

# **IGraf: Uma proposta de sistema para ensino de função via Web**

REGINALDO DO PRADO <sup>†</sup>

Orientador:

LEÔNIDAS DE OLIVEIRA BRANDÃO

DISSERTAÇÃO APRESENTADA AO  
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA  
DA UNIVERSIDADE DE SÃO PAULO  
COMO PARTE DOS REQUISITOS PARA OBTENÇÃO DO  
TÍTULO DE MESTRE EM CIÊNCIA DA COMPUTAÇÃO

**São Paulo**  
**Maio de 2008**

<sup>†</sup>Durante a elaboração deste trabalho o autor contou com apoio financeiro do Programa Bolsa Mestrado da Secretaria da Educação do Estado de São Paulo.



# Resumo

A Educação a Distância (EAD) tem experimentado um grande crescimento nos últimos anos e o crescimento do uso de Sistemas Gerenciadores de Curso (SGC) tem acompanhado essa tendência.

Esse fato cria uma demanda por ferramentas (componentes de software) que se integrem aos SGC's facilitando – pelo ponto de vista docente – a criação, gerenciamento e distribuição de conteúdo instrucional via *web* e – pelo ponto de vista do aprendiz – a interação com as atividades sem ter que buscar recursos auxiliares fora do ambiente gerenciador do seu curso.

Neste trabalho apresentamos o **iGraf - Gráficos Interativos na Internet**, um visualizador para gráficos de funções matemáticas que pode ser integrado a sistemas gerenciadores de cursos pela *web*.

Dentre os principais recursos desenvolvidos destacamos o sistema de animação e o sistema de edição que permitem ao usuário interagir com o desenho dos gráficos; a implementação de ferramentas relacionadas ao estudo do Cálculo Diferencial e Integral; o sistema de script que permite ao professor analisar os passos que levaram o aluno a um determinado resultado na solução de um exercício; o mecanismo de comunicação que permite ao aluno enviar seus resultados a um servidor para análise do professor e o sistema de avaliação automática, que mostra ao aluno, no momento do envio, a correção dos seus resultados.



# Abstract

Distance Learning has experienced an intensive growth in the last years and the use of Learning Management Systems (LMS) has followed this pattern.

This fact generates a demand for tools (software components) that integrates itself to LMS's making easier – by the teacher's point of view – the creation, management and distribution of instructional content by web and – by the learner point of view – the interaction with activities without have to look for complementary resources out of your course manager environment.

In this work we present the **iGraf – Interactive Graphics on Internet**, a mathematic function graphic visualizer which can be integrated to course management systems by web.

Among the main resources developed we point out the animation and the edition systems that allows the user interaction with the graphic plotting; the implementation of routines related to Differential and Integral Calculus study; the script system that allow teacher analyze the steps that get pupil to some result in the solution of an assignment; the communication engine that allows pupil send your results to a server for teacher analysis and the automatic evaluation system that gives immediate feedback to student about your results.



# Sumário

<b>Resumo</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>Lista de Figuras</b>	<b>ix</b>
<b>Lista de Tabelas</b>	<b>xi</b>
<b>Lista de Abreviaturas e Siglas</b>	<b>xiii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 História dos Sistemas Gerenciadores de Curso . . . . .	1
1.1.1 Instrução Programada . . . . .	2
1.1.2 SAKI . . . . .	2
1.1.3 PLATO . . . . .	3
1.2 Sistemas Gerenciadores de Cursos (SGC) Atuais . . . . .	3
1.3 Delimitação do Problema . . . . .	5
1.4 Justificativas . . . . .	6
1.5 Objetivos . . . . .	7
1.6 Organização do Texto . . . . .	8
<b>Introdução</b>	<b>1</b>

<b>2</b>	<b>Software gráfico para computadores</b>	<b>9</b>
2.1	CAS (Computer Algebra System) . . . . .	10
2.1.1	Maple . . . . .	10
2.1.2	Mathematica . . . . .	12
2.1.3	SAGE . . . . .	13
2.1.4	SciLab . . . . .	15
2.1.5	Maxima . . . . .	16
2.1.6	Yacas . . . . .	17
2.1.7	Singular . . . . .	18
2.2	Visualizadores gráficos <i>off-line</i> . . . . .	20
2.2.1	Graphmatica . . . . .	20
2.2.2	Winplot . . . . .	21
2.3	Visualizadores gráficos <i>on-line</i> . . . . .	22
2.3.1	Wessa's Equation Plotter . . . . .	23
2.3.2	WebGraphing . . . . .	23
2.3.3	Cornell Equation Plotter (relplot) . . . . .	24
2.3.4	QuickMath . . . . .	25
2.4	Arcabouços ( <i>Frameworks</i> ) . . . . .	25
2.4.1	Java Components for Math Project (JCM) . . . . .	25
2.4.2	Java Tools and Libraries for the Advancement of Sciences (jScience)	26
2.4.3	Meditor . . . . .	27
2.4.4	Java e-Learning Simulations (JeLSIM) . . . . .	27
<b>3</b>	<b>O Programa iGraf</b>	<b>29</b>
3.1	Tipos de Usuário . . . . .	29
3.2	Tipos de Gráfico . . . . .	30

3.3	A Interface Gráfica . . . . .	31
3.3.1	Barra de Título . . . . .	32
3.3.2	Área de Edição . . . . .	33
3.3.3	Menu de Botões . . . . .	33
3.3.4	Área de Desenho . . . . .	33
3.3.5	Barra de Mensagens . . . . .	34
3.4	Menu Gráfico . . . . .	34
3.4.1	Edição de Expressões . . . . .	34
3.5	Menu Cálculo . . . . .	35
3.5.1	Visualizar Derivada . . . . .	35
3.5.2	Visualizar Reta Tangente . . . . .	36
3.5.3	Visualizar Integral Indefinida . . . . .	36
3.5.4	Calcular Integral Definida . . . . .	38
3.6	Menu Animação . . . . .	39
3.6.1	Controle Parar / Animar . . . . .	41
3.7	Menu Área de Desenho . . . . .	44
3.7.1	Controles de <i>Zoom</i> . . . . .	44
3.7.2	Edição de Texto . . . . .	45
3.8	Menu Exercício . . . . .	45
3.8.1	Histórico . . . . .	45
<b>4</b>	<b>Autoria e Validação Automática de Exercícios</b>	<b>47</b>
4.1	Estudo de Função . . . . .	48
4.2	Modalidades de respostas em atividades com funções . . . . .	49
4.2.1	Dados obtidos . . . . .	52
4.3	Autoria de Exercícios . . . . .	52

4.3.1	Definição de Gabaritos . . . . .	54
4.3.2	Geração de Arquivos . . . . .	55
4.4	Critérios de Validação . . . . .	57
4.4.1	Tratamento para repostas numéricas . . . . .	58
4.4.2	Tratamento para respostas do tipo conjunto numérico . . . . .	58
4.4.3	Tratamento para respostas na forma de expressões algébricas . . . . .	59
4.4.4	Tratamento para outros tipos de respostas . . . . .	61
<b>5</b>	<b>Análise do iGraf</b>	<b>65</b>
5.1	Análise da Interface Gráfica . . . . .	65
5.1.1	Heurísticas de Nielsen . . . . .	66
5.2	Teste no laboratório do CEC . . . . .	69
5.2.1	A aplicação das atividades . . . . .	70
5.2.2	O questionário pós-teste . . . . .	70
5.2.3	Resultados da aplicação do questionário . . . . .	71
5.2.4	Conclusões do capítulo . . . . .	83
<b>6</b>	<b>Conclusões</b>	<b>85</b>
6.1	Contribuições . . . . .	86
6.2	Trabalhos Futuros e em Andamento . . . . .	87
6.2.1	Melhorias em Andamento . . . . .	87
6.2.2	Trabalhos Futuros . . . . .	87
	<b>Bibliografia</b>	<b>104</b>

# Lista de Figuras

2.1	Maple 10 . . . . .	11
2.2	Mathematica . . . . .	12
2.3	Sage . . . . .	14
2.4	Scilab . . . . .	15
2.5	Maxima . . . . .	16
2.6	Yacas . . . . .	17
2.7	Parametrização no Graphmatica . . . . .	21
2.8	Painel de parâmetros . . . . .	22
2.9	Saída gráfica . . . . .	22
3.1	Tela inicial do iGraf . . . . .	32
3.2	Menu de operações com gráficos . . . . .	35
3.3	Ferramentas do Cálculo Diferencial e Integral . . . . .	36
3.4	Tangente de $\cos(x)$ em $x = 1$ . . . . .	37
3.5	Configuração que gerou a figura 3.4 . . . . .	37
3.6	Dinstinção explícita entre integral e área . . . . .	39
3.7	Cálculo de área . . . . .	40
3.8	Pintura diferenciada das regiões “positiva” e “negativa” . . . . .	40
3.9	Opções para controle de animação . . . . .	41
3.10	Painel de configuração dos parâmetros da animação . . . . .	42

3.11	Quadros de uma animação controlada por <i>slider</i> . . . . .	43
3.12	Controles para o “papel” do gráfico . . . . .	44
3.13	Opções ligadas a exercícios . . . . .	46
4.1	Relação percentual entre os tipos de resposta . . . . .	53
4.2	Opções para a criação e edição de exercícios . . . . .	53
4.3	Janela de Configuração do Gabarito . . . . .	54
4.4	Opções de tipos de resposta . . . . .	55
4.5	Janela com os tipos possíveis de resposta . . . . .	56
4.6	Geração de página HTML: apenas um clique . . . . .	57
4.7	Configuração do gabarito: resposta numérica . . . . .	58
4.8	Configuração do gabarito: resposta ponto . . . . .	59
4.9	Configuração do gabarito: intervalo numérico . . . . .	60
4.10	Expressões . . . . .	61
4.11	Gabarito vazio: envio de respostas discursivas . . . . .	62
4.12	Opção de resposta de exercícios . . . . .	63
5.1	Frequência de uso do computador . . . . .	72
5.2	Opiniões sobre o visual do programa . . . . .	73
5.3	Opiniões sobre a facilidade de uso inicial . . . . .	74
5.4	Opiniões sobre a qualidade dos gráficos . . . . .	75
5.5	Opiniões sobre o texto do menu . . . . .	76
5.6	Opiniões sobre a quantidade de ferramentas . . . . .	77
5.7	Opiniões sobre a janela de envio . . . . .	78
5.8	Opiniões sobre a barra de mensagens . . . . .	79
5.9	Opiniões sobre a ajuda do programa . . . . .	80

5.10	Nível de aceitação do iGraf . . . . .	81
6.1	Tela de abertura do SAW . . . . .	89
6.2	Lista de exercícios . . . . .	90
6.3	Tela que precede o início do teste . . . . .	90
6.4	Exercício 01 . . . . .	91
6.5	Exercício 02 . . . . .	91
6.6	Exercício 03 . . . . .	92
6.7	Exercício 04 . . . . .	92
6.8	Exercício 05 . . . . .	93
6.9	Exercício 06 . . . . .	93
6.10	Exercício 07 . . . . .	94
6.11	Exercício 08 . . . . .	94



# Lista de Tabelas

2.1	Levantamento histórico sobre sistemas de Álgebra Computacional relacionado ao surgimento de algumas linguagens de programação. . . . .	19
3.1	Constantes admitidas no iGraf . . . . .	31
4.1	Quantidade <i>versus</i> tipo de exercício nas publicações analisadas . . . . .	52
5.1	Níveis de satisfação propostos por Likert . . . . .	71
5.2	Cálculo dos pontos obtidos na Questão 2 . . . . .	71
6.1	Recursos apresentados por alguns sistemas visualizadores gráficos . . . . .	86
6.2	Frequência de uso do computador . . . . .	95
6.3	Pelo aspecto visual, o programa é . . . . .	95
6.4	Começar a usar o programa é . . . . .	95
6.5	Os gráficos . . . . .	96
6.6	Os menus . . . . .	96
6.7	A quantidade de funcionalidades é . . . . .	96
6.8	O envio de respostas é . . . . .	96
6.9	A barra de mensagens foi útil? . . . . .	96
6.10	A ajuda facilitou o uso do programa? . . . . .	97
6.11	Satisfação geral do usuário . . . . .	97

# Lista de Abreviaturas e Siglas

ABED	Associação Brasileira de Educação a Distância
ABRAEAD	Anuário Brasileiro Estatístico de Educação Aberta e a Distância
CAS	Computer Algebra System
CEC	Centro de Ensino de Computação
EAD	Educação a Distância
HTML	Hypertext Markup Language
IEEE	Institute of Electrical and Electronics Engineers
IME	Instituto de Matemática e Estatística
INRIA	Institut National de Recherche em Informatique et em Automatique
JCM	Java Components for Mathematics
LMS	Learning Management System
MIT	Massachussets Institute of Technology
OA	Objeto de Aprendizagem
RENOTE	Revista Novas Tecnologias na Educação
SGC	Sistema Gerenciador de Cursos
USP	Universidade de São Paulo

# Capítulo 1

## Introdução

*“There are more people in the world than ever before, and a far greater part of them want an education. The demand cannot be met simply by building more schools and training more teachers. Education must become more efficient.”*

(34)

As universidades brasileiras estão, cada vez mais rapidamente, aderindo à Educação a Distância (EAD). Essa modalidade de ensino tem características peculiares que exigem conhecimentos, pessoal e ferramentas muito específicas ao seu desenvolvimento.

Dentre as ferramentas necessárias, tem lugar de destaque os Sistemas Gerenciadores de Cursos, assim nomeados como uma tradução da expressão inglesa Learning Management System (LMS). Um sistema gerenciador é, basicamente, o software que permite aos alunos a interação com materiais instrucionais utilizados em cursos ministrados total ou parcialmente a distância e aos professores a produção, gerenciamento e distribuição de tais materiais.

### 1.1 História dos Sistemas Gerenciadores de Curso

Skinner é categórico no primeiro parágrafo de seu artigo – *Teaching Machines* – citado na abertura deste capítulo. Levando-se em consideração que o artigo foi publicado em 1958, é fácil concluir que as atuais necessidades educacionais não são diferentes das que existiam na época. Ainda que esteja disponível um aparato tecnológico muito melhor do que o que existia naquela época, também o número de pessoas que necessitam de educação formal

cresceu significativamente.

A população mundial em 1960 era de aproximadamente 3 bilhões de pessoas. Em 2000 o número de habitantes do planeta já passava de 6 bilhões. Esse ritmo de crescimento populacional vem diminuindo, porém os efeitos dessa diminuição demorarão a fazer diferença na prática. O problema de fornecer o acesso à educação para grandes massas é um dos grandes desafios modernos e precisa de soluções de curto e médio prazo.

### 1.1.1 Instrução Programada

Uma das primeiras propostas de solução para este problema foi feita entre 1953 e 1956, por Burrhus Frederic Skinner. Trata-se da Instrução Programada, que pode ser considerada como a primeira experiência de ensino com o uso do computador. Sua proposta consistia da "... utilização de máquinas de ensinar como forma de resolver os impasses que surgem em decorrência das dificuldades de atender cada aluno. O acompanhamento poderia ser feito pela própria máquina, especialmente nas formas de avaliação, entendidas por ele como parte essencial da aprendizagem." (15) Na visão dele o uso das máquinas de ensinar apresenta várias vantagens sobre outros métodos. Estudantes podem compor sua própria resposta em lugar de escolhê-la em um conjunto de alternativas. Exige-se que lembrem mais, e não apenas que reconheçam - que dêem respostas e que também vejam quais são as respostas corretas. A máquina assegura que esses passos sejam dados em uma ordem cuidadosamente prescrita. Para Skinner, o uso de tais máquinas propicia ao aluno estudar e se desenvolver no seu próprio ritmo e esse desenvolvimento ainda pode ser melhorado caso o aprendiz tenha acesso a resultados imediatos sobre seu trabalho. (34)

### 1.1.2 SAKI

Na mesma época, em 1956, Gordon Pask e Robin McKinnon-Wood desenvolveram o SAKI (*Self-Adaptive Keyboard Instructor*) (28), o primeiro sistema adaptativo de ensino a ser produzido em escala comercial. O funcionamento do SAKI consistia em modelar o comportamento do treinando dando uma nota para o nível de eficiência obtido na realização de uma tarefa. Quando o aprendiz conseguia superar essa marca, o sistema o estimulava a realizar a mesma tarefa com um nível de dificuldade maior e fornecendo menos informações. Com esse sistema era possível terminar um treinamento em dois terços ou até na

metade do tempo usual. As bases para o gerenciamento de cursos mediado por máquinas estavam então surgindo.

### 1.1.3 PLATO

Os sistemas gerenciadores de curso, doravante apenas SGC, surgiram nos anos 60. O primeiro sistema que pode ser considerado um SGC surgiu exatamente em 1960, se chamava PLATO e foi criado na *University of Illinois at Urbana-Champaign*. O projeto PLATO pode ser considerado economicamente um erro e sua tecnologia estava em muito ultrapassada quando foi oficialmente encerrado em 2006. Independente disso, o projeto foi pioneiro na criação de conceitos chave para os SGC's atuais como fóruns, testes on-line, e-mail, salas de bate-papo, mensagens instantâneas, compartilhamento remoto de aplicativos e jogos *multiplayers online*. PLATO também introduziu a diferenciação de papéis para os usuários do sistema dividindo-os em categorias como aluno, professor e administrador. Oficialmente PLATO não significa nada, mas a escolha do nome, supostamente, seria um acrônimo para **P**rogrammed **L**ogic for **A**utomated **T**eaching **O**perations.

## 1.2 Sistemas Gerenciadores de Cursos (SGC) Atuais

SGC's são programas de computador que formam a infra-estrutura de cursos *online*. Sistemas gerenciadores de curso via *Web* são projetados para fornecer serviços *online* que contribuam para a melhoria da qualidade do ensino e do aprendizado. Os SGC's, podem ajudar o professor em seu trabalho com o ensino, assim como podem ajudar os alunos em suas atividades de aprendizado, mas isso não fica garantido pelo simples fato de se usar um sistema computacional. A criação de um curso de qualidade é de responsabilidade do professor e não depende da ferramenta utilizada.

Os componentes mínimos de um SGC atual incluem a funcionalidade de comunicação (e-mail, listas de discussão, fóruns e chats), mecanismos que permitam a criação e distribuição de conteúdo (textos, materiais multimídia, programas de simulação) e ferramentas administrativas. Os SGC's podem ser comerciais, produzidos por uma empresa com o objetivo de adquirir lucro com sua venda ou não-comerciais, de acesso público, livre de custo ou obrigações e geralmente acessíveis em páginas da Internet.

Os sistemas comerciais, embora geralmente forneçam um conjunto de opções didáticas

bastante diversificado têm a inflexibilidade como característica. O usuário não tem, geralmente, como modificar o que é fornecido pelo fabricante do sistema para fazer com que os recursos se adaptem da melhor maneira possível à realidade de seu curso, classe ou instituição. Em geral, o usuário deverá se adaptar a tais sistemas. São exemplos de sistemas comerciais os SGC's ANGEL Learning, Apex Learning, Desire2Learning, Blackboard e Litmos.<sup>1</sup>

Em contraste, as ferramentas não-comerciais permitem a flexibilização do sistema de tal modo que ele possa ser adaptado à realidade de cada instituição em que for utilizado. São exemplos de sistemas não-comerciais os SGC's .LRN (leia-se: *dot learn*), ATutor, Bodington, Claroline, Dokeos, Moodle, OLAT, Sakai e VClass. É possível destacar ainda os SGC's nacionais AulaNet, SAW, Teleduc e Tidia-AE<sup>2</sup>

Os SGCs não-comerciais, geralmente, são resultados de projetos de *open source*. A teoria por trás do *open source* é simples. O código-fonte dos programas produzidos com base nesse modelo de desenvolvimento é livre. Qualquer um pode melhorá-lo, modificá-lo e explorá-lo. Mas essas melhorias, modificações e explorações têm que ser tornadas públicas e livremente disponíveis. O projeto não pertence a ninguém e pertence a todos ao mesmo tempo. Projetos desse tipo se desenvolvem rápida e continuamente devido ao fato de várias pessoas trabalharem paralelamente com o mesmo objetivo. Torvalds citado por (28)

A adoção de uma filosofia semelhante na implementação de cursos a distância pode produzir um sistema aberto e compartilhado de gerenciamento de cursos cujo design pedagógico evolui a partir das necessidades da comunidade de educadores e que, se combinado com o uso de conteúdos compartilhados, poderá fornecer a base para que a tradição acadêmica de liberdade e opção individual existente na sala de aula possa ser levada também para o ensino *online*.<sup>3</sup> (23)

---

<sup>1</sup>Veja uma lista com mais de uma centena de LMS's no *site*: <http://elearning-india.com/content/blogcategory/19/38/>

<sup>2</sup>Em desenvolvimento

<sup>3</sup>Veja os resultados de um estudo comparativo entre um sistema comercial e outro não-comercial em <http://www.humboldt.edu/jdv1/moodle/all.htm>

## 1.3 Delimitação do Problema

Independente do modelo de distribuição adotado pelos seus desenvolvedores, os SGCs só podem ser considerados úteis se puderem efetivamente auxiliar no trabalho docente e facilitar o aprendizado. Para que essas necessidades sejam satisfeitas os SGCs contam com a possibilidade de complementação de seus recursos e conteúdo através da incorporação de materiais instrucionais que satisfaçam as necessidades de um curso ou tópico específico. Esses materiais são chamados de objetos de aprendizagem.

Objeto de Aprendizagem (OA) é "todo material – digital ou não – que pode ser usado e reutilizado para aprendizagem, educação ou treinamento". (4)

Objetos de aprendizagem para o estudo de funções matemáticas geralmente utilizam gráficos. Esses gráficos precisam ser desenhados com o auxílio de algum aplicativo apropriado e então inseridos no OA. Dois problemas fáceis de serem percebidos nessa abordagem é que o gráfico inserido no OA é quase sempre estático e não pode ser manipulado (ou tem pouca possibilidade de manipulação) pelo aluno e que a cada nova atividade será necessário ao professor criar um novo OA e, mais uma vez, será necessária a busca de recursos externos (ao ambiente de construção do OA) para a criação de um novo gráfico.

Depois de pronto o objeto de aprendizagem, se houver a intenção de integração com um SGC, ele ainda terá que satisfazer aos padrões esperados pelo sistema no qual será inserido. Esta forma de proceder requer do usuário o conhecimento das características de três ambientes: o visualizador gráfico, o sistema de padronização de OA e o mecanismo de inclusão de recursos do SGC.

Durante o desenvolvimento deste trabalho buscou-se solucionar os problemas mencionados acima e, como resultado desta busca, é apresentado um sistema visualizador gráfico integrável a SGCs e que pode ser manipulado com pouco ou nenhum conhecimento técnico específico. Além disso são propostas soluções para:

- Ausência de mecanismo eficiente de avaliação automática interno ao SGC e adaptado ao estudo de função
- Recuperação de dados dos momentos de interação (sessão) do aluno com o material instrucional através do SGC
- Aumento da interatividade em páginas *Web* que usam gráficos de função

A pesquisa, que visou simplificar o processo de criação e utilização de objetos de aprendizagem para o ensino de funções matemáticas e tópicos relacionados, começou a ser desenvolvida em 2004 no Instituto de Matemática e Estatística da Universidade de São Paulo (IME-USP) e culminou com a produção do **iGraf - Gráficos Interativos na Internet**.

Sob a coordenação do professor Leônidas de Oliveira Brandão, o projeto teve o objetivo de permitir que alunos e professores pudessem utilizar um visualizador gráfico que fosse adaptado ao estudo de funções matemáticas pela *Web* com acesso livre e gratuito. Esse visualizador deveria ainda poder ser integrado a um sistema gerenciador de cursos, o SAW (Sistema de Aprendizagem pela *Web*), de forma simples e intuitiva.

Para atingir esses objetivos a implementação do programa utilizou a linguagem Java (20) “por esta permitir grande portabilidade (possibilidade de uso em diferentes computadores/sistemas operacionais) e possibilitar seu uso diretamente em páginas [da] Internet, na forma de *applet* (programa Java especialmente projetado para a *Web*)” (19)

O foco deste trabalho foi, portanto, a concepção e implementação do iGraf atribuindo ao mesmo as funcionalidades mais comuns entre os visualizadores gráficos e acrescentando-lhe algumas outras. Dentre as novas possibilidades estão o mecanismo de animação interativa e a função de edição de expressões que permitem a observação dos efeitos da variação dos parâmetros de uma função ou equação, o mecanismo de geração de *scripts* que permite ao professor verificar quais foram os passos dados por um aluno até obter um determinado resultado, o mecanismo de avaliação automática, a ferramenta interativa de visualização das retas tangentes a uma curva e a visualização de gráficos de integral calculados numericamente, que não são comuns nesse tipo de programa.

## 1.4 Justificativas

Apesar da existência de vários sistemas visualizadores de gráficos de função, são poucos os que podem ser usados diretamente em páginas da Internet. Na verdade, os sistemas que podem ser utilizados via *Web* são bastante restritos em suas funcionalidades e não são facilmente integráveis a SGCs.

Essa integração se faz um tanto necessária visto que, normalmente, quando se trata da criação ou visualização de gráficos de funções ou equações em um ambiente virtual de ensino-aprendizagem, tanto o professor quanto o aluno precisam procurar por recursos

externos ao SGC. Além disso, são raros os sistemas que podem ser usados como visualizadores gráficos e que possuem algum mecanismo de comunicação que permita a submissão de resultados de exercícios via *Web* e pelo mesmo gerenciador do curso *online* em que o aluno está inscrito.

Outro fator complicador é que a maioria dos sistemas que oferecem alguma flexibilidade no seu uso via *Web* são proprietários. É o caso dos CAS<sup>4</sup> Maple e Mathematica que, embora ofereçam algum mecanismo de comunicação e interação pela Internet, são produtos que têm alto custo de aquisição.

De modo geral, a proposta deste trabalho foi preencher as lacunas acima mencionadas, visando criar a oportunidade para que professores e alunos possam desenvolver estudos sobre - ou que possam utilizar - gráficos de funções e equações em cursos presenciais, semi-presenciais e a distância pela Internet sem qualquer custo ou obrigação.

## 1.5 Objetivos

O objetivo geral deste trabalho foi criar um programa de computador, um visualizador gráfico genérico, que pudesse ser incorporado a sistemas gerenciadores de curso para o ensino e aprendizagem de temas que podem se beneficiar do uso de gráficos de funções e equações.

O programa desenvolvido se chama iGraf e seu principal diferencial é permitir o acesso a todas as suas funcionalidades pela *Web*. Essas funcionalidades são determinadas pelo papel do usuário e existem três papéis possíveis.

**Professor:** a ele é permitido criar e distribuir material instrucional para cursos que utilizem a *Web*.

**Aluno:** pode editar e visualizar gráficos, fazer exercícios e enviá-los para a análise do professor sem a necessidade de qualquer outro recurso auxiliar.

**Administrador:** responsável pela instalação, configuração e manutenção do iGraf quando integrado a algum SGC.

---

<sup>4</sup>CAS - *Computer Algebra Systems* serão tratados na seção 2.1

## 1.6 Organização do Texto

Este texto está dividido em 6 capítulos.

O capítulo 2 faz uma introdução à importância da visualização de gráficos de funções e equações matemáticas e apresenta algumas ferramentas que atualmente são utilizadas para a visualização desses gráficos com o uso do computador dividindo-as em três categorias – CAS, visualizadores gráficos *off-line* e visualizadores gráficos *on-line* – e analisando as características principais de cada uma.

O capítulo 3 apresenta o iGraf fazendo uma breve introdução aos recursos que nele foram implementados fazendo um paralelo com as funcionalidades encontradas nos sistemas discutidos no capítulo anterior.

O capítulo 4 apresenta algumas considerações sobre o estudo de função e mostra os resultados de uma pesquisa sobre os tipos de resposta possíveis para exercícios sobre o tema. Além disso, descreve os detalhes da implementação do mecanismo de avaliação automática do programa e as possibilidades de criação e distribuição de exercícios com o uso do iGraf.

O capítulo 5 apresenta os resultados de uma análise do software dividida em duas partes: a primeira baseada em um conjunto de heurísticas para análise de projetos de interfaces gráficas e, a segunda, em um experimento realizado com professores da Rede Pública Estadual de Ensino no laboratório CEC-IME-USP.

Finalmente, o capítulo 6 apresenta as conclusões que se podem inferir a partir dos estudos realizados para a concepção desta dissertação.

## Capítulo 2

# Software gráfico para computadores

*“A principal causa do equívoco da educação atual é o baixo índice de aceitação e incorporação da tecnologia no processo educacional”*

(6)

A transmissão de informação a seres humanos é facilitada com o uso de diagramas, figuras e gráficos em geral. Em Matemática, os gráficos de funções têm papel de destaque, pois ajudam na análise visual das relações entre as grandezas que por ventura estejam sendo representadas.

Os gráficos de funções matemáticas podem ser feitos manualmente por uma pessoa ou, automaticamente, por um programa de computador. Programas que permitem a visualização de gráficos de funções são chamados de **visualizadores gráficos**.

A estratégia utilizada pelo computador para fazer o gráfico de uma função é semelhante ao processo manual. Ele marca no plano cartesiano pontos com coordenadas na forma  $(x, f(x))$  para um certo número de valores igualmente espaçados de  $x$ . Depois conecta cada ponto com seu precedente, formando assim uma representação gráfica de  $f(x)$ . (36)

Este capítulo aborda as principais características de alguns programas de computador que permitem a visualização de gráficos de funções. Devido ao grande número de programas que permitem a visualização de gráficos disponíveis atualmente no mercado, foram selecionados e citados apenas alguns que receberam destaque na comunidade acadêmica e que apresentam características relevantes ao estudo.

## 2.1 CAS (Computer Algebra System)

Sistemas de Computação Algébrica, ou simplesmente CAS, são programas de computador que facilitam o trabalho com matemática simbólica. A manipulação de expressões matemáticas na forma simbólica é a principal funcionalidade dos CAS, embora a maioria deles também possa ser usada para a visualização de gráficos de funções.

Os CAS começaram a surgir em meados dos anos 70 e têm suas origens no trabalho pioneiro do físico holandês - vencedor do prêmio Nobel de Física de 1999 - Martinus J. G. Veltman. Ele criou em 1963 um sistema para estudos de física de alta energia e matemática simbólica chamado *Schoonship*. Anos mais tarde surgiram os primeiros sistemas comerciais Reduce, Derive e Macsyma. Este último tem uma versão de uso livre chamada Maxima que discutiremos na seção 2.1.5.

Os atuais líderes do mercado de CAS são os programas Maple e Mathematica ambos comumente usados por pesquisadores em Matemática, Engenharia e diversas outras áreas do conhecimento. Existem também alguns projetos de CAS que visam tornar público o acesso a esse tipo de ferramenta. É o caso dos projetos SAGE, Yacas, Singular, Scilab e do já citado Maxima.

### 2.1.1 Maple

Maple é um sistema de álgebra computacional comercial de uso genérico. Constitui um ambiente informático para a computação de expressões algébricas, simbólicas, e que permite o desenho de gráficos a duas ou a três dimensões. O seu desenvolvimento começou em 1980 pelo Grupo de Computação Simbólica na Universidade de Waterloo, na província de Ontário, Canadá.

Em 1982 o programa começou a ser utilizado por vários grupos locais tanto na pesquisa quanto no ensino. No final de 1983 havia 50 instalações externas do software. Em 1987 foram feitas 300 instalações no mundo todo. Esses números podem parecer pequenos, mas deve-se lembrar que essas instalações eram feitas apenas em universidades; o meio acadêmico serviu para demonstrar o potencial para o licenciamento e distribuição comercial do programa.

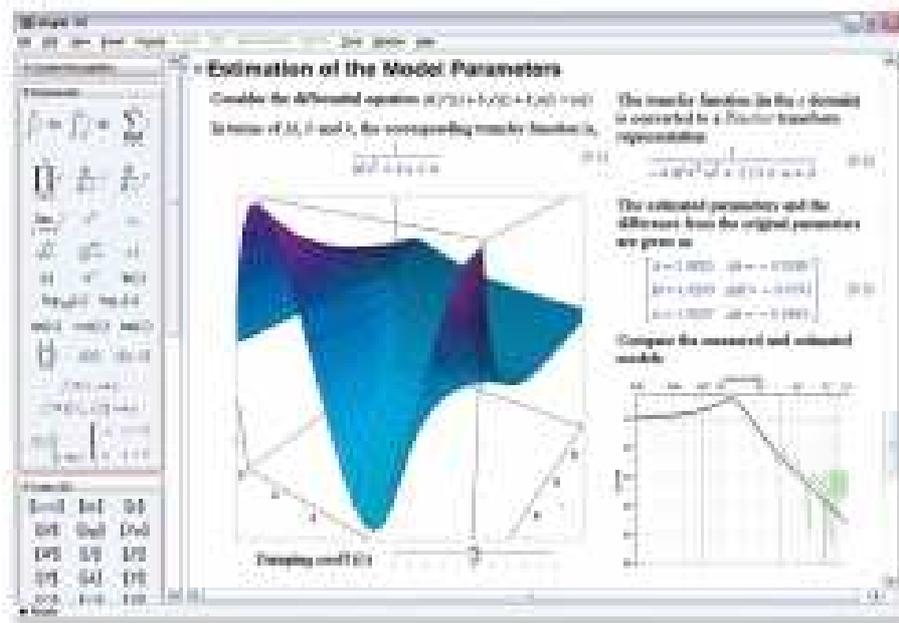


Figura 2.1: Maple 10

Desde 1988, o Maple tem sido desenvolvido e comercializado pela **Maplesoft**<sup>1</sup>, uma companhia canadense também baseada em Waterloo. A empresa comercializa o programa em duas versões: profissional e estudantil, sendo que as diferenças básicas entre as duas são o preço (maior para o pacote profissional) e a ausência de alguns recursos na versão estudantil.

A versão atual do Maple (versão 11.0) permite a publicação na *web* de documentos criados com o programa; para este fim foi criada uma interface chamada MapleNet. Os documentos criados no Maple são pequenas aplicações (chamadas de *mapplets*) voltadas para o estudo de conceitos específicos que podem ser exploradas *online* permitindo aos usuários modificar parâmetros usando componentes gráficos padrão como botões e controles deslizantes. A análise e os cálculos, no entanto, são feitos por uma cópia do Maple, que deve ser instalada no servidor que distribui as páginas da Internet com este conteúdo. (24) Tal abordagem impossibilita o acesso a outros recursos do programa já que os *mapplets* não podem ser modificados pelo usuário.

<sup>1</sup><http://www.maplesoft.com>

## 2.1.2 Mathematica

**Mathematica**<sup>2</sup> é um sistema de álgebra computacional, originalmente concebido por Stephen Wolfram. Sua primeira versão foi lançada em 1988.

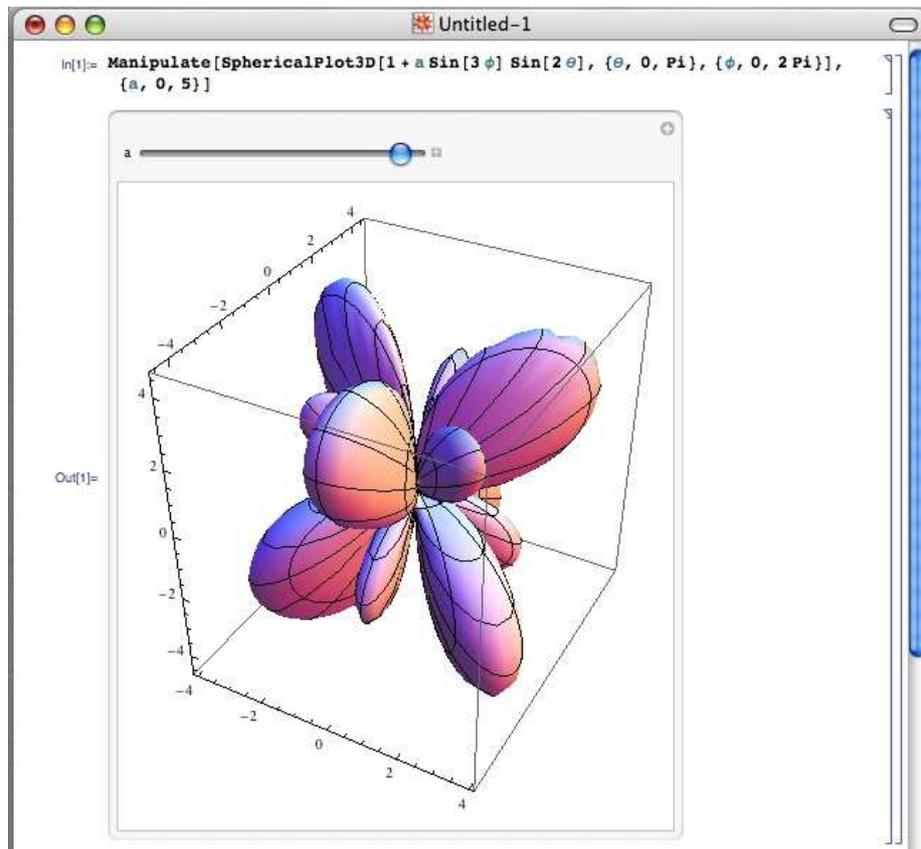


Figura 2.2: Mathematica

O programa possui rotinas para cálculos próprios de diversas áreas da engenharia e matemática, além de servir como um ambiente para desenvolvimento rápido de programas. As versões mais recentes permitem a troca de informação com programas escritos nas linguagens Java, C++, entre outras, usando bibliotecas de funções para comunicação entre aplicações possibilitando assim que um programa escrito no Mathematica, por exemplo, acesse a porta serial (onde se pode conectar *mouse*, impressora e outros periféricos) do computador onde está instalado e receba informações de equipamentos (hardware) externos.

<sup>2</sup><http://www.wolfram.com>

Assim como o seu maior concorrente, o Maple, este programa possui uma interface chamada *webMathematica*<sup>3</sup> que permite aos usuários criar páginas web com recursos interativos de cálculo e visualização de gráficos. Essa interessante forma de uso do software só é possível por meio de sua integração às mais recentes tecnologias em servidores *web* o que requer, portanto, conhecimento especializado. O programa ainda pode ser usado para a digitação de documentos com formatação matemática complexa.

### 2.1.3 SAGE

**SAGE** (*Software for Algebra and Geometry Experimentation*) é o nome de um projeto que está em desenvolvimento sob a coordenação do professor Willian Stein, do Departamento de Matemática da Universidade de Washington, com o objetivo de criar um sistema viável de álgebra computacional para pesquisa e experimentação em Álgebra, Geometria, Teoria dos Números, Criptografia e áreas relacionadas.

O projeto pretende produzir uma alternativa ao uso de software proprietário como Maple, Mathematica e outros. Ele se baseia na integração de diversas peças de software livre pré-existentes e de qualidade reconhecida como Singular, Maxima, PARI, GAP, gfan e etc, resultando em um produto final de alta qualidade e funcionalidades muito diversas, de código aberto e de uso livre. (33) O custo de tantas funcionalidades para o usuário é alto, já que a instalação do SAGE ocupará aproximadamente 2 Gb de espaço em disco e tomará algumas horas de download dependendo da velocidade da conexão utilizada. Em um sistema com velocidade média de recepção de 40 kbps foram gastas 4 horas para baixar todos os arquivos necessários à instalação.

A interface gráfica com o usuário do SAGE é atípica, pois opera através do web browser. O usuário pode criar, editar e avaliar documentos do SAGE por intermédio do navegador em modo off-line. É possível ainda - devido a presença de um servidor web interno no programa - que essa interação ocorra em modo on-line, mesmo que o usuário não possua uma instalação do programa em sua máquina. Assim, qualquer pessoa que tem o SAGE instalado em casa pode iniciá-lo no modo servidor, fazendo com que todas as funções do software possam ser acessadas pela web, sem restrições. Os distribuidores do programa recomendam o uso dos navegadores Firefox e Mozilla para permitir o pleno funcionamento do SAGE. Opera e Safari também podem funcionar, porém o Internet Explorer

---

<sup>3</sup><http://www.wolfram.com/products/webmathematica/whatis.html>

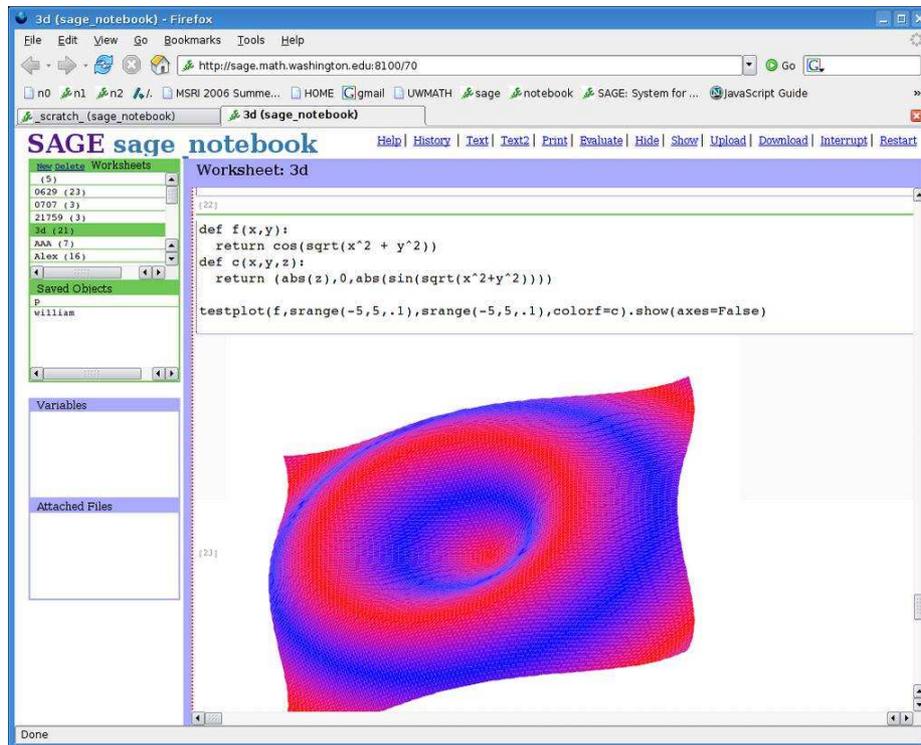


Figura 2.3: Sage

é incompatível.

A primeira versão do SAGE foi publicada em novembro de 2005<sup>4</sup>, ano do início de seu desenvolvimento e, curiosamente, nessa época SAGE significava (*Software for Arithmetic Geometry Experimentation*).

Hoje o projeto conta com a contribuição de desenvolvedores do mundo todo que se uniram por razões ideológicas. Segundo eles, duas das regras mais básicas da conduta acadêmica estão sendo violadas pelos criadores de CAS proprietário. As regras são: toda informação será transmitida sem custo e a correção dessa mesma informação poderá ser verificada por qualquer pessoa<sup>5</sup>.

<sup>4</sup>[http://modular.math.washington.edu/sage\\_old](http://modular.math.washington.edu/sage_old)

<sup>5</sup><http://modular.math.washington.edu/talks/2006-02-sage-digipen/current.pdf>

### 2.1.4 SciLab

O Scilab é um ambiente computacional aberto para resolução de problemas numéricos e criação de aplicações para engenharia e outras áreas técnicas. Foi criado e é mantido por um grupo de pesquisadores do INRIA (Institut National de Recherche em Informatique et em Automatique) e da ENPC (École Nationale des Ponts et Chaussées). (5)

O Scilab é gratuito (*free software*), de código aberto (*open source software*) e sua distribuição é feita livremente, junto com toda sua documentação, desde 1994.

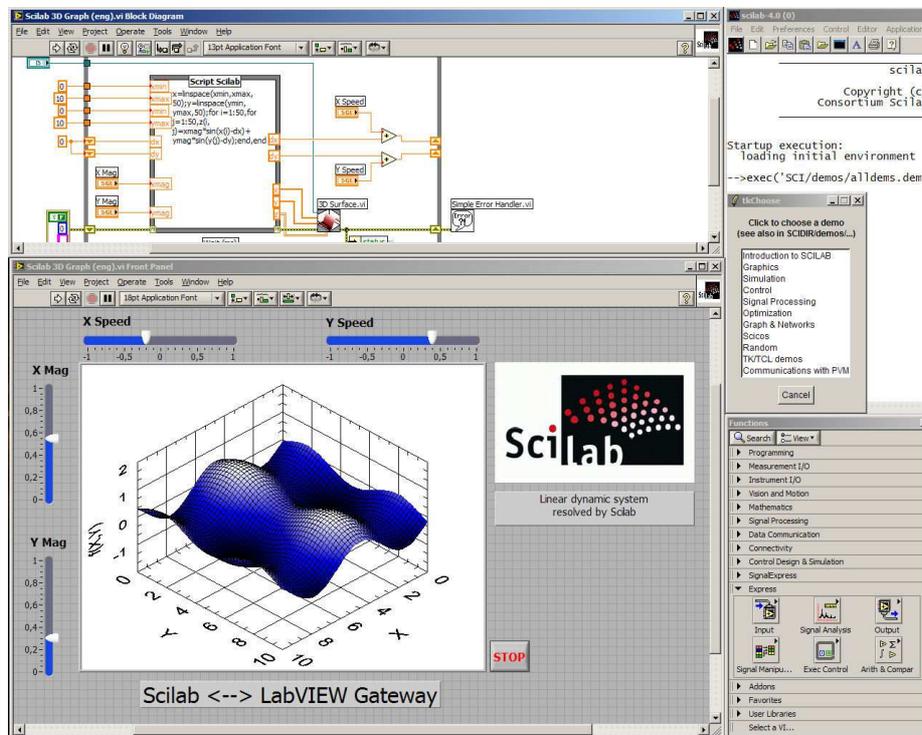


Figura 2.4: SciLab

Possui centenas de funções matemáticas pré-definidas e a possibilidade de se acrescentar ativamente funções e programas escritos em outras linguagens de programação como C, C++, Fortran e outras. Possui uma sofisticada estrutura de dados além de um interpretador e uma linguagem de programação de alto nível. (11)

Diferente dos programas já citados o Scilab não possui qualquer mecanismo que possibilite o seu uso através de páginas web.

## 2.1.5 Maxima

Maxima é um sistema de álgebra computacional de código aberto. É descendente do famoso programa Macsyma desenvolvido no MIT (Massachusetts Institute of Technology) em meados dos anos 60.

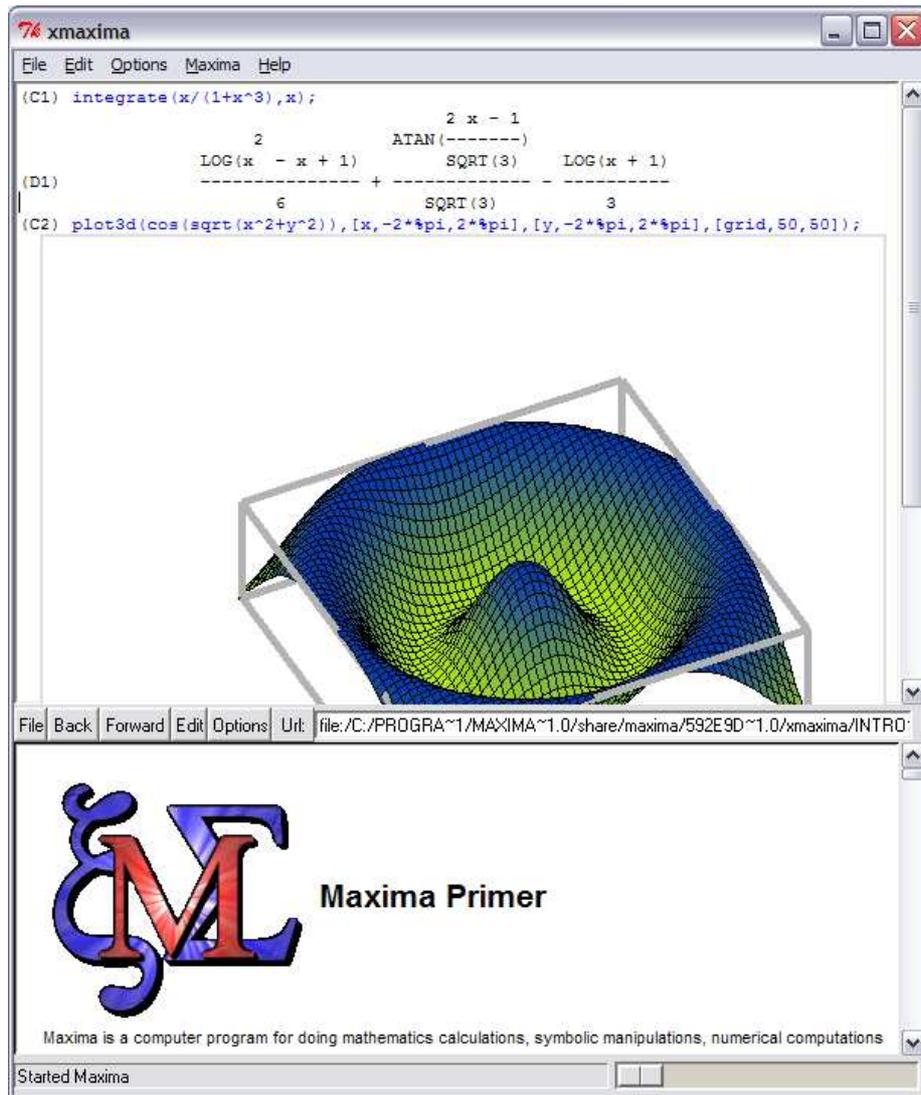


Figura 2.5: Maxima

Em 1982, a Symbolics Incorporating adquiriu licença exclusiva para comercialização do Macsyma. O MIT conseguiu na época uma licença do programa para o Departamento de Energia dos Estados Unidos criando assim uma ramificação em seu desenvolvimento

que daria origem ao Maxima. (29)

O software foi mantido pelo professor Willian Frederick Schelter da Universidade do Texas desde 1982 até 2001 quando veio o docente a falecer. Em 1998 ele obteve licença para distribuir o código fonte sob os termos da Licença Pública Geral. (22) Devido principalmente aos esforços do Dr. Schelter a sobrevivência do Maxima foi possível. (25)

O Maxima, assim como o Scilab, não possui meios de integração com a web.

### 2.1.6 Yacas

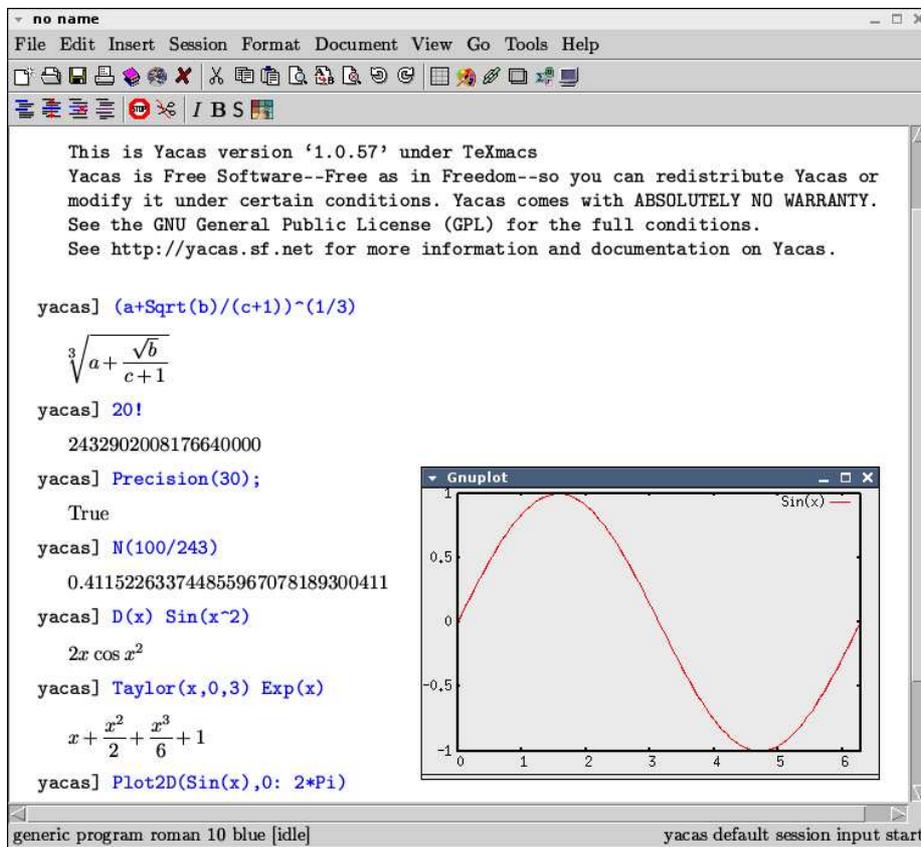


Figura 2.6: Yacas

Yacas (*Yet Another Computer Algebra System*) é um CAS que está em desenvolvimento desde 1999 sob coordenação de Ayal Z. Pinkus da Universidade de Trier na Alemanha. Conta atualmente com um grupo de dezenas de colaboradores do mundo todo. O objetivo do projeto é criar um sistema pequeno que permita o desenvolvimento e estudo de

algoritmos de matemática simbólica de forma simplificada. Uma característica interessante desse projeto é a criação de uma documentação para a descrição didática dos algoritmos implementados<sup>6</sup> no programa e dos detalhes do design<sup>7</sup> do sistema como um todo, o que permite aos mais interessados descobrirem como é construído um CAS.

O programa não possui uma interface gráfica em sua versão atual (versão 1.0.63), no entanto, pode ser experimentado on-line em uma versão Java Applet<sup>8</sup>. Possui basicamente as mesmas funcionalidades que estão implementadas em todos os sistemas citados. Seu desenvolvimento ainda se encontra no estágio inicial, embora o programa já seja plenamente funcional. (41)

### 2.1.7 Singular

Singular é um CAS para cálculos polinomiais com ênfase especial nas necessidades da álgebra comutativa, geometria algébrica e teoria da singularidade. É um software livre e sua distribuição se dá sob os termos da Licença Pública Geral. (22)

O software está sendo desenvolvido pelo Singular Team do Departamento de Matemática da Universidade de Kaiserslautern na Alemanha sob a direção de Gert-Martin Greuel, Gerhard Pfister e Hans Schönemann desde 1984. Foi vencedor do prêmio Richard D. Jenks por excelência em Engenharia de Software para Álgebra Computacional no ano de 2004. (17)

Possui uma extensa documentação on-line com detalhadas instruções sobre os algoritmos implementados e como utilizá-los, no entanto, não é possível a sua integração com a *web*. Além disso, o programa não possui uma interface gráfica embora possa ser utilizado em conexão com programas gráficos.

---

<sup>6</sup><http://yacas.sourceforge.net/Algomanual.html>

<sup>7</sup><http://yacas.sourceforge.net/NewDesignmanual.html>

<sup>8</sup><http://yacas.sourceforge.net/yacasconsole.html>

Desenvolvimento histórico dos sistemas de álgebra computacional, segundo Weinzierl.  
(39)<sup>9</sup>

The early days mainly LISP based systems	
1958	FORTRAN
1960	LISP
1965	MATHLAB
1967	SCHOOSHIP
1968	REDUCE
1970	SCRATCHPAD, evolved into AXIOM
1971	MACSYMA
1979	muMATH, evolved into DERIVE
Commercialization and migration to C	
1972	C
1981	SMP, with successor MATHEMATICA
1988	MAPLE
1992	MUPAD
Specialized systems	
1975	CAYLEY (group theory), with successor MAGMA
1985	PARI (number theory calculations)
1989	FORM (particle physics)
1992	MACAULEY (algebraic geometry)
A move to object-oriented design and open-source	
1984	C++
1984	SINGULAR
1995	Java
1999	GiNaC
2001	SCILAB
2005	SAGE
2007	YACAS

Tabela 2.1: Levantamento histórico sobre sistemas de Álgebra Computacional relacionado ao surgimento de algumas linguagens de programação.

<sup>9</sup>Os programas Singular, Scilab, SAGE e Yacas foram acrescentados pelo autor deste trabalho.

## 2.2 Visualizadores gráficos *off-line*

Existe uma categoria de programas cujo foco é a exibição de gráficos de funções e equações matemáticas. Esses programas são chamados de Visualizadores Gráficos (VG) e, geralmente, têm bem menos funcionalidades que os CAS.

A maioria dos VG's basicamente desenha gráficos bidimensionais e faz cálculos simples como a determinação de raízes de equações e a determinação de pontos de máximo e pontos de mínimo de uma função. Eventualmente são implementadas outras funcionalidades simples como cálculo de retas de regressão, gráficos de derivadas, traçado de retas tangentes e integração definida. Há VG's que fazem gráficos tridimensionais. Alguns desses programas contam com algum tratamento simbólico como é o caso dos dois que são analisados abaixo.

### 2.2.1 Graphmatica

**Graphmatica** é um visualizador gráfico desenvolvido em 1997 por Keith Hertzner, um programador graduado na Universidade de Berkeley na Califórnia. Existe uma versão do programa em português cuja tradução é de Carlos Malaca.

O programa faz gráficos de funções no sistema de coordenadas cartesianas e também em coordenadas polares. Desenha curvas definidas com equações paramétricas e gráficos de soluções de equações diferenciais ordinárias além de colorir regiões do plano cartesiano definidas por equações e inequações. Realiza diferenciação simbólica para a maioria das funções comuns e exibe o resultado na forma de gráfico (desenha na tela) e texto (escreve a função derivada na barra de *status*). (18)

Permite o estudo de famílias de gráficos por meio da inserção de um parâmetro variável  $a$  nas expressões desses gráficos. O uso desse recurso é bastante intuitivo. Basta que o usuário insira o parâmetro  $a$  em qualquer lugar da expressão onde poderia colocar um número. Ao detectar a existência desse parâmetro, o programa atribui-lhe um conjunto de valores que, por padrão, tem limite inferior igual a um, limite superior igual a três e passo de crescimento de uma unidade. O sistema permite a configuração manual desses valores.

Utilizando os valores padrão do parâmetro  $a$ , o gráfico de  $y = x + a$  será exibido como na figura 2.7.

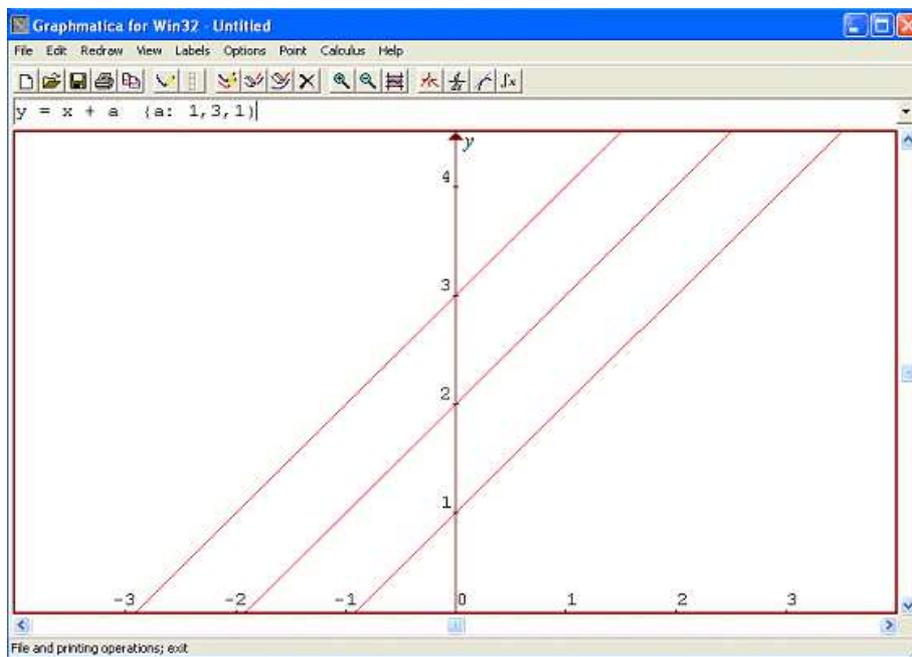


Figura 2.7: Parametrização no Graphmatica

O programa Graphmatica é distribuído como *shareware*<sup>10</sup> e a versão atual é a 2.0; não possui mecanismos que permitam a integração com a web.

### 2.2.2 Winplot

**Winplot** é um utilitário gráfico de propósito geral que pode desenhar curvas e superfícies determinadas por expressões matemáticas nas formas implícita, explícita, polar e paramétrica.

Desenvolvido e mantido pelo professor Richard L. Parris do Departamento de Matemática da Academia Phillips Exeter de New Hampshire, a sua primeira versão foi publicada em 2001. Atualizações são feitas periodicamente e a versão mais recente foi publicada em 06 de fevereiro de 2008. (27) Existe uma versão traduzida para o português pelo professor Adelmo Ribeiro de Jesus das Faculdades Jorge Amado, Bahia.

Permite o estudo de famílias de gráficos com o uso de animações pela atribuição de

<sup>10</sup>Shareware: programa distribuído de forma livre (sem o pagamento de licença de uso), em geral através de serviços on-line, Internet e revistas. Ao usar o shareware regularmente, solicita-se que se registre e pague uma taxa pela qual recebe-se completo acesso à versão comercial. (3)

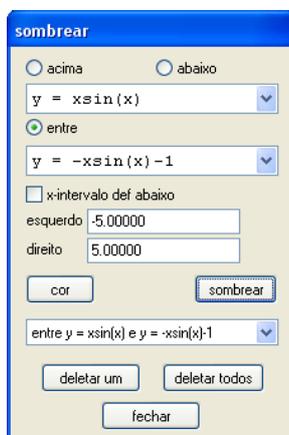


Figura 2.8: Painel de parâmetros

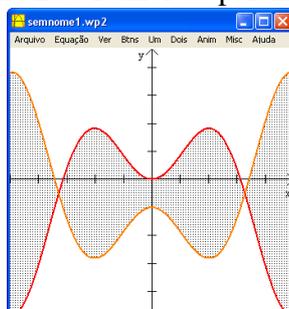


Figura 2.9: Saída gráfica

intervalos de valores para 26 parâmetros nomeados de A a W. É possível atribuir intervalos de valores para vários parâmetros e fazê-los variar simultaneamente.

Esse recurso não está presente no Graphmatica pois, na versão atual, apenas um parâmetro é suportado. No entanto, a utilização dessa funcionalidade no Winplot não é tão intuitiva.

Winplot também faz diferenciação simbólica e integração definida. Além disso, oferece a possibilidade de traçar o gráfico da integral indefinida calculado numericamente.

Embora o Winplot seja um software matemático bem mais completo que o Graphmatica, ele também não oferece a possibilidade de integração com a web.

## 2.3 Visualizadores gráficos *on-line*

A Internet oferece vários recursos que possibilitam o estudo de diversas áreas do saber. Dentre esses recursos podem ser encontrados alguns VG's que permitem ao usuário a ex-

ploração on-line. Foram selecionados, dentre os vários existentes, alguns que podem ser acessados sem custo ao usuário.

### 2.3.1 Wessa's Equation Plotter

**Wessa.net** é um sítio da Internet que conta com uma coleção de softwares para uso não comercial on-line e está na versão 1.1.21r4. Possui módulo para desenho de gráficos de funções, cálculos de regressão múltipla, cálculos financeiros e de probabilidades. O nome Wessa é uma abreviação para (*Web-Enabled Scientific Services and Applications*), embora também seja o sobrenome do criador do sítio, o professor belga Patrick Wessa.

O sistema permite o traçado de gráficos de funções explícitas bem como a visualização do gráfico da função derivada. Calcula integral definida numericamente e destaca a região de integração.

Permite o armazenamento e recuperação dos resultados das explorações dos usuários diretamente no servidor do sítio, sem qualquer custo. Tem uma interface bastante flexível e oferece ao usuário a possibilidade de salvar uma cópia de seus dados em formato de planilha eletrônica do MS Excel ou arquivo de texto do MS Word.

Possui uma ferramenta que permite a publicação dos resultados do usuário em um blog. Esses resultados são rotulados, catalogados, armazenados e estão permanentemente disponíveis para consulta. Visitantes do blog podem discutir, reproduzir e reutilizar os cálculos constantes neste arquivo, no entanto, o conteúdo só pode ser editado pelo seu criador.. (40)

Não possui, no entanto, mecanismos criados especificamente para permitir que professores organizem e gerenciem cursos pela web.

### 2.3.2 WebGraphing

O **WebGraphing** é um sítio da Internet no qual se pode encontrar várias ferramentas matemáticas que permitem, por exemplo, visualizar gráficos de funções, regiões do plano cartesiano definidas por equações ou por inequações e funções definidas por partes. É possível ainda visualizar gráficos tridimensionais, fazer divisões e multiplicações de polinômios, resolver sistemas de equações, participar de fóruns e ler textos sobre os conceitos matemáti-

cos abordados no sítio, mas nem todas essas funcionalidades são acessíveis a todos os usuários.

O “internauta” pode acessar o conteúdo do sítio como visitante ou, se preferir, pode se registrar no sistema. Visitantes têm acesso apenas ao desenho dos gráficos mais simples. Usuários registrados podem ser de dois tipos: membro básico e membro A+. Membros básicos não pagam pelo uso do sistema e, embora tenham mais opções que os visitantes sem registro, seu acesso ainda é restrito. Membros A+ têm acesso irrestrito, porém têm que pagar por isso. Eles escolhem um dentre vários planos de pagamento cujos valores são determinados pelo tempo que o usuário deseja utilizar todas as funcionalidades do sítio. Esse tempo pode variar de dois dias a um ano. (38)

É possível que um professor pague e registre todos os seus alunos para o uso de todos os recursos do sítio, no entanto o WebGraphing não conta com mecanismos que permitem ao professor monitorar a interação dos alunos com os materiais e recursos disponíveis.

### 2.3.3 Cornell Equation Plotter (relplot)

O **relplot** é um programa escrito por Andrew C. Mayers, professor do departamento de Ciência da Computação da Universidade de Cornell em Ithaca, Nova Iorque. Esse programa faz gráficos de funções no sistema de coordenadas cartesianas e em coordenadas polares, além de colorir regiões do plano definidas por equações e inequações. Permite a digitação de expressões matemáticas bastante complexas e está disponível para uso *online*. Diferente dos demais VG's, esse software não possui uma área de desenho para a exibição dos gráficos. Para cada gráfico ele gera um arquivo no formato PDF (*Portable Data Format*) ou PS (*Post Script*) e o exibe diretamente na janela do navegador. O usuário então pode optar por fazer o *download* do arquivo. O sítio é extremamente simples e possui apenas campos de texto onde se pode digitar a expressão matemática para a qual se pretende visualizar o gráfico e os limites da região retangular em que o gráfico será exibido. Apesar dessa simplicidade no visual da *web page* o programa relplot é bastante robusto e faz o desenho de vários gráficos que nenhum dos VG's estudados consegue fazer em suas implementações atuais.

### 2.3.4 QuickMath

O QuickMath pode ser usado como visualizador gráfico, porém seria mais correto considerá-lo como um serviço de respostas para problemas comuns de Matemática pela Internet. Podemos pensar sobre esse serviço como uma calculadora on-line que pode resolver questões básicas ligadas ao Cálculo Diferencial e Integral e à Álgebra, além de fazer cálculos com matrizes, gráficos de funções, e etc.

O sistema funciona por meio da submissão de questões a um servidor *web* onde ocorre o processamento, que é feito pelo CAS Mathematica. A integração da *web page* ao CAS utiliza a interface WebMathematica. (31)

Assim como os outros VG's analisados, o QuickMath não possui qualquer mecanismo que possibilite o seu uso integrado a cursos via *web*.

## 2.4 Arcabouços (*Frameworks*)

Existem algumas iniciativas de desenvolvimento de programas como, por exemplo, jScience, JCM, JelSim e Meditor que visam a criação e distribuição de bibliotecas (pacotes de procedimentos computacionais) que contêm implementações de funções matemáticas e científicas comuns a diversas áreas de estudo. Esse tipo de projeto é geralmente chamado de **arcabouço** – ou *framework* –, que segundo o dicionário *online* Priberam (30), onde também se pode encontrar a variante *arcoiboço*, é sinônimo de “armação interna” e também de “esqueleto de uma construção”. Assim, esses projetos definem estruturas básicas que podem ser usadas para o desenvolvimento de software matemático.

### 2.4.1 Java Components for Math Project (JCM)

Em agosto de 2001 o professor David J. Eck da Hobart and Willian Smith Colleges iniciou o projeto Java Components for Math (JCM) objetivando desenvolver um *framework* de componentes configuráveis de *software* escrito na linguagem de programação Java. A escolha dessa linguagem ocorreu devido ao fato de que os componentes do projeto JCM foram imaginados para serem usados como parte de material instrucional publicado em páginas *web*. (14)

Foram criados treze Java *applets* que servem como exemplo de utilização e que podem ser adicionados às páginas *web* de qualquer professor ou curso que possa se beneficiar do seu uso. Estes *applets* procuram demonstrar a flexibilidade dos componentes criados pelo projeto JCM.

Todo o material produzido para o projeto pode ser utilizado e distribuído sem qualquer restrição, inclusive com finalidade comercial. Para que isso seja possível é disponibilizada a documentação explicativa sobre cada componente e os respectivos códigos-fonte.

## 2.4.2 Java Tools and Libraries for the Advancement of Sciences (jScience)

O projeto jScience foi criado pelo Engenheiro Elétrico Jean-Marrie Dautelle em janeiro de 2005 e tem por objetivo fornecer à comunidade científica uma biblioteca escrita em linguagem Java que integre conhecimentos de diversas áreas como Matemática, Física, Biologia, Geografia, Economia, Química e etc.

É um projeto de código aberto (*open-source project*) e, por este motivo, conta com a participação de desenvolvedores do mundo todo, que colaboram para o seu desenvolvimento. Os módulos do jScience são de uso livre e irrestrito desde que sejam mantidos os créditos aos seus autores.

A versão atual, lançada em julho de 2007, conta com pacotes para o desenvolvimento de aplicações para Geografia, pacotes que modelam estruturas matemáticas avançadas como anéis, grupos, campos e espaços vetoriais, um pacote específico para desenvolvimento de aplicações que usam Álgebra Linear e um pacote voltado para aplicações que usam matemática simbólica. Além disso, no campo da Física, existem pacotes para tratamento de fenômenos relativísticos, física de alta energia, e física quântica. Há ainda um pacote para cálculos financeiros.

Esperava-se que ainda em 2007 fossem acrescentados ao projeto pacotes para o desenvolvimento de aplicações ligadas à Física Nuclear, algoritmos genéticos, redes neurais, transformadas rápidas de Fourier (FFT) e outros, no entanto, no final de fevereiro de 2008 quando foi feita a última consulta, os pacotes citados ainda estavam em desenvolvimento.

(7)

### 2.4.3 Meditor

O objetivo desse projeto é criar uma biblioteca de procedimentos de computação simbólica escrita em linguagem Java e um editor matemático que sirva como interface gráfica para essa biblioteca. A escolha da linguagem nesse caso se deu pelo fato da mesma possuir duas características fundamentais para o estudo: legibilidade e portabilidade.

A idéia básica é produzir código bem documentado e legível que permita a fácil compreensão por aquele que se prontificar a estudá-lo.

A versão atual do **Meditor** já conta com resolução de sistemas de equações, operações com vetores e matrizes, fatoração, diferenciação e integração, simplificação, Álgebra Geométrica e operações da Álgebra booleana. É possível ainda a exportação do conteúdo das folhas de trabalho no formato MathML e a geração automática de código Java. (26)

### 2.4.4 Java e-Learning Simulations (JeLSIM)

O projeto **JeLSIM** foi criado em julho de 2002 por um grupo de professores da Heriot-Watt University do Reino Unido. Segundo seus autores, o projeto visa à melhoria da qualidade da educação e treinamento on-line.

Dentre os projetos analisados é o que produziu a estrutura mais genérica e, por consequência, aquela de mais difícil aplicação prática. O resultado dessa iniciativa foi um conjunto de ferramentas para a produção e distribuição de simulações educacionais.

JeLSIM Builder é o nome desse *kit* de ferramentas. As ferramentas são escritas em linguagem Java e permitem ao usuário criar *applets* que podem ser distribuídos por meio de um navegador *web* padrão. A aplicação está dividida em três partes: *Model Wizard*, *Interface Builder* e *Deployment Manager*.

A criação das simulações com o uso desse *kit* requer a participação de um programador que possua conhecimentos da linguagem Java para fazer a modelagem computacional do sistema a ser simulado. O *Model Wizard* orienta o programador na criação do modelo computacional. Uma vez que esse modelo estiver pronto, a interface gráfica da simulação pode ser desenvolvida por professores e desenvolvedores de conteúdo que não tenham qualquer conhecimento de programação.

A criação da interface se dá por meio do uso de um ambiente onde se pode clicar e

arrastar elementos gráficos, o *Interface Builder*. Essa ferramenta permite inserir botões, campos de texto, áreas de texto, caixas de seleção e outros elementos na janela de exibição da aplicação. (1)

O JeLSIM Builder oferece suporte para os padrões SCORM e IMS. Uma vez criada a aplicação, é possível exportá-la como um SCO (*Shareable Content Object*), ou seja, um objeto de aprendizagem que obedece ao padrão SCORM. Essa exportação é feita através do *Deployment Manager*. O processo é bastante simples. Basicamente, o usuário escolhe *Scorm - sco* no menu *save to web* e será gerado todo o código (HTML e JavaScript) necessário para permitir que a página HTML contendo o applet se comunique com algum LMS (Learning Management System) compatível com o padrão SCORM.

## Capítulo 3

# O Programa iGraf

*“Através da análise dos gráficos, construídos por meio de alguns poucos comandos digitados no microcomputador, encontramos a possibilidade de tornar importantes e necessários os cálculos algébricos que, por si só, seriam, muitas vezes, maçantes e desprovidos de significado.”*

(3)

O iGraf, um programa visualizador gráfico para ensino de Funções pela *Web*, permite ao seu usuário desenhar com boa precisão gráficos de uma grande variedade de funções matemáticas, bem como visualizar animações sobre esses gráficos.

O início do desenvolvimento do iGraf ocorreu em meados de agosto de 2003 no Instituto de Matemática e Estatística da Universidade de São Paulo (IME/USP) coordenado pelo professor Dr. Leônidas de Oliveira Brandão tendo como desenvolvedor o autor desta dissertação. O objetivo principal era suprir a falta de programa similar que pudesse ser utilizado via *web* como ferramenta de apoio em cursos ministrados a distância pela Internet. Durante esta pesquisa não foram identificadas ferramentas semelhantes que se prestassem ao uso via *web* ou à integração a Sistemas Gerenciadores de Cursos(SGC).

### 3.1 Tipos de Usuário

A integração do iGraf a um SGC prevê a existência de três tipos de usuário: o aluno, o professor e o administrador do sistema.

Ao usuário aluno é possível a manipulação *on-line* do iGraf como ferramenta de apoio

ao estudo de temas que utilizem gráficos de função ou à realização de tarefas disponibilizadas pelo professor, bem como o envio de suas respostas, através de um SGC. Assim, ele não tem a necessidade de buscar ferramentas auxiliares para a realização de atividades que dependem – ou que podem se beneficiar – do uso de um visualizador gráfico.

O sistema permite ao usuário professor criar e disponibilizar aos seus alunos material instrucional relacionado ao estudo de funções e equações usando o mesmo SGC no qual se desenvolve seu curso. Ao professor ainda é possível acompanhar o desenvolvimento das atividades discentes visualizando os resultados de avaliações realizadas automaticamente pelo iGraf toda vez que um aluno envia os resultados de suas tarefas.

O programa foi projetado para que o administrador do sistema tenha um papel bastante reduzido, ou seja, a instalação e configuração do ambiente (servidor) no qual o iGraf será executado deve ser simples a ponto de poder ser feita por pessoas com apenas alguns conhecimentos básicos de computação. Eventualmente, o administrador do sistema será o próprio professor. As instruções para instalação e configuração do ambiente de execução do programa podem ser encontradas no *site* [www.matematica.br/igraf](http://www.matematica.br/igraf)

O gerenciador de cursos ao qual o iGraf se adapta é o SAW – Sistema de Aprendizagem pela Web. (9) O SAW é um SGC que permite a inclusão de Módulos de Aprendizagem (MA) na forma de *applets*<sup>1</sup> Java que obedecem o protocolo de inclusão desse gerenciador. O iGraf é um *applet* Java que obedece esse protocolo e, portanto, é um MA incorporável ao sistema.

## 3.2 Tipos de Gráfico

A versão atual do programa pode desenhar polígonos e gráficos de função. O iGraf reconhece o tipo do gráfico a ser desenhado pela análise da expressão matemática digitada pelo usuário. Estas expressões estão descritas abaixo.

As expressões que definem gráficos de função utilizam os operadores aritméticos (+) para adição, (-) para subtração, (\*) para multiplicação, (/) para divisão e (^) para potenciação e também podem incluir funções trigonométricas **sin** para seno, **cos** para cosseno, **tan** para tangente e as respectivas funções inversas **arcsin**, **arccos** e **arctan**. Além dessas, o iGraf ainda aceita as funções **sqrt** para raiz quadrada, **exp** para exponencial, **ln** para loga-

---

<sup>1</sup>programa escrito em linguagem Java que pode ser executado por navegadores *Web*

ritmo natural e **abs** para módulo ou valor absoluto de um número. Todas as funções devem ser seguidas por um par de parênteses envolvendo seus argumentos.

Uma expressão matemática válida para o desenho de gráficos de função ainda pode conter **parâmetros**. No iGraf, parâmetros são letras minúsculas do alfabeto latino às quais se associam valores numéricos reais. O iGraf conta com dois tipos de parâmetros: os **parâmetros constantes** – que são as letras '**d**', '**e**' e '**p**' – descritos na tabela 3.2 e os **parâmetros variáveis** que são usados em animações - descritos na seção 3.6 . Parâmetros podem ser usados em expressões que descrevem gráficos de função em qualquer lugar onde se poderia usar um número.

Constante	Valor	Uso
d	$\pi / 180$	conversão grau-radiano
e	2.7183	número de Euler
p	3.1415	valor aproximado de $\pi$

Tabela 3.1: Constantes admitidas no iGraf

O outro tipo de expressão aceita é usado para descrever polígonos. Um polígono, no iGraf, é definido por uma coleção de vértices representados na forma de par ordenado de números reais (**x**, **y**) colocados entre colchetes. Assim, um triângulo pode ser descrito pela expressão [(x1, y1)(x2, y2)(x3, y3)]. Pontos e segmentos de reta são tratados pelo programa como casos especiais de polígonos que possuem um e dois vértices respectivamente.

No iGraf a quantidade de gráficos é virtualmente ilimitada, embora o desenho de um grande número de gráficos possa tornar difícil a visualização e a análise do resultado. O sistema de desenho dos gráficos, por padrão, utiliza oito cores. Assim, cada gráfico desenhado terá uma cor diferente até o oitavo desenho; a partir do nono desenho as cores começam a se repetir. O usuário pode alterar a cor de um gráfico usando a ferramenta de edição cujas funcionalidades estão descritas na seção 3.4.1.

### 3.3 A Interface Gráfica

O iGraf é uma ferramenta de apoio para professores e alunos criada para ser usada durante o ensino e aprendizagem de tópicos relacionados a funções, conseqüentemente seu uso ocorrerá em momentos muito específicos de um curso. É necessário, portanto, que o tempo

gasto para aprender a usar o programa seja o menor possível. Por esse motivo foi criada uma interface gráfica simples, para que o usuário possa rapidamente se acostumar com os elementos gráficos e que o faça de maneira consistente para que, depois de algum tempo sem usar o programa, o aluno (ou o professor) consiga retomar o uso rapidamente.

Assim, a interface gráfica do programa, que pode ser vista na figura 3.1, está dividida em apenas cinco partes deliberadamente simples: na região superior a **barra de título**, a **área de edição** e o **menu de botões**, na região central a **área de desenho** e na região inferior a **barra de mensagens**.

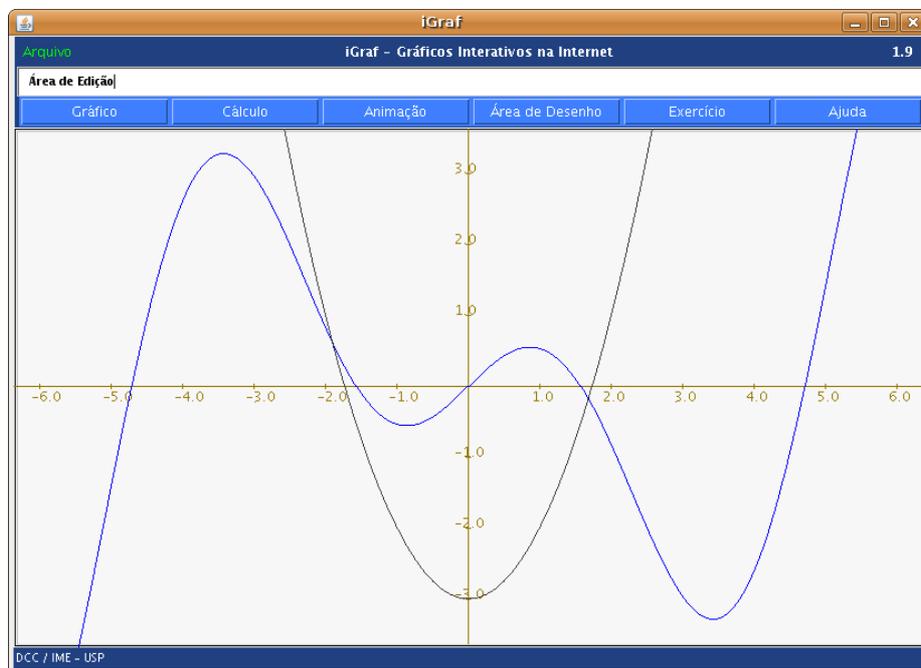


Figura 3.1: Tela inicial do iGraf

### 3.3.1 Barra de Título

A barra de título apresenta o nome do programa, sua versão e um menu (**Arquivo**) para acesso às funções de gravação e leitura de informações no computador. Esse menu possui uma funcionalidade chamada **Gerar Página HTML** que permite ao professor salvar no disco do computador o seu trabalho na forma de uma página própria para a exibição pela Internet com apenas um clique, sem usar qualquer conhecimento de programação. O menu

Arquivo só está disponível na versão aplicativo, pois a linguagem Java não permite que *applets* façam esse tipo de operação.

### 3.3.2 Área de Edição

Logo abaixo da barra de título encontra-se o campo de edição de expressões matemáticas. Nesse campo, o usuário digita a expressão para a qual deseja ver o gráfico correspondente, que pode ser um polígono ou uma função. A escolha do tipo de gráfico pelo usuário é feita no momento em que digita a expressão, ou seja, a sintaxe determina a forma; caso seja digitada uma expressão válida, nenhuma outra intervenção do usuário será necessária<sup>2</sup>.

### 3.3.3 Menu de Botões

O acesso às rotinas do programa é feito através do menu de botões que ao serem clicados, exibem uma lista na qual o usuário pode visualizar um grupo de opções disponíveis no programa. Cada uma das listas será descrita, em detalhes, a partir da seção 3.4.

### 3.3.4 Área de Desenho

No centro da janela de exibição do iGraf encontra-se uma representação do plano cartesiano, aqui nomeada de área de desenho, é nesta área que são exibidos os gráficos e animações produzidas pelo programa.

Por padrão, a área de desenho exibe a região retangular cujos vértices são  $[(-6,-3)(-6,3)(6,-3)(6,3)]$ . É possível, no entanto, selecionar e visualizar diferentes regiões do plano de três maneiras: utilizando-se as opções de *zooming* que estão no menu **Área de Desenho** (seção 3.7), clicando e “arrastando” a própria área de desenho ou usando as setas direcionais do teclado. Para movimentar a área de desenho com as teclas direcionais, é necessário que, antes, a mesma esteja em “foco”; um objeto na tela do computador obtém foco, geralmente, ao ser “clicado”. Portanto, o usuário deverá dar um clique na área de desenho antes de movimentá-la usando as teclas direcionais. Naturalmente, também é possível a visualização de uma região específica usando uma combinação dessas ações para fazer o ajuste dos limites visíveis de uma região retangular do plano.

---

<sup>2</sup>Veja a seção 3.2 para detalhes sobre a sintaxe das expressões possíveis

A área de desenho possui ainda um mecanismo de “movimento automático do plano” que é acionado pressionando-se a tecla **<alt>** combinada com uma das teclas direcionais. Um clique na área de desenho interrompe o movimento automático; pressionar a tecla **home** retorna o plano às configurações iniciais (origem no centro da tela).

### 3.3.5 Barra de Mensagens

Na parte inferior da janela encontra-se uma barra de mensagens cujo objetivo é informar ao usuário – com pequenos textos explicativos – qual é a função de cada item dos menus. Essa barra pode exibir ainda as expressões que descrevem os gráficos que estão na área de desenho, bastando para isso posicionar o cursor sobre a curva.

## 3.4 Menu Gráfico

O objetivo principal do iGraf é desenhar gráficos de função. Para isso, basta que o usuário digite uma expressão válida na área de edição, clique em **Gráfico** no menu de botões e selecione a opção **Desenhar**. É possível realizar a mesma ação apenas pressionando a tecla **<enter>** logo após o término da digitação da expressão matemática.

### 3.4.1 Edição de Expressões

O iGraf conta com uma ferramenta que permite a manipulação das expressões dos gráficos que estão na área de desenho. Os gráficos podem ser editados no modo *wysiwyg* (*what you see is what you get*), assim o usuário ao alterar os coeficientes das expressões pode ver imediatamente na tela a influência dessa alteração no formato do gráfico. A edição *on the fly* pode ser feita para gráficos de função, polígonos e animações. Durante a edição o usuário ainda pode escolher a cor do gráfico em uma lista ou mesmo compor uma cor personalizada atribuindo valores inteiros para as componentes RGB dessa cor, usando controles deslizantes ou digitando-os diretamente nos campos numéricos da janela de seleção de cores.

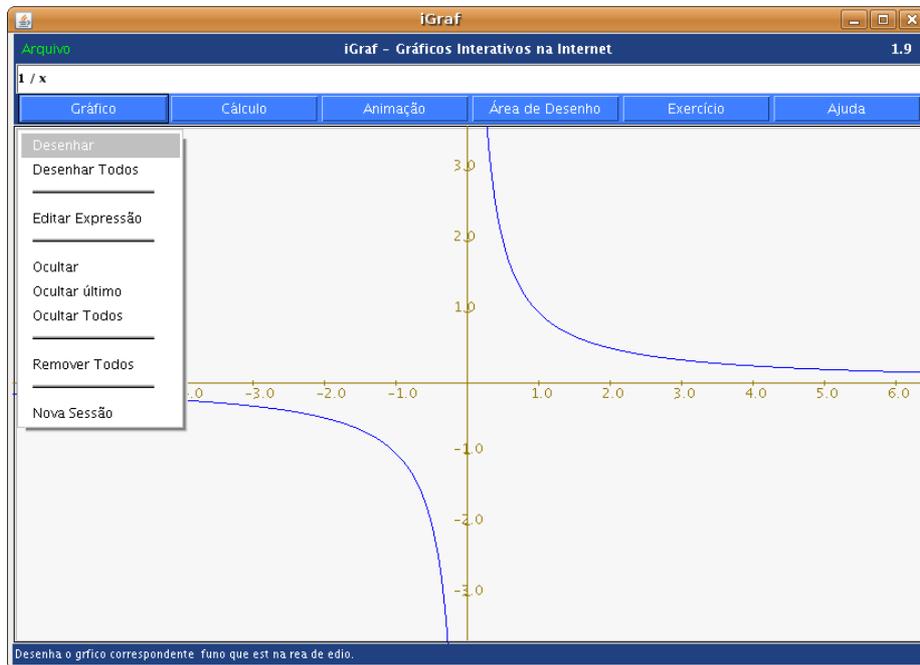


Figura 3.2: Menu de operações com gráficos

## 3.5 Menu Cálculo

Além de fazer gráficos com facilidade, rapidez e precisão razoável (3), visualizadores gráficos geralmente possuem ferramentas que apóiam o estudo do Cálculo Diferencial e Integral (CDI). Essas funcionalidades no iGraf podem ser acessadas clicando-se no botão **Cálculo** do menu de botões.

### 3.5.1 Visualizar Derivada

Para visualizar o gráfico da derivada  $f'(x)$ , basta digitar a função  $f(x)$  no campo de edição e clicar na opção **Visualizar Derivada**. Como resultado desta ação, além do gráfico na área de desenho será exibida a expressão algébrica de  $f'(x)$  no campo de edição. Não há sequer a necessidade do gráfico de  $f(x)$  ser previamente desenhado.

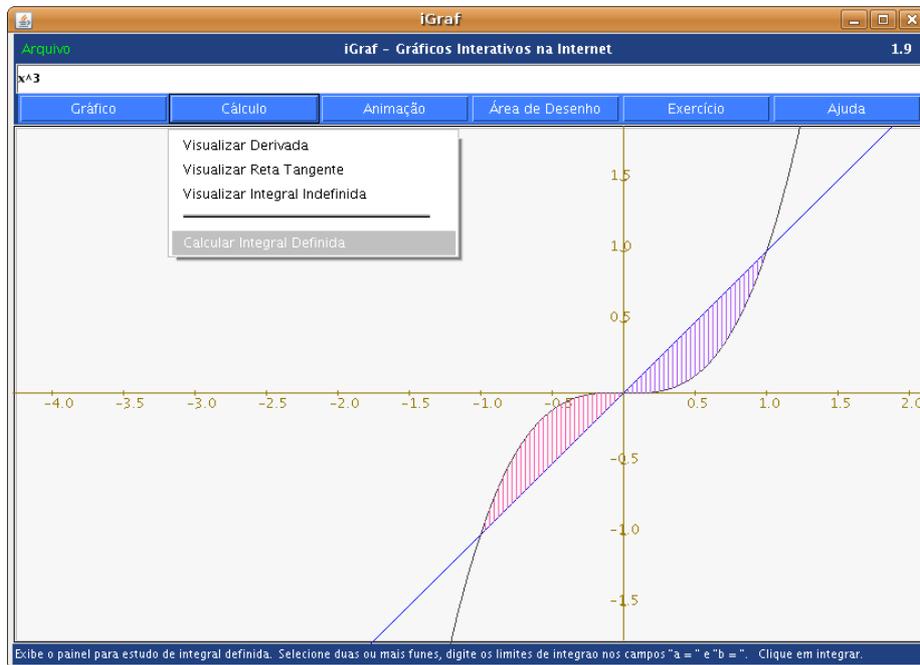


Figura 3.3: Ferramentas do Cálculo Diferencial e Integral

### 3.5.2 Visualizar Reta Tangente

Outra possibilidade é a análise das retas tangentes a uma curva. Um clique na opção **Visualizar Reta Tangente** exibe uma janela na qual é possível selecionar uma entre as funções recentemente editadas, atribuir um valor para  $x$  e visualizar, a reta tangente a  $f(x)$  no ponto  $x$  e sua equação reduzida. O valor de  $x$  pode ser determinado por digitação ou uso das teclas direcionais do teclado do computador. As teclas para cima e para a direita aumentam o valor de  $x$  e as teclas para baixo e para a esquerda diminuem o valor de  $x$ . Os incrementos e decrementos em  $x$  ocorrem em passos de 0,02.

O valor de  $f(x)$  também é calculado e exibido na mesma janela. Essa é uma das funcionalidades para a qual o autor deste trabalho durante sua pesquisa não detectou implementações similares.

### 3.5.3 Visualizar Integral Indefinida

A versão atual do iGraf não calcula analiticamente a primitiva  $F(x)$  dada por  $\int f(x)dx + C$ . No entanto, é possível a plotagem do gráfico de  $F(x)$  pelo do uso de procedimentos

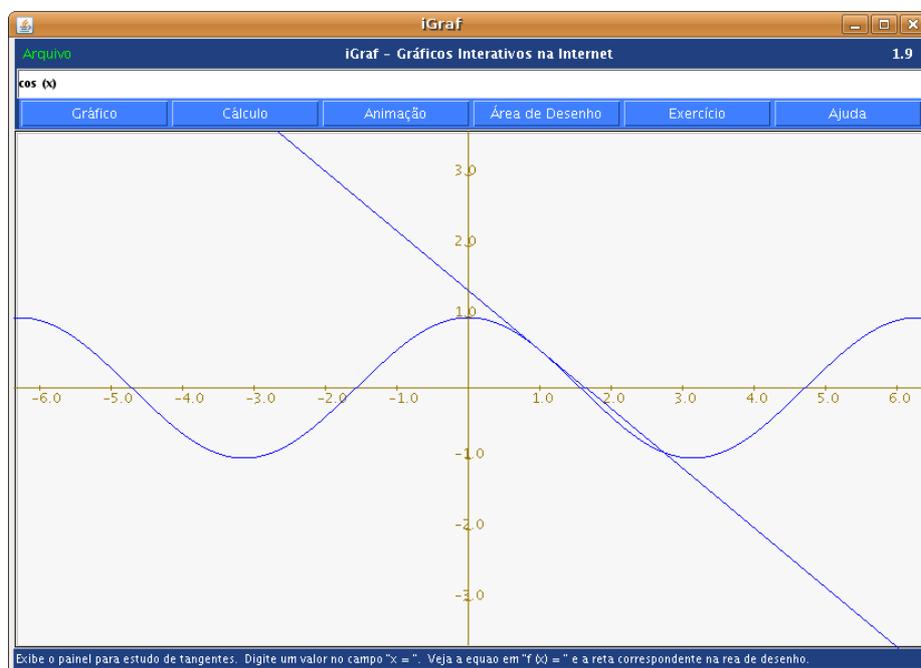
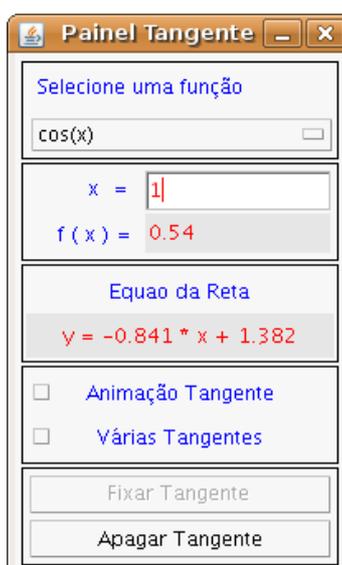
Figura 3.4: Tangente de  $\cos(x)$  em  $x = 1$ 

Figura 3.5: Configuração que gerou a figura 3.4

numéricos. É necessário apenas que a função para a qual se deseja desenhar o gráfico da integral esteja no campo de edição. Satisfeita essa condição, basta clicar no botão **Cálculo**, selecionar a opção **Visualizar Integral Indefinida** e o gráfico de  $F(x)$  será exibido na área de desenho. Dentre os vários visualizadores gráficos analisados apenas o **Winplot** possui funcionalidade semelhante.

O algoritmo utilizado pelo iGraf para calcular integrais é chamado de **Regra dos Trapézios Repetida** (32) e pode ser descrito pela equação 3.1. Além de ser usado para definir os pontos do gráfico da integral indefinida de uma função, o algoritmo também é usado para fazer integração definida e cálculo de área adotando-se, no entanto, estratégias distintas para o tratamento dos resultados em cada caso.

$$\int_a^b f(x)dx = \left(\frac{\Delta x}{2}\right) \left(f(a) + f(b) + 2 \sum_{i=1}^{n-1} f(x_i)\right) \quad (3.1)$$

onde

$$\Delta x = \left(\frac{(b-a)}{n}\right) \quad \text{e} \quad \{x \in \mathbf{R} \mid a \leq x \leq b\} \quad (3.2)$$

sendo  $n$  o número de trapézios com o qual se subdivide a região e  $a$  e  $b$  os limites laterais da região a ser integrada.

Para calcular a integral definida (veja seção 3.5.4) a fórmula é usada sem alterações. Para calcular área é necessário substituir na fórmula  $f(x)$  por  $|f(x)|$ . Já as coordenadas de cada ponto  $P$  da curva descrita por  $F(x)$  são dadas por:

$$P_i = (x_i, F(x_i)), \quad (3.3)$$

sendo  $F(x_i)$  a integral definida no intervalo de 0 a  $i$  calculada numericamente.

### 3.5.4 Calcular Integral Definida

Com o iGraf também é possível calcular integral definida ou a área de uma região delimitada por duas curvas. Para utilizar essa funcionalidade, basta clicar no botão **Cálculo** e selecionar a opção **Calcular Integral Definida**. Será exibida uma janela na qual o usuário poderá selecionar as funções que determinam as curvas limitantes superior e inferior. Por padrão, a curva limitante inferior é o eixo das abscissas definido pela função  $f(x) = 0$ . Para o cálculo de integral definida é necessária também a definição dos limites laterais  $x_0$

e  $x_1$  do intervalo de integração. (36) Tendo essas variáveis definidas, basta clicar no botão **Integrar** e será exibido o valor da integral (ou da área) definida por  $f$  e  $g$  dependendo do modo de cálculo selecionado na janela de integração. É importante destacar que o iGraf faz explícita distinção entre cálculo de área e integral definida, usando para isso um esquema de cores diferente para cada caso (veja figuras 3.6, 3.7 e 3.8). Essa distinção não foi encontrada em nenhum dos programas analisados durante a confecção deste trabalho.

### 3.6 Menu Animação

Uma característica interessante presente em alguns visualizadores gráficos é a possibilidade de parametrização das expressões matemáticas para que se possa estudar os efeitos que a variação nos valores dos parâmetros têm sobre a forma dos gráficos. Para Figueiredo,

*“O apoio computacional é importante na compreensão da influência dos parâmetros e do que ocorre quando se combinam funções por meio de operações matemáticas. Em particular, o recurso de produzir animações traz vida ao efeito da variação de um parâmetro sobre uma família de gráficos”*

(16)

O iGraf conta com um mecanismo simples de parametrização das expressões matemáticas que permite ao usuário visualizar uma animação sobre os seus gráficos. Para gerar uma animação, basta substituir um número de uma expressão por um dos parâmetros de animação.

O algoritmo que analisa as expressões detecta automaticamente a presença de um dos parâmetros variáveis permitidos pelo programa. Os parâmetros variáveis que podem ser

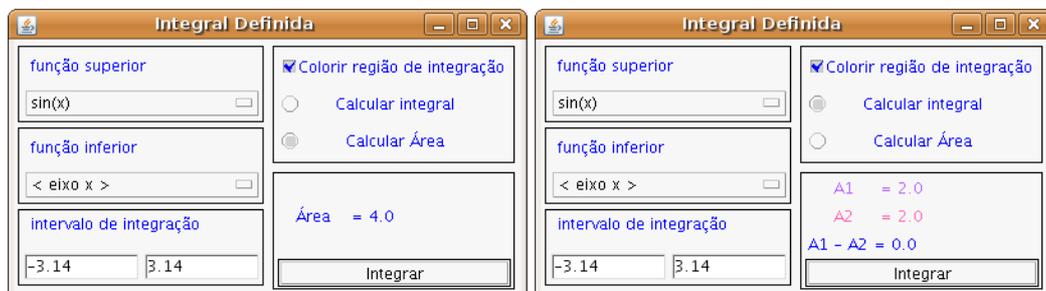


Figura 3.6: Dinstinção explícita entre integral e área

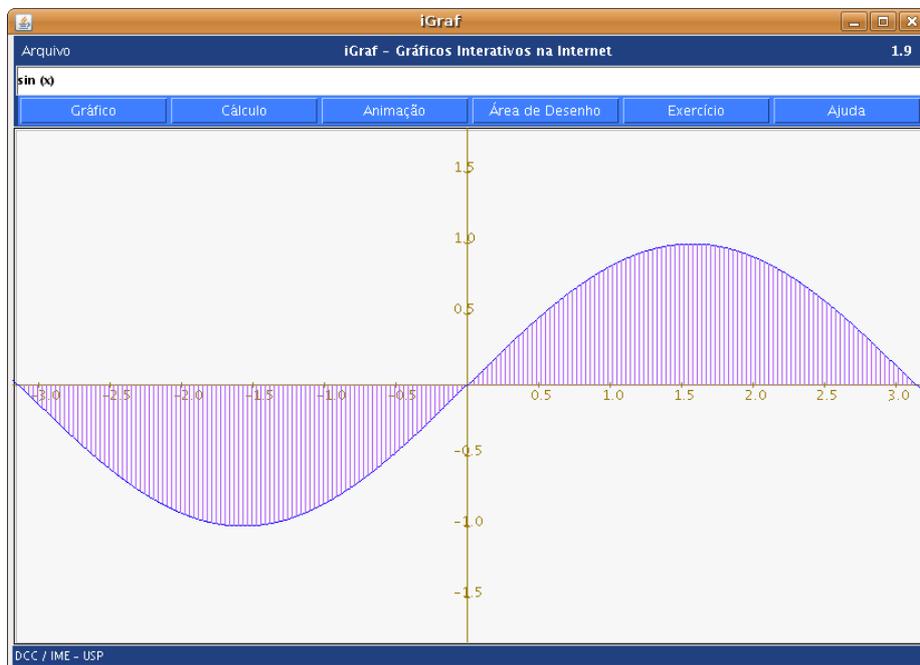


Figura 3.7: Cálculo de área

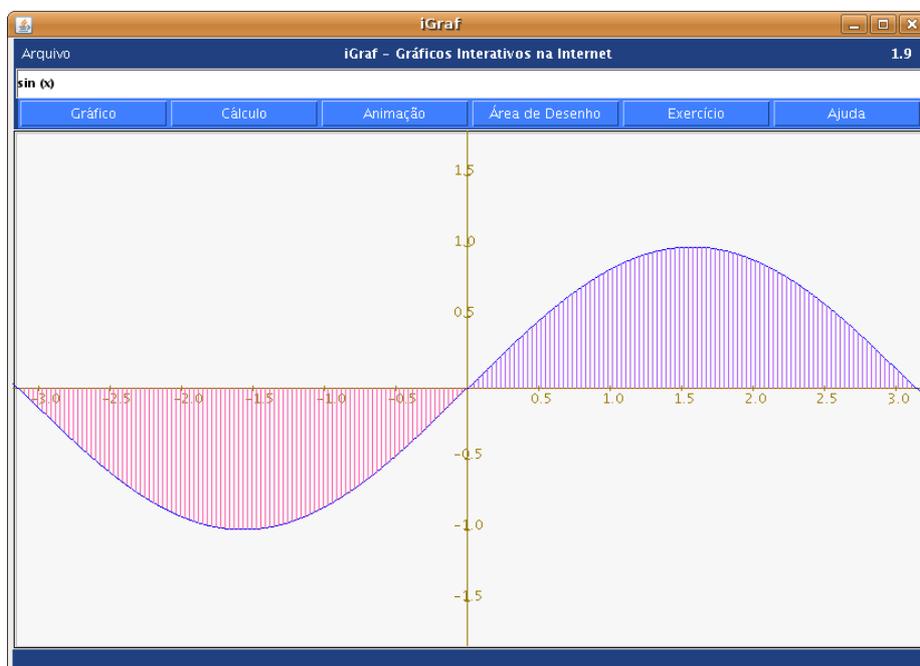


Figura 3.8: Pintura diferenciada das regiões “positiva” e “negativa”

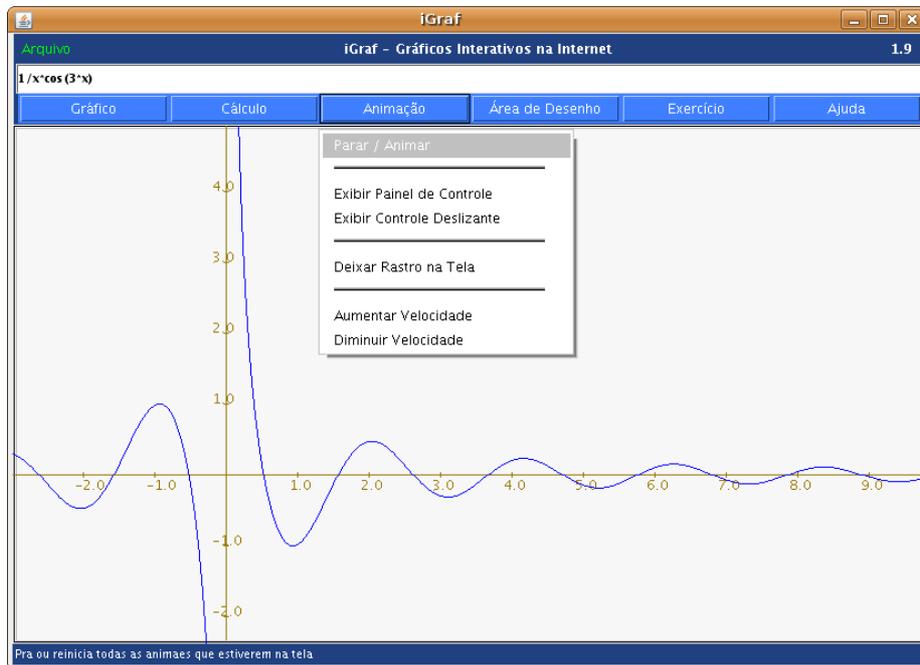


Figura 3.9: Opções para controle de animação

utilizados são as letras minúsculas  $a$ ,  $b$ ,  $c$ ,  $k$ ,  $m$  e  $n$ ; por padrão, o valor destes parâmetros varia no intervalo de -1 a 1. Para alterar estes valores ou mesmo desabilitar algum dos parâmetros basta clicar na área de desenho com o botão direito e usar o **Painel de Parâmetros**.

Quando um parâmetro variável é detectado, o programa assume que o usuário deseja ver uma animação. No iGraf, uma animação é o desenho seqüencial de uma série de gráficos que são gerados pela alteração dos parâmetros em um intervalo de valores determinado pelo usuário. Essa animação pode deixar ou não um rastro na tela, o que permite a visualização de quanto a forma do gráfico muda em relação à modificação do valor do parâmetro. Assim, a expressão  $a*\cos(x)$  ou qualquer outra que conte com um parâmetro variável, desencadeará o processo de animação. Para ver o rastro na tela o usuário do programa deve clicar no botão **Animação** e selecionar a opção **Deixar Rastro na Tela**.

### 3.6.1 Controle Parar / Animar

A animação pode ser automática ou controlada manualmente. A animação automática é aquela em que o usuário não interfere, ele pode assistir a exibição da animação e obser-

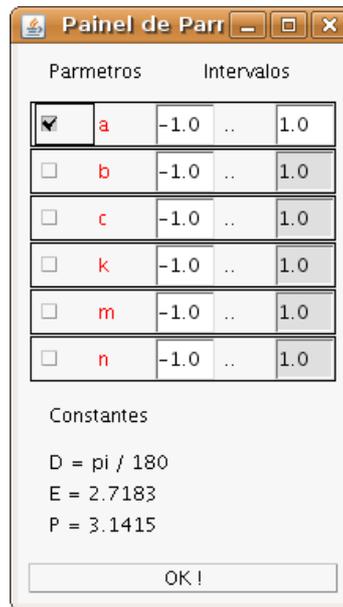
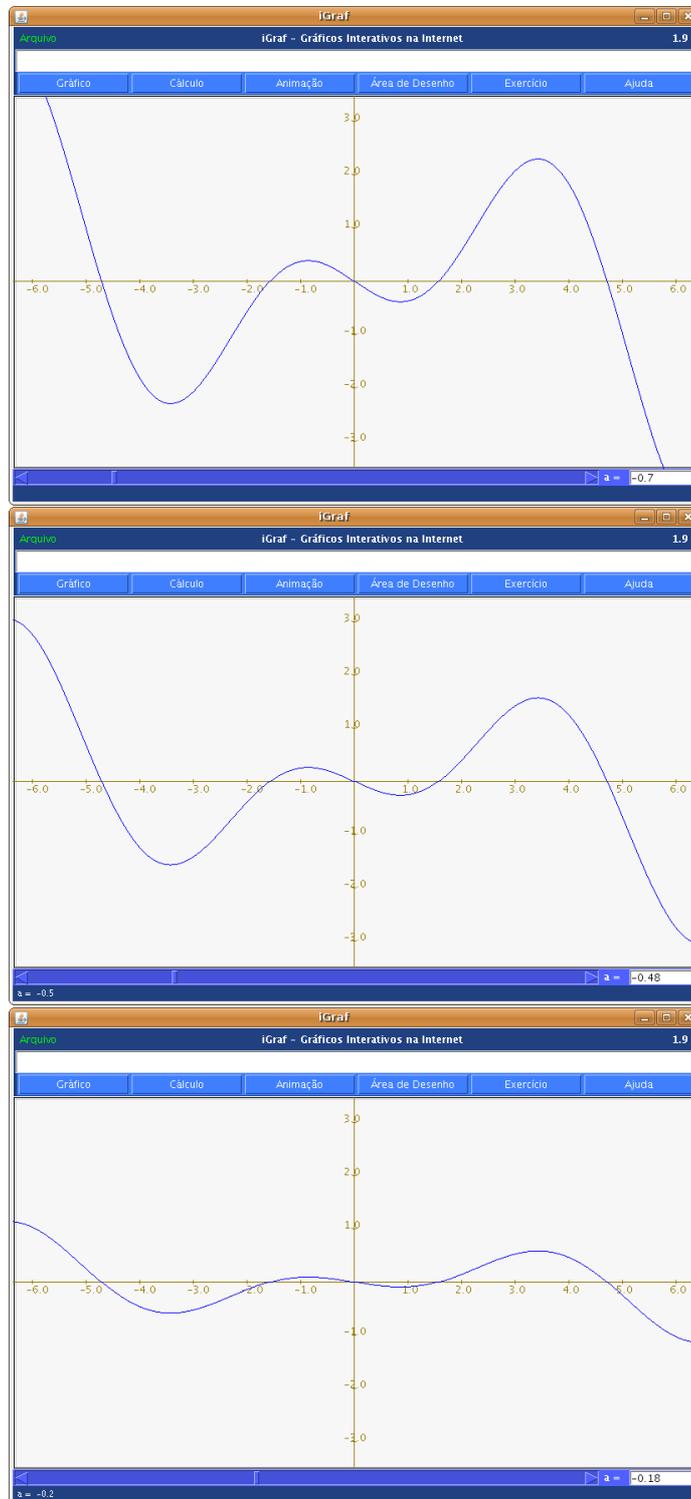


Figura 3.10: Painel de configuração dos parâmetros da animação

var o comportamento do gráfico para os valores que o parâmetro assume. A animação automática funciona por tempo indeterminado e ao usuário é permitido parar e reiniciar o processo quantas vezes desejar. Esse é o comportamento padrão das animações.

Em oposição à animação automática, o iGraf permite a animação controlada manualmente. O usuário pode utilizar um controle deslizante (*slider*) para fazer variar o valor que o parâmetro  $a$  assume. Durante a animação automática,  $a$  assume valores reais no intervalo  $[V_i..V_f]$  em passos de 0,1 onde  $V_i$  é o valor inicial e  $V_f$  é o valor final. Por padrão,  $V_i = -1$  e  $V_f = 1$ . A implementação da animação com controle manual, devido à menor exigência de recursos computacionais, permite a variação no valor de  $a$  em passos de 0,01 produzindo uma seqüência de desenhos mais próximos um do outro, produzindo assim uma animação mais “suave”.

O valor que os outros parâmetros assumem é indexado pelo valor de  $a$ . Assim, só é possível determinar  $V_i$  para os outros parâmetros;  $V_f$  fica determinado pelo módulo do intervalo  $[V_i..V_f]$  definido para o parâmetro  $a$ . Por exemplo, se o intervalo de variação de  $a$  está definido entre -1 e 1, o módulo do intervalo é 2. Assim, se for definido que um parâmetro  $b$ , por exemplo – tem  $V_i = 3$ , esse parâmetro terá obrigatoriamente  $V_f = 5$ . Em versões futuras essa forma de indexação deverá ser abandonada e a determinação de  $V_i$  e

Figura 3.11: Quadros de uma animação controlada por *slider*

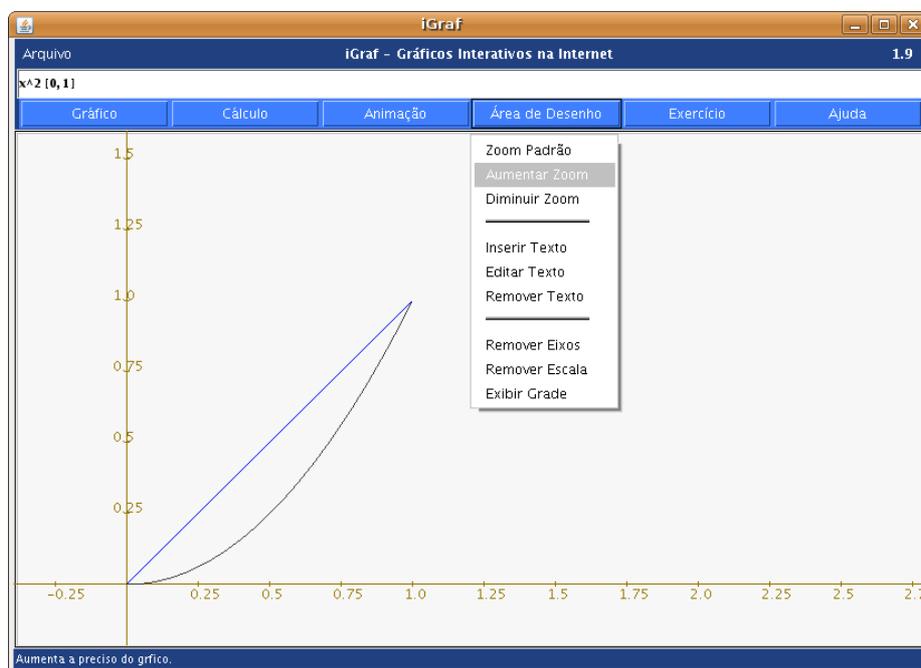


Figura 3.12: Controles para o “papel” do gráfico

$V_f$  para cada parâmetro deverá se tornar independente.

## 3.7 Menu Área de Desenho

Programas para edição de gráficos de função costumam apresentar vários recursos para padronização da forma de apresentação dos desenhos. Para que se possa ter acesso aos controles de padronização do “papel do gráfico”, o usuário deve clicar no botão *Área de Desenho*.

### 3.7.1 Controles de Zoom

O iGraf permite vários modos de exibição dos gráficos podendo o usuário escolher entre exibir ou não os eixos cartesianos e/ou seus rótulos, ocultar os eixos e exibir uma grade ou até mesmo mostrar os gráficos sem nada ao fundo. A escolha da forma de apresentação se dá pela combinação de cliques nas opções *Ocultar Eixos*, *Ocultar Escala* e *Exibir Grade*.

### 3.7.2 Edição de Texto

Outro recurso interessante é o que possibilita a inserção de texto na área de desenho. Esse texto pode ser desde um simples lembrete até o enunciado de um exercício escrito por um professor que pretenda disponibilizá-lo aos seus alunos. A ferramenta de texto permite a configuração da cor do texto e do tamanho da fonte. Para inserir um texto basta clicar no botão **Área de Desenho** e selecionar o item **Inserir Texto**, digitar o texto na janela que será apresentada, escolher as coordenadas do texto e clicar em **inserir**. Caso o usuário pretenda modificar a posição do texto na área de desenho, basta clicar sobre o texto e arrastá-lo para a nova posição ou selecionar a opção **Editar Texto** e alterar as coordenadas.

## 3.8 Menu Exercício

Ferramentas modernas usadas em ambientes gerenciadores de cursos geralmente contam com mecanismos de criação, resolução e/ou correção automática de exercícios, além de ferramentas para monitorar as interações de seus usuários. O iGraf possui um mecanismo para registro das ações dos usuários chamado histórico, descrito abaixo. Também conta com um conjunto simples de ferramentas que permitem a criação e avaliação automática de exercícios. O capítulo 4 faz uma descrição detalhada dessas ferramentas.

### 3.8.1 Histórico

As ações de um usuário do iGraf são armazenadas em um registro chamado **sessão**. A ferramenta *histórico* permite a visualização dos dados registrados na sessão de um usuário. Com o uso do histórico, um professor pode analisar todos os passos que um aluno realizou até atingir um determinado resultado e verificar a correção desses passos. Outro aspecto que pode ser explorado é a criação de pequenos “tutoriais”. O professor pode solucionar um problema e disponibilizar aos alunos o seu arquivo; os alunos poderão então rever a construção da solução passo-a-passo e por quantas vezes julgarem necessário.

Dentre os diversos visualizadores gráficos analisados durante esta pesquisa não foi possível detectar recurso similar.

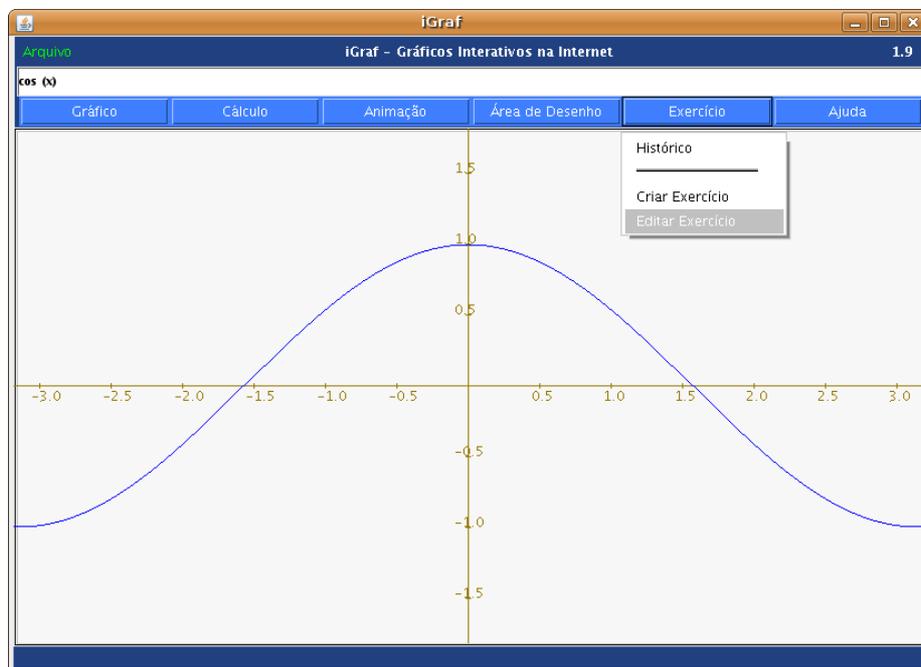


Figura 3.13: Opções ligadas a exercícios

## Capítulo 4

# Autoria e Validação Automática de Exercícios

O uso de sistemas computacionais modernos e da Internet como ferramentas de apoio ao ensino têm alterado de maneira positiva o sistema de ensino tornando possível o acesso de um grande número de pessoas à Educação. Por outro lado, o aumento na quantidade de alunos matriculados em cursos, tanto presenciais quanto a distância, tem incrementado a carga de trabalho dos professores na preparação e correção de atividades. É necessário, portanto, que essa mesma tecnologia, que possibilita um maior acesso à informação, possa também ser usada para facilitar o trabalho do professor.

Como já observado no capítulo 2, uma importante categoria de ferramentas utilizadas como suporte à Educação na *Web* é a dos Sistemas Gerenciadores de Cursos (SGC). Boa parte desses sistemas conta com ferramentas para o gerenciamento de conteúdo e monitoramento da interação do usuário, mantendo registros de quantos foram os acessos de um determinado aluno e quando ocorreram, qual foi o material visualizado e até mesmo quanto tempo o aluno esteve conectado. Vários SGCs também contam com mecanismos que permitem a criação de atividades que podem ser avaliadas automaticamente, embora essa avaliação ocorra, em geral, com o uso de questionários de múltipla escolha, permitindo apenas que o professor registre no sistema uma série de alternativas e que indique, entre elas, qual é a correta. Esse modelo de avaliação é particularmente insatisfatório em atividades de aprendizagem por não possibilitar ao professor entender o raciocínio do aluno. Alguns desses sistemas oferecem ainda possibilidades como edição de texto *online* e até envio de arquivos inteiros, mas, nesses casos, a avaliação automática se torna impraticável.

Ainda assim, embora essas ferramentas de avaliação automática apresentem restrições, elas têm o mérito de fornecer ao usuário uma resposta imediata, tornando o aluno consciente dos seus resultados, possibilitando, e até incentivando, que o mesmo reveja sua postura de aprendizagem quando tiver uma avaliação insatisfatória.

Este capítulo apresenta algumas considerações sobre o estudo de função e mostra os resultados de uma pesquisa sobre as categorias de resposta para exercícios sobre o tema. Além disso, apresenta as implementações feitas no iGraf, possibilitando a avaliação automática para as três categorias de resposta identificadas.

## 4.1 Estudo de Função

O tópico “função” é recorrente em livros de matemática do ensino médio e em livros introdutórios à matemática do ensino superior. A importância do estudo de funções fica evidenciada pela quantidade de livros que abordam o tema nas mais diversas áreas, como Matemática Financeira, Nutrição, além dos cursos clássicos de Ciências Exatas.

Outro tópico importante da Matemática é o Cálculo. Para Kleitman, “o estudo do Cálculo requer familiaridade com duas noções básicas: a noção de número e a noção de função” (21). Já Ávila diz que “todo o Cálculo diferencial e Integral se desenvolve em torno de dois conceitos fundamentais: o conceito de função e o conceito de limite”(37)p. 39. Stewart, por sua vez, reitera que o “objeto fundamental do Cálculo são as funções” (36)p. 11. Concluindo, Barnett afirma que “o conceito de função é uma das mais importantes idéias em Matemática” (2)p. 3.

De modo geral, o “...estudo de matemática além do nível elementar requer uma firme compreensão de uma lista básica de funções elementares, suas propriedades e seus gráficos” (2), conseqüentemente, o domínio dos conceitos relacionados ao estudo de função demanda que o estudante procure resolver exercícios sobre o tema. Essa necessidade fica clara pela grande quantidade de exercícios presente nos livros-texto de Cálculo.

Ao professor, nesse contexto, cabe a tarefa de corrigir um grande número de exercícios e fornecer aos seus alunos informações sobre seus desempenhos na realização de suas tarefas. Acrescente-se a este cenário o fato de que o aumento do acesso à educação formal, no ensino médio (13), no ensino superior presencial (12) e no ensino a distância de modo geral (8), tem levado a um incremento no número de alunos por turma indicando que

ferramentas de autoria e avaliação automática para exercícios são muito bem vindas.

Assim, o uso de avaliação automática se apresenta como uma alternativa com potencial para diminuir o tempo gasto com a correção e atribuição de notas às atividades dos alunos permitindo que o professor possa dedicar mais tempo à preparação de boas atividades didáticas, inclusive determinadas pelo diagnóstico dos exercícios aplicados às turmas. Porém, para que seja possível criar uma ferramenta avaliadora automática, é necessário conhecer os tipos de questões que podem ser formuladas sobre um determinado tema e o formato das possíveis respostas para estas questões. Para a obtenção desses dados sobre função foi feito o estudo cujos resultados são apresentados na seção 4.2.

## 4.2 Modalidades de respostas em atividades com funções

Com o objetivo de identificar os tipos de respostas possíveis para questões sobre função, foram analisadas as listas de exercícios relacionados ao tema das seguintes publicações:

- Livros de matemática do ensino médio
  - Iezzi - Matemática: Ciência e Aplicações vol. 1
  - Paiva - Matemática vol. 1
- Livros do ensino superior
  - Ávila - Cálculo I: diferencial e integral
  - Boulos - Introdução ao Cálculo
- Livros de atividades computacionais para ensino de função
  - Figueiredo - Cálculo com aplicações
  - Silva - Atividades para o estudo de funções em ambientes computacionais
- Páginas Web que tratam do tema “função”
  - Barufi - e-Cálculo <sup>1</sup>
  - Taylor e Fraser - Exercises in Math Readiness <sup>2</sup>

Como resultado dessa análise, concluiu-se que as categorias de resposta seriam: **resultados numéricos; conjuntos-resposta; expressões algébricas; e gráficos**. Esses tipos de

---

<sup>1</sup><http://www.cepa.if.usp.br/e-cálculo>

<sup>2</sup><http://math.usask.ca/emr/menu.html>

resultados adaptam-se à grande maioria dos exercícios encontrados nas obras analisadas, como indica a figura 4.1

Exercícios que contêm vários itens e que, portanto, requerem mais que uma resposta, foram classificados considerando-se cada item como um exercício independente, já que cada parte da questão pode requerer um tipo diferente de resposta; todas as respostas que não se enquadraram nos padrões acima foram incluídas na categoria **outros**.

- **Resultados Numéricos**

Um tipo de exercício bastante comum nos livros, tanto do ensino médio quanto do ensino superior, é aquele que demanda como resposta um valor numérico, que pode ser um escalar ou um vetor.

São comuns as questões que propõem ao estudante a determinação de pontos que maximizam ou minimizam uma função, pontos de inflexão ou pontos de intersecção entre curvas. Vejamos dois exemplos de Stewart:

“Se a reta tangente a  $y = f(x)$  em  $(4, 3)$  passa no ponto  $(0, 2)$ , encontre  $f(4)$  e  $f'(4)$ ”.

“Encontre os valores de máximo e mínimo local para  $f(x) = x^3 - 12x + 1$ ”.

Estes são dois tipos de exercícios comuns nos textos considerados e que, independentemente do grau de complexidade da resolução, solicitam ao estudante apenas uma resposta numérica.

- **Conjuntos Numéricos**

A análise das condições de existência de funções requer que, eventualmente, sejam determinados conjuntos sobre os quais uma função pode ou não ser aplicada. As respostas a esse tipo de questão devem ser dadas em alguma notação que expresse de forma sintética qual é o conjunto.

Os exercícios nessa categoria geralmente envolvem soluções de desigualdades, intervalos de crescimento e de decréscimo, domínio, contra-domínio e imagem de funções. Tipicamente esses exercícios requerem respostas na forma de intervalo numérico.

Um exemplo dessa categoria é o clássico exercício: “Qual é o domínio e a imagem da função  $f(x)$ ”, que pode ser encontrado em quase toda obra que trate do tema ‘função’.

- **Expressões algébricas**

O estudo de diversos fenômenos se vale da modelagem matemática, ou seja, da descrição idealizada dos mesmos através de equações ou funções matemáticas. A partir dessas expressões, é possível a utilização de ferramentas matemáticas, como o Cálculo, para a análise da situação descrita. Por essa característica é comum a existência de exercícios que solicitem que o estudante forneça como resultado, ou parte dele, uma expressão algébrica.

Também são comuns exercícios envolvendo cálculo de derivadas, integrações indefinidas, funções compostas e equações de retas tangentes. Tais exercícios requerem que o aluno busque respostas na forma de expressões algébricas. Por exemplo, o exercício: “Encontre uma função contínua positiva  $f$  tal que a área sob o gráfico de  $f$  [no intervalo] de 0 a  $t$  é  $A(t) = t^3$  para todo  $t > 0$ ” (36) requer como resposta uma expressão algébrica.

- **Gráficos**

“O método mais comum de visualizar uma função consiste em fazer seu gráfico. [...] O gráfico de uma função nos dá uma imagem proveitosa do comportamento ou história da vida de uma função” (36)p. 12 .

Questões que solicitam que o estudante faça um ou mais gráficos são comuns no estudo de função e em outros tópicos da Matemática. Em todos livros analisados foi possível encontrar exercícios com enunciados do tipo: *Esboce o gráfico da função  $f$ .*

Devido às características dessa categoria de exercícios, quando considerada com o uso de visualizadores gráficos, é necessário admitir-se como resposta a expressão algébrica que define o gráfico. Por esse motivo, exercícios que exigem resposta na forma de gráfico foram tabulados (veja tabela 4.2.1) junto àqueles que exigem expressões algébricas como resposta.

- **Outros tipos de resposta**

Existem questões para as quais o padrão de resposta é diferente dos formatos descritos acima. Dentre elas se encontram as discursivas, que solicitam ao estudante uma demonstração, uma análise ou mesmo a verificação de alguma propriedade. Foram incluídas ainda nesse grupo, as questões com respostas do tipo verdadeiro ou falso e de múltipla escolha. Ainda assim, o percentual encontrado para essa categoria foi de apenas de 15%.

Autor	num	conj	expr	outros	
Iezzi	274	164	104	74	616
Paiva	174	75	57	33	339
Ávila	235	26	317	85	663
Boulos	83	0	189	40	312
Silva	163	139	132	119	553
Figueiredo	67	6	109	55	237
Barufi	10	8	50	19	87
Taylor e Fraser	78	23	74	9	186
	1084	441	1032	434	2991

Tabela 4.1: Quantidade *versus* tipo de exercício nas publicações analisadas

### 4.2.1 Dados obtidos

A tabela 4.2.1 e a figura 4.1 mostram os resultados obtidos com a contagem dos tipos de resposta esperados para os exercícios analisados nas publicações citadas.

A coluna **num** representa os exercícios para os quais as respostas esperadas são valores numéricos. A coluna **conj** representa os exercícios para os quais as respostas esperadas são intervalos numéricos. A coluna **expr** representa os exercícios para os quais as respostas esperadas são expressões algébricas ou gráficos. A coluna **outros** representa os exercícios para os quais as respostas esperadas não podem ser classificadas em nenhum dos três padrões acima.

## 4.3 Autoria de Exercícios

A partir das três modalidades de resposta identificadas, foram implementadas no iGraf ferramentas que possibilitam a autoria e avaliação automática para cada modalidade. Essas implementações seguiram os princípios de usabilidade discutidos no capítulo 5 visando não demandar do professor/autor e nem do aluno a leitura de várias páginas de manuais.

Nas subseções seguintes serão explicadas as funcionalidades envolvidas na produção de exercícios (autoria), os métodos utilizados na avaliação automática de cada categoria de exercícios e como o iGraf deve ser usado para enviar solução de exercício.

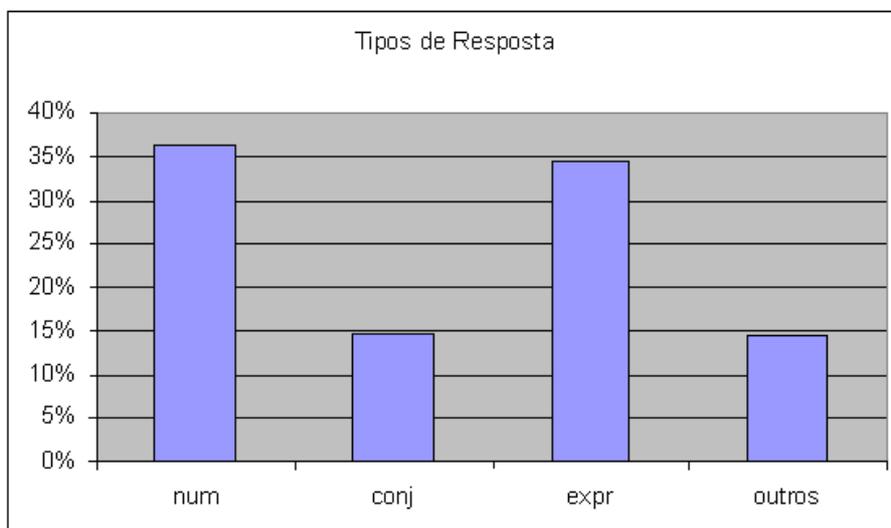


Figura 4.1: Relação percentual entre os tipos de resposta

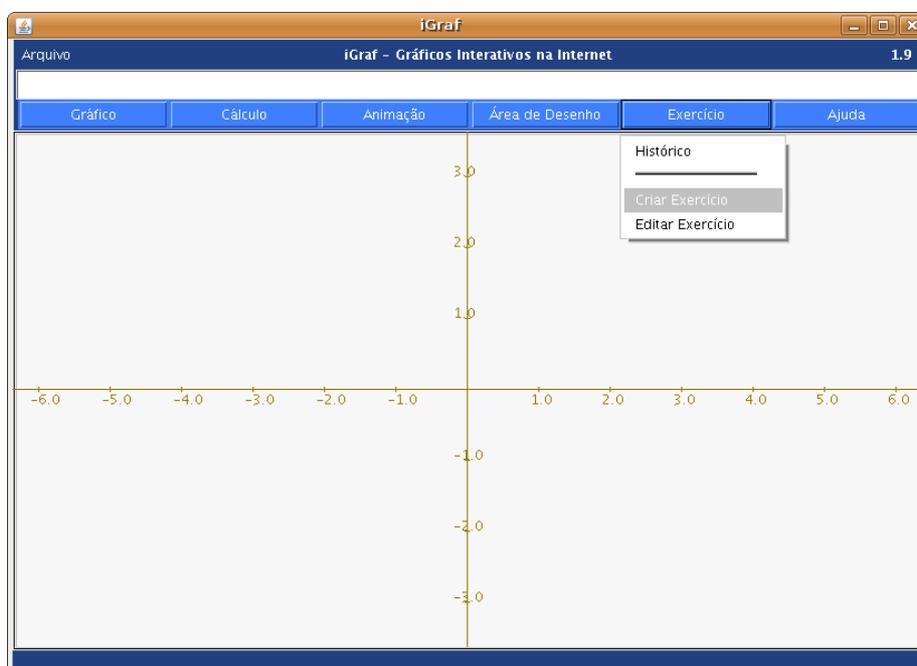


Figura 4.2: Opções para a criação e edição de exercícios

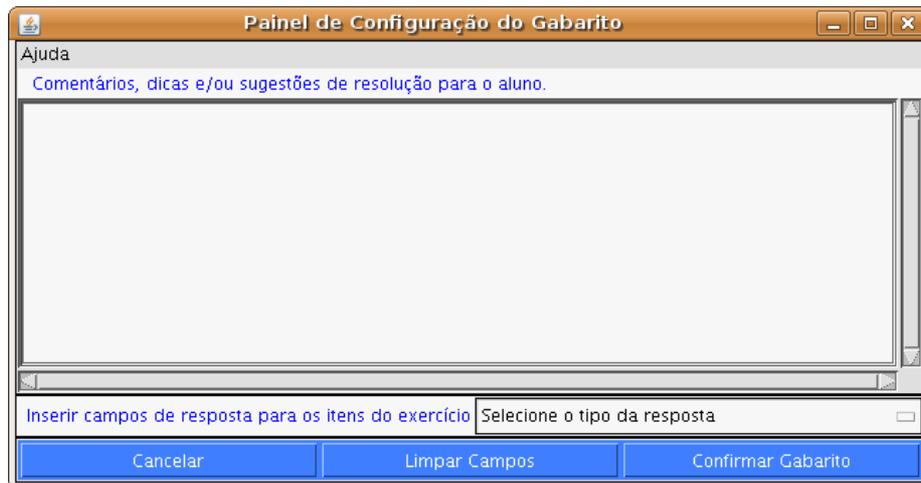


Figura 4.3: Janela de Configuração do Gabarito

### 4.3.1 Definição de Gabaritos

Para criar um exercício, o professor precisa definir um gabarito onde constam as respostas que espera receber de seus alunos. Ele pode inserir textos ou gráficos na área de desenho do iGraf, como parte do enunciado. Pode também deixar a tela em branco, caso o exercício requeira que o aluno “construa” gráficos. Depois de definido o conteúdo, o professor clica no menu **Exercício** mostrado na figura 4.2 e seleciona a opção **Criar Exercício** para iniciar a configuração do gabarito.

Em um mesmo gabarito, o professor pode inserir quantos itens desejar de uma das três categorias de exercícios, indicando em cada um desses itens qual é a resposta esperada. A inserção é feita a partir da **Janela de Configuração do Gabarito** (figura 4.3), quando o usuário seleciona um tipo de resposta dentre as opções da lista apresentada na figura 4.4. Note que aparecem quatro opções na lista, mas as duas primeiras pertencem à mesma categoria.

Supondo que um professor criou um exercício dividido em quatro partes, no momento de enviar a resposta o aluno verá uma tela parecida com aquela apresentada na figura 4.5. A ordem na qual serão exibidos os campos para o aluno encaminhar suas respostas é a mesma do gabarito produzido pelo professor.

Além de selecionar os tipos de respostas o professor pode inserir (na parte superior do gabarito) um texto com o objetivo de orientar os alunos. Caso o professor insira um tipo

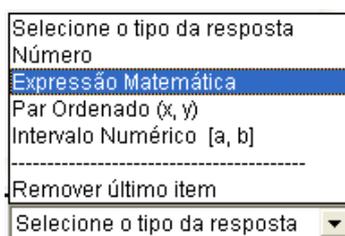


Figura 4.4: Opções de tipos de resposta

de resposta por engano, existe a opção de eliminar o último campo inserido. Para finalizar a configuração do gabarito, basta clicar em **Confirmar Gabarito**.

O iGraf permite que o professor formule questões discursivas e receba respostas na forma de texto. Nesse caso, naturalmente, não é necessária a seleção de tipos de resposta e basta ao professor confirmar o gabarito com a lista de opções vazia (4.3). No momento da resposta, o aluno deverá preencher a área de texto que fica na parte superior da janela de respostas.

### 4.3.2 Geração de Arquivos

Depois de terminar as configurações de um exercício, o professor deverá optar entre duas possibilidades de registro no disco de seu computador: gravar o exercício em um arquivo do iGraf (com extensão **.grf**) ou gravar na forma de página HTML.

Arquivos do iGraf podem ser carregados pela versão aplicativo (que é executada na máquina do usuário) do programa. Uma vez carregado, o conteúdo de um arquivo pode ser modificado e as mudanças registradas ou descartadas. O mecanismo de avaliação automática (que será discutido na seção 4.4) também funciona nesta situação e a única diferença é que a resposta do usuário não é enviada a um SGC (veja 3.1). Porém, se o usuário preencher os campos de resposta e clicar no botão **Enviar Resposta**, receberá a tela de verificação dos resultados informando se suas respostas estão corretas ou não.

Esta forma de uso do iGraf é interessante para a situação em que o professor pode estar com seus alunos em um laboratório de informática ou quando o mesmo não tem meios para publicar conteúdo em um servidor. (veja 3.1) O professor poderia gravar os exercícios em arquivos e disponibilizá-los durante a aula para que seus alunos pudessem resolvê-los. Outra vantagem é que, ao final da aula, os alunos poderiam levar os arquivos de resolução

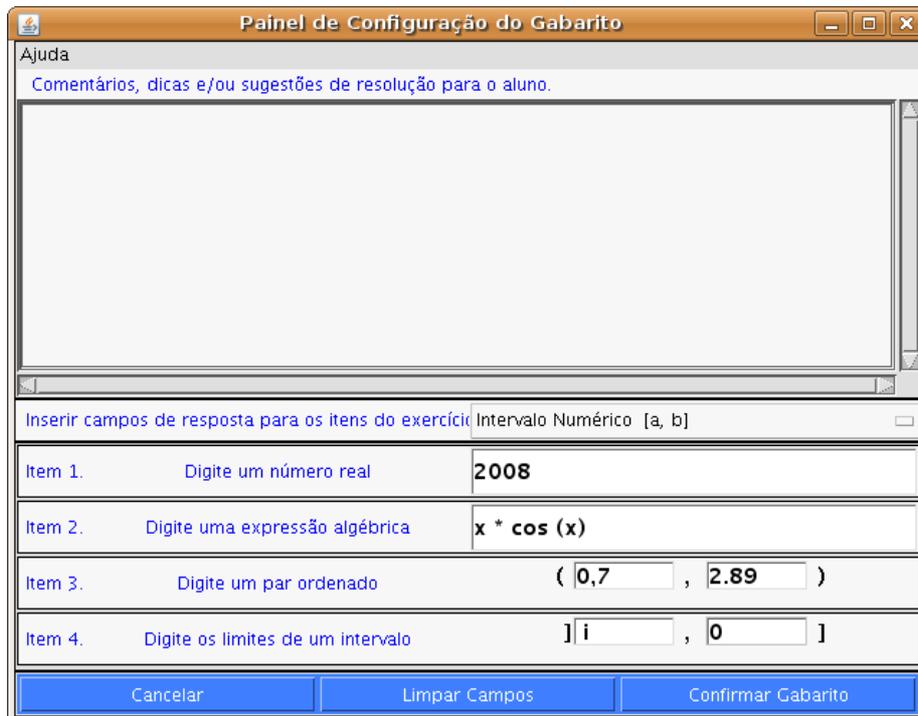


Figura 4.5: Janela com os tipos possíveis de resposta

ou exemplos para casa em algum dispositivo de armazenamento de dados digitais, como um disquete ou um *pen-drive*.

Além de gravar o conteúdo em arquivo, o iGraf permite a geração de páginas *Web*, que também podem conter exercícios ou servir apenas para a visualização de gráficos.

Uma página *Web* é um arquivo de texto que utiliza o formato HTML (*HyperText Markup Language*<sup>3</sup>) e que deve ter a extensão **.html**. Este tipo de arquivo pode ser visualizado com o uso de navegadores *Web* como o *Mozilla Firefox*, o *Internet Explorer*, dentre outros. Além disso, a página pode executar mini-aplicativos, mais conhecidos como *applets* (programas que podem ser executados pelo navegador). O iGraf, em sua versão aplicativo, pode gerar páginas HTML configuradas para carregar e executar uma cópia dele mesmo (sua versão *applet*) exibindo o conteúdo definido pelo usuário através da *Web*.

Assim, o professor pode gerar páginas HTML que apenas mostrem o trabalho realizado em uma sessão (3.8.1) de uso do iGraf (para, digamos, usar como exemplo em uma aula) ou páginas contendo exercícios, que poderão ser publicadas em um servidor *web* para que seus

<sup>3</sup>Saiba mais sobre HTML em <http://www.w3.org/html/>



Figura 4.6: Geração de página HTML: apenas um clique

alunos acessem pela Internet, resolvam os exercícios, tenham seus resultados registrados por um SGC e recebam um parecer imediato sobre suas soluções.

Para criar tais páginas, o professor não tem que fazer qualquer tipo de opção ou configuração. A grande vantagem oferecida pelo programa é que qualquer usuário pode gerar estas páginas sem ter o menor conhecimento de programação, bastando apenas clicar na opção **Gerar Página HTML** do menu **Arquivo**.

A resolução de exercícios pelo aluno é igualmente simples. Quando o iGraf carrega um arquivo que foi salvo após a configuração do gabarito ou é iniciado (na forma *applet*) em uma página HTML contendo parâmetros de exercício, ele detecta que deve utilizar o modo de resolução de exercício e oferece uma lista adequada de opções de menu com as quais o aluno interagirá. O trabalho do aluno, então, se resume a resolver o exercício, preencher as respostas e clicar no botão de envio.

## 4.4 Critérios de Validação

Uma vez que o professor criou um exercício (configurando um gabarito) o iGraf pode receber e avaliar automaticamente as respostas dos seus alunos. Nas subseções abaixo, serão descritas as estratégias utilizadas no programa para que isso seja possível.

### 4.4.1 Tratamento para repostas numéricas

A primeira categoria de respostas inclui números reais e pontos. Devido a pequenas variações que podem ocorrer em resultados na forma de números reais causadas por diferentes estratégias de cálculo e até mesmo pelo uso de máquinas com maior ou menor grau de precisão, a verificação desse tipo de resposta requer a determinação de um valor de erro máximo tolerável  $\varepsilon$  na resposta do aluno. Por padrão a tolerância é dada por  $\varepsilon = 0,01$ . Em versões futuras pretende-se que esse valor possa ser configurado pelo professor.

A verificação é feita a partir da distância entre a resposta do aluno e a do professor. O valor  $A$ , que é enviado pelo aluno, é comparado com o resultado  $R$  fornecido pelo professor. Para ser considerado correto, o resultado do aluno deve satisfazer a condição:

$$R - \varepsilon \leq A \leq R + \varepsilon$$

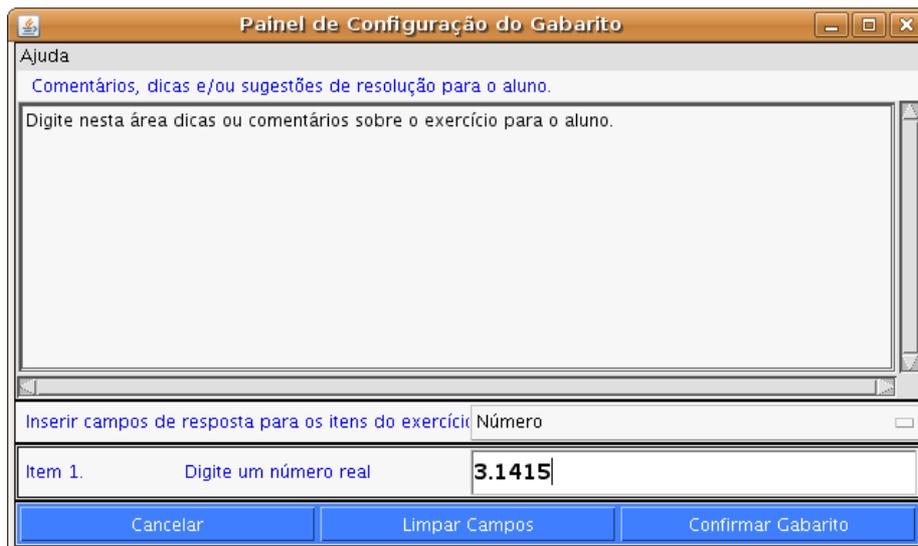


Figura 4.7: Configuração do gabarito: resposta numérica

No caso da solução ser um par ordenado, utiliza-se como critério de distância a métrica euclidiana.

### 4.4.2 Tratamento para repostas do tipo conjunto numérico

A notação adotada para a representação de conjuntos no iGraf utiliza números entre colchetes e a direção para onde apontam os colchetes indica se o conjunto inclui ou não os valores

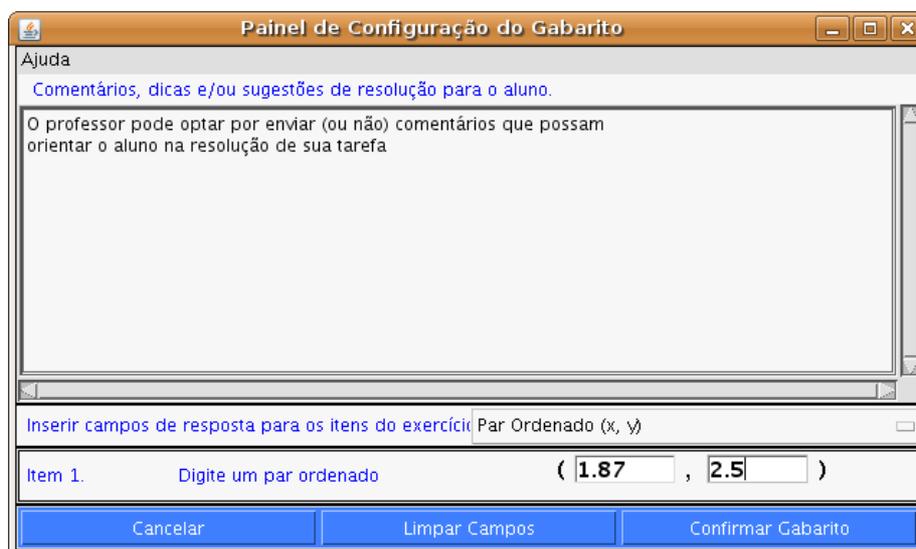


Figura 4.8: Configuração do gabarito: resposta ponto

das extremidades. O usuário do iGraf pode alterar a direção dos colchetes clicando sobre os mesmos.

Essa notação elimina a necessidade do uso de operadores lógicos e relacionais, permitindo o uso de uma estratégia simples para analisar a resposta. Tal estratégia trata o problema de avaliar se um conjunto numérico  $C_a$  fornecido pelo aluno está de acordo com a resposta  $C_p$  fornecida pelo professor dividindo-o em duas etapas: primeiro verifica os valores numéricos usando os critérios discutidos na seção 4.4.1. Depois compara as posições dos colchetes para verificar se representam intervalo fechado ou aberto em cada uma das extremidades.

Existem, no entanto, casos especiais de respostas que devem ser tratadas pelo programa, como os intervalos que tendem ao infinito, para a direita, para a esquerda ou em ambas as direções, além da necessidade de representação para conjuntos vazios. As soluções adotadas para resolver estes casos foram: usar a letra minúscula  $i$  para representar o infinito e  $[0, 0]$  para conjuntos vazios.

#### 4.4.3 Tratamento para respostas na forma de expressões algébricas

Quando um exercício requer uma resposta na forma de expressão algébrica alguns cuidados são necessários. Em primeiro lugar é preciso verificar se a sintaxe utilizada na formação

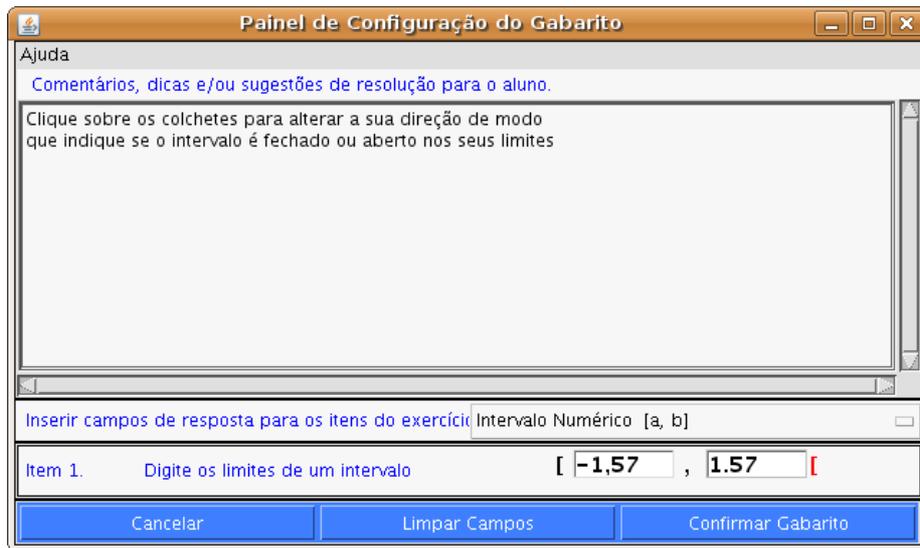


Figura 4.9: Configuração do gabarito: intervalo numérico

da mesma é aceita pelo sistema de avaliação automática. Caso a sintaxe utilizada pelo aluno tenha erro, o sistema lança uma mensagem indicando qual o erro detectado e em que campo ocorreu.

É preciso ainda garantir que expressões matematicamente equivalentes sejam avaliadas como respostas corretas. Por exemplo, se a resposta pré-definida de um exercício é  $x^2 + 2x$ , o sistema tem que aceitar a resposta  $x * x + x + x$  e também a resposta  $x^2 + 2*x$ . Para que isso seja possível, as expressões (resposta do aluno  $A(x)$  e resposta do professor  $P(x)$ ) são avaliadas (e comparadas) numericamente para um conjunto finito de valores (limitados pela porção visível do eixo  $x$ ) definido pelo professor no momento da inserção do exercício no sistema.

A necessidade de aceitação de expressões equivalentes sugere que, mesmo para esta modalidade de respostas é necessário o estabelecimento de um valor de erro máximo tolerável impondo algum nível de flexibilidade ao sistema, já que a avaliação de expressões distintas, embora equivalentes, pode resultar em *falso-erro* por motivo de precisão.

Adotou-se então, como solução para este problema, admitir um erro máximo de 5%, ou seja, se  $A(x)$  for comparado a  $P(x)$   $n$  vezes, em 5% das comparações o erro poderá ser maior que o valor  $\varepsilon$  (definido na seção 4.4.1).

As questões que solicitam esboço de gráficos como resposta, para efeito de tabulação

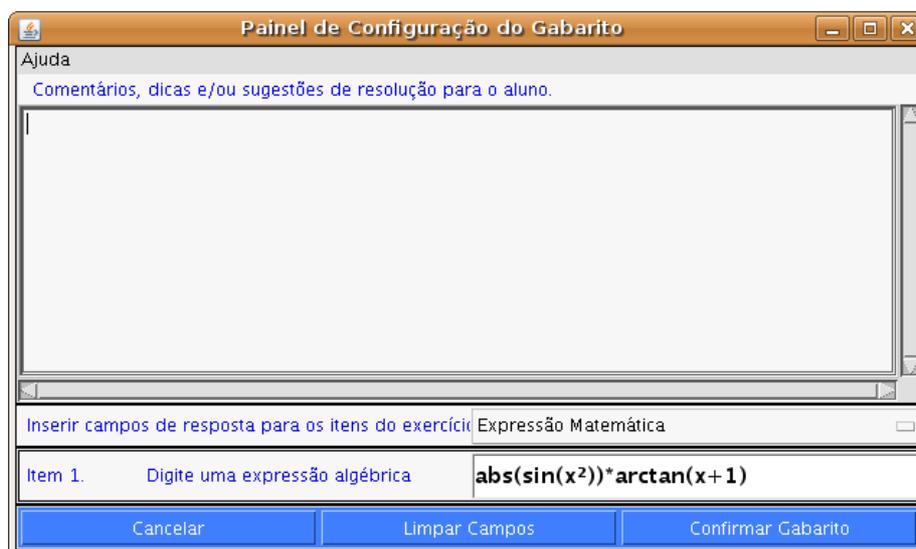


Figura 4.10: Expressões

dos dados, foram enquadradas nessa categoria pois, no iGraf, os gráficos só podem ser desenhados a partir de expressões matemáticas. No entanto, devido às características da implementação computacional, o envio de respostas na forma de gráfico será tratado na seção 4.4.4

#### 4.4.4 Tratamento para outros tipos de respostas

As questões de múltipla escolha (incluindo as de verdadeiro ou falso) podem ser facilmente adaptadas se forem usados números para especificar as opções disponíveis. Restam, no entanto, as questões discursivas, ou seja, aquelas cujas respostas devem ser dadas na forma de texto, para as quais a avaliação automática não se adapta muito bem.

Este tipo de resposta, no iGraf, pode servir aos mais diversos propósitos, desde demonstrações formais e justificativas até a troca de observações entre alunos e professor. Supondo que fossem usadas apenas para envio de informação matemática, ainda assim, seria bastante difícil criar uma maneira para automatizar a avaliação.

Devido a imprevisibilidade da expressão humana, um sistema que se propusesse a avaliar automaticamente questões com respostas discursivas deveria ter características de reconhecimento da linguagem natural, provavelmente baseadas em Redes Neurais Artificiais; pelo fato de estar além do escopo deste trabalho, tal abordagem não será discutida.

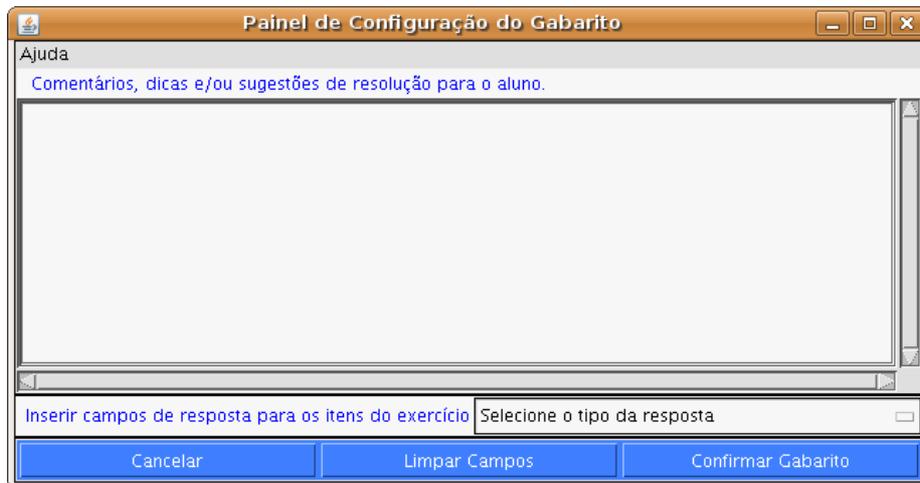


Figura 4.11: Gabarito vazio: envio de respostas discursivas

Conforme mencionado na seção 4.4.3, as respostas na forma de gráficos foram tabuladas junto com aquelas que requerem expressões algébricas, mas são descritas nesta seção devido às suas características peculiares.

Questões criadas com o iGraf e que pedem o esboço de gráficos podem ser simples e, portanto, respondidas apenas com uma expressão. Porém, o programa permite a criação de questões mais sofisticadas, que podem ser formuladas com o objetivo de ilustrar ou expressar um conceito, uma tendência ou uma característica que só fica bem definida pela sobreposição de gráficos. É possível solicitar, por exemplo, que o aluno encontre todas as retas tangentes à curva descrita pela função  $f$  e que se interceptam em um ponto determinado ou que sejam paralelas a uma reta dada. Provavelmente, receber expressões como resposta não dará uma idéia clara ao professor do trabalho do aluno... seria melhor “ver” este trabalho. Por esse motivo, o envio de gráficos como resposta é tratado como uma questão discursiva.

Tais respostas não serão avaliadas automaticamente, como as demais citadas, mas poderão ser registradas pelo mesmo SGC que recebe os outros tipos de resposta e, portanto, tornadas disponíveis para a análise do professor em momento oportuno.

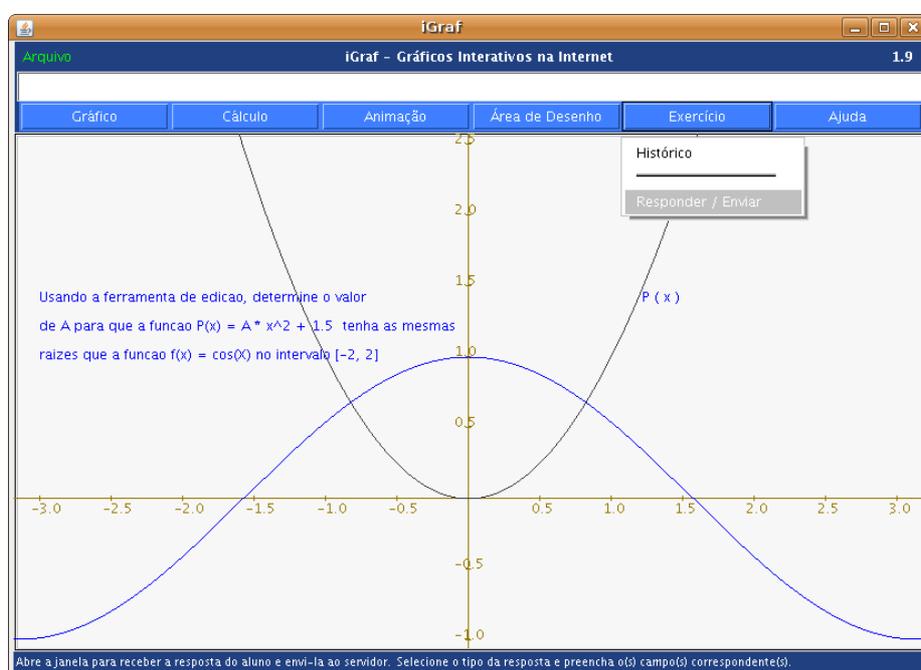


Figura 4.12: Opção de resposta de exercícios



# Capítulo 5

## Análise do iGraf

O projeto cuidadoso da interface gráfica com o usuário (*Graphical User Interface - GUI*) é parte essencial do projeto de software. É de fundamental importância que a GUI seja projetada para aproveitar as habilidades e experiências dos usuários do programa de modo a maximizar a produtividade e minimizar os erros. (35)

Neste capítulo, o iGraf é analisado a partir de dois diferentes pontos de vista: um primeiro teórico que utiliza um conjunto de heurísticas que induzem à reflexão sobre o projeto de sua interface gráfica e outro prático baseado em um experimento realizado no laboratório de informática do CEC-IME-USP com professores da rede pública do Estado de São Paulo.

### 5.1 Análise da Interface Gráfica

Muitos dos chamados ‘erros de usuário’ são causados por interfaces gráficas mal projetadas. Como consequência de um projeto gráfico ruim os usuários podem ser incapazes de acessar algumas características do sistema e sentir que o mesmo cria limitações e problemas ao invés de resolvê-los.(35) Esse tipo de situação precisa ser evitada em sistemas criados para apoiar o ensino.

Assim, a análise abaixo visa detectar possíveis defeitos que podem ter sido inadvertidamente introduzidos na GUI do iGraf, para corrigir esses defeitos rapidamente, se possível, ou propor uma solução a ser implementada no futuro. Tal análise se baseia em um conjunto de heurísticas desenvolvidas pelo engenheiro dinamarquês Jakob Nielsen.

### 5.1.1 Heurísticas de Nielsen

Defendendo tese sobre a interação homem-máquina, Nielsen concluiu doutorado na Universidade Técnica da Dinamarca. Trabalhou na Sun Microsystems – criadora da linguagem Java – de 1994 a 1998 onde foi responsável pelo desenvolvimento do projeto de usabilidade das páginas *web* da empresa. Atualmente, Nielsen é diretor da Nielsen Norman Group, uma empresa que presta serviços de consultoria sobre design de produtos e serviços para empresas que os comercializam pela Internet<sup>1</sup>.

Em 1990, Nielsen apresentou na ACM CHI 90 – *Human Factors in Computing Systems Conference*<sup>2</sup> um trabalho que se tornou referência para o desenvolvimento de interface gráfica e usabilidade. O artigo, *Heuristic evaluation of user interfaces*, escrito em parceria com Rolph Molich apresentava uma lista de dez heurísticas para a análise do projeto de interface gráfica de um *software*, as quais estão descritas abaixo junto com observações que procuram mostrar o grau de concordância do iGraf com cada tópico:

#### 1. Visibilidade do estado do sistema

*O sistema deve sempre manter o usuário informado sobre o que está acontecendo por meio de mensagens periódicas.*

Durante o uso do iGraf, o usuário é constantemente informado sobre o que *pode* acontecer caso clique em uma determinada região. O programa conta com uma barra de mensagens que indica ao usuário a função de cada um dos comandos dos menus sempre que o *mouse* é “posicionado” sobre eles. Cada vez que o cursor é movimentado para outra área, a mensagem é atualizada. Assim, o processo de informação é contínuo e não facultativo, embora também não seja invasivo, pois não interfere no uso de qualquer funcionalidade do programa.

#### 2. Concordância entre o sistema e o mundo real

*O sistema deve ‘falar’ a língua do usuário, com palavras, frases e conceitos familiares a ele ao invés de usar termos técnicos. Seguir convenções do mundo real, apresentando a informação em uma ordem natural e lógica.*

Pelo fato de ser um programa voltado para o ensino de função, a terminologia uti-

---

<sup>1</sup>Biografia de Jakob Nielsen: [www.useit.com/jakob/](http://www.useit.com/jakob/)

<sup>2</sup>[www.interaction-design.org/references/conferences/](http://www.interaction-design.org/references/conferences/)

lizada é muito bem definida e concordante com o que os usuários – alunos e professores – encontram nos tradicionais livros-texto e outras publicações sobre o tema:

### 3. Controle do usuário e liberdade

*Usuários frequentemente escolhem funções do sistema por engano e necessitam de uma ‘saída de emergência’ para corrigir o erro sem ter que ler uma longa orientação. O sistema deve ter um mecanismo do tipo ‘desfazer/refazer’*

Tal mecanismo não está implementado no iGraf na versão atual e poderá ser acrescentado em versões futuras.

### 4. Consistência e padrões

*Usuários não devem ter que adivinhar que diferentes palavras, situações ou ações significam a mesma coisa. As convenções da plataforma devem ser seguidas.*

O iGraf é escrito em linguagem Java e, portanto, independente de plataforma. Segue, no entanto, o padrão de interação *WIMP* – *Window, Icon, Menu and Pointer*, que se tornou popular a tal ponto que o usuário médio de computador necessita apenas de orientações básicas e dependentes de contexto para interagir com os elementos de uma GUI.

### 5. Prevenção de erros

*Melhor que boas mensagens de erro é um projeto cuidadoso que previne um problema antes que ocorra. Devem ser eliminadas operações propensas a erros ou, na impossibilidade de tal eliminação, devem ser exibidas mensagens solicitando confirmação da ação.*

As operações realizadas pelos usuários do iGraf são simples e não têm potencial para causar problemas sérios como grandes perdas de dados e outros. Ainda assim, o projeto conta com alguns elementos de prevenção de erros. Um exemplo é o uso de vírgula como separador decimal. Ao invés de orientar o usuário a não utilizá-las, caso sejam digitadas o programa as substitui por ponto, internamente, de modo totalmente transparente ao usuário.

### 6. Reconhecimento ao invés de lembrança

*A carga cognitiva do usuário deve ser minimizada tornando-se objetos, ações e opções visíveis. O usuário não deve ter que se lembrar de informações usadas*

*em outras partes do programa. Informações sobre o uso do sistema devem estar facilmente acessíveis.*

O iGraf não usa ícones (desenhos que representam ação) para indicar funcionalidades e sim palavras logicamente agrupadas em listas, assim, o usuário pode se orientar sobre o uso do programa apenas lendo os nomes dos comandos.

#### **7. Flexibilidade e eficiência de uso**

*Atalhos (ou teclas aceleradoras), pouco notados por usuários novatos, porém, podem melhorar o desempenho na interação dos usuários experientes, assim o sistema pode ser adaptado ao uso de ambos os grupos. As ações mais frequentes devem poder ser configuradas pelo usuário.*

As teclas aceleradoras ainda não foram implementadas no iGraf, mas esta é uma situação que estará resolvida até a entrega da versão final deste trabalho.

#### **8. Estética e projeto *minimalista***

*Diálogos não devem conter informações irrelevantes ou que sejam usadas muito raramente.*

O elemento principal de diálogo com o usuário é a barra de mensagens (parte inferior da tela) que, até por restrição de espaço, só exibe mensagens curtas, simples e objetivas.

#### **9. Ajuda ao usuário no reconhecimento de erros**

*Mensagens de erro devem ser escritas em linguagem simples, indicar precisamente o problema e sugerir uma solução.*

Os erros que os usuários podem cometer no iGraf estão ligados, principalmente, a digitação de expressões matemáticas; veja abaixo dois exemplos.

Quando ocorre um erro na entrada da expressão para o desenho de gráfico – e esse tipo de erro é comum – é exibido na barra de mensagem um alerta que indica qual foi o primeiro caractere não aceito pelo sistema. O usuário, ao ver que o gráfico não foi desenhado, naturalmente busca corrigir a expressão. Como ele já sabe exatamente o que está errado, a correção se torna mais fácil e rápida.

Já na tela de configuração de gabarito, se o programa aceitar um valor com o tipo de dado incorreto, poderá comprometer o resultado da avaliação de, possivelmente, vários alunos. A verificação nesse caso é bem mais rigorosa e o professor é impedido

de finalizar o gabarito enquanto cada campo de entrada não tiver o tipo de dado correto. Por exemplo, um campo numérico não aceitará uma expressão e a cada tentativa de finalização do gabarito nova verificação será feita; caso existam erros, o sistema exibirá uma janela de diálogo indicando qual foi o erro e em que campo ocorreu. As mesmas regras de validação dos dados são aplicadas às respostas dos alunos.

#### 10. Ajuda e documentação

*Seria melhor se o programa pudesse ser usado sem qualquer subsídio, mas pode ser necessário fornecer ajuda e documentação. Esses textos devem poder ser consultados facilmente, estar orientados para a tarefa do usuário e listar passos concretos para a sua execução, além de não ser muito longos.*

O iGraf conta com uma ajuda que busca explicar exatamente como realizar as tarefas possíveis ao usuário e nada mais... os detalhes são omitidos para tornar a leitura simples e rápida. Caso deseje ou necessite de informações mais aprofundadas, o usuário poderá encontrá-las no manual *online* do programa<sup>3</sup>.

## 5.2 Teste no laboratório do CEC

Em janeiro de 2008, com o objetivo de verificar a usabilidade do iGraf, foi realizado um teste prático no laboratório de informática do CEC-IME-USP com professores de matemática dos ensinos Fundamental e Médio participantes do LEM – Laboratório de Ensino de Matemática, um curso de Difusão Cultural oferecido anualmente como parte do Programa de Verão no IME-USP.

Nessa oportunidade foram aplicadas algumas atividades que os professores deveriam realizar usando o programa. Em seguida, todos receberam um questionário onde puderam emitir o seu parecer sobre as características, funcionalidades e viabilidade de uso do programa em suas aulas. Seguem abaixo o detalhamento das atividades e do questionário, bem como os resultados obtidos com essa experiência.

---

<sup>3</sup>[www.matematica.br/igraf/manual](http://www.matematica.br/igraf/manual)

### 5.2.1 A aplicação das atividades

Para avaliar a usabilidade de um *software* é preciso, antes, especificar inequivocamente a que se refere esse termo. Segundo a NBR 9241, define-se como usabilidade

*“O conjunto de atributos que evidenciam o esforço necessário para se poder utilizar o software, bem como o julgamento individual desse uso, por um conjunto explícito ou implícito de usuários.”*

(10)

Os exercícios apresentados aos professores durante o teste tiveram essa definição como premissa, pois, o programa seria avaliado e não o nível de conhecimentos dos participantes. Assim, os enunciados foram escritos com a preocupação de induzir os professores ao uso e observação das funcionalidades do iGraf. As questões propostas foram deliberadamente simples do ponto de vista matemático para que a atenção dos professores fosse direcionada aos atributos do programa.

Inicialmente os participantes foram informados verbalmente de todos os procedimentos e do objetivo do teste. Como as atividades foram realizadas com o apoio do SAW (descrito na seção 3.1) e, portanto, *online*, os detalhes técnicos do ambiente de aplicação também precisaram ser abordados. Durante a aplicação das atividades, cada participante usou um computador para resolver os exercícios propostos individualmente. Todas as páginas com os enunciados dos exercícios propostos estão no Anexo 1.

### 5.2.2 O questionário pós-teste

Assim que terminaram suas atividades os participantes começaram a responder o questionário pós-teste. Foram elaboradas questões de múltipla escolha com o objetivo de minimizar a subjetividade das opiniões e facilitar a tabulação dos dados. Para fazer o levantamento das preferências dos participantes foi utilizada a escala proposta por Renis Likert (10) que consiste em disponibilizar cinco conceitos de múltipla escolha que variam conforme a tabela abaixo. Após a aplicação do questionário pós-teste foi feita a tabulação dos dados obtidos e, em seguida, foram gerados gráficos baseados na contagem do número de ocorrências de cada alternativa oferecida em cada questão.

Péssimo = 1	Ruim = 2	Regular = 3	Bom = 4	Ótimo = 5
-------------	----------	-------------	---------	-----------

Tabela 5.1: Níveis de satisfação propostos por Likert

### 5.2.3 Resultados da aplicação do questionário

Para simplificar a análise foi calculado um valor, aqui chamado de **coeficiente de satisfação** ( $S$ ), com o objetivo de quantificar, utilizando a tabela de Likert, o grau de contentamento dos professores com cada recurso que lhes foi apresentado.

$$S = \frac{10 * p_i}{P}, \quad 2 \leq S \leq 10 \quad (5.1)$$

onde  $p_i$  é o total de pontos obtidos para a questão de índice  $i$  e  $P = 125$ , o número máximo de pontos para cada questão, já que foram 25 os participantes do teste. A tabela abaixo mostra, como exemplo, o cálculo de  $p_2$ . Nesse caso,  $S = 8,2$ .

pontos	1	2	3	4	5	
freqüência	0	0	1	20	4	total
freqüência * pontos	0	0	3	80	20	103

Tabela 5.2: Cálculo dos pontos obtidos na Questão 2

Seguem, nas próximas páginas, o enunciado e o objetivo de cada questão do questionário pós-teste, além dos gráficos criados a partir dos dados obtidos e uma breve análise dos resultados.

**Questão 01.** Com que frequência você usa o computador? (independente da finalidade).

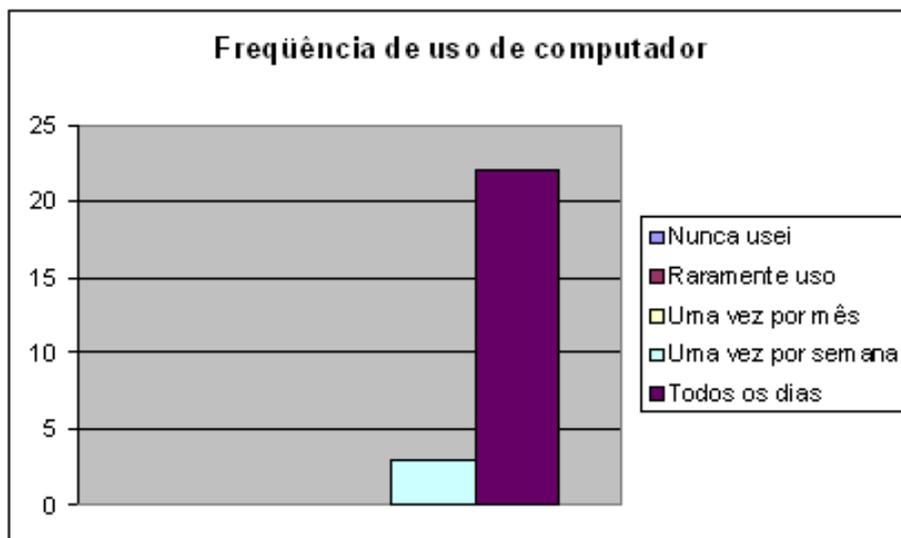


Figura 5.1: Frequência de uso do computador

**Objetivo:** Identificar se as dificuldades com o uso do programa são decorrentes de inexperiência com o uso da máquina.

**Coefficiente de Satisfação:** Não se aplica

**Análise do resultado:** Observando-se o gráfico, é possível notar que a maioria das pessoas que participaram do teste são usuários frequentes de computador e, em decorrência desse fato, que as dificuldades eventualmente apresentadas por essas pessoas, provavelmente, não estão ligadas à inexperiência com o uso do equipamento.

**Questão 02.** De acordo com o aspecto visual, o iGraf lhe parece um programa:

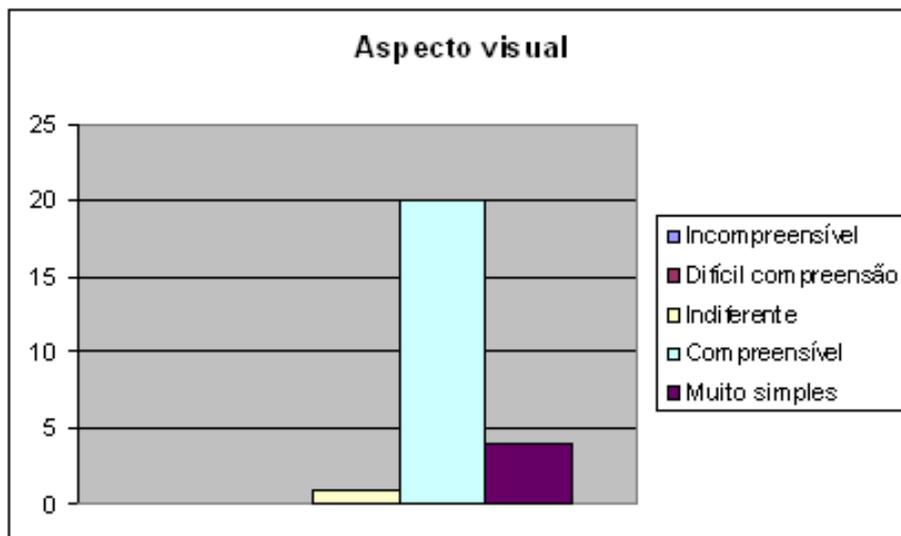


Figura 5.2: Opiniões sobre o visual do programa

**Objetivo:** Verificar se a aparência transmite ao usuário a idéia de que se trata de um programa de fácil compreensão.

**Coefficiente de Satisfação:** 8,2

**Análise do resultado:** O gráfico mostra que não houve opiniões negativas sobre o aspecto visual, e que, além disso, houve uma maior concentração de opiniões em torno da opção *compreensível*. A observação direta do gráfico permite concluir que o aspecto visual do programa foi aprovado pelos professores que participaram do teste.

**Questão 03.** Na sua opinião, para quem tem um primeiro contato, começar a usar o programa é:

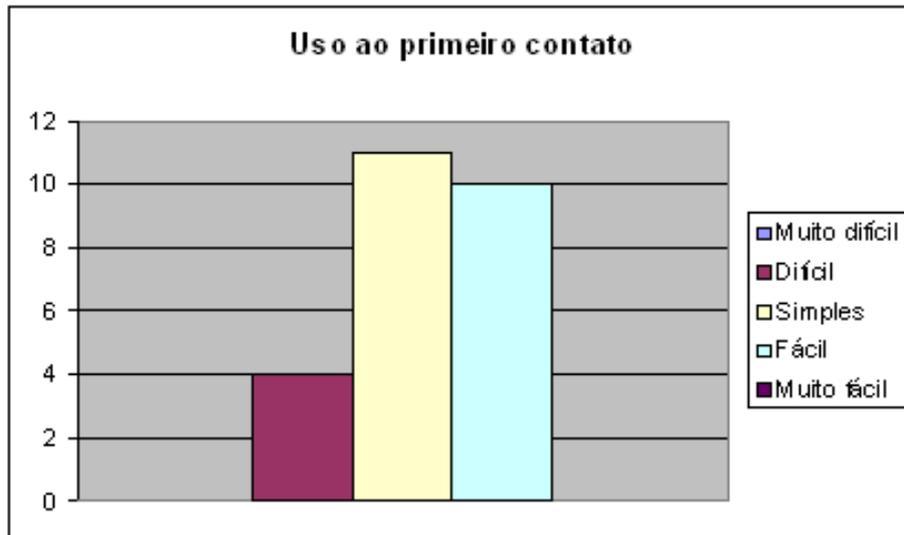


Figura 5.3: Opiniões sobre a facilidade de uso inicial

**Objetivo:** Descobrir o quão confortáveis se sentem as pessoas ao usar o programa pela primeira vez.

**Coefficiente de Satisfação:** 6,5

**Análise do resultado:** Ocorreram algumas opiniões negativas sobre a facilidade no uso inicial, porém essa foi a opção de um percentual pequeno, apenas 16% das pessoas. O baixo coeficiente de satisfação também aponta que melhorias devem ser feitas para tornar mais simples o primeiro contato do usuário com o programa. A maioria das pessoas, no entanto, considerou o uso do programa de *simples a fácil* já na primeira experiência.

**Questão 04.** O gráficos desenhados pelo iGraf:

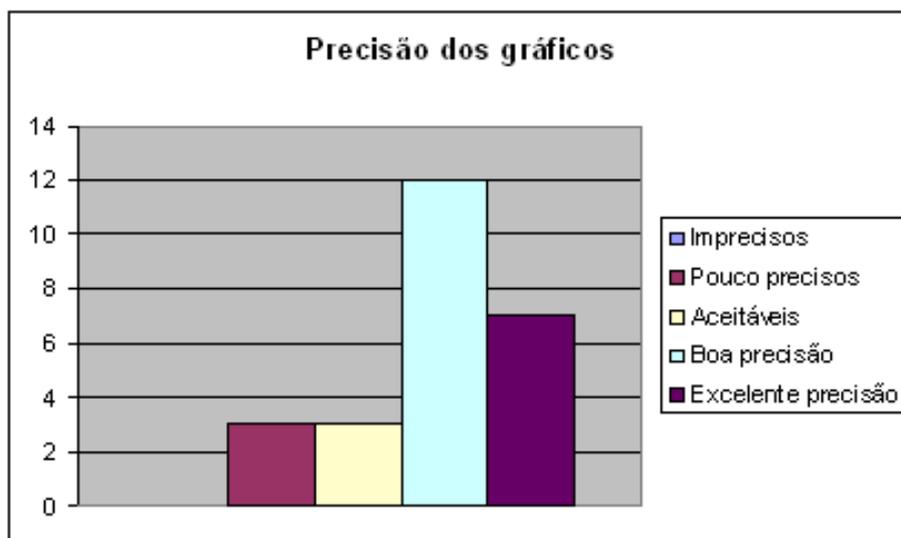


Figura 5.4: Opiniões sobre a qualidade dos gráficos

**Objetivo:** Verificar qual é a impressão que os gráficos criados pelo programa causam nos participantes do teste.

**Coefficiente de Satisfação:** 7,8

**Análise do resultado:** Um percentual elevado, 76% dos participantes acredita que a precisão dos gráficos é mais que aceitável, embora 12 % dos professores que participaram do teste acreditem que os gráficos gerados pelo programa são pouco precisos. Baseado na opinião dominante entre os professores que fizeram o teste, pode-se afirmar que o nível de precisão dos gráficos foi considerado suficientemente bom.

**Questão 05.** Os itens dos menus do iGraf pretendem ser auto-instrutivos. Na sua opinião eles:

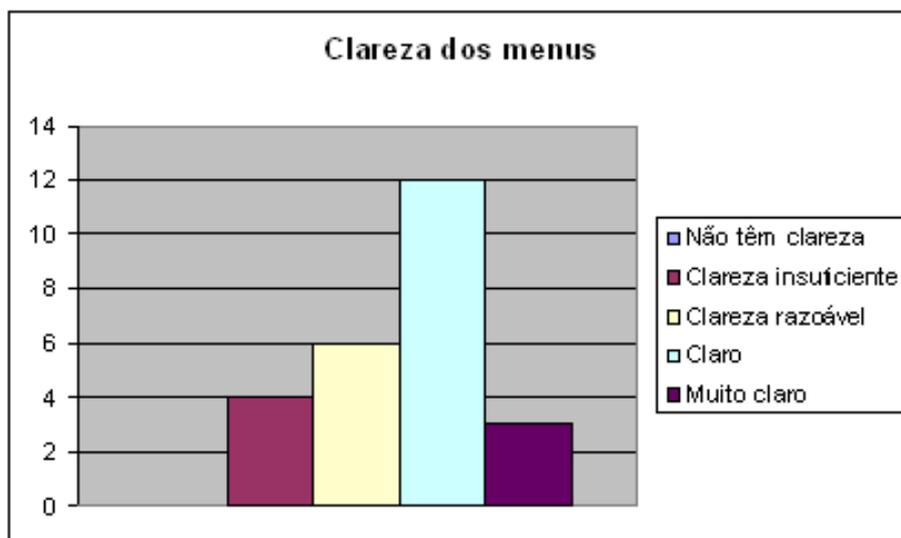


Figura 5.5: Opiniões sobre o texto do menu

**Objetivo:** Verificar se o texto dos itens dos menus é suficientemente claro.

**Coefficiente de Satisfação:** 7,1

**Análise do resultado:** O texto dos itens de menu foi considerado *claro* ou *muito claro* por 60% dos participantes. No entanto, existiu a reprovação de 16% das pessoas, por acreditarem que o texto não é claro o suficiente. Mais uma vez, a maioria das opiniões foi favorável, indicando que, embora haja correções a fazer, a implementação atual do menu possui um texto escrito de forma clara o suficiente para a maior parte das pessoas.

**Questão 06.** A quantidade de ferramentas / funcionalidades oferecidas é:

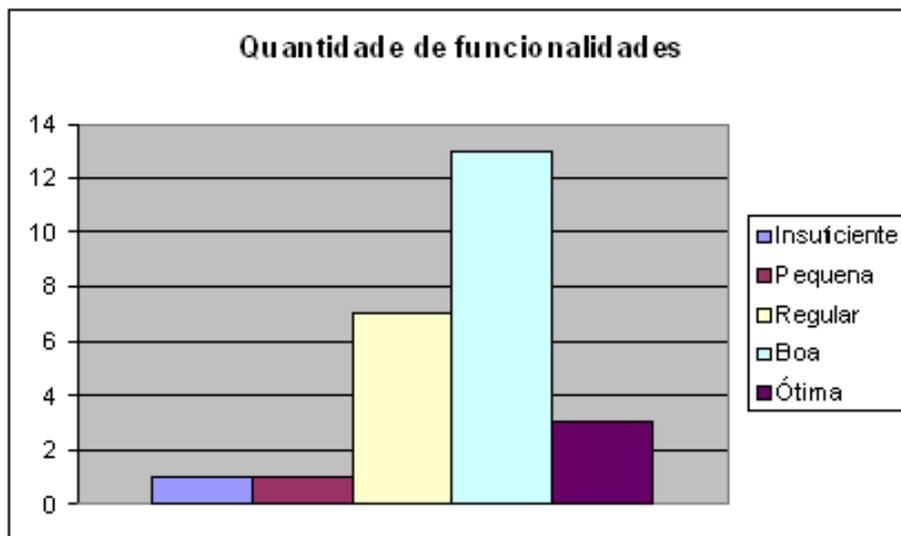


Figura 5.6: Opiniões sobre a quantidade de ferramentas

**Objetivo:** Descobrir se existe alguma demanda para a inclusão de funcionalidades que possam melhorar a experiência do professor no ensino de função com o uso do programa.

**Coefficiente de Satisfação:** 7,3

**Análise do resultado:** Pensando em usar o iGraf para o estudo ou ensino de função, 70% dos participantes afirmaram que a quantidade de funcionalidades do programa é *boa* ou *ótima*. Daí se pode concluir que a quantidade de ferramentas do iGraf é adequada ao ensino de função, embora ainda haja possibilidades de incrementos nessa área.

**Questão 07.** O uso da janela de envio de respostas te pareceu:

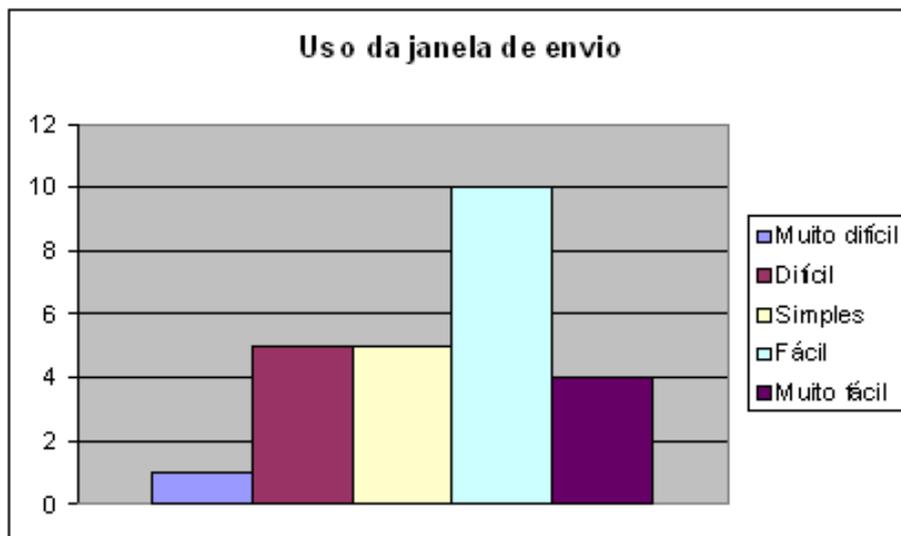


Figura 5.7: Opiniões sobre a janela de envio

**Objetivo:** Verificar se o uso da janela de envio de respostas é simples o suficiente.

**Coefficiente de Satisfação:** 6,9

**Análise do resultado:** Embora 76% das pessoas tenham achado pelo menos *fácil* usar a janela de envio de respostas, esse foi o item que teve a maior reprovação, com 24% das pessoas achando seu uso *difícil* ou *muito difícil*. Esses dados indicam que essa característica do programa pode e deve ser melhorada em versões futuras.

**Questão 08.** A barra de mensagens esclareceu suas dúvidas sobre o funcionamento de alguma opção?

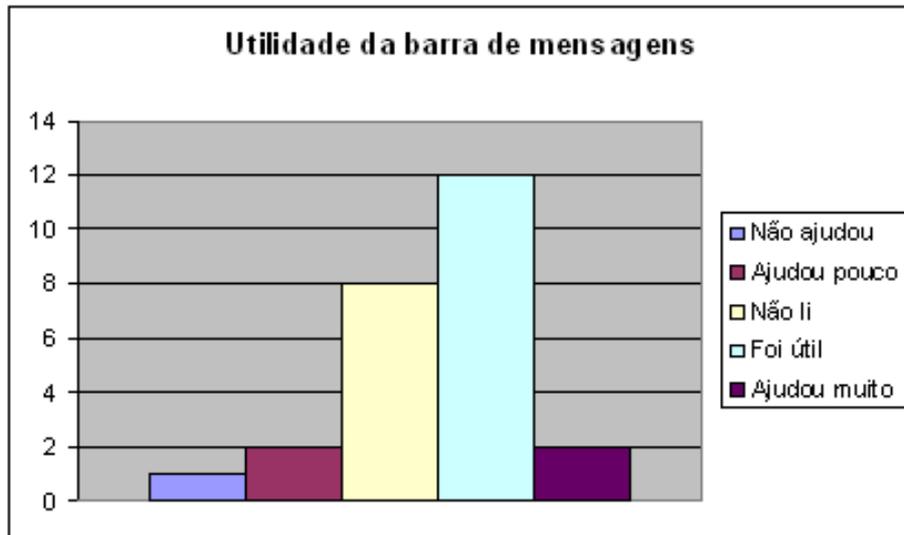


Figura 5.8: Opiniões sobre a barra de mensagens

**Objetivo:** Obter informações sobre a utilidade da barra de mensagens.

**Coefficiente de Satisfação:** 7,0

**Análise do resultado:** A leitura das informações registradas no gráfico acima é um pouco mais delicada devido à existência da opção *não li*, respondida por 32% dos participantes. Para interpretar o significado desse percentual, foi feito um cruzamento de informações e constatou-se que a maioria daqueles que não leram a barra de tarefas também acharam o uso do programa *razoavelmente simples* ou *fácil*. Como o percentual de reprovação também foi baixo, 7,5% apenas, conclui-se que a barra de mensagens foi útil na maioria dos casos em que foi efetivamente usada.

**Questão 09.** A ajuda do programa tornou seu uso mais fácil?

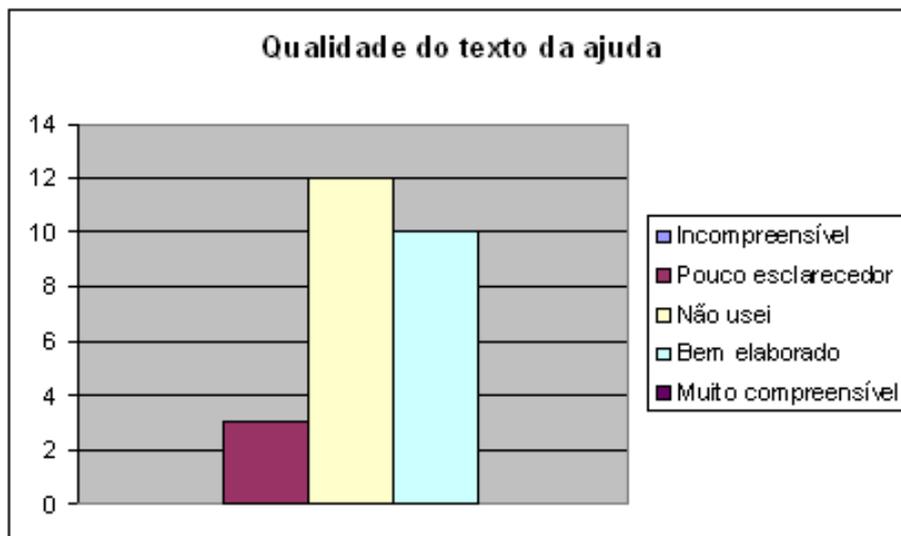


Figura 5.9: Opiniões sobre a ajuda do programa

**Objetivo:** Obter informações sobre a qualidade dos textos de ajuda.

**Coefficiente de Satisfação:** 6,6

**Análise do resultado:** Diferentemente da barra de mensagens que fica exposta a todos os usuários, o texto da ajuda só pode ser acessado clicando-se no menu de mesmo nome. Analisando, então, as opiniões dos que leram a ajuda é possível constatar uma reprovação de 23% contra uma aprovação de 77%, o que mostra que a ajuda também foi bem avaliada no teste prático.

**Questão 10.** Se você tivesse que usar um visualizador gráfico com seus alunos, você:

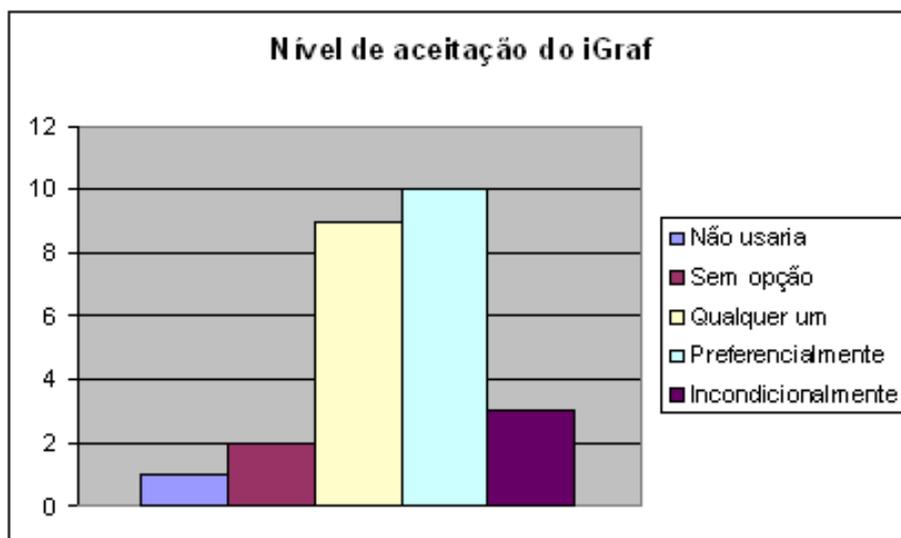


Figura 5.10: Nível de aceitação do iGraf

**Objetivo:** Descobrir se o professor acha que o uso do iGraf em sala de aula é plausível.

**Coefficiente de Satisfação:** 7,0

**Análise do resultado:** Mais da metade dos professores (52%) afirmaram que optariam pelo uso preferencial do iGraf em sala de aula. A reprovação incondicional, por sua vez, foi muito baixa. Apenas 2,5% dos participantes afirmam que não usariam o programa em hipótese alguma. Daí, é possível concluir que o programa teve boa aceitação entre os participantes do teste.

Além das questões objetivas, o questionário pós-teste contou ainda com três questões discursivas. Seguem abaixo o enunciado dessas questões e algumas frases escritas pelos participantes do teste. Nas frases, abaixo transcritas, foi mantida a grafia original.

**Questão 11.** O iGraf apresenta alguns recursos que não são comuns em outros visualizadores. Você observou algum desses recursos? Se sim, por gentileza, faça um breve comentário dizendo por que gostou ou não. Caso nunca tenha usado outro visualizador gráfico, desconsidere essa questão.

“A opção ‘exercícios’, pois facilita o EAD. E as ferramentas do cálculo, pois, apesar de aparecerem em outros visualizadores, são de fácil utilização.”

“A animação é interessante. O editor é bom, porém apresentou falhas ao desenhar todas”

“Gostei recursos animação e área de desenho são recursos diferenciados”

**Questão 12.** O iGraf está em fase de testes. Você detectou algum erro ou comportamento desagradável do programa. Se sim, por gentileza, descreva-o.

“Falta um botão de desfazer e refazer a última opção”

“Às vezes não selecionava a função ou não a desenhava”

“Sim, travava com frequência...”

**Questão 13.** Caso queira fazer algum elogio, comentário, crítica ou sugestão utilize as linhas abaixo.

“Gostei bastante do iGraf, usarei constantemente.”

“É um programa a ser investido”

“Não tive contato com outros visualizadores, mas achei o iGraf de fácil visualização e operação.”

“Gostaria que as raízes das funções fossem calculadas...”

### **5.2.4 Conclusões do capítulo**

Nesse capítulo foram apresentados dois métodos para avaliar as características do iGraf. Na análise pelas heurísticas de Nielsen foi possível observar que a implementação atual do iGraf tem algumas falhas, porém está de acordo com a maioria dos tópicos. Foram apresentadas ainda as questões utilizadas e também a estratégia para a coleta de dados realizada durante um experimento prático no laboratório do CEC-IME-USP. Foi feita uma análise desses dados que indicou a existência de pontos que devem ser melhorados no iGraf e que, em detrimento desses pontos, o programa foi bem aceito pelos professores participantes do teste prático.



# Capítulo 6

## Conclusões

A partir dos anos 90, a Internet passou a ser fortemente utilizada no ensino/aprendizagem, sendo o incremento nos últimos anos particularmente empurrado pelo surgimento de sistemas gerenciadores de cursos (SGC). Estes sistemas propiciaram o acompanhamento remoto (ou não), por parte dos professores, de atividades desenvolvidas pelos alunos, facilitando ao aluno seguir seu próprio ritmo de aprendizagem. Atualmente novas gerações de ferramentas facilitadoras têm surgido, como aquelas que permitem a incorporação de repositório de componentes didáticos e, também, ferramentas para autoria e para a avaliação automática de exercícios. Estas últimas, além de reduzir bastante as tarefas a serem executadas pelo professor, ajudam na aprendizagem do aluno por fornecer-lhe imediatamente uma avaliação sobre sua resposta.

Em relação ao ensino e a aprendizagem de tópicos ligados a funções, notou-se a inexistência de sistemas que pudessem ser integrados a SGC's e utilizados diretamente em navegadores *Web*. Desse modo, a proposta deste trabalho foi desenhar e implementar um sistema que, além de preencher essa lacuna, possibilitasse a autoria e a avaliação automática de exercícios.

O sistema derivado deste trabalho, o iGraf, teve seu núcleo desenvolvido visando que o mesmo pudesse ser utilizado em atividades didáticas do ensino médio e do início do superior. Dois de seus mais importantes requisitos não-funcionais são a facilidade para integrá-lo a SGC's, permitindo a apresentação e o envio de conteúdos, além de poder ser utilizado como um aplicativo local, na máquina do usuário, sem estar conectado à Internet.

O modelo de exercícios proposto, e já implementado no iGraf, foi baseado em uma

pesquisa em livros didáticos e sítios da Internet. Neste estudo foram identificados três (3) tipos de respostas a exercícios envolvendo funções: uma resposta que é um intervalo, um ponto do domínio ou uma expressão representando alguma função.

Algumas versões do iGraf foram testadas como Módulos de Aprendizagem (MA) de um SGC. O programa já pode ser livremente acessado (ou descarregado) a partir do endereço [www.matematica.br/igraf](http://www.matematica.br/igraf). Nessa página da Internet, também é possível ao usuário ler um manual no qual são descritas, detalhadamente, todas as funcionalidades do programa.

## 6.1 Contribuições

O iGraf, um dos resultados desta dissertação, apresenta uma série de funcionalidades que são comuns à maioria dos sistemas analisados. Além disso, durante os estudos que resultaram nesta dissertação, observou-se a possibilidade de acrescentar ao programa algumas características que poderiam ser facilitadoras do ensino e/ou da aprendizagem de tópicos ligados a funções matemáticas. A tabela abaixo destaca os principais recursos do iGraf fazendo um paralelo com os sistemas similares estudados:

Programa	AA	Anim	Calc	Com	Inter	Pág	Hist	Web	Licença
iGraf	x	x	x	x	x	x	x	x	gratuito
Graphmatica			x						shareware
Winplot		x	x		x				gratuito
Maple	x	x	x	x	x	x		x	comercial
Mathematica	x	x	x	x	x	x		x	comercial
Sage			x					x	gratuito
Scilab		x	x						gratuito
Maxima			x						gratuito
Wessa			x	x				x	GPL
WebGraphing			x					x	comercial
Relplot			x					x	gratuito
QuickMath			x					x	gratuito

Tabela 6.1: Recursos apresentados por alguns sistemas visualizadores gráficos

Na tabela 6.1, a coluna “AA” aponta os programas que possuem recursos para autoria e validação automática de exercícios. A coluna “Anim” destaca aqueles que permitem ao usuário visualizar animações sobre gráficos de funções. A coluna “Calc” mostra os pro-

gramas que têm implementadas ferramentas do Cálculo Diferencial e Integral. Na coluna “Com” estão aqueles que possuem recursos de comunicação. A coluna “Inter” indica os visualizadores que permitem ao usuário interagir com os gráficos alterando valores dos parâmetros das funções. A coluna “Pág” indica os programas que possuem a capacidade de gerar páginas da Internet para publicação de material instrucional. A coluna “Hist” indica programas que podem gerar registros sobre a atividade do usuário. Note que, nesse quesito, o iGraf é o único dentre os programas mencionados. Na coluna “Web” são apontados os programas que permitem o seu uso irrestrito diretamente em páginas *Web*. E, finalmente, a coluna “Licença” refere-se ao tipo de licença que cada um dos programas possui.

## 6.2 Trabalhos Futuros e em Andamento

O iGraf está em constante desenvolvimento e as possibilidades de aprimoramento desse programa são diversas. Algumas necessidades já foram reconhecidas e estão listadas abaixo, separadas em dois grupos: melhorias em andamento e trabalhos futuros.

### 6.2.1 Melhorias em Andamento

O desenvolvimento de sistemas computacionais geralmente segue a premissa: *primeiro fazer funcionar e, depois, fazer funcionar bem*. Com o iGraf, isso não foi diferente. Existem algumas rotinas implementadas que já funcionam corretamente e que, no entanto, podem ser melhoradas para que o programa se torne mais eficiente, ocupando menor quantidade de memória da máquina do usuário ou simplesmente realizando as mesmas operações mais rapidamente. Estão em andamento as seguintes melhorias:

- **Aprimoramento do sistema de desenho** - a estratégia atualmente adotada para a realização dos desenhos pelo iGraf é pouco eficiente. Por esse motivo estão sendo feitos estudos para a alteração das rotinas de geração e exibição de gráficos.

### 6.2.2 Trabalhos Futuros

Em discussões entre o autor desta dissertação e seu orientador e durante as seções de utilização do iGraf (uma delas está descrita na seção 5.2) com colegas professores e com

alunos do autor desta dissertação, algumas idéias foram sugeridas pelos participantes, falhas foram percebidas e, dessas observações e questionamentos, surgiram algumas propostas de trabalhos futuros. Dentre elas, destacam-se:

- **Uso em cursos reais** - o iGraf ainda precisa ser utilizado em situações didáticas reais. O programa poderia ser testado com diferentes grupos de alunos de dos níveis de ensino médio ou superior em disciplinas que possam se beneficiar da visualização de gráficos de função e da autoria e avaliação automática de exercícios.
- **Aprendizado cooperativo** - poderiam ser desenvolvidas novas funcionalidades com o objetivo de permitir a realização de atividades em grupo baseadas na colaboração entre os participantes. Para que isso fosse possível, seria necessária a criação de um modelo de interação que permitisse o trabalho sincronizado de várias pessoas sobre a mesma área de desenho. Tal modelo ainda não existe.
- **Introdução de novas funcionalidades** - a atual implementação do iGraf pode servir como ponto de partida para a criação de novas ferramentas que permitissem, por exemplo, o cálculo analítico de integrais, o desenho de curvas definidas com equações e cálculos associados ao ensino de Geometria Analítica, regressão linear e interpolação polinomial. A estrutura básica para a introdução dessas funcionalidades já existe. Outra possibilidade seria a visualização de gráficos tridimensionais, embora esse seja um objetivo um pouco mais difícil de ser atingido.

# Anexo 01

## Utilização do SAW

Seguem as imagens capturadas do sistema gerenciador de cursos SAW, ambiente no qual foram realizados os testes apresentados aos professores participantes do experimento no CEC-IME-USP

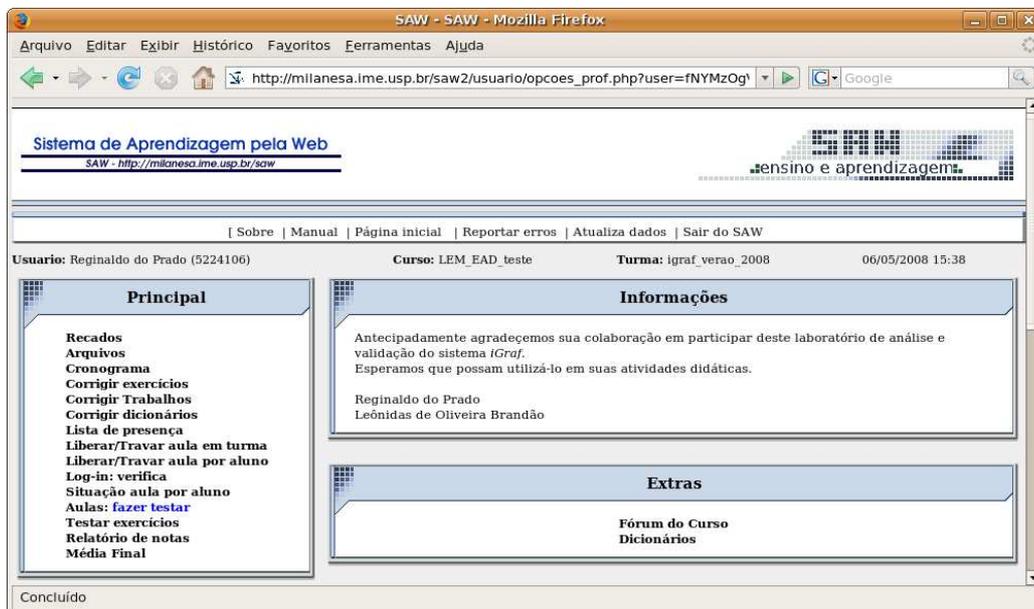


Figura 6.1: Tela de abertura do SAW

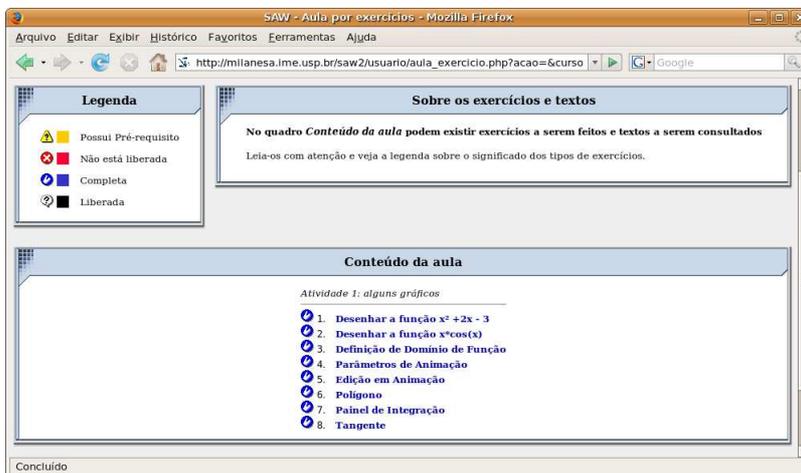


Figura 6.2: Lista de exercícios



Figura 6.3: Tela que precede o início do teste

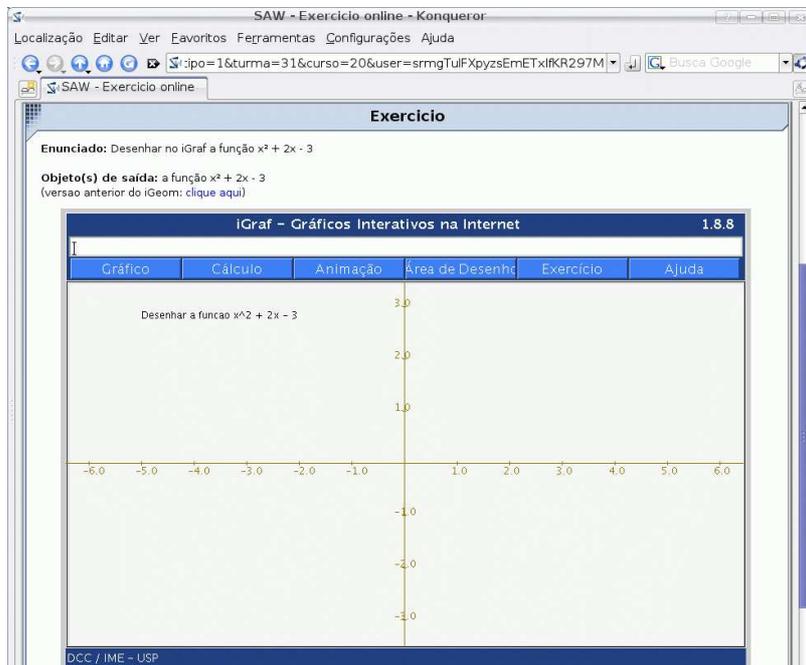


Figura 6.4: Exercício 01

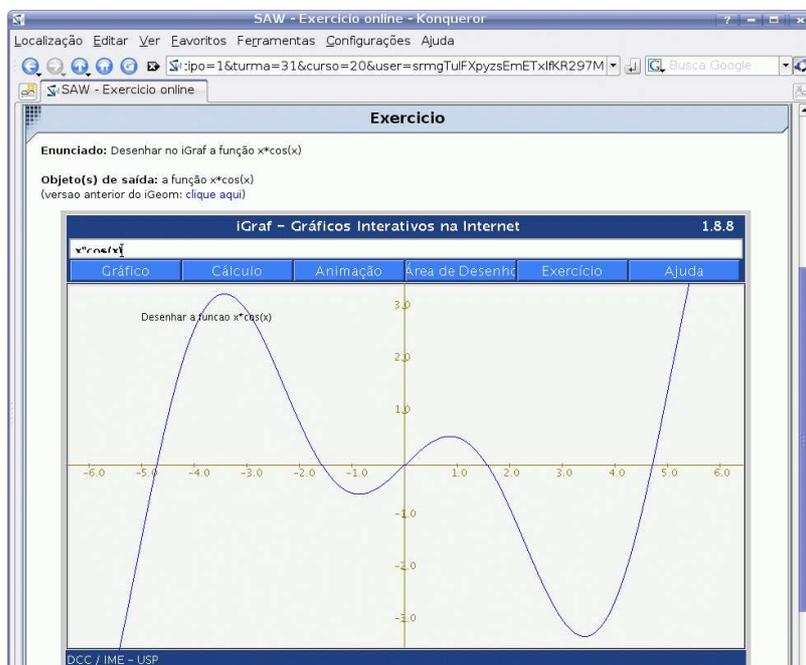


Figura 6.5: Exercício 02

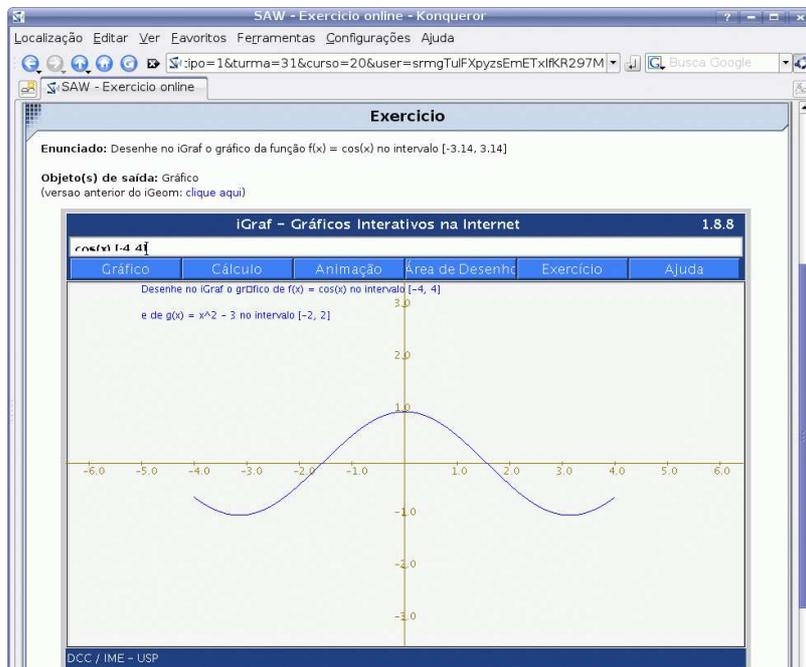


Figura 6.6: Exercício 03

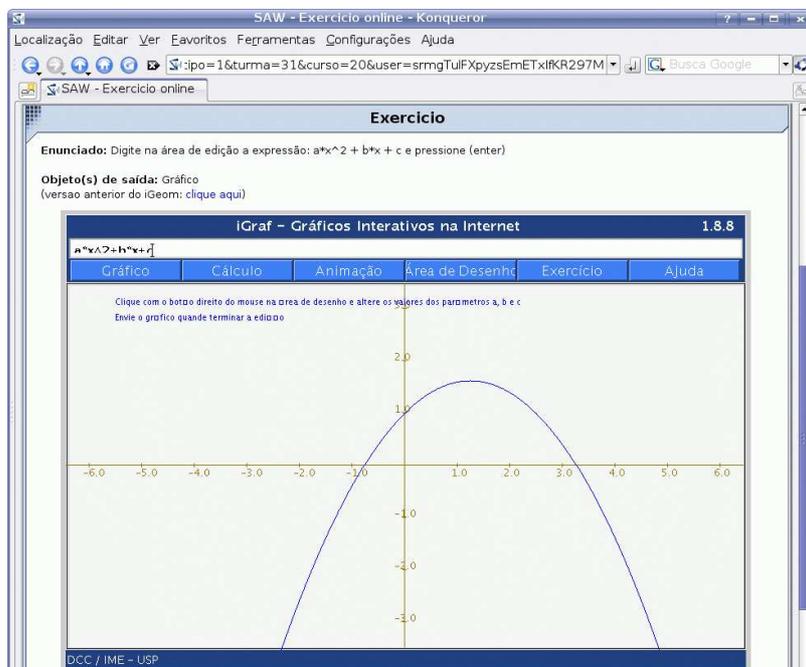


Figura 6.7: Exercício 04

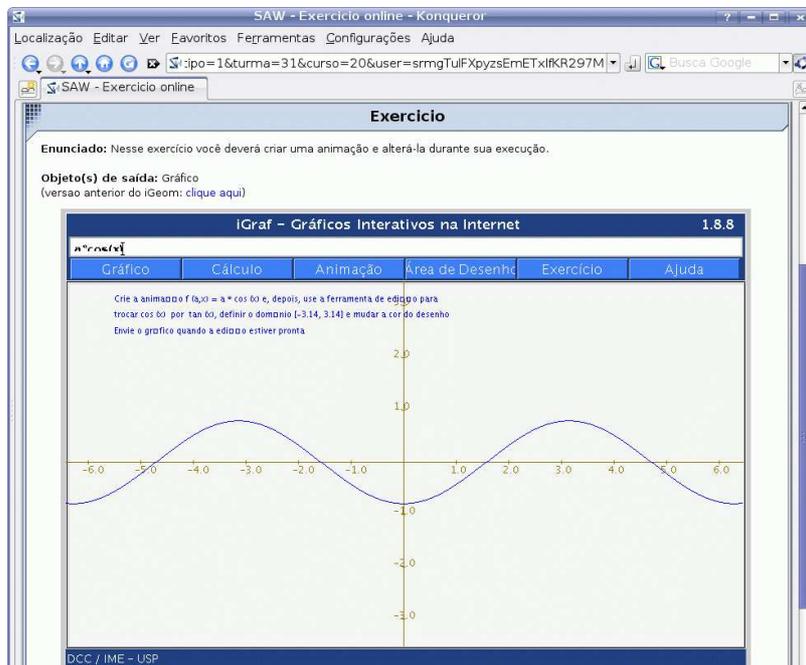


Figura 6.8: Exercício 05

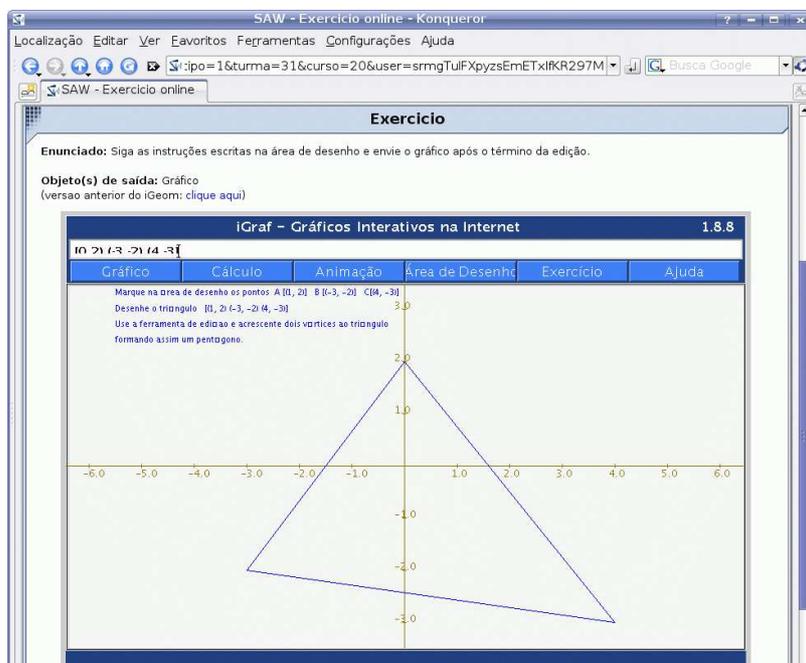


Figura 6.9: Exercício 06

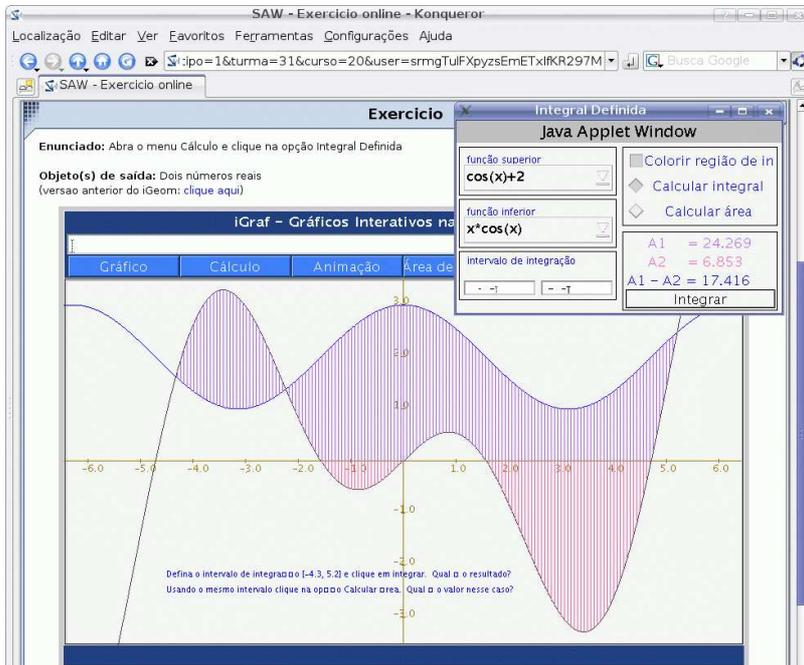


Figura 6.10: Exercício 07

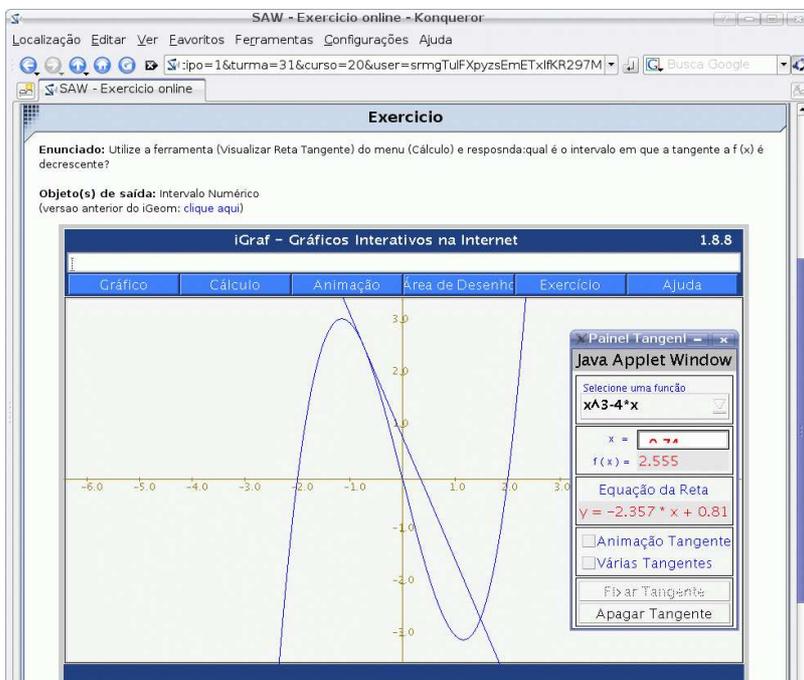


Figura 6.11: Exercício 08

## Anexo 02

### Dados obtidos nos questionários pós-teste

Tabela 6.2: Frequência de uso do computador

<b>Questão 01</b>	<b>Respostas</b>	<b>%</b>
Nunca usei	0	0
Raramente uso	0	0
Uma vez por mês	0	0
Uma vez por semana	0	0
Todos os dias	25	100,0

Tabela 6.3: Pelo aspecto visual, o programa é ...

<b>Questão 02</b>	<b>Respostas</b>	<b>%</b>
incompreensível	0	0
de difícil compreensão	0	0
indiferente ou não sei avaliar	1	4
facilmente compreensível	20	80
extremamente simples	4	16

Tabela 6.4: Começar a usar o programa é ...

<b>Questão 03</b>	<b>Respostas</b>	<b>%</b>
muito difícil	0	0
difícil	4	16
razoavelmente simples	11	44
fácil	10	40
muito fácil	0	0

Tabela 6.5: Os gráficos ...

<b>Questão 04</b>	<b>Respostas</b>	<b>%</b>
são imprecisos	0	0
são pouco precisos	3	12
têm precisão aceitável	3	12
têm boa precisão	12	48
têm precisão excelente	7	28

Tabela 6.6: Os menus ...

<b>Questão 05</b>	<b>Respostas</b>	<b>%</b>
não esclarecem sua função	0	0
deixam dúvidas sobre a sua função	4	16
indicam razoavelmente sua função	6	24
indicam bem sua função	12	48
indicam claramente sua função	3	12

Tabela 6.7: A quantidade de funcionalidades é ...

<b>Questão 06</b>	<b>Respostas</b>	<b>%</b>
insuficiente	1	4
pequena	1	4
regular	7	28
boa	12	48
excelente	3	12

Tabela 6.8: O envio de respostas é ...

<b>Questão 07</b>	<b>Respostas</b>	<b>%</b>
muito difícil	1	4
difícil	5	20
não usei esta funcionalidade	5	20
fácil	10	40
muito fácil	4	16

Tabela 6.9: A barra de mensagens foi útil?

<b>Questão 08</b>	<b>Respostas</b>	<b>%</b>
A barra não ajudou em nada	1	4
As mensagens da barra não são claras	2	8
Não li a barra de mensagens	8	32
A barra de mensagens foi útil em alguns momentos	12	48
A barra de mensagens resolveu todas as minhas dúvidas	2	8

Tabela 6.10: A ajuda facilitou o uso do programa?

<b>Questão 09</b>	<b>Respostas</b>	<b>%</b>
O texto da ajuda é incompreensível	0	0
O texto só esclarece alguns poucos pontos	3	12
Não usei a ajuda	12	48
O texto da ajuda é bem elaborado	10	40
O texto da ajuda eliminou todas as dúvidas	0	0

Tabela 6.11: Satisfação geral do usuário

<b>Questão 10</b>	<b>Respostas</b>	<b>%</b>
Não usaria o iGraf	1	4
Usaria o iGraf apenas se não tivesse outra opção	2	8
Usaria qualquer um, incluindo o iGraf	9	36
Optaria pelo iGraf preferencialmente	10	40
Optaria pelo iGraf incondicionalmente	3	12

## Anexo 03

### Relatos obtidos nos questionários pós-teste

A seguir são apresentados os comentários feitos pelos participantes do teste com o iGraf. Estes comentários estão documentados, por escrito, nos próprios questionários. Com o objetivo de manter a fidelidade na transcrição dos comentários, nenhuma correção ortográfica ou gramatical foi feita sobre o texto original... ele apenas foi digitalizado.

**11. O iGraf apresenta alguns recursos que não são comuns em outros visualizadores. Você observou algum desses recursos? Se sim, por gentileza, faça um breve comentário dizendo por quê gostou ou não.**

“A opção exercícios, pois facilita o EAD. E as ferramentas de Cálculo, pois, apesar de aparecerem em outros visualizadores, são de fácil utilização.”

“A animação é interessante. O editor é bom porém apresentou falhas ao desenhar todas”

“Gostei recursos animação e área de desenho são recursos diferenciados”

“Trabalhar com duas variáveis, como a equação da circunferência p/ que possamos desenvolver/demonstrar equações trigonométricas.”

“O recurso visualizar derivada, integral”

“Gostei da movimentação dos gráficos”

**12. O iGraf está em fase de teste. Se você detectou algum erro ou comportamento desagradável do programa, por gentileza, descreva-o**

“Falta um botão de ‘desfazer’ e ‘refazer’ a última ação.”

“Durante a nevegação (aproximadamente 3 horas) não foi detectado erros.”

“Penso que deveria constar as definições dos conteúdos.”

“O programa por muitas vezes ‘trava’ e isso atrapalha o andamento do trabalho”

“Sim. O programa trava com frequência”

“Travou várias vezes”

“Ao clicar determinado ícone, ele não obedecia ao comando do mouse ou executava a função anterior. Faltou também o zoom”

“Não consegui ocultar um gráfico anterior ao último”

“Sim, ele trava com muita frequência e não permite digitar as funções”

“Não muda a cor da função quando requisitado”

**13. *Caso queira fazer algum elogio, comentário, crítica ou sugestão utilize as linhas abaixo***

“Gostei bastante do iGraf, usarei constantemente. Qualquer problema lhe aviso. Parabéns!”

“Impressiona bastante, mas está muito instável”

“Deslocar a escala do eixo das ordenadas”

“apresentar as setas no plano cartesiano”

“acho que deveria, na barra de ferramentas, ser acrescentado dois botões: um para ponto e outro para função, pois poderia usar a notação de função e não uma expressão”

“É um programa a ser investido”

“Parabéns pelo trabalho e pela aula teste.”

“Parabéns! O programa é muito bom.”

“Melhorar a visualização”

“Bom eu ja usava o Grafmatica e achei este igraf muito mais facil de usar e com muito mais recursos... muito Bom”

“De modo geral e por ser gratuito vamos fazer bom uso”

“Não tive contato com outros visualizadores, mas achei o iGraf de fácil visualização e operação”

“Gostaria que as raízes das funções fossem calculadas e sejam mostradas com o passar do mouse em uma caixa de diálogos”

“parabens pela iniciativa de contribuir para a melhoria do ensino, continue aperfeiçoando para melhorias futuras”

“Programa muito bom, continuem assim pois esse é o futuro da educação”



## Referências Bibliográficas

- [1] Bill Austin, Colin Milligan, and Ruth Thomas, *Java e-learning and simulation*, Heriot-Watt University, <http://www.jelsim.org/>, 2002.
- [2] Raymond A. Barnett, *Calculus for business, economics, life sciences, and social sciences*, 10 ed., Pearson, New Jersey, 2005.
- [3] Maria Cristina Bonomi Barufi and Maira Mendias Lauro, *Funções elementares, equações e inequações: Uma abordagem utilizando microcomputador*, CAEM - IME/USP, Imprensa Oficial, São Paulo, 2000.
- [4] Learning Technology Standards Committee, *Learning object metadata*, Institute of Electrical and Electronics Engineers, [http://ltsc.ieee.org/wg12/files/LOM\\_1484\\_12\\_1\\_v1\\_Final\\_Draft.pdf](http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf), 2002.
- [5] Paulo Sérgio da Mota Pires, *Introdução ao scilab*, Universidade Federal do Rio Grande do Norte, <http://www.scilab.org/publications/SCIPOINT/sciport.pdf>, 2001.
- [6] Ubiratan D'Ambrósio, *Informática, ciências e matemática*, Proinfo: Programa Nacional de Informática na Educação (<http://www.proinfo.mec.gov.br/site/biblioteca.php>), Ministério da Educação e Cultura - MEC, 2000.
- [7] Jean-Marrie Dautelle, *Java tools and libraries for the advancement of sciences*, Open Source Project, <http://jscience.org/>, 2005.
- [8] Associação Brasileira de Educação a Distância (ed.), *Anuário brasileiro estatístico de educação aberta e a distância*, Instituto Monitor, 2007.

- [9] Janine Gomes de Moura and Leônidas de Oliveira Brandão, *Saw – sistema de aprendizagem pela web baseado em applets*, Proceedings of the Webmedia & LA-Web Joint Conference, 2004, pp. 280–283.
- [10] André Luís Lima de Oliveira, *Avaliação comparativa de diferentes modelos de interfaces gráficas empregadas no ensino de geometria, segundo os conceitos de usabilidade.*, Master's thesis, Universidade de São Paulo, 2005.
- [11] Institut National de Recherche en Informatique et em Automatique, *Scilab - sistema de Álgebra computacional (cas)*, École Nationale des Ponts et Chaussées, <http://www.scilab.org/>, 2007.
- [12] José Marcelino de Rezende Pinto, *O acesso a educação superior no brasil*, Revista Educação e Sociedade (<http://www.cedes.unicamp.br/revista/rev/rev88.htm>), vol. 25, Universidade de Campinas - Unicamp, Centro de Estudos Educação & Sociedade, outubro 2004, acessado em 15/03/2008, pp. 727–756.
- [13] Linda Taranto Goulart e outros, *O desafio de universalização do ensino médio*, maio 2006, Acessado em 15/03/2008.
- [14] David J. Eck, *Java components for mathematics*, Hobart and William Smith Colleges, <http://math.hws.edu/javamath/>, 2001.
- [15] Jairo Ferreira, *O comportamentalismo e a instrução programada*, Universidade Federal do Rio Grande do Sul - UFRGS, 1998.
- [16] Vera L X Figueiredo, Margarida P Melo, and Sandra A Santos, *Cálculo com aplicações: Atividades computacionais e projetos*, Coleção IMECC, Unicamp IMECC, São Paulo, 2005.
- [17] Gerhard Pfister e Hans Schönemann Gert-Martin Greuel, *Singular 3.0 - a computer algebra system for polynomial computations.*, Centre for Computer Algebra, University of Kaiserslautern - Deutsch, 2005.
- [18] Keith Hertzner, KSoft, <http://www8.pair.com/ksoft/>, 2007.
- [19] Seiji Isotani, *Desenvolvimento de ferramentas no igeom: utilizando a geometria dinâmica no ensino presencial e a distância*, Master's thesis, Universidade de São Paulo, 2005.

- [20] Java, *Linguagem de programação java*, Sun Microsystems, <http://java.sun.com/>, 2007.
- [21] Daniel J. Kleitman, *What is calculus and why do we study it?*, Calculus for Begginers and Artists ([http://www-math.mit.edu/~djk/calculus\\_beginners/chapter01/section02.html](http://www-math.mit.edu/~djk/calculus_beginners/chapter01/section02.html)), Massachusetts Institute of Technology - MIT, Department of Mathematics, 2008, Acessado em 15/03/2008.
- [22] General Public License, GNU Project, <http://www.fsf.org/licenses/licenses/gpl.html>, 2007.
- [23] Thomas Maloy, Gary Jensen, Alison Regan, and May Reddick, *Open courseware and shared knowledge*, Behavior Research Methods Instruments & Computers, no. 34, California State University, 2002.
- [24] Maple, *Sistema de Álgebra computacional (cas)*, MapleSoft Incorporated, <http://www.maplesoft.com/>, 2007.
- [25] Maxima, *Sistema de Álgebra computacional (cas)*, Open Source Project, <http://maxima.sourceforge.net/>, 2007.
- [26] Meditor, *Sistema de Álgebra computacional (cas)*, Open Source Project, <http://sourceforge.net/projects/jscl-meditor/>, 2007.
- [27] Richard L Parris, *Winplot*, Academia Phillips Exeter, <http://math.exeter.edu/rparris/winplot.html>, julho 2007.
- [28] Ashok Patel, Bernard Scott, and Kinshuk, *Intelligent tutoring: From saki to byzantium*, Kibernetes, no. 30, ISSN 0368-492X.
- [29] Richard Petti, *Macsyma saga*, Texas University, 2003.
- [30] Priberam, *Língua portuguesa online*, Priberam Informática S/A, <http://priberam.com>, maio 2008.
- [31] QuickMath, *Automatic math solutions*, Quick Math, <http://www.quickmath.com/>, 2007.

- [32] Marcia A. Gomes Ruggiero and Vera Lucia da Rocha Lopes, *Cálculo numérico*, Pearson Makron Books, São Paulo, 2004.
- [33] SAGE, *Sistema de Álgebra computacional (cas)*, Open Source Project, <http://www.sagemath.org/>, 2007.
- [34] Burrhus Frederic Skinner, *Teaching machines*, Science, vol. 128, 10 1958, pp. 969–977.
- [35] Ian Sommerville, *Engenharia de software*, Pearson Addison-Wesley, São Paulo, 2007.
- [36] James Stewart, *Cálculo*, vol. 1, Thomson Learning, São Paulo, 2003.
- [37] Geraldo Ávila, *Cálculo i: diferencial e integral*, Livros Técnicos e Científicos - LTC, Rio de Janeiro, 1978.
- [38] WebGraphing, *Online graphing calculators*, WebGraphing.com, <http://webgraphing.com>, maio 2008.
- [39] Stefan Weinzierl, *Computer algebra in particle physics*, proceedings of XI Seminario Nazionale di Fisica Teorica (<http://www.pr.infn.it/snft/2002/SNFT-2002.htm>), Istituto Nazionale di Fisica Nucleare, Università di Parma, 2002.
- [40] Patrick Wessa, *Wessa - web-enabled scientific services and applications*, Free Statistics Software, Office for Research Development and Education, <http://www.wessa.net>, maio 2008.
- [41] Yacas, *Sistema de Álgebra computacional (cas)*, Open Source Project, <http://yacas.sourceforge.net/>, 2007.