* com validação de exercícios, juntamente com a solução do aluno é enviada a informação resultante da avaliação, certo ou errado. Na hipótese de solução errada, o *iGeom* apresenta uma configuração da solução do licenciando na qual o erro fica mais evidente. Os licenciandos foram solicitados a opinar sobre a imediata avaliação de suas soluções e as respostas revelaram que 85% acharam a funcionalidade muito boa ou boa. O ponto negativo apontado por alguns alunos foi o fato

de o *iGeom* não mostrar exatamente em que ponto a construção estava errada. Excetrandose um dos alunos, os demais acharam a aula válida e 92% afirmaram terem compreendido como utilizar o sistema *SAW*.

Um dado curioso foi que 65% responderam que não fariam um curso totalmente à distância. A justificativa de mais de 50% destes dizia respeito à importância do contato com o professor em sala de aula. Dos alunos que fizeram curso a distância, nenhum utilizou sistema de gerenciamento que possuísse ferramentas de cooperação e/ou colaboração. Os que relataram terem feito curso a distância utilizaram somente mídias, como vídeos e textos, durante o curso.

Quanto ao conhecimento de construções geométricas utilizando régua e compasso, 60% afirmaram terem pouco ou nenhum conhecimento. Este dado chama a atenção uma vez que o tópico “construções geométricas” faz parte do programa oficial do ensino fundamental. Porém a deficiência dos alunos em relação às construções foi comprovada em sala de aula, pois muitos não conseguiram resolver problemas simples de construção geométrica.

O segundo questionário foi aplicado no último dia de aula, após uma avaliação individual, com 31 respostas. Neste questionário procuramos medir o uso do *SAW* e as dificuldades encontradas nas ferramentas utilizadas. O questionário foi baseado na escala de Likert (Likert, 1932). Nas questões relativas à avaliação do *SAW* usamos a classificação 1=fraco, 2=regular, 3=bom, 4=muito bom e 5=excelente. Nas questões relativas às ferramentas do dicionário webmídia usamos a classificação 1=não utilizei, 2=fraco, 3=regular, 4=bom, 5=muito bom (figura 4.2).

Dos que responderam ao questionário, 80% consideraram o *SAW* um sistema ótimo, muito bom ou bom e nenhum aluno considerou o *SAW* um sistema fraco. O número de alunos que acreditam que o *SAW* estimulou o estudo na disciplina *MAC118* foi de 80%.

Uma característica do sistema que também foi aprovada pela maioria dos entrevistados foi a apresentação do resultado certo ou errado das soluções de atividades enviadas pelo sistema. Neste caso, 71% consideram ótimo, muito bom ou bom. Relativamente à usabilidade do sistema, 87% não tiveram dificuldade em encontrar as atividades disponibilizadas pelo professor e 35% relataram dificuldade em compreensão dos enunciados propostos. Em relação ao modo de apresentação dos exercícios, todos exceto um, classificaram como ótimo, muito bom ou bom. O ponto de maior discussão foi em relação à ferramenta dicionário webmídia.

Esta funcionalidade foi utilizada em uma atividade cuja avaliação correspondia a 30% da

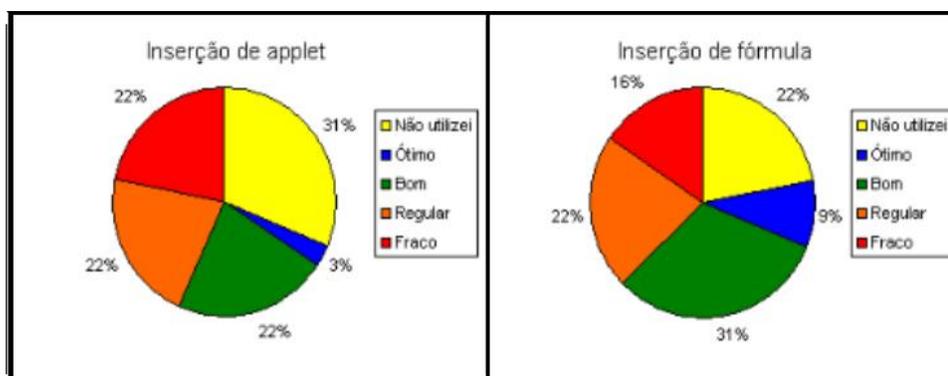


Figura 4.2: Análise de uso: inserção de applets e fórmulas

nota final devido a complexidade a ela associada. Nesta atividade os alunos deveriam escolher um tema de geometria e, após aprovação do tema pelo professor, elaborar um hipertexto com demonstrações de teoremas utilizando o editor do *SAW* (MaTE) com todos seus recursos (inserção de fórmulas, *MA* e imagens). Muitos alunos não tinham conhecimento da linguagem $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, o que dificultou a inserção de fórmulas matemáticas. Além disso, a inserção de *applets* também apresentou problemas, pois o editor permitia a inserção de, no máximo, 10 *applets* o que foi considerado insuficiente por alguns alunos.

2. Resultados da avaliação quantitativa

Para a avaliação quantitativa foram analisados: as atividades realizadas durante a aula e suas respectivas notas; as provas realizadas durante o semestre; e a *atividade final*, realizada utilizando a funcionalidade *Dicionário Webmídia*.

No início de *MAC118* haviam 65 alunos sendo que 7 alunos desistiram, 33 alunos foram aprovados, 22 alunos ficaram de recuperação e 3 alunos foram reprovados. O alto índice de recuperação foi devido à cópias no trabalho final.

Devido ao alto índice de participação dos alunos na resolução dos exercícios, a análise relacionando esta participação com o índice de aprovação não permitiu qualquer conclusão. O percentual médio de atividades realizadas por aluno foi de 88%, sendo que 60% obtiveram mais de 95% de acerto. Apesar disso, o número de alunos em recuperação foi alto. Isso se deveu ao fato de que, dentre os alunos que ficaram de recuperação, 95% teve sua atividade final classificada como plágio. O professor da turma relatou que estes copiaram frases *ipsis-literis* de livros e de páginas Web, sem nem mesmo citar a fonte.

Ao longo do semestre a disciplina teve 28 aulas práticas e 2 aulas de avaliação individual

(prova), totalizando 120h/aula em laboratório. Em cada aula prática eram realizados pelo menos 2 atividades usando o *iGeom*, totalizando 60 atividades propostas ao final do semestre. As aulas de avaliação individual consistiram da proposição de 3 atividades envolvendo o *iGeom*, realizadas em laboratório, com 1 aluno por computador. Nestas avaliações, apenas 2 alunos obtiveram nota abaixo da mínima para aprovação. Entre os alunos que obtiveram, nas avaliações individuais, média maiores que 8, 98% obtiveram média maiores que 9 (figura 4.3). Este dado é indicativo de que a resolução de um maior número de atividades em sala de aula (laboratório) contribui para melhor desempenho em avaliações individuais (provas).

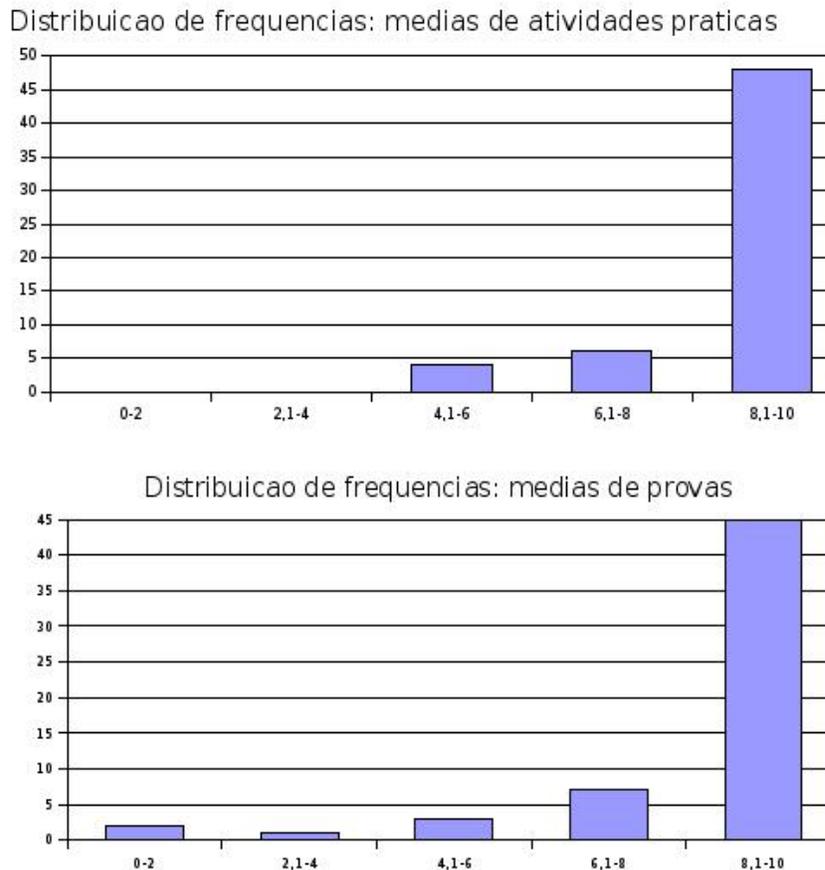


Figura 4.3: Comparação: médias de exercícios e provas

O uso da funcionalidade *Dicionário Webmídia* foi avaliado a partir da atividade final proposta no curso, descrita no item anterior. Todos os alunos criaram pelo menos um hipertexto relativo a um vocábulo de geometria e enviaram ao sistema, totalizando 59 hipertextos enviados. Dos hipertextos enviados, 42% incluíam *MA*, 80% incluíam imagens,

63% incluíam fórmulas matemáticas e 61% incluíam bibliografia usando os recursos de edição do *SAW* (figura 4.4).

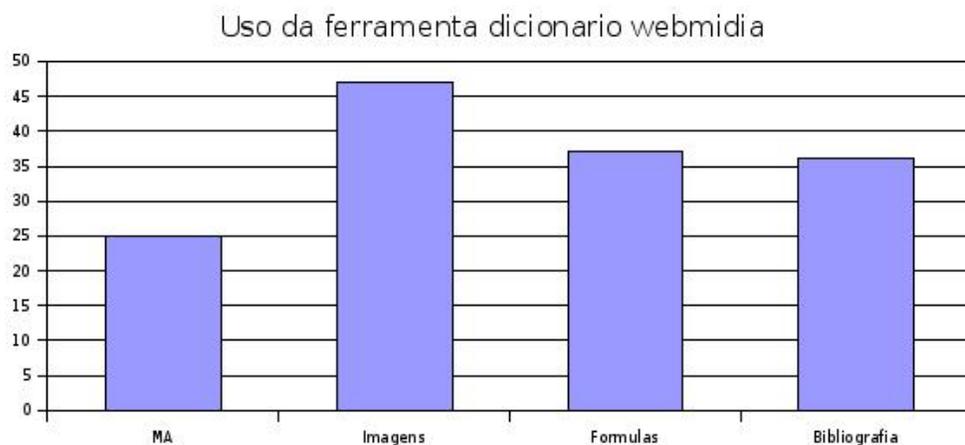


Figura 4.4: Utilização relativa de ferramentas na edição da atividade final

Estes dados mostram que a inserção de conteúdos pouco explorados pelos alunos no dia-a-dia (*MA*, fórmulas matemáticas e bibliografia) precisa ser melhor explorada, por exemplo com a realização de atividades de produção de textos ao longo do curso.

Capítulo 5

Conclusões

Neste trabalho apresentamos o *SAW* e suas funcionalidades. Discutimos algumas aplicações deste tipo de sistema e as motivações que apontaram a necessidade das funcionalidades desenvolvidas no *SAW*.

Uma parte importante deste trabalho foi o desenvolvimento do *SAW*, um sistema de gerenciamento de cursos, que permite a incorporação de módulos de aprendizagem que podem incrementar a interatividade e ajudar o professor no acompanhamento dos alunos. Os módulos de aprendizagem (MA) são *applets* Java que permitem que partes das operações de avaliação do aluno ocorram na própria máquina cliente, liberando o servidor para outras tarefas. Além disso, os MAs podem ser facilmente inseridos e possibilitam a troca de domínio sem a necessidade de mudanças internas do sistema, conferindo-lhe maior flexibilidade e interatividade no auxílio ao ensino/aprendizagem do aluno.

Neste trabalho foi apresentado o uso do *iGeom* como módulo de aprendizagem, porém outros dois MA estão em desenvolvimento para serem acoplados ao sistema: um para o ensino funções, o *iGraf*, e outro para o ensino de programação, o *iCG*, que pode ser testado a partir do endereço Web <http://www.matematica.br/programas/icg>. Sendo que este último já testado em uma disciplina de introdução à programação (MAC0110) no curso de licenciatura em matemática do IME-USP.

O uso do *SAW* com o MA *iGeom* em laboratório de informática resultou num envolvimento muito maior dos alunos nas atividades. Na disciplina *MAC118*, por exemplo, em semestres anteriores ao uso do *SAW*, poucos alunos apresentavam sugestões para resolver os exercícios propostos, enquanto que após a introdução do *SAW* com o MA *iGeom*, mais alunos passaram a fazê-lo. Devido à individualização possibilitada pelo sistema, o envolvimento dos alunos com

as atividades propostas foi incrementado. Um exemplo disso pôde ser medido com a própria introdução do conceito de Geometria Dinâmica. Desde o início de 2005, na primeira aula de *MAC118* tem sido apresentados 10 exercícios, com o objetivo de levar os alunos descobrirem o que é Geometria Dinâmica através do iGeom. Nesta aula é mostrado ao aluno apenas como acessar o ambiente (sua conta) e como enviar suas respostas aos exercícios. Em 2005 e 2006, o número médio de exercícios resolvidos e encaminhados com sucesso nesta atividade foi de aproximadamente 90%.

Apresentamos alguns resultados de uso da geometria dinâmica no ensino, apontando alguns problemas de nível de conhecimento de geometria nas turmas consideradas. Estas aplicações motivaram o desenvolvimento de um novo recurso no *SAW*, que foi inserido no sistema como uma funcionalidade para o desenvolvimento de dicionários interativos, visando fornecer um espaço para que o aluno possa escrever texto com formalismo matemático. A partir dos estudos de caso realizados pudemos notar que a utilização da funcionalidade “Dicionário Webmídia” pode ajudar a combater um engano que se mostrou comum aos alunos com pouca experiência matemática: que uma construção da geometria dinâmica, na qual se observa determinada propriedade, não é uma demonstração.

Também mostramos os resultados obtidos num estudo de caso conduzido com uma turma de *MAC118* no primeiro semestre de 2006. Neste estudo de caso foram avaliadas algumas funcionalidades do *SAW* e seu impacto no processo de ensino/aprendizagem da disciplina. Dos itens avaliados neste estudo através da realização de questionários, citamos dois que merecem destaque:

- forte indicativo de que a utilização de MAs com validador automático de atividades agilize o processo de ensino/aprendizagem;
- a necessidade de explorar a redação de textos/demonstrações matemáticas, como ficou evidenciado no fraco desempenho dos alunos na atividade final.

Relativamente ao sistema propriamente dito, os resultados apontam indícios de que a utilização do *SAW* como ferramenta motivadora e facilitadora no processo de ensino/aprendizagem da disciplina *MAC118* propiciou:

- grande participação dos alunos em todas as atividades propostas;
- melhor aproveitamento dos alunos;

- diminuição da carga de trabalho do professor, em relação ao acompanhamento do aluno;
- necessidade de estender o editor de textos MaTE, para que o mesmo tivesse características WYSIWYG. Esta última observação nos motivou a desenvolver uma nova versão do editor de hipertexto do *SAW* a partir do editor *RTE* (*Rith Text Editor*) (rte, 2007), que passamos a denominar *MaRTE* (*Math Rich Text Editor*).

5.1 Trabalhos Futuros

Em versões futuras do *SAW* a forma de armazenar os dados poderá ser modificada para que o sistema permita o uso de XML, com utilização de algum padrão (como SCORM (SCORM, SCORM)) que facilite a migração de conteúdos entre o *SAW* e outros *SGC*.

Em relação a extensão deste trabalho, destacamos duas direções de interesse. Numa delas, relativa ao desenvolvimento de sistemas e funcionalidades seria interessante:

- incorporar os recursos do *SAW* para gerenciamento de componentes com dependências e módulos de aprendizagem em sistemas abertos como o Moodle (Moodle, 2007) ou como o Tidia (TIDIA, 2007).
- desenvolver novos *MA*s que possam ser incorporados ao *SAW*. Isso implica na pesquisa de formas interessantes de se conseguir avaliações automáticas nestes *MA*s.

Noutra direção, relativa à aprendizagem, existem várias questões interessantes a serem investigadas, como:

- analisar diferenças de aprendizagem entre a Geometria Dinâmica (GD) e ao uso da régua e compasso, particularmente se o uso da GD permite uma abstração mais rápida ou não.
- examinar em mais turmas o efeito da geometria dinâmica no entendimento de demonstração matemática.
- realizar novos estudos de caso com a extensão do editor de textos MaTE não apenas na disciplina *MAC118*, mas também em disciplinas de Geometria, Cálculo Diferencial e Integral e Introdução a Computação.

Apêndice A

Arquitetura

<i>Camada de Tarefas</i>		
<i>Autoria</i>		<i>Adaptação</i>
<i>Professor</i>	<i>Administrador</i>	
alterar_turma.php	alterar_applet.php	resolucao.php
alterar_turma2.php	alterar_applet2.php	analisa.php
alterar_componente.php	alterar_curso.php	processa.php
alterar_componente2.php	alterar_curso2.php	confere.php
alterar_usuario.php	alterar_usuario.php	
alterar_usuario2.php	alterar_usuario2.php	
alterar_arquivo.php	alterar_turma.php	
alterar_arquivo2.php	alterar_turma2.php	
alterar_trabalho.php	inserir_applet.php	
alterar_trabalho2.php	inserir_applet2.php	
alterar_menu_aluno.php	inserir_curso.php	
alterar_menu_aluno2.php	inserir_curso2.php	
alterar_aula_turma.php	inserir_usuario.php	
alterar_aula_turma2.php	inserir_usuario2.php	
alterar_dicionario.php	inserir_usuario_arquivo.php	
alterar_dicionario2.php	inserir_usuario_arquivo2.php	
inserir_usuario.php		
inserir_usuario2.php		
inserir_componente.php		
inserir_componente2.php		
inserir_arquivo.php		
inserir_arquivo2.php		
inserir_trabalho.php		
inserir_trabalho2.php		
inserir_dicionario.php		
inserir_dicionario2.php		

Tabela A.1: Relação de páginas com a arquitetura na *Camada de Tarefas*

<i>Camada de Comunicação</i>
funcoes.php
funcoes_aluno.php
funcoes_js.js
funcoes_usuario
definicoes.inc
configuracoes.inc
funcoes_admin.php
funcoes_dicionario.php
funcoes_textos.php
funcoes_recados.php

Tabela A.2: Relação de páginas com a arquitetura na *Camada de Comunicação*

<i>Camada de Interação</i>			
<i>Professor</i>	<i>Aluno</i>	<i>Administrador</i>	<i>Geral</i>
menu_prof.php	menu_aluno.php	menu.php	menu_usuario.php
opcoes_prof.php	meu_curso.php		home_user.php
libera_aula.php	insere_arquivo.php		logoff_total.php
libera_aula2.php	insere_arquivo2.php		saida.php
libera_vocabulo.php	comentar_vocabulo.php		reportar.php
libera_vocabulo2.php	comentar_vocabulo2.php		index.php
libera_aula_aluno.php	alterar_recado.php		
libera_aula_aluno2.php	alterar_recado2.php		
libera_componente.php	incluir_recado.php		
libera_componente2.php	incluir_recado2.php		
exercicio_prof.php	incluir_vocabulo.php		
grava_arquivo.php	incluir_vocabulo2.php		
grava_resposta.php	alterar_vocabulo.php		
relatorio_aluno.php	alterar_vocabulo2.php		
relatorio_aluno2.php	dicionario.php		
relatorio_turma1.php	dicionario2.php		
relatorio_turma2.php	aula_exercicio.php		
relatorio_opcao.php	exercicio.php		
ver_resposta.php			
ver_resposta2.php			
ver_exercicio.php			
visualiza_texto.php			
publicar_dicionario.php			
publicar_dicionario2.php			
comentar_vocabulo.php			
comentar_vocabulo2.php			
alterar_recado.php			
alterar_recado2.php			
incluir_recado.php			
incluir_recado2.php			
incluir_topico.php			
incluir_topico2.php			
alterar_topico.php			
alterar_topico2.php			
inserir_nota.php			
inserir_nota2.php			
alterar_nota.php			
alterar_nota2.php			

Tabela A.3: Relação de páginas com a arquitetura na *Camada de Interação*

Apêndice B

Manual do Professor

B.1 Primeiro passo

No *SAW* todo usuário deve possuir cadastro com *login* e senha. Para solicitar cadastro no *SAW*, deve-se clicar no *link* “**Solicitar cadastro**” na página inicial do *SAW*, figura B.1.

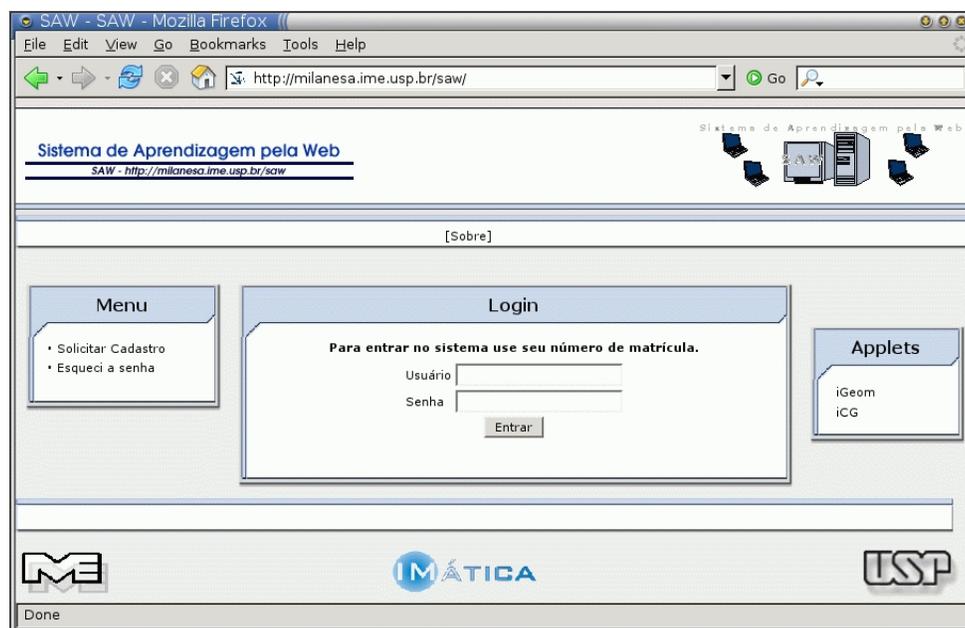


Figura B.1: Interface inicial do SAW

B.1.1 Cadastro de disciplina

No *SAW* uma disciplina é cadastrada como curso, onde um curso pode ter várias turmas. Para cadastrar um curso ou turma no *SAW*, o professor deve enviar um email ao administrador

do sistema informando: nome e sigla do curso e nome da turma. Após receber o email de confirmação de cadastramento, acesse o sistema e procure o curso/turma. Um mesmo usuário pode estar cadastrado em vários cursos/turmas. Ao selecionar alguma turma aparecerão todas as funcionalidades que o professor pode utilizar em seu curso, figura B.2.

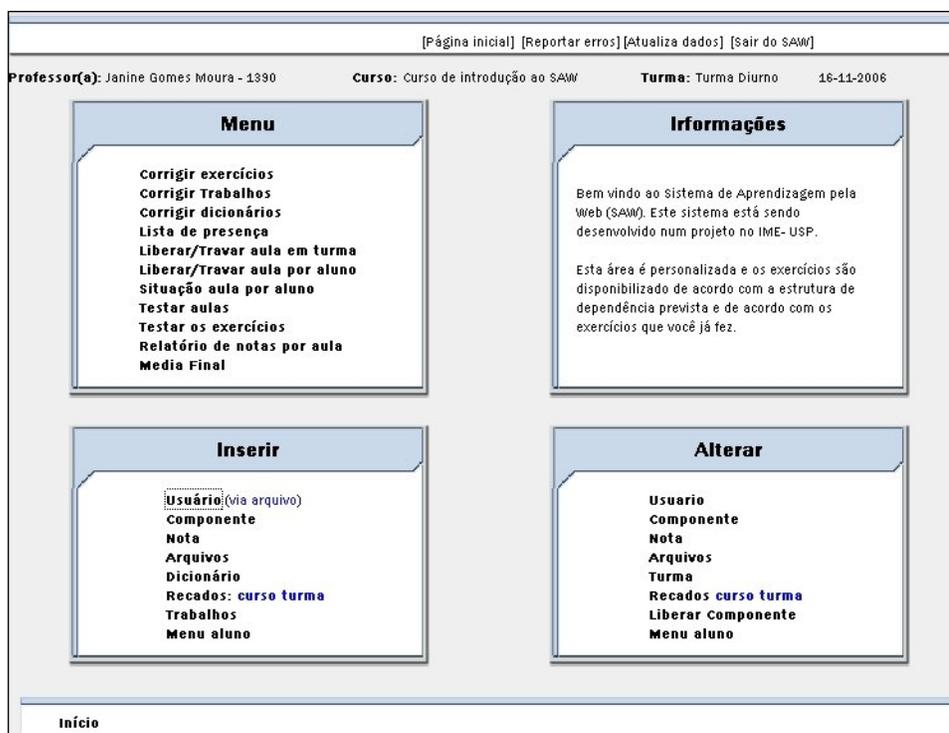


Figura B.2: Interface com o menu do professor

A tela inicial do curso é composta por módulos. A seguir é apresentado o módulo de inserção para o professor.

B.2 Módulo Inserir

B.2.1 Usuário

O *SAW* possui quatro tipos de usuários: administrador, professor, monitor e aluno. O professor pode inserir usuários do tipo monitor e aluno. Existem duas maneiras de inserir usuários no *SAW*:

- **Individual**

Para cadastrar um aluno no sistema o professor deve utilizar o *link* “usuário” no módulo

Inserir. Ao entrar na interface de inserção o professor deve digitar o nome completo do usuário, o email e um número de *login*. O número de *login* tem que ser único e deve conter somente números. É com este número que o usuário poderá se conectar ao sistema. Se o quadro **ativa** estiver clicado, significa que o cadastrado será válido a partir do momento que o usuário for cadastrado. Caso contrário o usuário será cadastrado no sistema mas não poderá acessar até que o professor o libere. Por último deve-se escolher o tipo de usuário (monitor ou aluno) que está sendo inserido e clicar em “**cadastrar**”. A interface de inserção é mostrada na figura B.3.

Cadastro de usuario

O usuário será cadastrado no curso: **Curso de introdução ao SAW** e turma: **Turma Diurno**

Nome:

Login: (a senha inicial do usuário é o mesmo do login)

E-mail:

Tipo de usuário: ▼
Selecione um tipo de usuário
Monitor
Aluno

Ativa:

Figura B.3: Interface de inserção de usuários

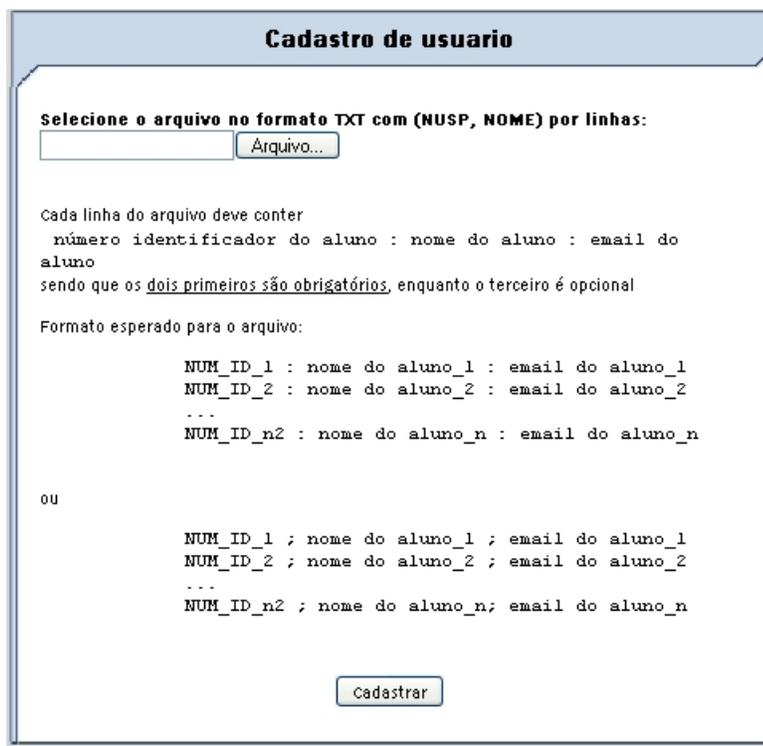
- **Em grupo**

O professor tem a opção de cadastrar um grupo de usuários alunos de uma só vez. Para isto deve-se criar um arquivo (extensão txt) contendo:

```
número identificador do aluno : nome do aluno : email do aluno
```

Utilizando o *link* “**usuário - via arquivo**”, no módulo **Inserir**, o professor tem acesso a interface de inserção de alunos por arquivo, figura B.4. Para inserir o grupo de alunos basta inserir o arquivo criado anteriormente e clicar no botão “**cadastrar**”.

Se um usuário já cadastrado é inserido no *SAW*, o sistema informa que o usuário já está cadastrado e verifica se o usuário possui cadastrado no curso e turma desejado, se o usuário está cadastrado mas não está inscrito no curso/turma desejado o sistema faz a inserção solicitada. Desta maneira um mesmo usuário pode fazer parte de diversos cursos e turmas com diferentes



Cadastro de usuario

Selecione o arquivo no formato TXT com (NUSP, NOME) por linhas:

Arquivo...

Cada linha do arquivo deve conter
número identificador do aluno : nome do aluno : email do aluno
sendo que os dois primeiros são obrigatórios, enquanto o terceiro é opcional

Formato esperado para o arquivo:

```
NUM_ID_1 : nome do aluno_1 : email do aluno_1
NUM_ID_2 : nome do aluno_2 : email do aluno_2
...
NUM_ID_n2 : nome do aluno_n : email do aluno_n
```

ou

```
NUM_ID_1 ; nome do aluno_1 ; email do aluno_1
NUM_ID_2 ; nome do aluno_2 ; email do aluno_2
...
NUM_ID_n2 ; nome do aluno_n ; email do aluno_n
```

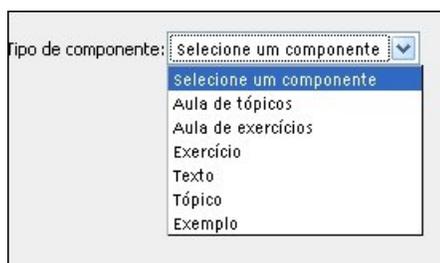
Cadastrar

Figura B.4: Interface de inserção de alunos por grupo

tipos de usuários.

B.2.2 Componentes

Os conteúdos no *SAW* são armazenados em componentes. O *SAW* possui seis tipos de componentes: texto, exercícios, exemplo, aula, tópico e curso. Para inserir componentes o professor deve acessar o *link* “**componentes**” no módulo de **inserção**. Ao entrar na interface de inserção o professor deve selecionar qual o componente que deseja inserir, figura B.5. A seguir faremos uma descrição de como inserir cada um dos componentes.



Tipo de componente: Seleccione um componente ▼

- Seleccione um componente
- Aula de tópicos
- Aula de exercícios
- Exercício
- Texto
- Tópico
- Exemplo

Figura B.5: Interface de inserção de componentes

B.2.3 Texto - Editor MaRTE

Para inserir um hipertexto no *SAW* o professor deve utilizar o editor MaRTE, figura B.6.

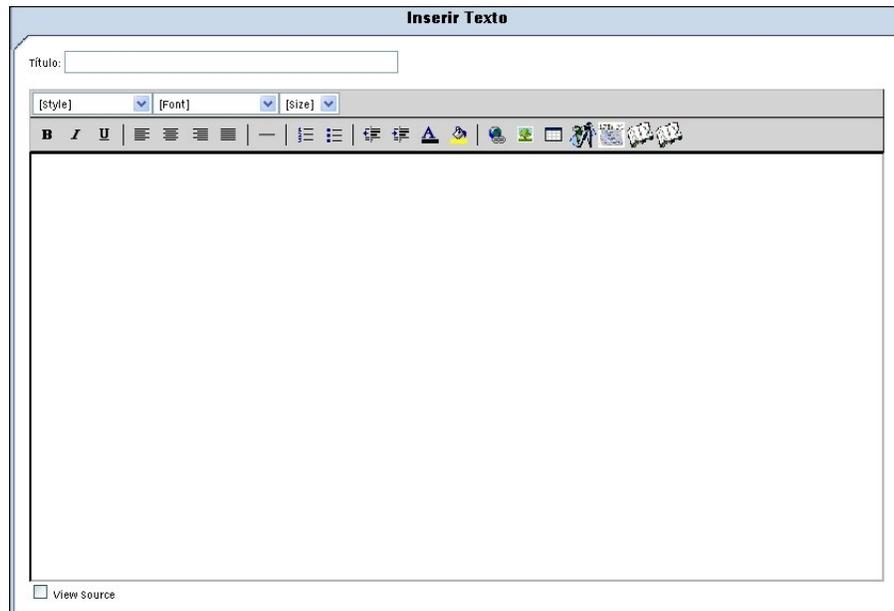


Figura B.6: Interface de inserção de textos

O editor MaRTE possui os seguintes recursos:

1. *Edição de textos*

O Marte possui recursos para alinhamento e formatação de texto. Como recurso para alinhamento o editor possui os botões: alinhamento à esquerda, centralizado, alinhamento à direita e justificado (figura B.7).



Figura B.7: Editor MaRTE, recursos de alinhamento

Como recurso de formatação o MaRTE possui os botões: negrito, itálico, sublinhado, cor de fonte, cor de fundo, tipo de letra, tamanho da letra e estilo da letra (figura B.8).

Para utilizar estes recursos o usuário deve selecionar o texto e clicar no botão com o recurso desejado ou clicar no botão e depois escrever o texto que deseja inserir o recurso.

2. *Inserir imagem*

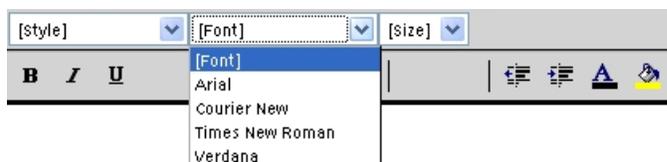


Figura B.8: Editor MaRTE, recursos de formatação

Para inserir uma imagem o usuário deve clicar no botão imagem e selecionar a imagem no computador, figura B.9. Ao clicar no botão “ok” a imagem é inserida. O usuário pode ainda dimensionar a figura para o tamanho que achar mais adequada.

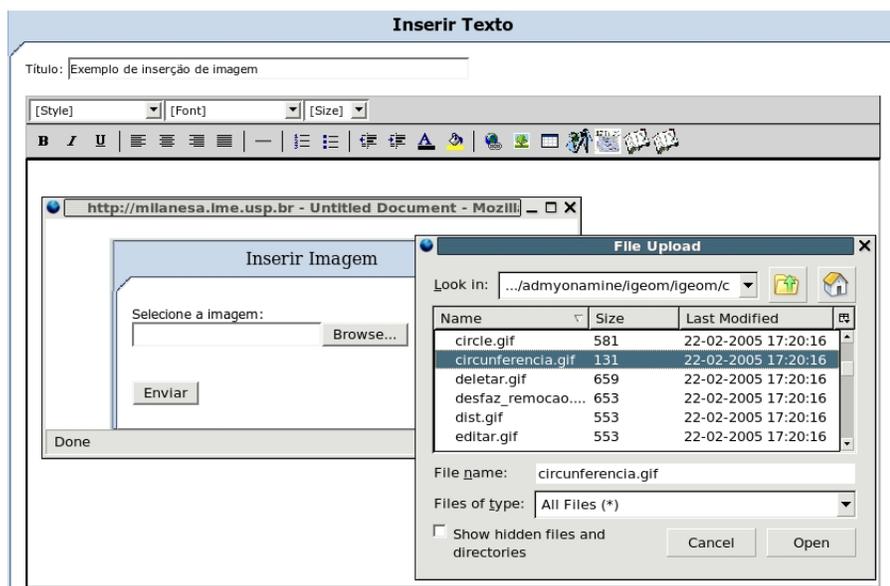


Figura B.9: Editor MaRTE, inserir imagem

3. *Inserir módulo de aprendizagem (imagem interativa)*

Para inserir uma imagem interativa no hipertexto, o usuário deve clicar no botão “inserir applet”, selecionar o *MA* a ser utilizado, selecionar o arquivo no computador e colocar as dimensões do *MA*, figura B.10. Ao clicar no botão “ok” o *MA* é inserido.

4. *Inserir bibliografia e citação*

Para inserir uma bibliografia, deve-se clicar no botão “inserir bibliografia” (figura B.11), e na próxima tela inserir:

- Nome do autor ou autores;
- Título da bibliografia;

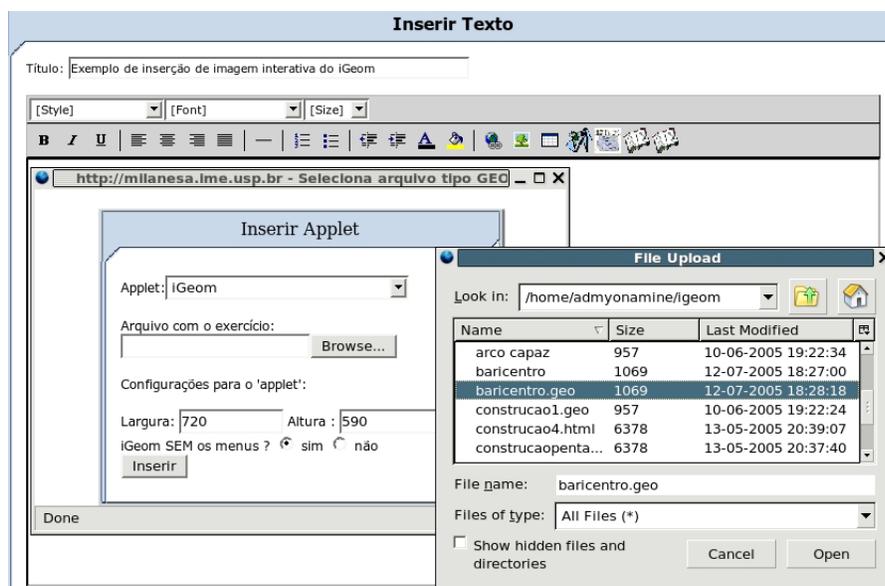


Figura B.10: Editor MaRTE, inserir imagem interativa

- Editora ou publicação;
- Ano de publicação.

Para inserir citação deve-se clicar no botão “inserir citação” e selecionar uma bibliografia, figura B.12. Se a bibliografia não estiver na lista o usuário deve primeiramente fazer a inserção da bibliografia.

5. *Inserir fórmula matemática*

Para inserir fórmula matemática no texto, o usuário deve clicar no botão “inserir fórmula” (figura B.13), e digitar a expressão matemática em formato $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.

B.2.4 Exercício

Para inserir um exercício no *SAW*, o professor deve completar o formulário de inserção, figura B.14, onde:

- Título: Inserir um título para o exercício.
- Enunciado: Digitar o enunciado do exercício. O texto do enunciado deve claro e não deve permitir ambiguidade de resposta.
- Objeto de saída: Inserir qual(is) são o(s) objeto(s) que o aluno deve encontrar como resposta do exercício.

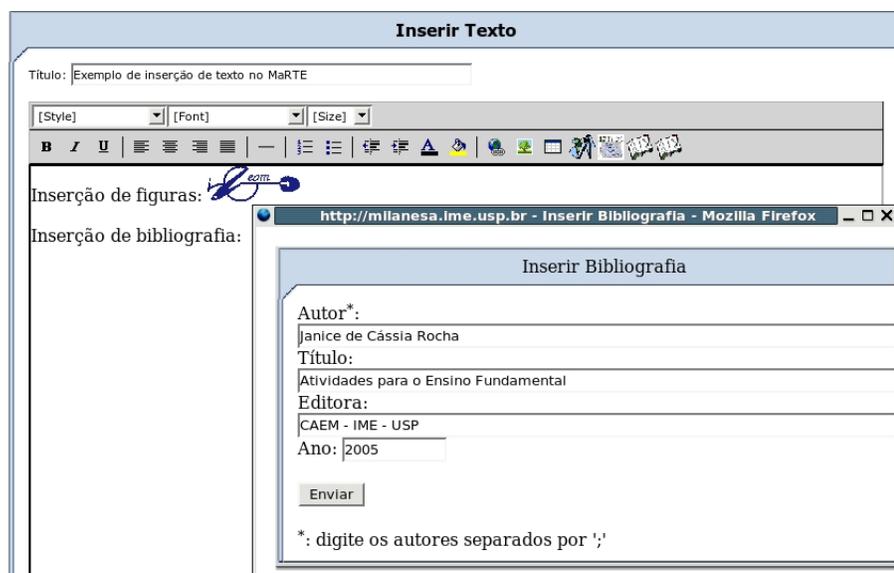


Figura B.11: Editor MaRTE, inserir bibliografia

- Applet: Selecionar qual o *MA* que será utilizado para inserção do exercício.
- Mostrar resposta para o usuário: Se este item estiver selecionado, e o *MA* tiver o recurso de avaliação automática, o aluno saberá se acertou ou errou o exercício assim que enviar o mesmo para correção.
- Comentário: Deve-se inserir um comentário sobre o exercício, somente outros professores cadastrados no *SAW* terão acesso ao comentário do exercício. Este item é essencial para que haja o reuso de exercícios.
- Ativa: Se este item estiver selecionado o exercício estará disponível para o professor utilizar em seu curso.

A inserção do exercício utilizando o *MA*, pode ser feita de duas maneiras, enviando o arquivo com o exercício ou fazendo o exercício na hora. Abaixo apresentamos um exemplo de como inserir exercícios utilizando o *MA iGeom*:

- **Arquivo com o exercício:** O professor pode fazer o exercício e o gabarito em seu computador local utilizando a versão aplicativa do *iGeom*. Depois basta selecionar o caminho que o exercício foi salvo e inserir o arquivo com o exercício, figura B.15
- **Fazer o exercício online:** O professor pode fazer a construção do exercício de maneira *online*, utilizando a versão *applet* do *iGeom*. Ao clicar no botão *online*, aparecerá a

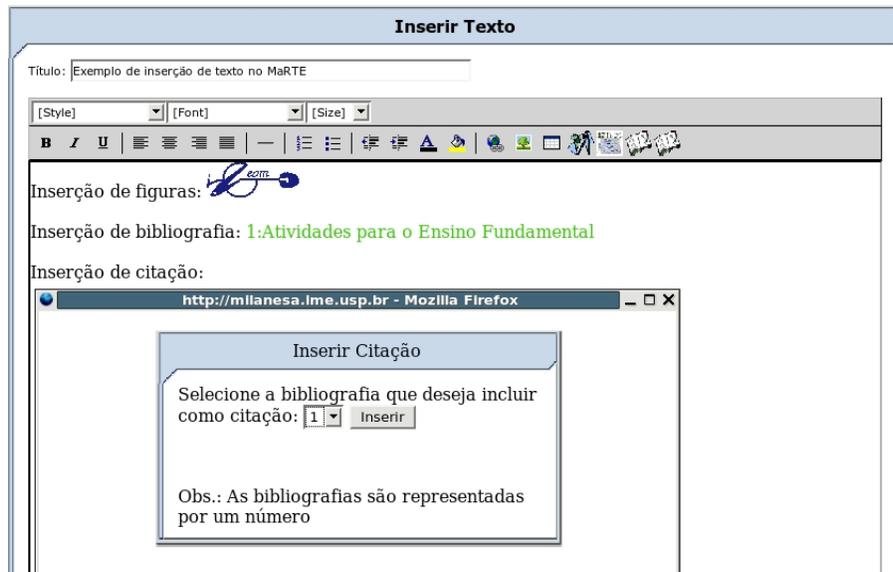


Figura B.12: Editor MaRTE, inserir citação

interface do programa *iGeom*, o professor deve fazer a construção do exercício, figura B.16, e construir o gabarito.

Para elaborar o gabarito deve-se clicar no botão “Criar exercício”, selecionar os objetos de entrada, selecionar os objetos para resposta do exercício, selecionar os botões do *iGeom* que devem aparecer para o aluno e clicar no botão “Ok”, figura B.17.

B.2.5 Exemplo

Para inserir um exemplo, usando uma construção interativa, o professor deve seguir o mesmo procedimento de inserção de exercício. Porém, para inserção de exemplo não é necessário elaborar o gabarito de exercício. A interface de inserção de exemplo é mostrada na figura B.18.

B.2.6 Tópico

Uma aula de tópico é composta pelos componentes exercícios, exemplos ou textos. Para inserir um tópico no *SAW*, o professor deve completar o formulário de inserção, figura B.19, onde:

- Título: Inserir um título para o tópico.
- Componentes que farão parte do tópico: Selecionar os componentes que deseja inserir no tópico e clicar no botão “>>”.

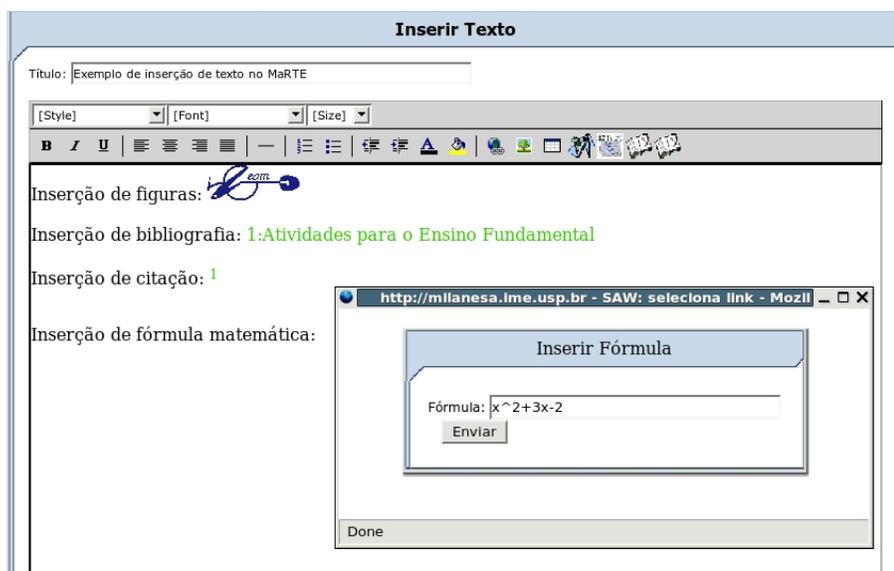


Figura B.13: Editor MaRTE, inserir expressão matemática

- Inserir pré-requisito: Com os componentes selecionados é possível inserir pré-requisitos entre os mesmos, figura B.20. Por exemplo, se o professor selecionou três exercícios A, B e C, mas deseja que o aluno resolva o exercício B antes de resolver o exercício A, deve-se colocar o exercícios B como pré-requisito do exercício A.
- Número mínimo de componentes para ir para outro tópico: O professor deve digitar o número de componentes que o usuário deve completar para que o sistema entenda que o usuário adquiriu o conhecimento necessário e possa liberá-lo para outra aula ou tópico.
- Comentário: Inserir um comentário sobre o tópico, somente outros professores cadastrados no *SAW* terão acesso ao comentário do tópico. Este item é essencial para que haja o reuso de componentes.
- Ativa: Se este item estiver selecionado o tópico estará disponível para o professor utilizar em seu curso.

B.2.7 Aula

No *SAW* há dois tipos de aula:

- **Aula de exercícios:** composta pelos componentes exercícios, exemplos ou textos.

The image shows a web form titled "Inserir Exercício". The form contains the following elements from top to bottom: a "Título:" text input field; an "Enunciado:" text area; an "Objetos de saída:" text input field; an "Applet:" dropdown menu with the text "por favor, selecione um applet" and a blue selection bar; a "Mostra" checkbox which is checked; an "Arquivo com o exercício:" text input field followed by an "Arquivo..." button; an "Fazer o exercício on line:" checkbox which is checked; a "Comentário:" text area; an "Ativa:" checkbox which is checked; and finally a "cadastrar" button at the bottom left.

Figura B.14: Interface de inserção de exercícios

- **Aula:** composta por componentes tópicos.

Para inserir uma aula no *SAW*, o professor deve completar o formulário de inserção, figura B.21, onde:

- **Título:** Inserir um título para a aula.
- **Componentes que farão parte da aula:** Selecionar os componentes que deseja inserir na aula e clicar no botão “>>”.
- **Inserir pré-requisito:** Com os componentes selecionados é possível inserir pré-requisitos entre os mesmos.
- **Número mínimo de componentes para ir para outra aula:** O professor deve digitar o número de componentes que o usuário deve completar para que o sistema considere que o usuário adquiriu o conhecimento necessário e possa liberá-lo para outra aula.
- **Comentário:** Deve inserir um comentário sobre a aula, somente outros professores cadastrados no *SAW* terão acesso ao comentário da aula. Este item é essencial para que haja o reuso de componentes.

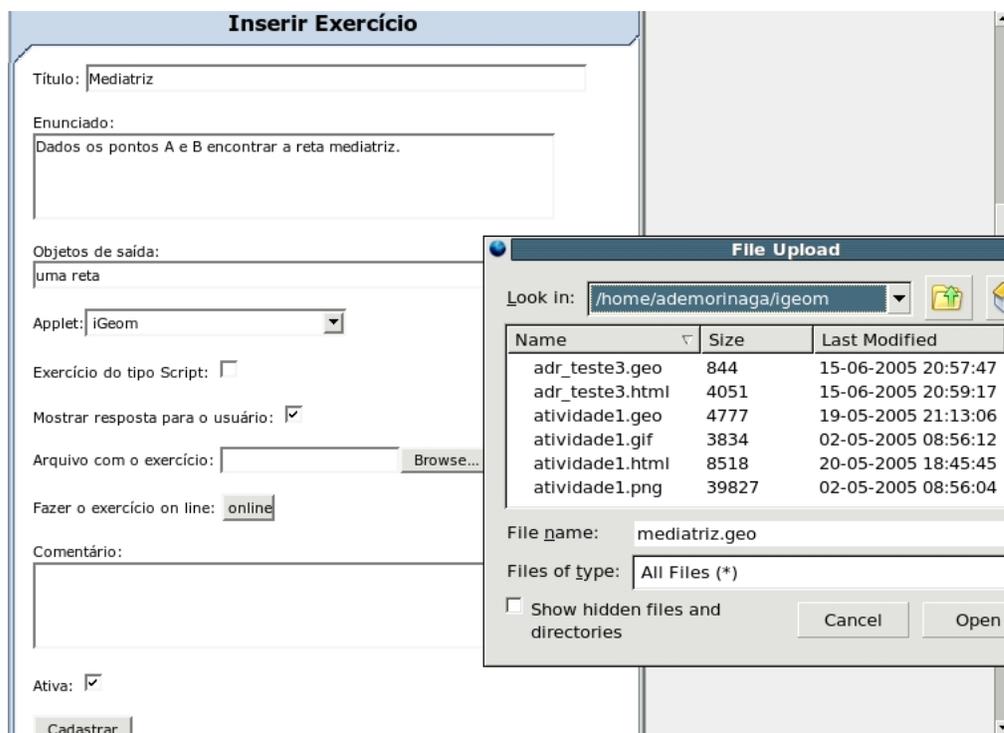


Figura B.15: Inserção de exercício por arquivo

- **Ativa:** Se este item estiver selecionado a aula estará disponível para o professor utilizar em seu curso.

B.2.8 Nota

No *SAW* é possível inserir 5 tipos de notas:

- **Nota de aula:** Para cada aula o professor pode solicitar uma nota. O *SAW* calculará as notas que os alunos adquiriram em cada exercício (se o aluno acertou é nota 10 e se o aluno errou é nota 0) e calculará a média aritmética. A figura B.22 mostra a interface para a inserção de nota por aula.
- **Nota de dicionário:** O professor pode atribuir uma nota para cada vocábulo criado pelos alunos em cada *dicionário webmídia*.
- **Nota de trabalho:** O professor pode atribuir uma nota para cada trabalho enviado pelo aluno através da funcionalidade *trabalhos*.
- **Nota externa:** O professor tem a opção de inserir no *SAW* notas referentes a atividades

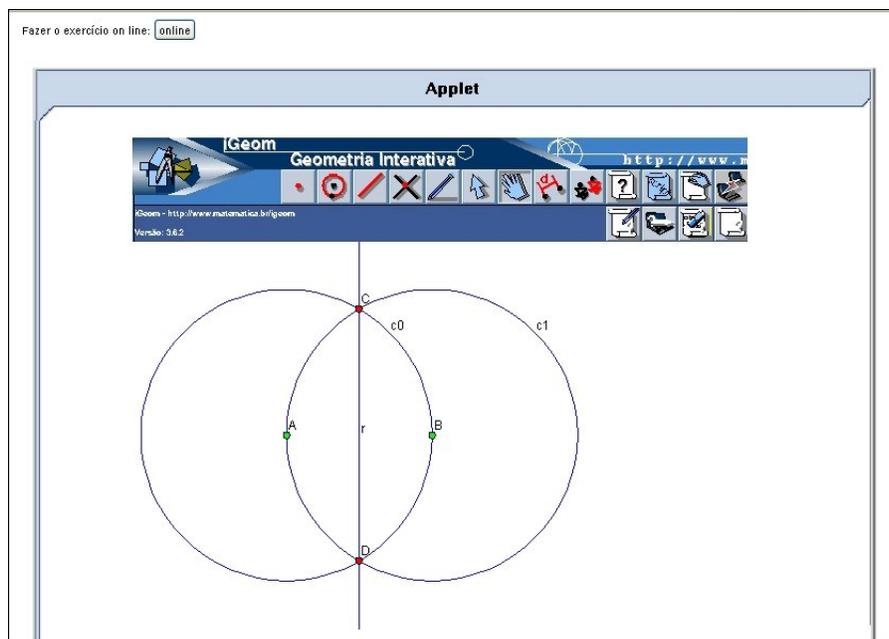


Figura B.16: Inserção de exercício online

que tenha sido realizada fora do *SAW*.

- **Média Final:** O professor pode gerar uma média final, escolhendo quais das notas criadas anteriormente deseja inserir e atribuindo peso para cada uma.

Em todos os casos, cada aluno só tem acesso a sua nota.

B.2.9 Arquivo

Esta é uma área para compartilhamento de arquivos entre a turma. Para inserir algum arquivo basta clicar no botão “Procurar”, selecionar o arquivo que deseja inserir e colocar uma descrição sobre o arquivo. A figura B.23 mostra a interface de inserção de arquivo.

B.2.10 Dicionário Webmídia

Este recurso permite o desenvolvimento de conteúdo webmídia a partir da criação de hipertextos, onde cada hipertexto é considerado um *vocábulo*. Para utilizar este recurso o professor deve criar um dicionário, através do *link* “Dicionário” no módulo inserir, figura B.24. Cada texto produzido no dicionário será um vocábulo. Cada vocábulo é de responsabilidade de seu autor (um aluno ou professor), mas todos os demais participantes no desenvolvimento do conteúdo (demais alunos

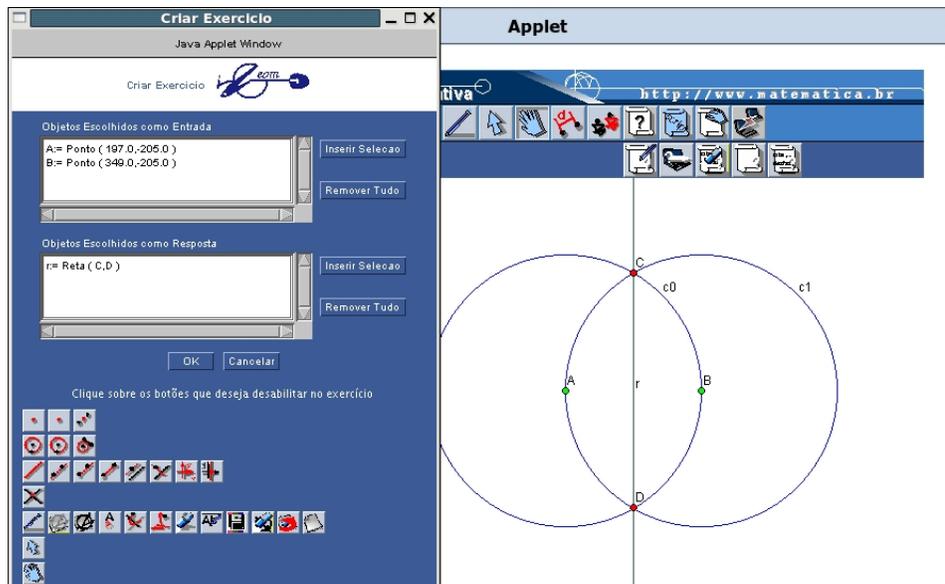


Figura B.17: Construção do gabarito do exercício

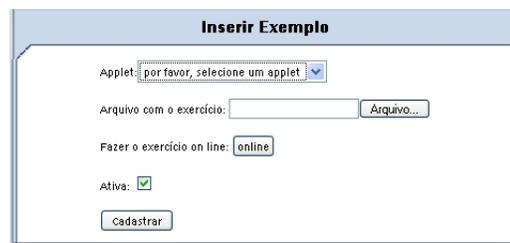


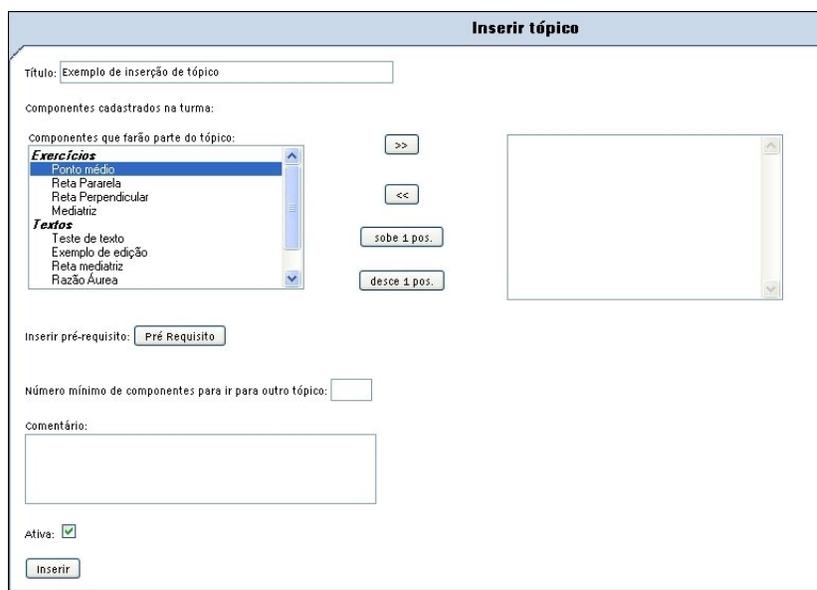
Figura B.18: Interface de inserção de exemplo.

e professor) podem contribuir para o aperfeiçoamento de um vocábulo através da postagem de comentários.

B.2.11 Painel de Avisos e Recados

Esta funcionalidade fornece a possibilidade do professor inserir avisos ou comentários para um determinado aluno ou para todos os alunos em um turma cadastrada no *SAW*. No caso de um professor enviar um recado para um determinado aluno, somente o aluno terá acesso ao recado. O usuário do tipo aluno tem permissão de enviar ou responder mensagens individuais para o professor ou para outro aluno, cadastrado na mesma turma.

A edição de mensagens é feita pelo editor de hipertexto MaRTE. O professor seleciona o(s) usuário(s) que deseja enviar algum recado, insere um título e escreve a mensagem, figura B.25. As mensagens são disponibilizadas através de *links*. Assim que o usuário entrar no *SAW*, o sistema



A interface de inserção de tópico, intitulada "Inserir tópico", apresenta os seguintes elementos:

- Um campo de texto para o "Título" com o valor "Exemplo de inserção de tópico".
- Uma seção "Componentes cadastrados na turma:" que contém uma lista de componentes dividida em "Exercícios" (Ponto médio, Retas Paralelas, Retas Perpendiculares, Mediatriz) e "Textos" (Teste de texto, Exemplo de edição, Retas mediatriz, Razão Áurea).
- Botões de navegação: ">>" e "<<" para mover componentes entre as listas, e "sobe 1 pos." e "desce 1 pos." para mover itens dentro de uma lista.
- Um campo para "Inserir pré-requisito:" com o valor "Pré Requisito".
- Um campo para "Número mínimo de componentes para ir para outro tópico:".
- Um campo de texto para "Comentário:".
- Um campo "Ativa:" com uma caixa de seleção marcada.
- Um botão "Inserir" no canto inferior esquerdo.

Figura B.19: Interface de inserção de tópico

verificará se há avisos/recados não lidos e em caso positivo o *link* "Recados" aparecerá em azul no menu. A figura B.26 mostra um exemplo de usuário que recebeu e enviou vários recados.

B.2.12 Trabalhos

Esta funcionalidade permite que o professor proponha trabalhos para serem feitos fora do sistema. Esta funcionalidade possui uma área onde o professor pode digitar o enunciado do trabalho (usando o editor MaRTE) e inserir uma data limite para a entrega. O aluno faz o trabalho fora do *SAW* e pode encaminhar um arquivo com seu trabalho (*upload*) até a data limite. O professor tem ainda a possibilidade de corrigir os trabalhos e inserir uma nota no sistema. A figura B.27 mostra a interface de inserção de trabalho.

B.2.13 Menu do aluno

O *SAW* permite que o professor escolha quais as funcionalidades deseja liberar para os alunos utilizar durante o curso. O professor deve clicar no *link* "Menu do aluno" no módulo inserir e selecionar as atividades de deseja liberar, figura B.28. A qualquer momento o professor pode fazer a alteração das funcionalidades para um determinada turma cadastrada no sistema.

Inserir tópico

Título:

Componentes cadastrados na turma:

Componentes que farão parte do tópico:

Exercícios

- Ponto médio
- Reta Paralela
- Reta Perpendicular
- Mediatriz

Textos

- Exemplo de hipertexto utilizando o editor MaRi

Exemplos

>>

<<

sobe 1 pos.

desce 1 pos.

Teste de texto

Exemplo de edição

Reta mediatriz

Razão Áurea

Componente	Teste de texto	Exemplo de edição	Reta mediatriz	Razão Áurea
Teste de texto	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Exemplo de edição	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Reta mediatriz	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Razão Áurea	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Número mínimo de componentes para ir para outro tópico:

Comentário:
Comentários sobre o tópico para que outro professor possa utilizar este componente

Ativa:

Figura B.20: Interface de inserção de tópico com pré-requisitos

B.2.14 Liberar componentes

Cada componente criado por um professor passa por um período de teste*. Este período serve para o professor verificar se o componente criado não possui erros e fazer atualizações. No período de teste somente o professor que criou o componente poderá utilizá-lo. Após este período aconselhasse que o professor libere o componente para que possa ser utilizado por outros professores cadastrados no *SAW*.

Para liberar um componente o professor deve clicar no *link* “Liberar Componente” e selecionar os componentes. Os componentes liberados não poderão ser atualizados ou excluídos.

*O período de teste é determinado pelo professor

Inserir aula de exercícios

Título:

Componente aula de exercícios cadastrados na turma:

Componentes que farão parte da aula:

Exercícios

- Reta Paralela
- Reta Perpendicular
- Mediatriz

Textos

- Reta mediatriz
- Razão Áurea
- Exemplo de hipertexto utilizando o editor MaR

Exemplos

>>

<<

sobe 1 pos.

desce 1 pos.

Teste de texto

Exemplo de edição

Ponto médio

Componente	Teste de texto	Exemplo de edição	Ponto médio
Teste de texto	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Exemplo de edição	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ponto médio	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Número mínimo de componentes para ir para outra aula:

Comentário:

Ativa:

Figura B.21: Interface de inserção de aula

Notas por aula

A nota só aparecerá para o aluno após inseri-la em **Menu | Relatório de notas** e incluí-la no curso em **Menu ! Média Final**

Título

Peso da nota:

Libera esta nota:

Selecione as aulas que farão parte da nota

1. Introdução ao iGeom
2. Aula sobre ângulo, mediatriz, bissetriz,...
3. Sobre distâncias mínimas
4. Aula sobre distâncias mínimas e pontos notáveis
5. Aula sobre pontos notáveis num triângulo
6. Construção básicas: circ. tangentes, segm. com mes
7. Trigonometria
8. Trigonometria
9. Multiplicação geométrica
10. Função 1/x
11. Reta 2x e raiz quadrada de a*b
12. Parábola, elipse e hipérbole
13. Curva de Nicomedes
14. Razão especial (sqrt(5)/2)
15. Polígonos regulares 1
- 16-17: Avaliação OL1_42-1
- 16-17: Avaliação OL1_48-1
- 16-17: Avaliação OL1_42-2
- 16-17: Avaliação OL1_48-2
18. Locus "especial" (qual é?)
19. Exercício tipo caixa preta 1

>>

<<

sobe 1 pos.

desce 1 pos.

Figura B.22: Interface de inserção de nota por aula

Área para compartilhamento de arquivos entre a turma
pode-se também manter arquivos que os demais alunos não vejam (basta não liberá-lo)

Nome	Remove/edita	Autor	Tamanho	Última alteração
Pasta pessoal na turma: área para arquivos na turma Turma 42				
Arquivo:	<input type="text"/>	<input type="button" value="Procurar..."/>		
Descrição do arquivo:	<input type="text"/>			
<small>(ao passar o mouse sobre o nome do arquivo inserido aparecerá esta descrição)</small>				
Libera este item para a turma toda? <input type="checkbox"/> (clique para resposta afirmativa)				
<input type="button" value="Enviar"/>				

Figura B.23: Interface de inserção de arquivos

Cadastro de Dicionário

Título do dicionário:

Descrição/Objetivo::

Ativa

Figura B.24: Interface de inserção de dicionário webmídia

Destinatário ..

Emilio Hirokazu Taniguchi
Érika Andersen
Everson de Jesus Lapola Gomes Santos
Fabio Pires Benatti
Fernando Gomes Garcia

Envia para a turma toda?

seu texto

Editor hipertexto SAW

[style] [font] [size]

View source

Figura B.25: Exemplo da funcionalidade Painel de Avisos e Recados

Recados	
Legenda:	convenções de cores para recados
não lidos:	recados ainda não lidos pelo usuário (destinatário)
lidos:	recados já lidos pelo usuário (destinatário)
Termo:	recados do professor/monitor para o turma toda
Leônidas de Oliveira Brandão	Alunos das listas 4 e 5: exercícios
Leônidas de Oliveira Brandão	Dica sobre navegadores (use o Firefox)
Leônidas de Oliveira Brandão	Texto sobre programação: introdução
Leônidas de Oliveira Brandão	Nota final no SAW e JúpiterWeb
Enviados para:	recados ENVIADOS pelo usuário (Leônidas de Oliveira Brandão)
Louise Franco de Oliveira	Sobre "Avaliação"
Pedro Leonardo Silva de Almeida	Re: Deixei uma ajuda ai...
usuario 1	mais um teste, com fórmula
usuario 1	mais um teste, com fórmula
André de Camargo Fernandes	Sobre: Por favor,avise os professores
Anelise Eichinger	EP2
Thays Tagliatieri de Graziá	Re: Notas das Provas
Abigail Rodrigues Ozzoto	Re: Você NÃO está matriculado(a) em MAC110 segundo o JúpiterWeb
Bruno César Spósito	Re: Você NÃO está matriculado(a) em MAC110 segundo o JúpiterWeb
Bruno César Spósito	Re: Você NÃO está matriculado(a) em MAC110 segundo o Júpiter
Abigail Rodrigues Ozzoto	Re: Você NÃO está matriculado(a) em MAC110 segundo o Júpiter
Recebidos de:	recados específicos RECEBIDOS pelo usuário (Leônidas de Oliveira Brandão)
usuario 1	Apenas um teste...
Louise Franco de Oliveira	Avaliação
Pedro Leonardo Silva de Almeida	Deixei uma ajuda ai...
Louise Franco de Oliveira	Avaliação

Figura B.26: Exemplo da funcionalidade Painel de Avisos e Recados

Usuário Leônidas de Oliveira Brandão 1 13-11-2008

Inserir "arquivo/trabalho a serem enviados"

Nome:

Data/hora limite para entrega:
 Data de entrega (dd/mm/aaaa) Hora de entrega (hh:mm)

Número de integrantes:

Extensão do arquivo: (deixe em branco se quiser aceitar arquivos com qualquer extensão)

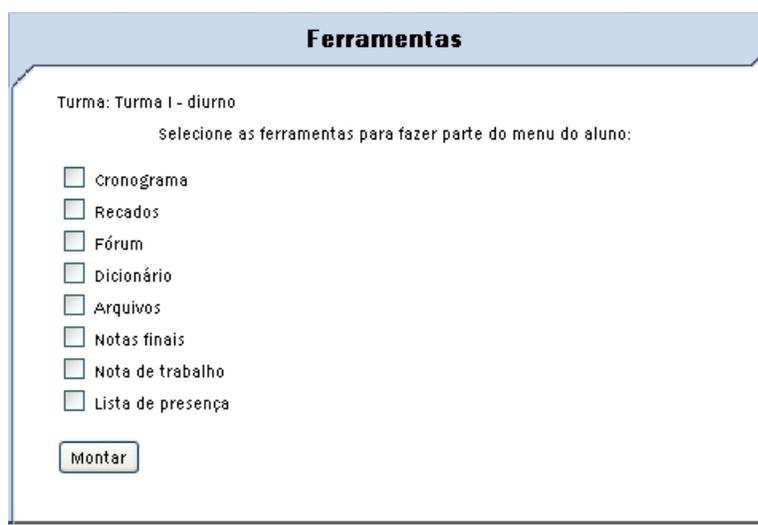
Peso:

[style] [font] [size]

View Source

Ativa:

Figura B.27: Exemplo da funcionalidade Painel de Avisos e Recados



Ferramentas

Turma: Turma I - diurno

Selecione as ferramentas para fazer parte do menu do aluno:

- Cronograma
- Recados
- Fórum
- Dicionário
- Arquivos
- Notas finais
- Nota de trabalho
- Lista de presença

Figura B.28: Interface de liberação de funcionalidades para o aluno

Apêndice C

Tabelas do Banco de Dados

Tabela C.1: Estrutura da Tabela saw_admin_admin

Campo	Tipo	Nulo	Padrão
id	bigint(20)	Não	0
nome	varchar(100)	Não	
tipo	int(11)	Não	0
id_cadastro	bigint(20)	Sim	NULL
data_cadastro	datetime	Sim	NULL

Tabela C.2: Estrutura da Tabela saw_admin_applet

Campo	Tipo	Nulo	Padrão
<i>id</i>	bigint(20)	Não	
nome	varchar(20)	Não	
extensao	varchar(30)	Não	
descricao	text	Não	
arquivo_jar	varchar(20)	Não	
largura	int(11)	Não	0
altura	int(11)	Não	0
arquivo_class	varchar(30)	Não	
id_cadastro	bigint(20)	Não	0
data_cadastro	datetime	Sim	NULL
ativa	int(11)	Sim	NULL

Tabela C.3: Estrutura da Tabela saw_admin_curso

Campo	Tipo	Nulo	Padrão
<i>id</i>	bigint(20)	Não	
nome	varchar(255)	Não	

Tabela C.3: Estrutura da Tabela saw_admin_curso (continued)

Campo	Tipo	Nulo	Padrão
ativa	int(1)	Não	0
logo	varchar(20)	Sim	NULL
comentario	longtext	Sim	NULL
informacao	longtext	Sim	NULL
bemvindo	text	Sim	NULL
id_forum	bigint(20)	Sim	NULL
id_cadastro	bigint(20)	Não	0
data_cadastro	datetime	Não	0000-00-00 00:00:00

Tabela C.4: Estrutura da Tabela saw_admin_saw

Campo	Tipo	Nulo	Padrão
informacao	text	Sim	NULL
bemvindo	text	Sim	NULL

Tabela C.5: Estrutura da Tabela saw_curso_aluno_aula

Campo	Tipo	Nulo	Padrão
id_aluno	bigint(10)	Não	0
id_aula	bigint(10)	Não	0
ativa	int(1)	Sim	NULL
completo	int(1)	Sim	NULL
id_turma	int(10)	Não	0
liberado	tinyint(4)	Sim	NULL
n_tentativas	int(11)	Sim	NULL
nota	float	Sim	NULL

Tabela C.6: Estrutura da Tabela saw_curso_aluno_curso

Campo	Tipo	Nulo	Padrão
id_aluno	bigint(10)	Não	0
id_curso	bigint(10)	Não	0
ativa	int(1)	Sim	NULL
n_tentativas	bigint(10)	Sim	NULL

Tabela C.7: Estrutura da Tabela saw_curso_aluno_exercicio

Campo	Tipo	Nulo	Padrão
id_aluno	bigint(10)	Não	0
id_exercicio	bigint(10)	Não	0
id_turma	bigint(20)	Não	0
ativa	int(1)	Sim	NULL
completo	int(1)	Sim	NULL
valor	int(1)	Sim	NULL
arquivo	longtext	Sim	NULL
resposta	longtext	Sim	NULL
mostra	int(1)	Sim	NULL
n_tentativas	int(11)	Sim	NULL
acao	longtext	Sim	NULL
data	datetime	Sim	NULL
ip	varchar(15)	Sim	NULL
avaliado	varchar(10)	Sim	NULL

Tabela C.8: Estrutura da Tabela saw_curso_aluno_nota

Campo	Tipo	Nulo	Padrão
id_aluno	bigint(20)	Não	0
id_nota	bigint(20)	Não	0
nota	float	Sim	NULL

Tabela C.9: Estrutura da Tabela saw_curso_aluno-presenca

Campo	Tipo	Nulo	Padrão
data_dia	date	Sim	NULL
id_aluno	int(11)	Sim	NULL
id_curso	int(11)	Sim	NULL
presente	tinyint(4)	Sim	NULL
data_hora	time	Sim	NULL

Tabela C.10: Estrutura da Tabela saw_curso_aluno-presenca_prof

Campo	Tipo	Nulo	Padrão
id_aluno	bigint(20)	Sim	NULL
id_curso	int(11)	Não	0
id_data	bigint(11)	Sim	NULL
presente	int(11)	Sim	NULL

Tabela C.11: Estrutura da Tabela saw_curso_aluno_topico

Campo	Tipo	Nulo	Padrão
id_aluno	bigint(10)	Não	0
id_topico	bigint(10)	Não	0
ativa	int(1)	Sim	NULL
completo	int(1)	Sim	NULL
liberado	tinyint(4)	Sim	NULL
n_tentativas	int(11)	Sim	NULL

Tabela C.12: Estrutura da Tabela saw_curso_aluno_trabalho

Campo	Tipo	Nulo	Padrão
id_aluno	int(11)	Não	0
id_trabalho	int(11)	Não	0
turma	int(11)	Sim	NULL
data_entrega	datetime	Sim	NULL
nome_arquivo	text	Sim	NULL
nome_no_diretorio	text	Sim	NULL
tamanho	bigint(20)	Sim	NULL
nota	float(10,2)	Sim	NULL
comentario	longtext	Sim	NULL
id_corrighu	int(11)	Sim	NULL

Tabela C.13: Estrutura da Tabela saw_curso_arquivo

Campo	Tipo	Nulo	Padrão
<i>id</i>	bigint(20)	Não	
nome	varchar(100)	Não	
descricao	varchar(100)	Sim	NULL
tamanho	bigint(20)	Sim	NULL
download	bigint(20)	Sim	NULL

Tabela C.14: Estrutura da Tabela saw_curso_aula_turma

Campo	Tipo	Nulo	Padrão
id_aula	int(11)	Não	0
id_turma	int(11)	Não	0
ativa	tinyint(4)	Sim	NULL
inicio	datetime	Sim	NULL
termino	datetime	Sim	NULL
online	tinyint(4)	Sim	NULL
pre_requisito	varchar(20)	Sim	NULL

Tabela C.14: Estrutura da Tabela saw_curso_aula_turma
(continued)

Campo	Tipo	Nulo	Padrão
teste	int(11)	Sim	NULL
ip	varchar(30)	Sim	NULL

Tabela C.15: Estrutura da Tabela saw_curso_aula_turma_aluno

Campo	Tipo	Nulo	Padrão
id_aluno	bigint(20)	Não	0
id_aula	bigint(20)	Não	0
id_turma	bigint(20)	Não	0
ativa	tinyint(4)	Sim	NULL
inicio	datetime	Sim	NULL
termino	datetime	Sim	NULL
teste	int(11)	Sim	NULL
ip	varchar(30)	Sim	NULL
online	int(11)	Sim	NULL

Tabela C.16: Estrutura da Tabela saw_curso_data_aula

Campo	Tipo	Nulo	Padrão
<i>id_data</i>	bigint(20)	Não	
data	date	Sim	NULL
ativa	tinyint(4)	Sim	NULL
id_curso	int(11)	Sim	NULL

Tabela C.17: Estrutura da Tabela saw_curso_data_prof

Campo	Tipo	Nulo	Padrão
<i>id_data</i>	bigint(20)	Não	
data	datetime	Sim	NULL
id_curso	bigint(20)	Sim	NULL

Tabela C.18: Estrutura da Tabela saw_curso_dic_comentario

Campo	Tipo	Nulo	Padrão
<i>id_comentario</i>	bigint(20)	Não	0
id_vocabulo	bigint(20)	Sim	NULL
comentario	text	Sim	NULL

Tabela C.18: Estrutura da Tabela saw_curso_dic_comentario
(continued)

Campo	Tipo	Nulo	Padrão
id_cadastro	bigint(20)	Sim	NULL
data_cadastro	datetime	Sim	NULL
id_turma	bigint(20)	Sim	NULL

Tabela C.19: Estrutura da Tabela saw_curso_dic_dicionario

Campo	Tipo	Nulo	Padrão
<i>id</i>	bigint(20)	Não	
nome	varchar(100)	Não	
objetivo	text	Sim	NULL
id_curso	bigint(20)	Não	0
id_turma	varchar(50)	Não	
id_cadastro	int(11)	Não	0
data_cadastro	text	Não	
ativa	int(11)	Sim	0

Tabela C.20: Estrutura da Tabela saw_curso_dic_publicacao

Campo	Tipo	Nulo	Padrão
<i>id</i>	bigint(20)	Não	
nome	varchar(100)	Sim	NULL
publicacao	int(11)	Sim	NULL
id_tipo	bigint(20)	Sim	NULL
turma_id	bigint(20)	Sim	NULL
vetor_publicacao	varchar(100)	Sim	NULL
id_cadastro	bigint(20)	Sim	NULL
data_cadastro	datetime	Sim	NULL
id_curso	bigint(20)	Sim	NULL

Tabela C.21: Estrutura da Tabela saw_curso_dic_topico

Campo	Tipo	Nulo	Padrão
<i>id</i>	bigint(20)	Não	
nome	varchar(100)	Sim	NULL
id_pai	bigint(20)	Sim	NULL

Tabela C.22: Estrutura da Tabela saw_curso_dic_vocabulo

Campo	Tipo	Nulo	Padrão
<i>id_vocabulo</i>	bigint(20)	Não	0
id_autor	bigint(20)	Não	0
data_cadastro	datetime	Sim	NULL
id_dicionario	bigint(20)	Não	0
id_turma	bigint(20)	Não	0
ativa	int(11)	Sim	NULL
nota	float	Sim	NULL
comentario	text	Sim	NULL
id_corrighu	bigint(20)	Sim	NULL
id_topico	varchar(100)	Sim	NULL
data_cadastro2	datetime	Sim	NULL

Tabela C.23: Estrutura da Tabela saw_curso_menu_aluno

Campo	Tipo	Nulo	Padrão
<i>id</i>	bigint(20)	Não	0
nome	varchar(250)	Não	
descricao	text	Sim	NULL
ativa	int(11)	Sim	NULL
endereço	varchar(100)	Sim	NULL
id_turma	bigint(20)	Não	0

Tabela C.24: Estrutura da Tabela saw_curso_notas

Campo	Tipo	Nulo	Padrão
<i>id</i>	bigint(20)	Não	
ord	int(11)	Não	0
peso	double(16,4)	Sim	NULL
nome	text	Sim	NULL
id_aulas	varchar(100)	Sim	NULL
id_curso	bigint(20)	Sim	NULL
tipo	int(11)	Sim	NULL

Tabela C.25: Estrutura da Tabela saw_curso_notas_curso

Campo	Tipo	Nulo	Padrão
<i>id</i>	int(11)	Não	
id_curso	bigint(20)	Sim	NULL
id_turma	int(11)	Sim	NULL
ano	varchar(4)	Sim	NULL

Tabela C.25: Estrutura da Tabela saw_curso_notas_curso
(continued)

Campo	Tipo	Nulo	Padrão
id_notas	varchar(255)	Sim	NULL
data	date	Sim	NULL
usuario	bigint(20)	Sim	NULL

Tabela C.26: Estrutura da Tabela saw_curso_recado_curso

Campo	Tipo	Nulo	Padrão
<i>id</i>	bigint(20)	Não	
nome	varchar(255)	Sim	NULL
id_textos	varchar(255)	Sim	NULL
id_cadastro	bigint(20)	Sim	NULL
data_cadastro	date	Sim	NULL
alteracao	date	Sim	NULL

Tabela C.27: Estrutura da Tabela saw_curso_recado_turma

Campo	Tipo	Nulo	Padrão
id_recado	bigint(20)	Não	0
ano	varchar(4)	Sim	NULL
id_turma	int(11)	Não	0
ativa	int(1)	Sim	NULL
id_cadastro	bigint(20)	Sim	NULL
alteracao	date	Sim	NULL

Tabela C.28: Estrutura da Tabela saw_curso_recado_usuario

Campo	Tipo	Nulo	Padrão
id_recado	bigint(20)	Não	0
de	bigint(20)	Não	0
para	bigint(20)	Não	0
id_turma	bigint(20)	Não	0
id_texto	bigint(20)	Não	0
ativa	char(1)	Não	
visto	varchar(16)	Não	0
edicao	varchar(16)	Não	
data	varchar(16)	Não	

Tabela C.29: Estrutura da Tabela saw_curso_trabalho

Campo	Tipo	Nulo	Padrão
<i>id</i>	int(11)	Não	
extensao	tinytext	Não	
nome	longtext	Não	
descricao	mediumtext	Sim	NULL
liberado	tinyint(4)	Sim	NULL
data_limite	datetime	Sim	NULL
integrantes	tinyint(4)	Sim	NULL
peso	int(11)	Sim	NULL
id_cadastro	int(11)	Sim	NULL
data_cadastro	datetime	Sim	NULL

Tabela C.30: Estrutura da Tabela saw_curso_trabalho_turma

Campo	Tipo	Nulo	Padrão
id_turma	int(11)	Não	0
id_trabalho	int(11)	Sim	NULL
ativa	tinyint(4)	Sim	NULL

Tabela C.31: Estrutura da Tabela saw_curso_turma

Campo	Tipo	Nulo	Padrão
<i>id</i>	bigint(20)	Não	
nome	varchar(30)	Não	
ativa	tinyint(4)	Sim	NULL
id_cadastro	bigint(20)	Não	0
data_cadastro	datetime	Sim	NULL
id_responsavel	bigint(20)	Sim	NULL
informacao	text	Sim	NULL

Tabela C.32: Estrutura da Tabela saw_user_applets

Campo	Tipo	Nulo	Padrão
<i>id</i>	bigint(20)	Não	
id_texto	bigint(20)	Não	0
autor	bigint(20)	Sim	NULL
id_applet	bigint(20)	Não	0
posicao	int(11)	Não	0
arquivo	longtext	Não	
nome_arq	varchar(128)	Não	
width	int(11)	Sim	NULL

Tabela C.32: Estrutura da Tabela saw_user_applets (continued)

Campo	Tipo	Nulo	Padrão
height	int(11)	Sim	NULL
menu	varchar(10)	Sim	NULL

Tabela C.33: Estrutura da Tabela saw_user_aula

Campo	Tipo	Nulo	Padrão
<i>id</i>	bigint(20)	Não	
nome	text	Não	
ativa	int(1)	Não	0
pre_requisito	int(1)	Não	0
num_minimo	int(11)	Sim	0
id_cadastro	bigint(20)	Sim	NULL
data_cadastro	datetime	Sim	NULL
tipo	tinyint(4)	Sim	NULL
comentario	longtext	Sim	NULL
liberado	int(11)	Não	0

Tabela C.34: Estrutura da Tabela saw_user_aula_componente

Campo	Tipo	Nulo	Padrão
id_aula	bigint(10)	Não	0
id_componente	bigint(10)	Não	0
ativa	int(1)	Sim	0
tipo_componente	int(2)	Sim	NULL
pre_requisito	varchar(100)	Sim	NULL

Tabela C.35: Estrutura da Tabela saw_user_aula_pre_requisito

Campo	Tipo	Nulo	Padrão
id	bigint(10)	Não	0
id_requisito	bigint(10)	Não	0
ativa	int(1)	Não	0

Tabela C.36: Estrutura da Tabela saw_user_bibliografia

Campo	Tipo	Nulo	Padrão
<i>id</i>	bigint(20)	Não	

Tabela C.36: Estrutura da Tabela saw_user_bibliografia (continued)

Campo	Tipo	Nulo	Padrão
id_texto	bigint(20)	Não	0
id_cadastro	bigint(20)	Sim	NULL
posicao	int(11)	Não	0
titulo	longtext	Não	
autor	varchar(200)	Sim	
editora	varchar(200)	Sim	NULL
ano	int(11)	Sim	NULL

Tabela C.37: Estrutura da Tabela saw_user_exemplo

Campo	Tipo	Nulo	Padrão
<i>id</i>	bigint(20)	Não	
nome	varchar(100)	Não	
ativa	int(1)	Não	0
id_cadastro	bigint(20)	Sim	0
data_cadastro	datetime	Não	0000-00-00 00:00:00
id_applet	bigint(20)	Não	0
liberado	int(11)	Não	0

Tabela C.38: Estrutura da Tabela saw_user_exercicio

Campo	Tipo	Nulo	Padrão
<i>id</i>	bigint(20)	Não	
nome	varchar(100)	Não	
ativa	int(1)	Não	0
enunciado	text	Não	
pre_requisito	int(1)	Não	0
gabarito	longtext	Não	
comentario	longtext	Sim	NULL
feedback	tinyint(4)	Não	0
saida	longtext	Sim	NULL
script	int(11)	Não	0
id_cadastro	bigint(20)	Sim	0
data_cadastro	datetime	Não	0000-00-00 00:00:00
id_applet	bigint(20)	Não	0
liberado	int(11)	Não	0

Tabela C.39: Estrutura da Tabela saw_user_formulas

Campo	Tipo	Nulo	Padrão
<i>id</i>	bigint(20)	Não	
id_texto	bigint(20)	Não	0
autor	bigint(20)	Não	0
formula	varchar(255)	Não	
img	longblob	Sim	NULL

Tabela C.40: Estrutura da Tabela saw_user_menu_aluno

Campo	Tipo	Nulo	Padrão
<i>id</i>	bigint(20)	Não	
nome	varchar(250)	Não	
descricao	text	Sim	NULL
ativa	int(11)	Sim	NULL
endereco	varchar(100)	Sim	NULL

Tabela C.41: Estrutura da Tabela saw_user_texto

Campo	Tipo	Nulo	Padrão
<i>id</i>	bigint(20)	Não	
nome	varchar(100)	Não	
texto	text	Sim	NULL
id_cadastro	bigint(20)	Não	0
data_cadastro	datetime	Não	0000-00-00 00:00:00
liberado	int(11)	Sim	0

Tabela C.42: Estrutura da Tabela saw_user_topico

Campo	Tipo	Nulo	Padrão
<i>id</i>	bigint(20)	Não	
ativa	int(1)	Não	0
pre_requisito	int(1)	Sim	0
codigo_pre_requisito	bigint(10)	Sim	0
nome	varchar(100)	Não	
num_minimo	int(11)	Não	0
id_cadastro	bigint(20)	Sim	0
data_cadastro	datetime	Sim	0000-00-00 00:00:00
comentario	longtext	Sim	NULL
liberado	int(11)	Não	0

Tabela C.43: Estrutura da Tabela saw_user_topico_componente

Campo	Tipo	Nulo	Padrão
id_topico	bigint(10)	Não	0
id_componente	bigint(10)	Não	0
ativa	int(1)	Sim	0
tipo_componente	int(2)	Não	0
pre_requisito	varchar(100)	Sim	NULL

Tabela C.44: Estrutura da Tabela saw_user_usuario

Campo	Tipo	Nulo	Padrão
<i>id</i>	bigint(10)	Não	0
nome	varchar(100)	Não	
email	varchar(100)	Não	
senha	varchar(10)	Não	
conectado	int(1)	Não	0
ativa	int(1)	Não	0
tipo	int(11)	Não	0
id_cadastro	bigint(20)	Sim	NULL
data_cadastro	datetime	Sim	NULL

Tabela C.45: Estrutura da Tabela saw_user_usuario_criptografado

Campo	Tipo	Nulo	Padrão
id_user	bigint(10)	Não	0
cript	varchar(50)	Não	
ativa	int(1)	Sim	0
id_aula	int(10)	Sim	0
id_topico	int(10)	Sim	0
id_exercicio	int(10)	Sim	0
id_curso	int(10)	Sim	0
tempo	datetime	Não	0000-00-00 00:00:00
id_aula_exercicio	bigint(20)	Sim	NULL
id_texto	bigint(20)	Sim	NULL

Tabela C.46: Estrutura da Tabela saw_user_usuario_curso

Campo	Tipo	Nulo	Padrão
id_usuario	int(11)	Não	0
id_curso	int(11)	Não	0

Tabela C.46: Estrutura da Tabela saw_user_usuario_curso
(continued)

Campo	Tipo	Nulo	Padrão
id_turma	varchar(20)	Não	0
tipo	bigint(20)	Não	0
ativa	int(11)	Sim	0
n_tentativas	int(11)	Sim	0

Tabela C.47: Estrutura da tabela saw_phpbb_auth_access

Campo	Tipo	Nulo	Padrão
group_id	mediumint(8)	Não	0
forum_id	smallint(5)	Não	0
auth_view	tinyint(1)	Não	0
auth_read	tinyint(1)	Não	0
auth_post	tinyint(1)	Não	0
auth_reply	tinyint(1)	Não	0
auth_edit	tinyint(1)	Não	0
auth_delete	tinyint(1)	Não	0
auth_sticky	tinyint(1)	Não	0
auth_announce	tinyint(1)	Não	0
auth_vote	tinyint(1)	Não	0
auth_pollcreate	tinyint(1)	Não	0
auth_attachments	tinyint(1)	Não	0
auth_mod	tinyint(1)	Não	0

Tabela C.48: Estrutura da tabela saw_phpbb_banlist

Campo	Tipo	Nulo	Padrão
<i>ban_id</i>	mediumint(8)	Não	
ban_userid	mediumint(8)	Não	0
ban_ip	varchar(8)	Não	
ban_email	varchar(255)	Sim	NULL

Tabela C.49: Estrutura da tabela saw_phpbb_categories

Campo	Tipo	Nulo	Padrão
<i>cat_id</i>	mediumint(8)	Não	
cat_title	varchar(100)	Sim	NULL
cat_order	mediumint(8)	Não	0

Tabela C.50: Estrutura da tabela saw_phpbb_config

Campo	Tipo	Nulo	Padrão
<i>config_name</i>	varchar(255)	Não	
config_value	varchar(255)	Não	

Tabela C.51: Estrutura da tabela saw_phpbb_confirm

Campo	Tipo	Nulo	Padrão
<i>confirm_id</i>	char(32)	Não	
<i>session_id</i>	char(32)	Não	
code	char(6)	Não	

Tabela C.52: Estrutura da tabela saw_phpbb_disallow

Campo	Tipo	Nulo	Padrão
<i>disallow_id</i>	mediumint(8)	Não	
disallow_username	varchar(25)	Não	

Tabela C.53: Estrutura da tabela saw_phpbb_formula_forum

Campo	Tipo	Nulo	Padrão
<i>id</i>	bigint(20)	Não	
id_post	mediumint(9)	Não	0
posicao	int(11)	Não	0
nome_arq	tinytext	Não	
texto_form	text	Não	
id_user	bigint(20)	Não	0

Tabela C.54: Estrutura da tabela saw_phpbb_forum_prune

Campo	Tipo	Nulo	Padrão
<i>prune_id</i>	mediumint(8)	Não	
forum_id	smallint(5)	Não	0
prune_days	smallint(5)	Não	0
prune_freq	smallint(5)	Não	0

Tabela C.55: Estrutura da tabela saw_phpbb_forums

Campo	Tipo	Nulo	Padrão
<i>forum_id</i>	smallint(5)	Não	0

Tabela C.55: Estrutura da tabela saw_phpbb_forums (continued)

Campo	Tipo	Nulo	Padrão
cat_id	mediumint(8)	Não	0
forum_name	varchar(150)	Sim	NULL
forum_desc	text	Sim	NULL
forum_status	tinyint(4)	Não	0
forum_order	mediumint(8)	Não	1
forum_posts	mediumint(8)	Não	0
forum_topics	mediumint(8)	Não	0
forum_last_post_id	mediumint(8)	Não	0
prune_next	int(11)	Sim	NULL
prune_enable	tinyint(1)	Não	0
auth_view	tinyint(2)	Não	0
auth_read	tinyint(2)	Não	0
auth_post	tinyint(2)	Não	0
auth_reply	tinyint(2)	Não	0
auth_edit	tinyint(2)	Não	0
auth_delete	tinyint(2)	Não	0
auth_sticky	tinyint(2)	Não	0
auth_announce	tinyint(2)	Não	0
auth_vote	tinyint(2)	Não	0
auth_pollcreate	tinyint(2)	Não	0
auth_attachments	tinyint(2)	Não	0
id_curso	bigint(20)	Sim	NULL
id_turma	bigint(20)	Não	0

Tabela C.56: Estrutura da tabela saw_phpbb_groups

Campo	Tipo	Nulo	Padrão
<i>group_id</i>	mediumint(8)	Não	
group_type	tinyint(4)	Não	1
group_name	varchar(40)	Não	
group_description	varchar(255)	Não	
group_moderator	mediumint(8)	Não	0
group_single_user	tinyint(1)	Não	1

Tabela C.57: Estrutura da tabela saw_phpbb_posts

Campo	Tipo	Nulo	Padrão
<i>post_id</i>	mediumint(8)	Não	
topic_id	mediumint(8)	Não	0

Tabela C.57: Estrutura da tabela saw_phpbb_posts (continued)

Campo	Tipo	Nulo	Padrão
forum_id	smallint(5)	Não	0
poster_id	mediumint(8)	Não	0
post_time	int(11)	Não	0
poster_ip	varchar(8)	Não	
post_username	varchar(25)	Sim	NULL
enable_bbcode	tinyint(1)	Não	1
enable_html	tinyint(1)	Não	0
enable_smilies	tinyint(1)	Não	1
enable_sig	tinyint(1)	Não	1
post_edit_time	int(11)	Sim	NULL
post_edit_count	smallint(5)	Não	0

Tabela C.58: Estrutura da tabela saw_phpbb_posts_text

Campo	Tipo	Nulo	Padrão
<i>post_id</i>	mediumint(8)	Não	0
bbcode_uid	varchar(10)	Não	
post_subject	varchar(60)	Sim	NULL
post_text	text	Sim	NULL

Tabela C.59: Estrutura da tabela saw_phpbb_privmsgs

Campo	Tipo	Nulo	Padrão
<i>privmsgs_id</i>	mediumint(8)	Não	
privmsgs_type	tinyint(4)	Não	0
privmsgs_subject	varchar(255)	Não	0
privmsgs_from_userid	mediumint(8)	Não	0
privmsgs_to_userid	mediumint(8)	Não	0
privmsgs_date	int(11)	Não	0
privmsgs_ip	varchar(8)	Não	
privmsgs_enable_bbcode	tinyint(1)	Não	1
privmsgs_enable_html	tinyint(1)	Não	0
privmsgs_enable_smilies	tinyint(1)	Não	1
privmsgs_attach_sig	tinyint(1)	Não	1

Tabela C.60: Estrutura da tabela saw_phpbb_privmsgs_text

Campo	Tipo	Nulo	Padrão
<i>privmsgs_text_id</i>	mediumint(8)	Não	0
privmsgs_bbcode_uid	varchar(10)	Não	0
privmsgs_text	text	Sim	NULL

Tabela C.61: Estrutura da tabela saw_phpbb_ranks

Campo	Tipo	Nulo	Padrão
<i>rank_id</i>	smallint(5)	Não	
rank_title	varchar(50)	Não	
rank_min	mediumint(8)	Não	0
rank_special	tinyint(1)	Sim	0
rank_image	varchar(255)	Sim	NULL

Tabela C.62: Estrutura da tabela saw_phpbb_search_results

Campo	Tipo	Nulo	Padrão
<i>search_id</i>	int(11)	Não	0
session_id	varchar(32)	Não	
search_array	text	Não	

Tabela C.63: Estrutura da tabela saw_phpbb_search_wordlist

Campo	Tipo	Nulo	Padrão
<i>word_text</i>	varchar(50)	Não	
word_id	mediumint(8)	Não	
word_common	tinyint(1)	Não	0

Tabela C.64: Estrutura da tabela saw_phpbb_search_wordmatch

Campo	Tipo	Nulo	Padrão
post_id	mediumint(8)	Não	0
word_id	mediumint(8)	Não	0
title_match	tinyint(1)	Não	0

Tabela C.65: Estrutura da tabela saw_phpbb_sessions

Campo	Tipo	Nulo	Padrão
<i>session_id</i>	char(32)	Não	
session_user_id	mediumint(8)	Não	0
session_start	int(11)	Não	0
session_time	int(11)	Não	0
session_ip	char(8)	Não	0
session_page	int(11)	Não	0
session_logged_in	tinyint(1)	Não	0

Tabela C.66: Estrutura da tabela saw_phpbb_smilies

Campo	Tipo	Nulo	Padrão
<i>smilies_id</i>	smallint(5)	Não	
code	varchar(50)	Sim	NULL
smile_url	varchar(100)	Sim	NULL
emoticon	varchar(75)	Sim	NULL

Tabela C.67: Estrutura da tabela saw_phpbb_themes

Campo	Tipo	Nulo	Padrão
<i>themes_id</i>	mediumint(8)	Não	
template_name	varchar(30)	Não	
style_name	varchar(30)	Não	
head_stylesheet	varchar(100)	Sim	NULL
body_background	varchar(100)	Sim	NULL
body_bgcolor	varchar(6)	Sim	NULL
body_text	varchar(6)	Sim	NULL
body_link	varchar(6)	Sim	NULL
body_vlink	varchar(6)	Sim	NULL
body_alink	varchar(6)	Sim	NULL
body_hlink	varchar(6)	Sim	NULL
tr_color1	varchar(6)	Sim	NULL
tr_color2	varchar(6)	Sim	NULL
tr_color3	varchar(6)	Sim	NULL
tr_class1	varchar(25)	Sim	NULL
tr_class2	varchar(25)	Sim	NULL
tr_class3	varchar(25)	Sim	NULL
th_color1	varchar(6)	Sim	NULL
th_color2	varchar(6)	Sim	NULL
th_color3	varchar(6)	Sim	NULL
th_class1	varchar(25)	Sim	NULL

Tabela C.67: Estrutura da tabela saw_phpbb_themes (continued)

Campo	Tipo	Nulo	Padrão
th_class2	varchar(25)	Sim	NULL
th_class3	varchar(25)	Sim	NULL
td_color1	varchar(6)	Sim	NULL
td_color2	varchar(6)	Sim	NULL
td_color3	varchar(6)	Sim	NULL
td_class1	varchar(25)	Sim	NULL
td_class2	varchar(25)	Sim	NULL
td_class3	varchar(25)	Sim	NULL
fontface1	varchar(50)	Sim	NULL
fontface2	varchar(50)	Sim	NULL
fontface3	varchar(50)	Sim	NULL
fontsize1	tinyint(4)	Sim	NULL
fontsize2	tinyint(4)	Sim	NULL
fontsize3	tinyint(4)	Sim	NULL
fontcolor1	varchar(6)	Sim	NULL
fontcolor2	varchar(6)	Sim	NULL
fontcolor3	varchar(6)	Sim	NULL
span_class1	varchar(25)	Sim	NULL
span_class2	varchar(25)	Sim	NULL
span_class3	varchar(25)	Sim	NULL
img_size_poll	smallint(5)	Sim	NULL
img_size_privmsg	smallint(5)	Sim	NULL

Tabela C.68: Estrutura da tabela saw_phpbb_themes_name

Campo	Tipo	Nulo	Padrão
<i>themes_id</i>	smallint(5)	Não	0
tr_color1_name	char(50)	Sim	NULL
tr_color2_name	char(50)	Sim	NULL
tr_color3_name	char(50)	Sim	NULL
tr_class1_name	char(50)	Sim	NULL
tr_class2_name	char(50)	Sim	NULL
tr_class3_name	char(50)	Sim	NULL
th_color1_name	char(50)	Sim	NULL
th_color2_name	char(50)	Sim	NULL
th_color3_name	char(50)	Sim	NULL
th_class1_name	char(50)	Sim	NULL
th_class2_name	char(50)	Sim	NULL
th_class3_name	char(50)	Sim	NULL
td_color1_name	char(50)	Sim	NULL

Tabela C.68: Estrutura da tabela saw_phpbb_themes_name
(continued)

Campo	Tipo	Nulo	Padrão
td_color2_name	char(50)	Sim	NULL
td_color3_name	char(50)	Sim	NULL
td_class1_name	char(50)	Sim	NULL
td_class2_name	char(50)	Sim	NULL
td_class3_name	char(50)	Sim	NULL
fontface1_name	char(50)	Sim	NULL
fontface2_name	char(50)	Sim	NULL
fontface3_name	char(50)	Sim	NULL
fontsize1_name	char(50)	Sim	NULL
fontsize2_name	char(50)	Sim	NULL
fontsize3_name	char(50)	Sim	NULL
fontcolor1_name	char(50)	Sim	NULL
fontcolor2_name	char(50)	Sim	NULL
fontcolor3_name	char(50)	Sim	NULL
span_class1_name	char(50)	Sim	NULL
span_class2_name	char(50)	Sim	NULL
span_class3_name	char(50)	Sim	NULL

Tabela C.69: Estrutura da tabela saw_phpbb_topics

Campo	Tipo	Nulo	Padrão
<i>topic_id</i>	mediumint(8)	Não	
forum_id	smallint(8)	Não	0
topic_title	char(60)	Não	
topic_poster	mediumint(8)	Não	0
topic_time	int(11)	Não	0
topic_views	mediumint(8)	Não	0
topic_replies	mediumint(8)	Não	0
topic_status	tinyint(3)	Não	0
topic_vote	tinyint(1)	Não	0
topic_type	tinyint(3)	Não	0
topic_first_post_id	mediumint(8)	Não	0
topic_last_post_id	mediumint(8)	Não	0
topic_moved_id	mediumint(8)	Não	0

Tabela C.70: Estrutura da tabela saw_phpbb_topics_watch

Campo	Tipo	Nulo	Padrão
topic_id	mediumint(8)	Não	0

Tabela C.70: Estrutura da tabela saw_phpbb_topics_watch
(continued)

Campo	Tipo	Nulo	Padrão
user_id	mediumint(8)	Não	0
notify_status	tinyint(1)	Não	0

Tabela C.71: Estrutura da tabela saw_phpbb_user_group

Campo	Tipo	Nulo	Padrão
group_id	mediumint(8)	Não	0
user_id	mediumint(8)	Não	0
user_pending	tinyint(1)	Sim	NULL

Tabela C.72: Estrutura da tabela saw_phpbb_users

Campo	Tipo	Nulo	Padrão
<i>user_id</i>	mediumint(8)	Não	
user_active	tinyint(1)	Sim	1
username	varchar(25)	Não	
user_password	varchar(32)	Não	
user_session_time	int(11)	Não	0
user_session_page	smallint(5)	Não	0
user_lastvisit	int(11)	Não	0
user_regdate	int(11)	Não	0
user_level	tinyint(4)	Sim	0
user_posts	mediumint(8)	Não	0
user_timezone	decimal(5,2)	Não	0.00
user_style	tinyint(4)	Sim	NULL
user_lang	varchar(255)	Sim	NULL
user_dateformat	varchar(14)	Não	d M Y H:i
user_new_privmsg	smallint(5)	Não	0
user_unread_privmsg	smallint(5)	Não	0
user_last_privmsg	int(11)	Não	0
user_emailtime	int(11)	Sim	NULL
user_viewemail	tinyint(1)	Sim	NULL
user_attachsig	tinyint(1)	Sim	NULL
user_allowhtml	tinyint(1)	Sim	1
user_allowbbcode	tinyint(1)	Sim	1
user_allowsmile	tinyint(1)	Sim	1
user_allowavatar	tinyint(1)	Não	1
user_allow_pm	tinyint(1)	Não	1
user_allow_viewonline	tinyint(1)	Não	1

Tabela C.72: Estrutura da tabela saw_phpbb_users (continued)

Campo	Tipo	Nulo	Padrão
user_notify	tinyint(1)	Não	1
user_notify_pm	tinyint(1)	Não	0
user_popup_pm	tinyint(1)	Não	0
user_rank	int(11)	Sim	0
user_avatar	varchar(100)	Sim	NULL
user_avatar_type	tinyint(4)	Não	0
user_email	varchar(255)	Sim	NULL
user_icq	varchar(15)	Sim	NULL
user_website	varchar(100)	Sim	NULL
user_from	varchar(100)	Sim	NULL
user_sig	text	Sim	NULL
user_sig_bbcode_uid	varchar(10)	Sim	NULL
user_aim	varchar(255)	Sim	NULL
user_yim	varchar(255)	Sim	NULL
user_msnm	varchar(255)	Sim	NULL
user_occ	varchar(100)	Sim	NULL
user_interests	varchar(255)	Sim	NULL
user_actkey	varchar(32)	Sim	NULL
user_newpasswd	varchar(32)	Sim	NULL

Tabela C.73: Estrutura da tabela saw_phpbb_vote_desc

Campo	Tipo	Nulo	Padrão
<i>vote_id</i>	mediumint(8)	Não	
topic_id	mediumint(8)	Não	0
vote_text	text	Não	
vote_start	int(11)	Não	0
vote_length	int(11)	Não	0

Tabela C.74: Estrutura da tabela saw_phpbb_vote_results

Campo	Tipo	Nulo	Padrão
vote_id	mediumint(8)	Não	0
vote_option_id	tinyint(4)	Não	0
vote_option_text	varchar(255)	Não	
vote_result	int(11)	Não	0

Tabela C.75: Estrutura da tabela saw_phpbb_vote_voters

Campo	Tipo	Nulo	Padrão
vote_id	mediumint(8)	Não	0
vote_user_id	mediumint(8)	Não	0
vote_user_ip	char(8)	Não	

Tabela C.76: Estrutura da tabela saw_phpbb_words

Campo	Tipo	Nulo	Padrão
<i>word_id</i>	mediumint(8)	Não	
word	char(100)	Não	
replacement	char(100)	Não	

Apêndice D

Questionário para o estudo de caso em MAC118 - PRÉ-TESTE

Questionário para o aluno

Caro aluno, para que possamos continuar a desenvolver sistemas e atividades dentro do IME-USP, gostaríamos de contar com sua valiosa colaboração respondendo ao questionário abaixo.

1. Idade:
2. Com que frequência utiliza o computador?:
3. Onde utiliza o computador mais frequentemente?
4. Já lecionou? Em qual escola?
5. Já usou o computador para lecionar? Qual programa? Como foi o uso?
6. Já fez algum curso à distância? Qual?
7. Como foi a experiência? Procure explicar resumidamente o porquê:
8. Qual Ambiente Virtual utilizou? (cite o nome do sistema computacional que gerenciava o curso)
9. Você faria um curso de graduação ou pós-graduação à distância? Procure explicar resumidamente o porquê:
10. Você utilizou régua e compasso (com construções) no ensino médio:

11. Quais construções geométricas você se lembra de ter construído com régua e compasso?
(citar algumas)
12. Já conhecia a termo Geometria Dinâmica - GD?
13. Já conhecia o iGeom ou algum software de GD? Qual?
14. Em relação à régua e compasso quais as semelhanças/dificuldades encontradas nos exercícios de hoje com o iGeom? (use como exemplo o exercício da construção do ponto médio)
15. Sobre a avaliação automática dos exercícios no iGeom+SAW, o que achou? Por que?
Sobre a aula de hoje:
16. Achou a aula válida? Procure explicar resumidamente o porquê:
17. Entendeu o sistema gerenciador *SAW*? Procure explicar resumidamente o porquê:
18. Gostou da experiência? Procure explicar resumidamente o porquê:
19. Quais as dificuldades encontradas?
20. Sugestões ou reclamações:

Apêndice E

Questionário para o estudo de caso em MAC118 - PÓS-TESTE

Questionário para o aluno

Para que possamos continuar a desenvolver sistemas e atividades dentro do IME-USP, gostaríamos de contar com sua colaboração respondendo ao questionário abaixo.

1. Você considera o *SAW* um sistema:

Excelente Muito Bom Bom Regular Fraco Insuficiente

2. Sobre o uso do *SAW* com o *iGeom* estimular seu estudo na disciplina *MAC118*:

Excelente Muito Bom Bom Regular Fraco Insuficiente

3. Sobre o uso do *SAW* com o *iGeom* estimular sua participação durante as aulas

Excelente Muito Bom Bom Regular Fraco Insuficiente

4. Sobre o retorno (correção) das atividades realizadas no *SAW*:

Excelente Muito Bom Bom Regular Fraco Insuficiente

Comente:

5. A utilização do *SAW* fora do horário da aula foi:

Excelente Muito Bom Bom Regular Fraco Insuficiente

Comente:

6. Como você se avalia em relação à disciplina *MAC118*?

Excelente Muito Bom Bom Regular Fraco Insuficiente

Comente:

7. Sobre a facilidade de encontrar as informações disponibilizadas pelo professor no *SAW*:

Excelente Muito Bom Bom Regular Fraco Insuficiente

Comente:

8. Sobre a facilidade de compreensão do enunciado e a realização dos exercícios das aulas:

Excelente Muito Bom Bom Regular Fraco Insuficiente

9. Em relação ao modo de apresentação, como foi a realização dos exercícios?

Excelente Muito Bom Bom Regular Fraco Insuficiente

Comente:

10. Sobre o uso do *SAW* com o *iGeom* em relação ao ensino/aprendizagem em geometria:

Excelente Muito Bom Bom Regular Fraco Insuficiente

11. Em relação à ferramenta dicionário do *SAW*, conseguiu resolver os trabalhos propostos?

Excelente Muito Bom Bom Regular Fraco Insuficiente

Comente:

12. No uso do editor de texto do dicionário, dê sua opinião em relação aos itens abaixo:

Avaliação	Não utilizei	Ótimo	Bom	Regular	Fraco
Inserção de applet					
Inserção de figura					
Inserção de fórmulas					
Inserção de links					
Edição de textos					

13. Você faria outro curso que utilizasse o *SAW*? Excelente Muito Bom Bom Regular Fraco Insuficiente

Comente:

14. Quais as vantagens que você encontrou no *SAW*?

15. Quais as desvantagens que você encontrou no *SAW*?

16. Sugestões:

Referências Bibliográficas

- (2007). *BlackBoard*. <http://www.blackboard.com/us/index.aspx/>. Acesso em: 23 jan. 2007.
- (2007). *Boycott blackboard*. Disponível em: <http://www.boycottblackboard.org/index.php/>. Acesso em: 23 jan. 2007.
- (2007). *EduWeb*. <http://www.eduweb.com.br/>. Acesso em: 23 jan. 2007.
- (2007). *FCKeditor*. <http://www.fckeditor.net/>. Acesso em: 23 jan. 2007.
- (2007). *HtmlArea*. <http://www.htmlarea.com/>. Acesso em: 23 jan. 2007.
- (2007). *Interbase*. <http://www.borland.com/interbase/>. Acesso em: 23 jan. 2007.
- (2007). *LaTeX: A document preparation system*. <http://www.latex-project.org/>. Acesso em: 23 jan. 2007.
- (2007). *Microsoft SQL*. <http://www.microsoft.com/sql/>. Acesso em: 23 jan. 2007.
- (2007). *Modellus - Interactive Modelling with mathematics*. <http://phoenix.sce.fct.unl.pt/modellus/>. Acesso em: 23 jan. 2007.
- (2007). *MySQL*. <http://www.mysql.com/>. Acesso em: 23 jan. 2007.
- (2007). *Oracle*. <http://www.oracle.com.br/>. Acesso em: 23 jan. 2007.
- (2007). *Perl*. <http://www.perl.com/>. Acesso em: 23 jan. 2007.
- (2007). *PHP*. <http://www.php.net/>. Acesso em: 23 jan. 2007.
- (2007). *PostgreSQL*. <http://www.postgresql.org/>. Acesso em: 23 jan. 2007.
- (2007). *Rich Text Editor*. <http://www.dynamicdrive.com/dynamicindex16/richtexteditor/>. Acesso em: 23 jan. 2007.

- (2007). *TinyMCE*. <http://tinymce.moxiecode.com/index.php/>. Acesso em: 23 jan. 2007.
- (2007a). *Wingeo*. <http://math.exeter.edu/rparris/winggeom.html/>. Acesso em: 23 jan. 2007.
- (2007b). *Winplot*. <http://math.exeter.edu/rparris/winplot.html/>. Acesso em: 23 jan. 2007.
- (2007). *WYSIWYG*. <http://whatis.techtarget.com/definition/>. Acesso em: 23 jan. 2007.
- Aronsson, L. (2002). Operation of a large scale, general purpose wiki website: Experience from susning.nu's first nine months in service. Verlag für Wissenschaft und Forschung Berlin. Disponível em: <http://aronsson.se/wikipaper.html>, pp. 27–37. Proceedings of the 6th International ICCC/IFIP Conference on Electronic Publishing held in Karlovy Vary.
- Balbino, J. (2006). *Patentearam o ensino a distância...* Disponível em: <http://www.cipsga.org.br/article.php?sid=8208>. Acesso em: 23 jan. 2007.
- Basso, M. V. d. A., L. da Cruz Fagundes, L. M. R. Tarouco, and A. C. da Rocha Costa (1999). Educação tecnológica e/na educação matemática - aplicações da matemática elementar na sala de aula. <http://www.nied.unicamp.br/oea>. Revista de Informática na Educação.
- Brusilovsky, P. (1996). *Methods and techniques of adaptive hypermedia*. Journal User Modeling and User-Adapted Interaction: Springer Netherlands.
- Brusilovsky, P. (1998). Adaptive educational systems on the world wide web: a review of available technologies. pp. 16–19. Proceedings of Workshop WWW-Based Tutoring at 4th International Conference on Intelligent Tutoring Systems - ITS98.
- Brusilovsky, P. (2000). Adaptive hypermedia: From intelligent tutoring systems to web-based. pp. 1–7. Proceedings of the 5th International Conference on Intelligent Tutoring Systems - ITS02.
- Buschmann, F., R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal (1996). *Pattern-Oriented Software Architecture*. Hardcover: Proc ACM symposium on Operating System Principles.
- Cabeda, M. (2005). O chat-forum : Uma idéia de uso híbrido, síncrono e assíncrono, através de uma única ferramenta normalmente assíncrona, o fórum virtual. Florianópolis - SC. 12º Congresso Internacional de Educação a distância - ABED.
- Cavaroli, J. T. and J. M. A. Coello (2004). Alerts fórum: Um sistema emissor de alertas a partir da avaliação de mensagens categorizadas em fóruns de discussão. Actas do VII Congresso IberoAmericano de Informática Educativa. Monterrey.

- ClassWeb (2007). *ClassWeb Open Source*.
- Cole, J. R. (2005). *Using Moodle: Teaching with the Popular Open Source Course Management System*. O'Reilly & Associates Inc.
- Costa, M., H. Gandra, L. Andrade, and C. Lima (2004). Aplicação de softwares para educação matemática numa turma de licenciatura em matemática da universidade federal do rio de janeiro - ufrj. In *Anais do VIII Encontro Nacional de Educação Matemática*, Recife. VIII Encontro Nacional de Educação Matemática.
- Depow, J. (2003). Open source software: Two learning management systems. Volume 4. International Review of Research in Open and Distance Learning.
- Dijkstra, E. W. (1968). The structure of the 'the' - multiprogramming system. Volume 11, <http://www.cs.virginia.edu/zaher/classes/CS656/p341-dijkstra.pdf>. Proc ACM symposium on Operating System Principles.
- Dougiamasé, M. (2006). *Philosophy*. <http://docs.moodle.org/en/Philosophy/>. Acesso em: 23 jan. 2007.
- EdTechPost (2007). *Open Source Course Management Systems Search*. <http://www.edtechpost.ca/>. Acesso em: 23 jan. 2007.
- education patents, N. (2006). *English Translation of the Blackboard Patent Claims*. Disponível em: <http://noedupatents.org/wiki/index.php?title=Blackboard>. Acesso em: 23 jan. 2007.
- EduWeb (2007). *AulaNet - Sistema AulaNet de Ensino a Distância*. <http://www.eduweb.com.br/portugues/home.asp/>. Acesso em: 23 jan. 2007.
- Ellis, R. K. (2001). *LCMS Roundup*. Disponível em <http://www.learningcircuits.org/2001/aug2001/ttools.html>.
- Ferreira, A. B. d. H. (2004). *Novo dicionário Aurelio da língua portuguesa*. Editora Positivo.
- Filho, A. R. P. (2005). *Ambiente de Aprendizagem Moodle UnB Manual do Professor*. Universidade de Brasília.
- Fuks, H., M. L. Cunha, M. A. Gerosa, and C. J. P. Lucena (2003). *Participação e Avaliação no Ambiente Virtual AulaNet da PUC-Rio*. Educação Online: Teorias, Práticas, Legislação e Formação Corporativa.

- Fuks, H., M. Gerosa, and M. Pimentel (2003). Projeto de comunicação em groupware: Desenvolvimento, interface e utilização. In *XXII Jornada de Atualização em Informática, Anais do XXIII Congresso da SBC*, Volume 2, Cap. 7, pp. 295–338.
- Fuks, H., M. Gerosa, A. Raposo, and C. Lucena (2004). O modelo de colaboração 3c no ambiente aulanet. *Informática na Educação: Teoria e Prática* 7(1), 25–48.
- Fuks, H., M. A. Gerosa, and C. J. P. de Lucena (2002). Usando a categorização e estruturação de mensagens textuais em cursos pelo ambiente aulanet. In *Revista Brasileira de Informática na Educação*, Volume 10. Sociedade Brasileira de Computação.
- Fussell, S. R., R. E. Kraut, F. J. Lerch, W. L. Schertis, M. M. McNally, and J. J. Cadiz (1998). Coordination, overload and team performance: effects of team communication strategies. In *Proceedings of CSCW '98*.
- GO, C. (2007). *Centro Federal de Educação Tecnológica de Goiás*. <http://www.cefetgo.br/>. Acesso em: 23 jan. 2007.
- Goldberg, M. W. and S. Salari (1997). An update on webct (world-wide-web course tools) - a tool for the creation of sophisticated web-based learning environments.
- Gravina, M. A. (1996). Geometria dinamica - uma nova abordagem para o aprendizado da geometria. In *Anais do VII Simpósio Brasileiro de Informática na Educação*, pp. 1–13. Simpósio Brasileiro de Informática na Educação.
- Grothman, R. (2007, jan). *C.A.R - Compass And Rules*. Disponível em: http://mathsrv.ku-eichstaett.de/MGF/homes/grothmann/java/zirkel/doc_en/index.html.
- Guerra, J. H. L. (2000). Utilização do computador no processo de ensino-aprendizagem: Um aplicação em planejamento e controle da produção. Dissertação de mestrado em engenharia de produção, Universidade de São Paulo.
- Haydt, R. C. (2000). *Avaliação do processo ensino aprendizagem*. Ática.
- IME-USP (2007). *Instituto de Matemática e Estatística da Universidade de São Paulo*. <http://www.ime.usp.br/>. Acesso em: 23 jan. 2007.
- Inc., B. (2006). *Blackboard Awarded Patent on e-Learning Technology*. <http://www.blackboard.com/company/press/release.aspx?id=887622/>. Acesso em: 23 jan. 2007.

- Isotani, S. (2005). Desenvolvimento de ferramentas no igeom: utilizando a geometria dinâmica no ensino presencial e a distância. Dissertação para mestre em ciência da computação, Universidade de São Paulo - USP.
- Isotani, S. and L. de Oliveira Brandão (2003). Uma ferramenta para ensino de geometria dinâmica na internet: igeom. In *Anais do Workshop sobre Informática na Escola*, pp. 1476–1487. XXIII Congresso da Sociedade Brasileira de Computação.
- Itmazi, J. A., M. G. Megías, P. Paderewski, and F. L. G. Vela (2005). A comparison and evaluation of open source learning management systems. In *IADIS AC*, pp. 80–86.
- Jackiw, N. (1995). *The Geometer's Sketchpad*, Volume 3.0. Berkeley: Key Curriculum Pres.
- Java (2007). *Linguagem de Programação Java*. <http://java.sun.com/>. Acesso em: 23 jan. 2007: Sun Microsystems.
- Javascript (2007). *Javascript*. <http://www.w3.org/TR/html4/interact/scripts.html/>. Acesso em: 23 jan. 2007.
- Jones, K. (2003). Using the internet in the teaching and learning of mathematics: a research bibliography. *Micromath 19*(2), 43–44.
- Kaplan-Leiserson, E. (2006). *Glossary*. Disponível em <http://www.learningcircuits.org/glossary.html>.
- Kennedy, D. M. (2005, June). Student managed learning management systems: Teachers as designers. In P. Kommers and G. Richards (Eds.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2005*, Montreal, Canada, pp. 3172–3178. AACE.
- Kernighan, B. W. and D. M. Ritchie (1988). *The C Programming Language* (2 ed.). Prentice-Hall.
- Kortenkamp, U. (1999). *Foundation of Dynamic Geometry*. Dissertation for ph.d. of technical sciences, Swiss Federal Institute of Technology Zurich, Disponível em <http://kortenkaamps.net/papers/diss.pdf>.
- Ksoft (2007). *Graphmatica*. <http://www8.pair.com/ksoft/>. Acesso em: 23 jan. 2007.
- Laborde, J. M. and F. Bellemain (1997). *Cabri Geometry II*. Dallas: Texas Instruments.

- LES (2007). *Laboratório de Engenharia de Software*. <http://www.les.inf.puc-rio.br/>. Acesso em: 23 jan. 2007.
- License, G. G. P. (2007). *GPL*. <http://www.gnu.org/copyleft/gpl.html/>. Acesso em: 23 jan. 2007.
- Likert, R. (1932). *A Technique for the Measurement Attitudes*. Archives of Psychology.
- Litto, F. M., A. Filatro, and C. André (2004). Brazilian research on distance learning, 1999-2003: A state-of-the art study. In *Proceedings of International Congress of Distance Education*, <http://www.abed.org.br/congresso2004/por/pdf/180-TC-D4.pdf>.
- Manhattan (2007). *What is it?* <http://manhattan.sourceforge.net/>. Acesso em: 23 jan. 2007.
- Melo, L., J. M. Ferreira, and J. D. A. Pontes (2000). Um software educacional para o descobrimento de propriedades matemáticas. In *Anais do XX Congresso da Sociedade Brasileira de Computação*, pp. 1476–1487.
- Menezes, R. A., H. Fuks, and A. C. B. Garcia (1998). Utilizando agentes no suporte à avaliação informal no ambiente de instrução baseada na web - aulanet. In *IX Simpósio Brasileiro de Informática na Educação*, Fortaleza.
- Microsoft (2007). *Access*. <http://office.microsoft.com/access/>. Acesso em: 23 jan. 2007.
- Miranda Júnior, M. d. R. (2005). Introdução ao uso da informática no ensino de física no ensino médio. Dissertação de mestrado em ensino de física, Universidade Federal do Rio Grande do Sul.
- Moodle (2007). *Course Management System*. <http://moodle.org/>. Acesso em: 23 jan. 2007.
- Moodle, D. G. (2004). A free, open source course management system for online learning.
- Moraes, M. B. d. S. A. and R. M. R. Teixeira (2005). Uma proposta inovadora para o ensino de física no nível médio: Utilização de novas tecnologias no ensino de eletrodinâmica. IV Encontro Ibero-Americano de Coletivos Escolares e Redes de Professores que Fazem Investigação na sua Escola.
- Moura, J. G. and L. de Oliveira Brandão (2005a). Aplicações no saw - sistema de aprendizagem pela web. In *Anais do XVI Simpósio Brasileiro de Informática na Educação - SBIE2005*.

Moura, J. G. and L. de Oliveira Brandão (2005b). Saw - sistema de aprendizagem pela web: incorporando novos recursos. In *Anais do XI Simpósio Brasileiro de Sistemas Multimídia - Webmedia2005*.

MuPAD (2007). *Mathematics mastered with MuPAD*. Disponível em: <http://www.sciface.com/>. Acesso em: 23 jan. 2007.

Nelson, G. E. (1998). On-line evaluation: Multiple choice, discussion questions, essay, and authentic projects. In: *Third Teaching in the Community Colleges Online Conference*, Kapiolani Community College.

Nichani, M. (2001). *LCMS = LMS + CMS*. http://www.elearningpost.com/articles/archives/lcms_lms_cms_rlos

Oeiras, J. Y. Y. (2005). Design de ferramentas de comunicação para colaboração em ambientes de educação a distância. Dissertação de doutorado em computação, Universidade Estadual de Campinas.

Otsuka, J. L., R. L. Lachi, T. B. Ferreira, and H. V. da Rocha (2002). Suporte à avaliação formativa no ambiente de educação a distância teleduc. *Revista Brasileira de Informática na Educação - RBIE*.

Patent, U. S. and T. Office (2006). *Blackboard Patent*. <http://patft.uspto.gov/>. Acesso em: 23 jan. 2007.

Pereira, J. C. L. and M. P. Bax (2002). Introdução à gestão de conteúdos. In *Anais. 1o. Congresso Anual da Sociedade Brasileira de Gestão do Conhecimento*, São Paulo. 3º Workshop Brasileiro de Inteligência Competitiva e Gestão do Conhecimento.

phpBB. *phpBB*. <http://www.phpbb.com/>. Acesso em: 23 jan. 2007.

Pimentel, M. G., H. Fuks, and C. J. P. de Lucena (2003). Debatí, debati... aprendi? investigações sobre o papel educacional das ferramentas de bate-papo. In *Anais do XXIII Congresso da Sociedade Brasileira de Computação - WIE2003*, Campinas - SP. IX Workshop de Informática na Escola.

Pope, J. (2007). *Patent fight rattles academic computing*. <http://www.usatoday.com/tech/products/software/>. Acesso em: 23 jan. 2007.

- Prado, R. d. and L. de Oliveira Brandão (2006). *igraf: uma proposta de sistema para ensino/aprendizagem de função na web*. In *Anais do III HTEM - História e Tecnologia no Ensino de Matemática*.
- PUCSP (2007). *Pontifícia Universidade Católica de São Paulo*. <http://www.pucsp.br/>. Acesso em: 23 jan. 2007.
- Rocha, H. V. d. (2002). *Projeto teleduc: Pesquisa e desenvolvimento de tecnologia para educação à distância*. Associação Brasileira de Educação a Distância - ABED.
- Romani, L. A. S. (2000). *Intermap: Ferramenta para visualização da interação em ambientes de educação a distância na web*. Dissertação de mestrado em ciência da computação, Universidade Estadual de Campinas.
- Scilab (2007). *The open source platform for numerical computation*. Disponível em: <http://www.scilab.org>. Acesso em: 23 jan. 2007.
- SCORM. *Advanced Distributed Learning*. <http://www.adlnet.gov/index.cfm/> Acesso em: 23 jan. 2007.
- Tancredi, R. M. S. P. (2002). *O acompanhamento do processo ensino-aprendizagem através das provas escritas*. Projeto Integrado de Ciências e Matemática para professores da rede pública - UFSCar.
- TeLEduc (2007). *Ambiente de suporte para ensino-aprendizagem a distância*. <http://teleduc.nied.unicamp.br/teleduc/>. Acesso em: 23 jan. 2007.
- Thomas, M. D., P. R. Patel, A. D. Hudson, and D. A. B. Jr (1997). *Programando em Java para Internet*. Makron Books.
- TIDIA (2007). *Programa de Tecnologia da Informação no Desenvolvimento da Internet Avançada*. <http://www.tidia.fapesp.br/portal/>. Acesso em: 23 jan. 2007.
- Togni, A. C. (2005). *Estudando e avaliando atividades pedagógicas em matemática utilizando softwares matemáticos e o ambiente teleduc*. IV Encontro Ibero-Americano de Coletivos Escolares e Redes de Professores que Fazem Investigação na sua Escola.
- Tools, E. (2007). <http://www.edutools.info/>. Acesso em: 23 jan. 2007.
- UFCG (2007). *Universidade Federal de Campina Grande*. <http://www.ufcg.edu.br/>. Acesso em: 23 jan. 2007.

- UFP (2007). *Universidade Federal da Paraíba*. <http://www.ufpb.br/>. Acesso em: 23 jan. 2007.
- UFRGS (2007). *Universidade Federal do Rio Grande do Sul*. <http://www.ufrgs.br/ufrgs/>. Acesso em: 23 jan. 2007.
- UFSCar (2007). *Universidade Federal de São Carlos*. www.ufscar.br/. Acesso em: 23 jan. 2007.
- UnB (2007). *Universidade de Brasília*. <http://www.unb.br/>. Acesso em: 23 jan. 2007.
- Unicamp (2007). *Universidade Estadual de Campinas*. <http://www.unicamp.br/>. Acesso em: 23 jan. 2007.
- UNIFESP (2007). *Universidade Federal de São Paulo*. www.unifesp.br/. Acesso em: 23 jan. 2007.
- USF (2007). *Universidade São Francisco*. <http://www.usf.br/>. Acesso em: 23 jan. 2007.
- WebCT (2007). *Web Course Tools*. <http://www.webct.com/>. Acesso em: 23 jan. 2007.
- Yin, R. K. (2005). *Estudo de caso: planejamento e métodos*. Editora Bookman.

Índice Remissivo

Adaptação

Adaptação do conteúdo, 25

Navegação adaptativa, 25

Arquitetura, 22

Camada de dados, 22

Camada de Tarefas, 24

Camada de Comunicação, 26

Camada de Interação, 27

Módulos de aprendizagem, 27

Avaliações automáticas, 10

Camada de dados

Dados de conteúdo, 23

Dados de curso, 23

Dados gerais, 23

Modelo do aluno, 23

Camada de Tarefas

Adaptação, 25

Autoria, 24

Colaboração, 9

colaborativo, 13

Componentes, 23

Conteúdo específico

Graphmatica, 11

Modellus, 11

Winggeom, 11

Winplot, 11

Conteúdo específico, 11

Dicionário Webmídia, 38

tópicos, 38

vocábulos, 38

Funcionalidades, 8

Bate-papo, 9

Disponibilização de conteúdo, 9

Fórum de discussão, 8

Prova online, 10

funcionalidades, 12

Gabarito, 11

Geometria Dinâmica - GD, 1

Cabri, 3

GSP, 3

iGeom, 3

Geometria Dinâmica, 11

GPL, 39

Interbase, 13

Java, 12

módulos, 13

Módulos de aprendizagem

Adaptação, 28

Autoria, 28

Avaliador, 28

Comunicação, 28

iCG, 30

- iGeom, 29
- iGraf, 30
- Interface, 28
- Núcleo, 28
- Moodle, 13
- MySQL, 12

- ODBC, 13
- online, 13

- Perl, 12
- PHP, 12
- PostgreSQL, 13

- SGC, 2
 - CMS, 2, 5
 - LCMS, 2, 5
 - LMS, 2, 5
- síncrono, 9
- sistema livre, 13
- sistemas, 5
 - Aulanet, 5
 - Blackboard/WebCT, 5
 - ClassWeb, 5
 - Manhattan, 5
 - Moodle, 5
 - TelEduc, 5

- TelEduc, 12

- Web, 1
 - Aberto, 1
 - Fechado, 1