

UNIVERSIDADE FEDERAL DE ALFENAS
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Geraldo Lucas Jardim Cordeiro

**INTERFACE PARA FERRAMENTA DE BUSCA DE CONTEÚDO
MATEMÁTICO**

Alfenas, 02 de Julho de 2010.

UNIVERSIDADE FEDERAL DE ALFENAS
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**INTERFACE PARA FERRAMENTA DE BUSCA DE CONTEÚDO
MATEMÁTICO**

Geraldo Lucas Jardim Cordeiro

Monografia apresentada ao Curso de Bacharelado em
Ciência da Computação da Universidade Federal de
Alfenas como requisito parcial para obtenção do Título de
Bacharel em Ciência da Computação.

Orientador: Prof. Flavio Barbieri Gonzaga

Alfenas, 02 de Julho de 2010.

Geraldo Lucas Jardim Cordeiro

**INTERFACE PARA FERRAMENTA DE BUSCA DE CONTEÚDO
MATEMÁTICO**

A Banca examinadora abaixo-assinada aprova a monografia apresentada como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação pela Universidade Federal de Alfenas.

Prof. Luiz Eduardo da Silva

Universidade Federal de Alfenas

Profa. Mariane Moreira de Souza

Universidade Federal de Alfenas

Prof. Flavio Barbieri Gonzaga (Orientador)

Universidade Federal de Alfenas

Alfenas, 02 de Julho de 2010.

Dedico este **Trabalho de Conclusão de Curso**
à todos que de alguma forma fizeram parte da minha vida.

Aos meus **Pais, Geraldo e Maria**,
meus primeiros mestres,
que sempre acreditaram em mim,
inclusive orando por mim.

Aos **Familiares**,
que nos momentos de minha ausência,
sempre me deram forças e torceram pelo meu sucesso.

Àquelas pessoas que quando deveriam ser simplesmente **Professores**,
foram **Mestres**.
Que quando deveriam ser **Mestres**,
foram **Amigos**.

Aos verdadeiros **Amigos**,
que sempre me animaram nos momentos difíceis
e que por mais complicado que fosse o obstáculo,
me deram forças para nunca parar.

AGRADECIMENTO

*Aos meus pais **Geraldo Nunes Cordeiro e Maria Raimunda Jardim Cordeiro**, por terem me ajudado em todos os aspectos.*

*A **Universidade Federal de Alfenas (UNIFAL-MG)**, juntamente com o **Laboratório de Inteligência Computacional (LInC)**, meu reconhecimento e agradecimento pela a oportunidade de formar no curso de **Bacharelado em Ciência da Computação (BCC)**.*

*Ao Prof. **Flavio Barbieri Gonzaga**, pela orientação no desenvolvimento deste projeto e na elaboração desta monografia, guiando de forma sábia os meus passos.*

*Ao Prof. **Humberto César Brandão de Oliveira**, por acreditar e apoiar o desenvolvimento de trabalhos científicos.*

*Ao Prof. **Luís Eduardo da Silva**, por guiar todos seus alunos sempre com bom humor e paciência, sendo um professor exemplar.*

*Aos meus amigos de República, **João Martins, Danilo Bispo, Adriano Luís, Vinícius Diniz**, além das agregadas da enfermagem **Rafaela, Marina e Samara**, que foram a minha família em Alfenas.*

*Aos demais **amigos**, que, ajudaram de alguma forma na elaboração desta monografia e que enriqueceram meus conhecimentos.*

Muito obrigado a todos!

"O universo escreve-se em linguagem matemática."

Galileu Galilei

RESUMO

Recentemente, projetos de bibliotecas matemáticas digitais disponibilizam informações de diversas áreas da matemática, sendo fontes de pesquisa para diferentes perfis de usuários. As bibliotecas existentes hoje oferecem ao usuário a busca textual pelas informações, de modo que um usuário que deseje obter informações sobre uma determinada expressão precisa conhecer o nome da mesma. Assim, essas bibliotecas não oferecem suporte à busca por equações ou expressões matemáticas com base na estrutura das mesmas. Exemplificando, um pesquisador que encontre uma determinada expressão matemática, não terá como efetuar a busca pela mesma em virtude de não saber se a expressão possui um nome associado. Em virtude de problemas dessa natureza, a busca por expressões matemáticas já vêm recebendo a atenção de pesquisadores em diferentes partes do mundo.

Assim, o foco do presente trabalho é o desenvolvimento de uma interface web que ofereça ao usuário o suporte para montar uma expressão matemática sem que seja necessário o conhecimento de nenhuma linguagem para representação matemática, como por exemplo as linguagens *MathML* e *TeX*. A interface proposta poderá então ser integrada com ferramentas de busca específicas de conteúdo matemático futuramente.

Palavras-Chave: Bibliotecas matemáticas digitais, linguagens matemáticas, interface, *MathML*.

ABSTRACT

Recently, math libraries projects offer digital information in several areas of mathematics, and research sources for different user profiles. Actually, existing libraries offer the user to search for textual information, so that a user wishing to obtain information about a particular expression must know the name of it. Thus, these libraries do not support the search for mathematical expressions or equations based on the structure of them. For instance, a researcher who finds a particular mathematical expression will not make the search for the same because they do not know if the expression has a name associated. Because of such problems, the search for mathematical expressions has been receiving attention from researchers in different parts of the world.

Thus, the focus of this work is to develop a web interface that provides user support to formulate a mathematical expression without requiring knowledge of any language to mathematical representation, such as languages MathML and TeX. The proposed interface can then be integrated with specific search tools of mathematical content in the future.

Keywords: Mathematical digital libraries, mathematical languages, interface, MathML.

LISTA DE FIGURAS

FIGURA 2.1- EXEMPLO DE UM CÓDIGO XML (W3SCHOOLS, 2010).....	32
FIGURA 2.2 - ESQUEMA EM ÁRVORE DO EXEMPLO DE CÓDIGO APRESENTADO NA FIGURA 2.1 (W3SCHOOLS, 2010).....	32
FIGURA 2.3 - EXEMPLO DE UMA EXPRESSÃO MATEMÁTICA.	33
FIGURA 2.4 - MARCAÇÃO UTILIZANDO A LINGUAGEM <i>MATHML</i> NO SEU FORMATO DE APRESENTAÇÃO A PARTIR DO EXEMPLO DE EXPRESSÃO MATEMÁTICA DEMONSTRADA NA FIGURA 2.3.....	34
FIGURA 2.5 - MARCAÇÃO UTILIZANDO A LINGUAGEM <i>MATHML</i> NO SEU FORMATO DE CONTEÚDO A PARTIR DO EXEMPLO DE EXPRESSÃO MATEMÁTICA DEMONSTRADA NA FIGURA 2.3.....	35
FIGURA 2.6 - EXEMPLO DE UMA EXPRESSÃO MATEMÁTICA EM <i>TeX</i>	37
FIGURA 3.1 - RELAÇÃO DE AFINIDADE DO USUÁRIO QUANTO A LINGUAGENS MATEMÁTICAS E OUTROS MÉTODOS DE ENTRADA DE EXPRESSÕES (ZHAO ET AL., 2008).	41
FIGURA 3.2 - PROTÓTIPO DA <i>MATH WEB SEARCH</i>	43
FIGURA 3.3 - TELA MOSTRADA PARA A EXPRESSÃO AMBÍGUA $\sin ? = ?$	44
FIGURA 3.4 - PROTÓTIPO DA <i>WHELP</i>	45
FIGURA 3.5 - TELA DA FERRAMENTA EM MODO SIMPLES.....	46
FIGURA 3.6 - TELA DA FERRAMENTA EM MODO AVANÇADO UTILIZANDO UMA EXPRESSÃO MATEMÁTICA COMO ENTRADA.	47
FIGURA 3.7 - INTERFACE DA FERRAMENTA DE BUSCA MATEMÁTICA <i>MATHDEX</i>	48
FIGURA 4.1 - DIAGRAMAS DEFINIDOS PELA <i>UML</i> (BEZERRA, 2007).....	54
FIGURA 4.2 - CASO DE USO DA INTERFACE.....	56
FIGURA 4.3 - DIAGRAMA DE CLASSE REFERENTE A VISÃO DE INTERFACE.....	57
FIGURA 4.4 - DIAGRAMA DE SEQUÊNCIA DA VISÃO DE INTERFACE.....	57
FIGURA 4.5 - CASO DE USO PARA A VISÃO MENU.	58
FIGURA 4.6 - DIAGRAMA DE CLASSE PARA A VISÃO DE MENU.	58
FIGURA 4.7 - DIAGRAMA DE SEQUÊNCIA PARA A VISÃO DE MENU.	59
FIGURA 4.8 - CASO DE USO DA VISÃO SÍMBOLO.....	59
FIGURA 4.9 - DIAGRAMA DE CLASSE DA VISÃO SÍMBOLO.	60
FIGURA 4.10 - DIAGRAMA DE SEQUÊNCIA DA VISÃO SÍMBOLO.	61
FIGURA 4.11 - CASO DE USO DA VISÃO FUNÇÃO.	61
FIGURA 4.12 - DIAGRAMA DE CLASSE DA VISÃO FUNÇÃO.	62
FIGURA 4.13 - DIAGRAMA DE SEQUÊNCIA DA VISÃO FUNÇÃO.	63
FIGURA 4.14 - CASO DE USO DA VISÃO DE EDIÇÃO DE UMA FUNÇÃO.....	63
FIGURA 4.15 - DIAGRAMA DE CLASSE DA VISÃO DE EDIÇÃO DE UMA FUNÇÃO.	64
FIGURA 4.16 - DIAGRAMA DE SEQUÊNCIA DA VISÃO DE EDIÇÃO DE UMA FUNÇÃO.	64
FIGURA 5.1 - EXEMPLO DA DIFERENÇA DE TAGS USADAS NA COMPOSIÇÃO DE OPERADORES, NÚMEROS E MATRIZES.....	67
FIGURA 5.2 - CONSULTA À TAGLIB PARA CRIAÇÃO DE UM NOVO BOTÃO PARA O OPERADOR "SOMA".	68
FIGURA 5.3 - TELA DA INTERFACE.....	69
FIGURA 5.4 - ELEMENTOS MATEMÁTICOS INSERIDOS NA BARRA DE BUSCA.....	70

FIGURA 5.5 - INSERÇÃO DE UM VALOR DO TIPO VARIÁVEL.	71
FIGURA 5.6 - EDIÇÃO DE UMA FUNÇÃO, QUE NO CASO TEM-SE O EXEMPLO DE UMA POTENCIAÇÃO.	71
FIGURA 5.7 - PAINÉIS SEPARADOS DOS SUBMENUS.	72
FIGURA 5.8 - EXPRESSÃO MATEMÁTICA RENDERIZADA A PARTIR DO CÓDIGO <i>MATHML</i> DA MESMA.	72
FIGURA 5.9 - VISUALIZAÇÃO AUMENTADA DO ELEMENTO EDITADO.	73
FIGURA 5.10 - CÓDIGO <i>MATHML</i> RESULTANTE DA EXPRESSÃO FORMULADA.	74
FIGURA 5.11 - ESTRUTURA DE UM SISTEMA DE BUSCA MATEMÁTICA (GONZAGA, 2010).	75

LISTA DE TABELAS

TABELA 3.1 - PONTOS POSITIVOS DOS MÉTODOS, CONVENCIONAL E UTILIZANDO LINGUAGENS MATEMÁTICAS.	42
TABELA 3.2 - SEMELHANÇAS E DIFERENÇAS DENTRE AS INTERFACES DAS FERRAMENTAS APRESENTADAS.....	49

LISTA DE ABREVIACÕES

AJAX	<i>Asynchronous Javascript And XML</i>
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CAS	<i>Computer Algebra Systems</i>
CSS	<i>Cascading Style Sheets</i>
DHTML	<i>Dynamic HTML</i>
DLMF	<i>Digital Library of Mathematical Functions</i>
DOM	<i>W3C Document Object Model</i>
GNU	<i>GNU is Not Unix</i>
GWT	<i>Google Web Toolkit</i>
HELM	<i>Hypertextual Electronic Library of Mathematics</i>
HTML	<i>HyperText Markup Language</i>
IDE	<i>Integrated Development Environment</i>
LGPL	<i>GNU Lesser General Public License</i>
MathML	<i>Mathematical Markup Language</i>
NIST	<i>National Institute of Standards and Technology</i>
RI	Recuperação da Informação
UML	<i>Unified Modeling Language</i>
W3C	<i>World Wide Web Consortium</i>
xHTML	<i>eXtensible Hypertext Markup Language</i>
XML	<i>eXtensible Markup Language</i>
YUI	<i>Yahoo! User Interface</i>

SUMÁRIO

1 INTRODUÇÃO	23
1.1 JUSTIFICATIVA E MOTIVAÇÃO	24
1.2 PROBLEMATIZAÇÃO	25
1.3 OBJETIVOS	26
1.3.1 Gerais	26
1.3.2 Específicos	26
1.4 ORGANIZAÇÃO DA MONOGRAFIA	26
2 BIBLIOTECAS DIGITAIS E LINGUAGENS MATEMÁTICAS	27
2.1 CONSIDERAÇÕES INICIAIS	27
2.2 BIBLIOTECAS DIGITAIS	28
2.3 BIBLIOTECAS MATEMÁTICAS DIGITAIS	29
2.4 LINGUAGENS MATEMÁTICAS	31
2.4.1 XML	31
2.4.2 MathML	33
2.4.3 TeX	36
2.5 CONSIDERAÇÕES FINAIS	37
3 REVISÃO BIBLIOGRÁFICA	39
3.1 CONSIDERAÇÕES INICIAIS	39
3.2 ENTRADA DE INFORMAÇÕES	39
3.3 FERRAMENTAS DE BUSCA	42
3.3.1 Math Web Search	42
3.3.2 Whelp	43
3.3.3 ActiveMath	45
3.3.4 Mathdex	47
3.4 CONSIDERAÇÕES FINAIS	48
4 INTERFACE PARA FERRAMENTA DE BUSCA MATEMÁTICA	51
4.1 MODELAGEM ORIENTADA A OBJETOS	51
4.2 LINGUAGEM DE MODELAGEM UNIFICADA (UML)	53
4.3 PROJETO	54
4.3.1 Escopo do projeto	54
4.3.2 Modelagem	56
4.3.2.1 Visão de Interface	56
4.3.2.2 Visão de Menu	58
4.3.2.3 Visão Símbolo	59
4.3.2.4 Visão Função	61
4.3.2.5 Visão de edição de uma função	63
5 RESULTADOS	65
5.1 CONSIDERAÇÕES INICIAIS	65
5.2 EXTJS	66
5.3 TAGLIB	67
5.4 DESENVOLVIMENTO	68

5.4.1 Interface resultante	69
5.5 CONSIDERAÇÕES FINAIS	74
6 CONCLUSÕES	77
7 TRABALHOS FUTUROS	79
8 REFERÊNCIAS BIBLIOGRÁFICAS	81

1

Introdução

Este capítulo apresenta a monografia de maneira geral, sendo a Seção 1.1 Justificativa e motivação, na Seção 1.2 é apresentado o problema, em seguida na Seção 1.3 são listados os objetivos do trabalho e por fim é apresentada a organização da monografia na Seção 1.4.

Nos últimos anos houve um grande empenho por parte de pesquisadores no desenvolvimento de bibliotecas digitais de conteúdo matemático e científico. Projetos como a *Digital Library of Mathematical Functions - DLMF* (Nist, 2010), o *MathWorld* (Weisstein, 2007) e a *Wikipedia* (Wikipedia, 2010) já disponibilizam conteúdo de diversas áreas da matemática, sendo fontes de pesquisa tanto para estudantes de ensino médio quanto para pesquisadores, matemáticos e cientistas.

Segundo Youssef (2006), para que os usuários se beneficiem dos conteúdos destas bibliotecas, é necessário que haja facilidade e efetividade não só para pesquisas de textos, mas também para equações e construções matemáticas.

Atualmente existem linguagens que permitem a formulação de expressões matemáticas e que embasam o conteúdo das bibliotecas *DLMF*, *MathWorld* e *Wikipedia*. Nesse aspecto, pode-se citar as linguagens *MathML* (W3Ca, 2010) e *TeX* (Knuth, 1985), sendo que a primeira é utilizada na *DLMF*, enquanto que a segunda é a base das linguagens utilizadas no *MathWorld* e *Wikipedia*. A *MathML* foi criada a partir da necessidade de se expressar conteúdo matemático em navegadores (tanto do ponto de vista de conteúdo, quanto do ponto de vista gráfico). Baseada no *XML* (W3Cb, 2010), a linguagem possui *tags* específicas para representar componentes de equações, como números e operadores. A linguagem *TeX* por sua vez, foi desenvolvida com o propósito de ser utilizada na produção de conteúdo científico, com a característica de ser reproduzida da mesma forma em diferentes

computadores. Por se tratar de uma linguagem comumente utilizada em trabalhos de pesquisa, a mesma possui suporte à representação de expressões matemáticas.

A proposta deste trabalho é desenvolver uma ferramenta que seja a interface entre o usuário e o mecanismo de busca de bibliotecas matemáticas, auxiliando-os de forma que, construída uma equação visualmente através das opções da interface, obtenha-se como saída a representação correspondente da mesma em *MathML*, para que então seja enviada ao sistema de busca.

1.1 Justificativa e Motivação

Atualmente a Internet possui um grande acervo de informações das mais diversas, encontradas em arquivos em vários formatos. Estes arquivos podem ser obtidos a partir de pesquisas efetuadas em sistemas de busca disponíveis na *Web*. Entretanto, há uma grande dificuldade de se obter informações consistentes e de fontes confiáveis que satisfaçam o resultado esperado pelo usuário. Neste momento surgem as bibliotecas digitais, com o intuito de organizar os vários conteúdos obtidos da *Web*, de forma que facilite nas pesquisas de temas científicos. No entanto os sistemas de busca disponíveis nestas bibliotecas ainda não oferecem ao usuário a possibilidade de se montar uma expressão matemática de forma gráfica, devendo o mesmo fazer a busca textual convencional (ou seja, informando o nome da expressão por exemplo), ou utilizar da linguagem matemática da biblioteca (*TeX*, *MathML*, dentre outras, que serão melhor explicadas no Capítulo 2).

Este projeto visa a implementação de uma interface que permita ao usuário montar expressões de forma gráfica, sem a necessidade do conhecimento de nenhuma linguagem matemática. A interface proposta gera então como saída uma estrutura da equação no formato de *tags* da linguagem *MathML* e, a partir desta, alimenta-se o serviço de busca podendo ser convertida posteriormente em outras

linguagens. A escolha da linguagem *MathML* e questões relacionadas à compatibilidade com navegadores será discutida no Capítulo 4.

1.2 Problematização

Os mecanismos de busca de textos disponíveis em *sites* da Internet foram desenvolvidos para, dado uma entrada textual, encontrar uma frase ou expressão semelhante. No entanto, quando um usuário possui uma informação específica não textual, as ferramentas textuais não oferecem suporte à recuperação de informação dessa natureza, uma vez que as mesmas não apresentam uma interface de entrada adequada para que o usuário possa representar a informação a ser requisitada.

Além da interface de entrada não ser adequada, os algoritmos de indexação e recuperação de informação para a busca textual não são adequados para o uso em busca matemática. Por exemplo, o dicionário de sinônimos, que oferece suporte para caso o usuário informe uma palavra 'cão', e a ferramenta encontre um sinônimo como 'cachorro', esse resultado também seja listado, não se aplica a ferramentas matemáticas.

Logo, como o usuário deve formular uma expressão matemática para ser passada para a ferramenta de busca utilizada? O foco do presente trabalho é então o desenvolvimento de uma interface de entrada adequada para esse tipo de busca, ficando a parte de indexação e recuperação da informação como uma funcionalidade a ser implementada na ferramenta de busca em trabalhos futuros.

1.3 Objetivos

1.3.1 Gerais

Desenvolver um modelo de interface para ferramentas de busca de conteúdo matemático.

1.3.2 Específicos

Os objetivos específicos deste projeto são os seguintes:

- Estudar a linguagem *XML (Extensible Markup Language)*.
- Estudar a linguagem matemática *MathML*.
- Obter conhecimento sobre a área de Interface Homem-Computador.
- Desenvolver uma interface que permita ao usuário construir expressões matemáticas de forma visual, ou seja, sem a necessidade do conhecimento em linguagens específicas.
- Desenvolvimento da monografia.

1.4 Organização da Monografia

No Capítulo 2 é feita a apresentação das bibliotecas matemáticas digitais que, por sua vez, são embasadas em linguagens específicas. No Capítulo 3 é feita a revisão bibliográfica. No Capítulo 4 é apresentada a proposta deste trabalho. No Capítulo 5 é mostrado como foi o desenvolvimento da interface. No Capítulo 6 são dadas as conclusões do trabalho. São apresentadas no Capítulo 7 algumas idéias para trabalhos futuros e, finalmente, as referências bibliográficas no Capítulo 8.

2

Bibliotecas Digitais e Linguagens Matemáticas

Este capítulo apresenta, na Seção 2.1, a contextualização da área de Recuperação da Informação, seguido da Seção 2.2 que fala sobre Biblioteca Digital e sua importância, apresentando alguns exemplos. Na Seção 2.3 são mostradas algumas Bibliotecas Matemáticas Digitais, analisando logo em seguida na Seção 2.4 as linguagens matemáticas usadas na apresentação do conteúdo destas bibliotecas. Alguns pontos que devem ser levados em consideração a cerca do que foi apresentado são tratados na Seção 2.5.

2.1 Considerações Iniciais

Recuperação da Informação (RI) é a obtenção de material que, geralmente, são documentos de natureza não-estruturados (texto), que satisfaça uma necessidade de informação a partir de grandes coleções (geralmente armazenadas em computadores).

Durante a última década houve intensa pesquisa na área de RI, buscando aperfeiçoar a eficácia das informações obtidas de sistemas de busca na web. Isso possibilitou novos níveis de qualidade, levando à satisfação de pessoas na maioria de suas buscas, tornando uma fonte de pesquisa preferencial.

Por exemplo, em Fallows (2004) constataram que “92% dos usuários de Internet dizem que a Internet é um bom lugar para se obter informações diárias”. Para surpresa de muitos, a área de RI deixou de ser uma questão essencialmente acadêmica para uma base de acesso à informação da maioria das pessoas.

Desta forma, tornou-se muito mais cômodo ao usuário buscar informações rápidas através da Internet do que consultar um livro na biblioteca. As informações apresentadas e outras relacionadas a área de RI estão disponíveis em Manning et al. (2008).

2.2 Bibliotecas Digitais

Atualmente a Internet abrange uma vasta quantidade de conteúdos científicos e tecnológicos dos mais diversos temas, tipos e formatos. Em paralelo a isso, foram desenvolvidos sistemas de recuperação de informação, como por exemplo o buscador de conteúdo em sites da *Google* (Brin & Page, 1998).

Tendo como objetivo o armazenamento de documentos e a recuperação automática de informação associada a eles, os buscadores web atuais, dependendo da informação a ser procurada, ainda lidam com o problema de refinamento do conteúdo retornado ao usuário. Soluções como o *Google Scholar* (Kesselman & Watsen, 2005) foram desenvolvidas no intuito de suprir essa necessidade de organização de conteúdo. Este mecanismo efetua uma busca otimizada para se ter como retorno artigos revisados por especialistas, teses, livros, resumos e artigos de editoras acadêmicas, organizações profissionais, bibliotecas de pré-publicações, universidades e outras entidades acadêmicas. Surgem então as chamadas Bibliotecas Digitais que, baseadas nas bibliotecas tradicionais, têm como objetivo a organização do acervo de conteúdos relevantes às áreas de pesquisa, disponíveis na Rede Mundial de Computadores.

A Biblioteca Digital surge como uma aliada da Biblioteca Tradicional. Sua organização requer os mesmos princípios empregados em uma biblioteca tradicional, dentre eles: seleção da fonte de informação, aquisição de informações que preencham as necessidades do usuário, fornecimento de material atualizado;

boa organização e classificação das aquisições com a finalidade de auxiliar a orientação do usuário na seleção dos itens de seu interesse, agrupamento de fontes de acordo com sua função e seu conteúdo, entre outros. A mudança mais óbvia que ocorre como resultado da inovação tecnológica está expressa concisamente no acesso, em lugar de propriedade (Pereira, 1995).

Instituições de ensino vêm criando portais oferecendo materiais didáticos, cursos e livros, seguindo o ideal da educação à distância que é a universalização da informação e o rompimento dos limites físicos para se obter o conhecimento. Atualmente o Portal CAPES (Capes, 2010) oferece acesso aos textos completos de artigos selecionados de mais de 15.475 revistas internacionais, nacionais e estrangeiras, e 126 bases de dados com resumos de documentos em todas as áreas do conhecimento, incluindo também uma seleção de importantes fontes de informação acadêmica com acesso gratuito na Internet. O uso do Portal é livre e gratuito para os usuários das instituições participantes. No entanto estas bibliotecas ainda estão em constante fase de crescimento, tanto de conteúdo como de facilidade ao manuseio. Surge então um grande desafio que é além de ampliar sua popularidade, a biblioteca digital também deve se adequar aos diversos níveis de usuários de forma que estes fiquem satisfeitos com os resultados apresentados nas suas pesquisas.

Dentre as diversas temáticas envolvidas em cada Biblioteca Digital, destacam-se as Bibliotecas Matemáticas Digitais, contendo informações que abrangem várias áreas da matemática e suas correlacionadas.

2.3 Bibliotecas Matemáticas Digitais

Atualmente existem diversas bibliotecas *online* que disponibilizam materiais voltados para pesquisa na área da matemática, dentre elas destacam-se:

- *MathWorld*,
- *DLMF*,
- *Wikipedia*.

A *MathWorld* foi desenvolvida por Eric Weisstein (Weisstein, 2007) e lançada no ano de 1995. Esta se tornou referência tanto na área de educação quanto entre os matemáticos. Atualmente a biblioteca conta com mais de 15.000 artigos, podendo os usuários proporem novos artigos, bem como contribuir para os artigos já escritos. As expressões matemáticas que embasam o conteúdo desta biblioteca são representadas em imagens no formato *GIF*. Porém, todas possuem uma representação alternativa em forma textual, que pode ser obtida no código *HTML* da página.

A *NIST Digital Library of Mathematical Functions* é um projeto que há pouco tempo foi finalizado. A sua organização é basicamente a divisão dos capítulos do livro *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables* (Abramowitz & Stegun, 1964), contendo em cada capítulo os tópicos internos. O diferencial da biblioteca *DLMF* está na sua representação. De forma semelhante à *MathWorld*, a mesma possui a representação gráfica e textual, mas nesse caso a representação textual está disponível tanto em linguagem *TeX* quanto em uma linguagem de marcação específica para a representação de conteúdo matemático na Internet, denominada *MathML*, apresentada na Seção 2.4.2 deste Capítulo.

A *Wikipedia* é uma enciclopédia livre que está em constante atualização de suas informações através da colaboração de milhares de pessoas de todas as partes do mundo. Baseado no conceito de *Wiki Wiki* (termos *wiki* e *Wiki Wiki* são utilizados para identificar um tipo específico de coleção de documentos em hipertexto ou o software colaborativo usado para criá-lo.), significando que qualquer usuário pode editar o conteúdo de quase todos os artigos mostrados nas páginas do site. O projeto foi iniciado em 15 de janeiro de 2001, na versão em língua inglesa. Em

apenas um ano de existência, esta versão já possuía quase 10 mil artigos. Até hoje já foram criados mais de 14 milhões de artigos em centenas de línguas e dialetos, sendo todo este conteúdo coberto pela licença de documentação livre *GNU*.

Estas bibliotecas digitais possuem um mecanismo de busca de conteúdo, onde o usuário insere dados sobre um determinado assunto e obtém como resultado páginas relacionadas à informação buscada. No entanto, caso o usuário queira efetuar uma busca de uma expressão matemática, surge então o problema de como modelar tal informação, pois, o padrão de formulação de expressões utilizada no conteúdo das páginas pode não coincidir com o padrão de formulação usada pelo usuário, ocasionando erro na busca. E mais, o usuário pode até conhecer o padrão de formulação, no entanto sua estrutura, dependendo do tamanho da expressão ou da linguagem matemática utilizada, pode se tornar bem complexa.

2.4 Linguagens Matemáticas

2.4.1 XML

Com o enorme crescimento do conteúdo da *Web*, tornou-se necessária a criação de padrões para a comunicação de dados. Com isso, foram criadas linguagens de marcação para serem utilizadas com a internet. *XML* (W3Cb, 2010), sigla de *Extensible Markup Language*, é uma linguagem de marcação que foi criada para suprir vários problemas existentes na computação atual. Através desta linguagem, tornou-se possível a estruturação de documentos que fornece um significado semântico.

```

<livraria>
  <livro categoria = "CULINARIA">
    <titulo lingua = "pt"> TITULO 1 </titulo>
    <autor> AUTOR 1 </autor>
    <ano> 2010 </ano>
    <preco> 0.00 </preco>
  </livro>
  <livro categoria = "INFANTIL">
    <titulo lingua = "pt"> TITULO 2 </titulo>
    <autor> AUTOR 2 </autor>
    <ano> 2010 </ano>
    <preco> 0.00 </preco>
  </livro>
  <livro categoria = "PROGRAMACAO">
    <titulo lingua = "pt"> TITULO 3 </titulo>
    <autor> AUTOR 3 </autor>
    <ano> 2010 </ano>
    <preco> 0.00 </preco>
  </livro>
</livraria>

```

Figura 2.1- Exemplo de um código XML (W3Schools, 2010).

Na Figura 2.1 é mostrado um exemplo de estruturação de um código na linguagem de marcação XML. Para melhor entendimento, é representado logo abaixo na Figura 2.2 um esquema em árvore do código demonstrado na Figura 2.1.

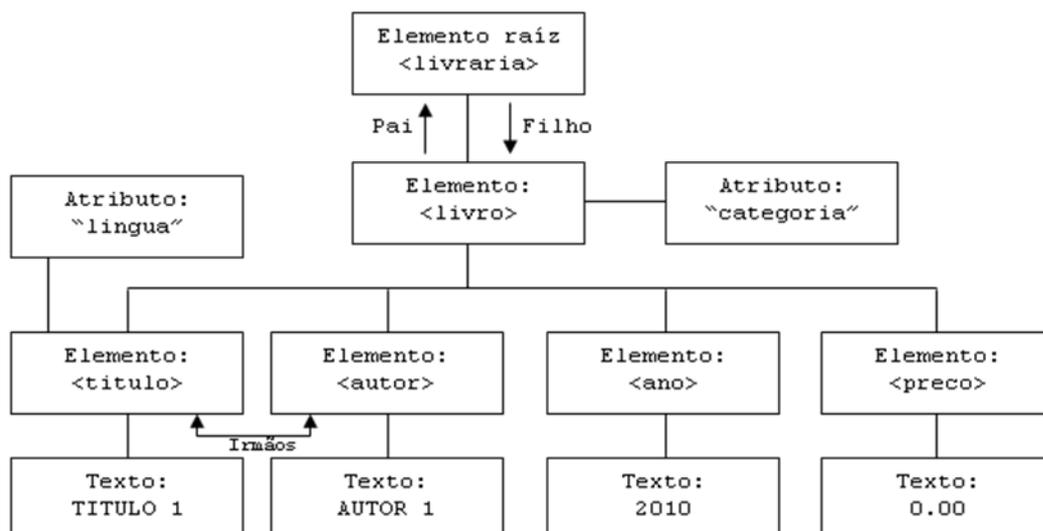


Figura 2.2 - Esquema em árvore do exemplo de código apresentado na Figura 2.1 (W3Schools, 2010).

Seguindo o esquema da Figura 2.2 na forma do topo para baixo, tem-se a livraria como raiz da árvore seguido dos nós Língua, Livro e Categoria. Abaixo de Língua, todos os elementos terão como atributo o tipo de língua escolhida. Ao lado direito de Livro, tem-se no mesmo nível o nó contendo a Categoria do livro. Abaixo de Livro, têm-se os nós Título, Autor, Ano e Preço seguidos logo abaixo por suas folhas com os textos correspondentes, respectivamente.

2.4.2 MathML

A linguagem *MathML* (*Mathematical Markup Language*), baseada na linguagem *XML*, foi desenvolvida para descrever notações matemáticas e capturar suas estruturas e conteúdos. A meta da *MathML* é disponibilizar matemática para ser processada através da *World Wide Web*, da mesma forma que a *HTML* tem disponibilizado esta funcionalidade para texto.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Figura 2.3 - Exemplo de uma expressão matemática.

A especificação da versão 1.01 da linguagem foi liberada em julho de 1999 e a versão 2.0 apareceu dentro de Fevereiro de 2001. Em Outubro 2003, a segunda edição da versão 2.0 do *MathML* foi publicada como versão final pelo grupo *W3C*, sendo a versão amplamente utilizada atualmente. Em junho de 2006, a *W3C* reiniciou pesquisas envolvendo o *MathML* para aprimorar a linguagem, lançando recentemente a recomendação *MathML* 3.0. Nesta nova versão foram introduzidas mais funcionalidades ao *MathML* 2, descritas em *W3Cc* (2010).

Esta linguagem possui dois conjuntos de elementos de marcação, um para apresentação e outro para conteúdo.

Para a codificação de conteúdo existem em torno de 75 elementos, com aproximadamente uma dúzia de atributos. Muitos desses elementos são agrupados em famílias e representam operações matemáticas e funções tais como "mais" (*plus*) e "seno" (*sin*), além de outros objetos matemáticos como "conjunto" (*set*) e "vetor" (*vector*). A codificação de conteúdo está mais voltada para aplicações tais como os sistemas algébricos tipo *Maple*¹ ou *Mathematica*² ou ainda sintetização da voz, enquanto que a codificação de apresentação se preocupa mais como a forma como as expressões serão exibidas na tela de um computador ou impressas em papel.

A seguir tem-se a formulação da equação apresentada na Figura 2.3 utilizando a linguagem *MathML*, no seu formato de conteúdo.

```

<apply>
  <eq/>
  <ci>x</ci>
  <apply>
    <divide/>
    <apply>
      <csymbol>±PlusMinus;</csymbol>
      <apply>
        <minus/>
        <ci>b</ci>
      </apply>
    </apply>
    <apply>
      <root/>
      <degree><cn>2</cn></degree>
    </apply>
    <apply>
      <minus/>
      <apply>
        <power/>
        <ci>b</ci>
        <cn>2</cn>
      </apply>
    </apply>
    <apply>
      <times/>
      <cn>4</cn>
      <ci>a</ci>
      <ci>c</ci>
    </apply>
  </apply>
</apply>
</apply>
<apply>
  <times/>
  <cn>2</cn>
  <ci>a</ci>
</apply>
</apply>
</apply>

```

Figura 2.5 - Marcação utilizando a linguagem *MathML* no seu formato de conteúdo a partir do exemplo de expressão matemática demonstrada na Figura 2.3.

XXXVXXXV—

¹ <http://www.maplesoft.com/products/Maple/index.aspx>

² <http://www.wolfram.com/products/mathematica/index.html>

Nota-se a partir da análise da Figura 2.4 e Figura 2.5 que a formulação de uma equação na linguagem *MathML*, por se tratar de uma linguagem de marcação, pode ser algo complexo. Estudos mostram que o aprendizado de lógica de programação é um desafio para estudantes de Computação. Esperar que usuários leigos aprendam uma linguagem de marcação para realizar a busca em um sistema tende a ser uma interface inadequada (Baeza-Yates, 2000), (Tobar et al., 2001), (Haden & Mann, 2003), (Neves & Coello, 2006) e (Costa et al., 2010).

2.4.3 TeX

Outra linguagem matemática comumente utilizada pela comunidade científica é a linguagem *TeX*, que apresenta uma estrutura diferenciada da linguagem *MathML* contendo elementos dispostos de maneira simples e seqüencial.

A linguagem *TeX* foi projetada para lidar com expressões matemáticas complexas de tal forma que a maioria delas sejam fáceis de serem dadas como entrada, de forma textual.

A fórmula mais simples é uma única letra como 'x', ou um único número, como '2', sendo estes circundados, respectivamente, por cifrões '\$' para serem reconhecidos pela linguagem em seu modo matemático. Caso haja algum cifrão a mais ou a menos, será apresentada uma mensagem de erro. Expressões também podem ser formuladas como o exemplo '\$(x+y) / (x-y)\$', resultando em '(x + y)/(x - y)'. Além disso, outros elementos matemáticos podem ser representados, como as letras gregas. Um exemplo é a entrada '\$\$\alpha, \beta, \gamma, \delta;\$\$', resultando nas letras gregas *alpha*, *beta*, *gamma* e *delta* em minúsculo, respectivamente,

$$\alpha, \beta, \gamma, \delta;$$

ou ainda em maiúsculo como o exemplo '\$\Gamma\$', resultando no *gamma* maiúsculo 'Γ'. A seguir um exemplo de expressão matemática em *TeX*.

```
\lim_{x\rightarrow\infty} \frac{a^x}{x^p}
```



$$\lim_{x \rightarrow \infty} \frac{a^x}{x^p}$$

Figura 2.6 - Exemplo de uma expressão matemática em *TeX*.

Analisando a estrutura da expressão matemática da Figura 2.6 nota-se a sequencialidade dos elementos da expressão. Esta disposição dos elementos assemelha-se com valores de entrada em sistemas de busca, adequando-se aos mesmos. No entanto, à medida que a expressão cresce acaba se tornando complexa, dificultando a localização de erros ou a manutenção da mesma.

2.5 Considerações Finais

Dadas as linguagens matemáticas apresentadas, nota-se a dificuldade para construir expressões matemáticas para que então sejam buscadas dentro do conteúdo das bibliotecas digitais apresentadas. Nestas circunstâncias, o usuário acaba optando por buscar pelo título do tema e não pelas fórmulas, tendo que posteriormente ler por todo o conteúdo encontrado para achar o ponto que tenha a expressão a ser analisada.

Logo, há a necessidade de que um sistema de busca de conteúdo tenha uma interface que ajude o usuário a formular equações, sem que este conheça previamente uma linguagem matemática. O capítulo 3 apresenta algumas propostas desenvolvidas na literatura e suas funcionalidades.

3

Revisão Bibliográfica

Este capítulo apresenta uma revisão bibliográfica sobre o tema abordado na monografia. Na Seção 3.1 têm-se algumas considerações sobre o que será abordado no capítulo. Na Seção 3.2 apresenta alguns métodos de entrada de informações nas ferramentas de busca e uma análise de usabilidade. Quatro ferramentas de busca de conteúdo matemático e suas interfaces são analisadas na Seção 3.3 e têm-se algumas considerações finais a cerca do que foi apresentado na Seção 3.4.

3.1 Considerações Iniciais

A especialização de sistemas de busca na Internet para um determinado tipo de conteúdo tem movido o interesse de diversos pesquisadores ao redor do mundo. Com a matemática não foi diferente, através da busca por expressões matemáticas em bibliotecas digitais, como é visto em alguns trabalhos, dentre os quais podem ser citados a *Math Web Search* (Kohlhase & Sucas, 2006), a *Whelp* (Asperti et al., 2006), *ActiveMath* (Melis et al., 2009) e a *Mathdex* (Miner & Munavalli, 2007). Antes de apresentar o funcionamento das interfaces destas ferramentas de busca, serão analisados métodos de entrada de informações.

3.2 Entrada de informações

Analisando de maneira geral, cada sistema de busca tem como meta proporcionar ao usuário uma experiência satisfatória, desde a entrada de uma informação até o resultado da busca. Deste modo, cada interface é projetada para se adequar ao tipo

de conteúdo que será procurado, seja um livro, uma música, um software, tendo assim filtros e campos auxiliares para serem utilizados na busca de forma que o resultado seja o mais próximo do que o usuário deseja.

De acordo com Zhao et al. (2008), a busca por expressões matemáticas pode ser dividida em três tipos:

- Matemática-Inconsciente: neste tipo de abordagem a ferramenta de busca ignora a estrutura da expressão e parte para a comparação de símbolos literais, como acontece no *Google Scholar*, *Zentralblatt Math* (Kruger, 2004) e *MathWorld*. Por exemplo, no *MathWorld* pode-se inserir expressões em *TeX*, sendo admitidas pelo sistema de busca como símbolos que serão comparados com as informações contidas nos documentos.
- Matemática-Consciente Sintaticamente: sistemas como a *Mathdex* e *ActiveMath* têm como entrada uma expressão cuja estrutura sintática será recuperada e assim admitida como uma expressão matemática. Isso possibilita variar a precisão das pesquisas, além do fato de trabalhar com expressões no nível de sintaxe.
- Matemática-Consciente Semanticamente: sistemas nesta categoria admitem não somente a análise sintática, mas também a semântica contida nas expressões, como acontece nos sites *Math Web Search* e *Wolfram Functions* (Wolfram, 2010). Através da análise semântica, podem-se manipular expressões para resolver a equivalência entre as mesmas, onde uma difere da outra pela análise sintática, entretanto semanticamente idênticas.

Outro fator importante é a linguagem matemática usada na entrada de expressões. Isso influenciará tanto na construção da interface quanto na experiência do usuário. As linguagens matemáticas mais comuns dentre os sistemas de buscas apresentados são o *MathML* e *TeX*. A escolha de uma ou outra determinará a metodologia de entrada e a manipulação das expressões pelo sistema. Como o *TeX* segue uma estrutura parecida com a entrada de palavras, sites utilizam essa

linguagem para diminuir a dificuldade com a qual o usuário irá lidar ao formular uma expressão, como ocorre no *MathWorld* e, em contrapartida, outros *sites* utilizam o *MathML* aproveitando sua representação gráfica das expressões, como ocorre no *Math Web Search*. Neste caso, para evitar a necessidade do conhecimento da linguagem e suas *tags*, estes *sites* proporcionam uma interface contendo os elementos já pré-definidos e bastará somente que o usuário escolha o que inserir na equação, deixando a cargo do sistema montar as *tags MathML* (exemplo *Mathdex*, *ActiveMath* e *Wolfram Functions*).

A afinidade do usuário com tais linguagens pode ser representada pela seguinte figura:

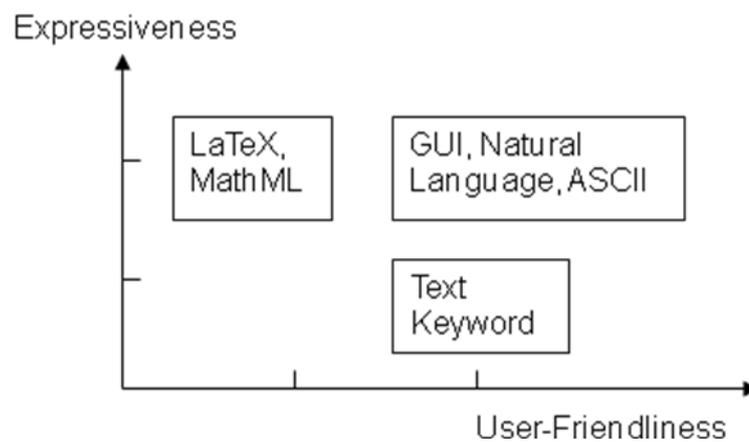


Figura 3.1 - Relação de afinidade do usuário quanto a linguagens matemáticas e outros métodos de entrada de expressões (Zhao et al., 2008).

Analisando o gráfico nota-se que à medida que aumenta horizontalmente o método de entrada torna-se mais amigável ao usuário, ou seja, ele encontra uma maior facilidade de uso ao trabalhar com este tipo de metodologia, sendo ela por meio de busca textual tradicional ou através de elementos gráficos, enquanto que quando o gráfico cresce verticalmente aumenta o nível de capacidade de expressão do usuário ao método de entrada, que no caso enquadram-se as linguagens matemáticas e afins.

Tanto o método convencional quanto o método utilizando linguagens matemáticas na entrada de informações em um sistema de busca tem pontos que favorecem um ou o outro, o que pode ser observado na tabela a seguir:

Tabela 3.1 - Pontos positivos dos métodos, convencional e utilizando linguagens matemáticas.

	Método de entrada convencional	Método usando linguagem matemática
Pontos Positivos	<ul style="list-style-type: none"> • Amplamente utilizado em sistemas de busca; • Facilidade na digitação; • Não necessita aprender uma linguagem específica. 	<ul style="list-style-type: none"> • Busca por soluções de um determinado problema; • Visualização da expressão matemática de entrada; • Linguagens como o <i>TeX</i> já são utilizadas na escrita de artigos científicos.

Evidentemente, as desvantagens são os pontos em que um tem e o outro não tem. A partir da análise destes pontos, vários trabalhos foram desenvolvidos a fim de obter uma interface que seja adequada para a maioria dos usuários, além do próprio sistema em si.

3.3 Ferramentas de busca

As ferramentas de busca que serão apresentadas neste capítulo são a *Math Web Search*, *Whelp*, *ActiveMath* e *Mathdex*. Estas terão, primeiramente a cada tópico, uma descrição resumida e em seguida o foco principal que são suas interfaces.

3.3.1 Math Web Search

A *Math Web Search* é uma ferramenta de busca de conteúdo matemático que tem como base de dados o repositório *CONNEXIONS Project* (Connexions, 2010), mais de 3.400 artigos, e cerca de 53.000 termos (ou 77.000 se forem considerados os

subtermos). O projeto inclui ainda um índice contendo 87.000 fórmulas em formato *MathML*, obtidas na página *functions* do *Wolfram*. Nesta ferramenta o usuário monta a expressão de acordo com os elementos apresentados. Iniciada a busca, expressões semelhantes são encontradas através da técnica de n-grama, detalhada em Kukich (1992).

A interface da *Math Web Search* possui um menu com botões no qual o usuário escolhe as estruturas para montar a sua expressão. Após montada a expressão, o usuário pode fazer modificações na mesma de forma textual. Em seguida, o usuário seleciona a linguagem de entrada e então executa a busca.

Na Figura 3.2 tem-se uma imagem da ferramenta.

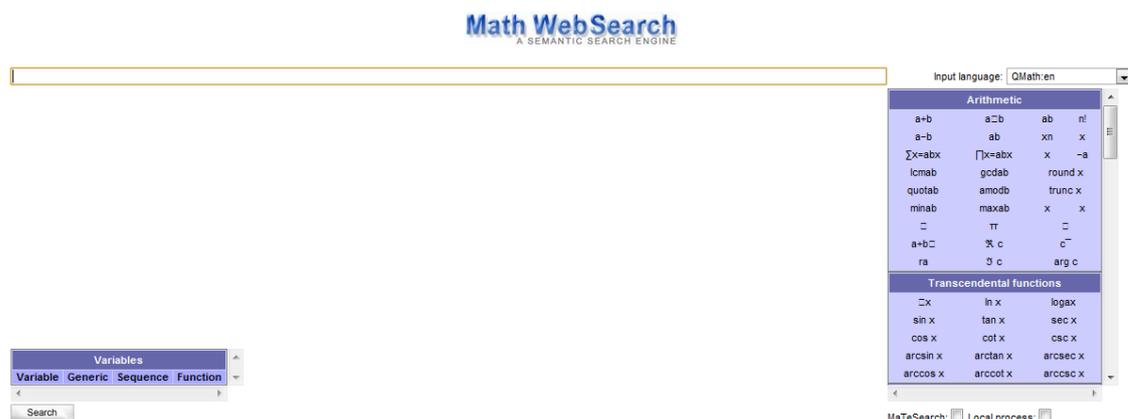


Figura 3.2 - Protótipo da *Math Web Search*.

3.3.2 Whelp

A *Whelp* é um protótipo de uma ferramenta desenvolvida para buscar expressões na biblioteca de conteúdo matemático do *Coq*, que é um gerenciador formal de provas. O *Coq* provê uma linguagem formal para a escrita de definições matemáticas, algoritmos executáveis e teoremas, juntamente com um ambiente interativo para o desenvolvimento de provas matemáticas. Para filtragem de expressões esta ferramenta utiliza metadados, que são uma forma de descrever

arquivos extraídos inicialmente de forma eletrônica e automática, como o exemplo do ID3 *tags* contido em arquivos mp3, armazenando informações como taxa de reprodução, nome do autor e álbum da música, dentre outras.

A ferramenta carrega algumas funcionalidades do *software Matita* (Asperti et al., 2007), um provador de teorema desenvolvido pela equipe do *HELM* (Asperti et al., 2001), tendo como foco o trabalho de resolver ambigüidades de expressões informadas pelo usuário. Um exemplo é a expressão $\sin ? = ?$ que ao ser inserida na busca e submetida, retorna uma tela informando ser uma equação ambígua, como é mostrado na Figura 3.3.

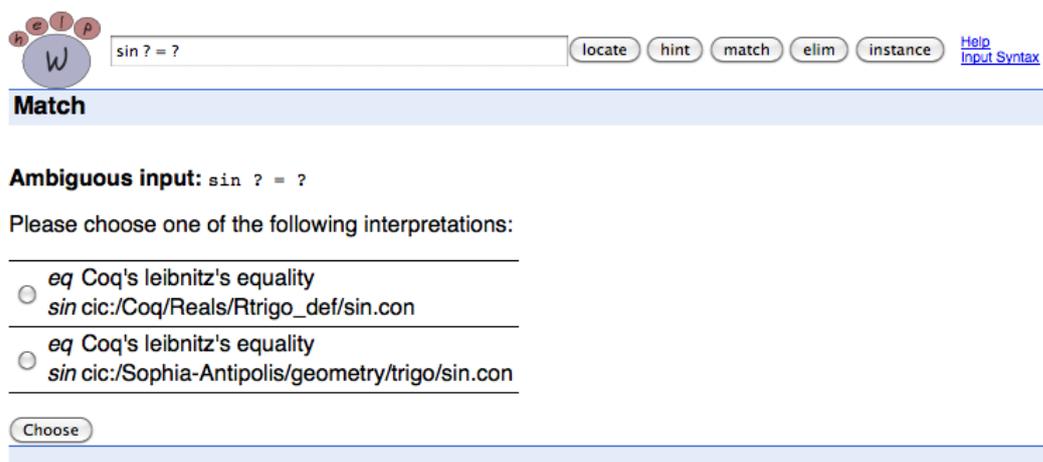


Figura 3.3 - Tela mostrada para a expressão ambígua $\sin ? = ?$.

Ao contrário da interface da *Math Web Search*, a *Whelp* não possui uma interface gráfica para construção de expressões. Um usuário que acesse a ferramenta pode não conhecer a sintaxe de entrada da mesma, que é a mesma usada no *Coq* (Bertot & Casteran, 2004).

O protótipo da *Whelp* é exibido na Figura 3.4.

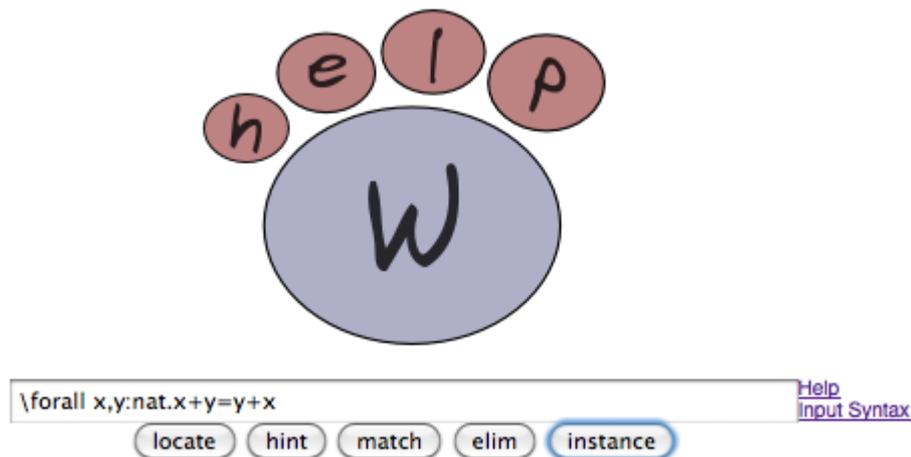


Figura 3.4 - Protótipo da *Whelp*.

3.3.3 ActiveMath

ActiveMath é uma plataforma inteligente de ensino virtual do qual apresenta uma série de recursos da *Web Semântica*. É baseado no projeto *LeActiveMath* (Libbrecht & Melis, 2006), fundado pela *EU & ATuF* (ME 1136/5-1) que por sua vez é suportada pela *German National Science*. Possui várias formas de busca, usando somente texto ou expressões matemáticas formuladas através de uma interface parecida com a do software *Microsoft Equation*. Utiliza a linguagem *OpenMath* (Caprotti et al., 2002) devido à sua estrutura de marcação semântica para fórmulas matemáticas, além de ser suportada por diversos sistemas de álgebra computacional (CAS) como por exemplo *Maple*, *Mathematica*, *Maxima*³, *Yacas*⁴, *WIRIS*⁵, sendo as três últimas utilizadas nas buscas do sistema da *ActiveMath*.

Sua interface de busca possui dois modos principais de busca: simples e avançado. Na busca simples tem como entrada uma caixa de texto, podendo ser inserido títulos de conteúdos, palavras ou expressões em forma textual, sem o uso

xlvxlv_____

³ <http://maxima.sourceforge.net/>

⁴ <http://yacas.sourceforge.net/homepage.html>

⁵ <http://www.wiris.com/>

de uma linguagem matemática específica. A Figura 3.5 apresenta a interface em seu modo simples.

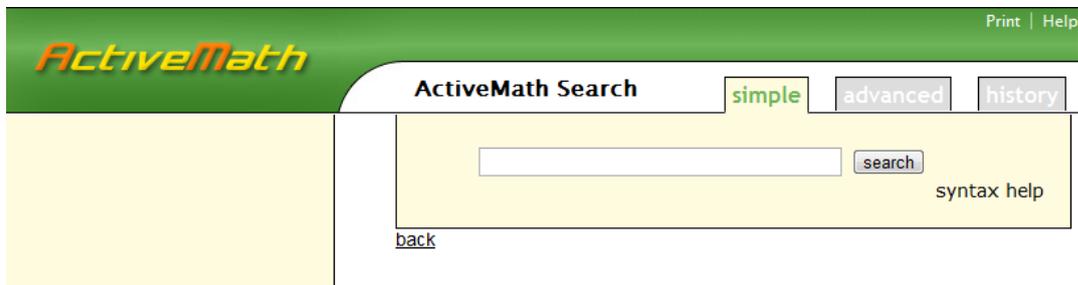


Figura 3.5 - Tela da ferramenta em modo simples.

Já na busca avançada apresenta mais algumas funcionalidades, como a possibilidade de inserir expressões matemáticas através de um editor de equações. O usuário digita os números e variáveis, podendo escolher visualmente funções, letras gregas e demais estruturas através dos botões do menu, que por sua vez é subdividido em abas, de acordo com a estrutura desejada. Além disso, tem como opção a verificação da sintaxe da expressão, ver os histórico de buscas e a escolha da língua no qual fará a busca. Para ter acesso a ferramenta é necessário efetuar um cadastro gratuito.

A Figura 3.6 apresenta a interface em modo avançado da ferramenta utilizando uma expressão matemática.

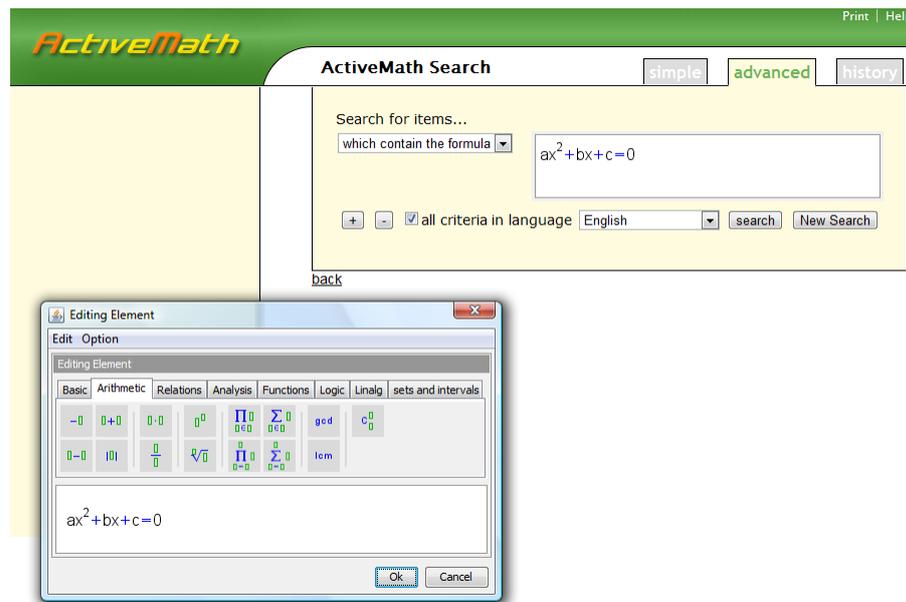


Figura 3.6 - Tela da ferramenta em modo avançado utilizando uma expressão matemática como entrada.

3.3.4 Mathdex

A *Mathdex* é um protótipo de uma ferramenta de busca de conteúdo matemático baseada em um algoritmo de busca e *workflow* de normalização, descritos em (Miner & Munavalli, 2007). O usuário entra com expressões matemáticas através de um editor gráfico de equações, podendo adicionar textos em uma caixa de texto *HTML* padrão. Os resultados são apresentados ao usuário em forma de lista, com resumos dos documentos e a expressão mais semelhante à buscada, para cada documento resultante. As expressões matemáticas contidas nas páginas resultantes são baseados na sintaxe de apresentação da linguagem *MathML* ou em imagens renderizadas⁶. Em 2007, esta ferramenta de busca já indexava em torno de 25.000 documentos do *arXiv* (Warner, 2005), 12.000 páginas da *Wikipedia* de conteúdo

xlviixlvii

⁶ Renderizar é uma palavra proveniente do verbo “*to render*” adaptada para o português que significa transformar um conjunto de dados $n -$ dimensionais em uma imagem bidimensional por meio de técnicas de projeção.

matemático, aproximadamente 1.300 páginas do *Connexions* e em torno de 1000 páginas da *MathWorld*.

A interface desta ferramenta apresenta uma área de edição das expressões matemáticas e, ao topo desta, uma barra contendo botões sendo que, para cada botão, é apresentado um submenu contendo elementos de um determinado grupo de termos matemáticos. Abaixo da área de edição, existe uma barra de texto para inserção de palavras chave para caso o usuário queira especificar mais ainda a sua pesquisa. Como a ferramenta não está disponível publicamente, não foi possível executar testes de formulação das equações, bem como explorar as suas funcionalidades.

A Figura 3.7 apresenta a interface da *Mathdex*.



Figura 3.7 - Interface da ferramenta de busca matemática *Mathdex*.

3.4 Considerações finais

Fazendo uma análise geral das ferramentas apresentadas, notam-se algumas semelhanças e diferenças dentre suas interfaces, como pode ser visto na tabela 3.2 a seguir:

Tabela 3.2 - Semelhanças e diferenças dentre as interfaces das ferramentas apresentadas.

Ferramenta de busca	Método de entrada	Linguagem matemática	Composição do Menu
<i>Math Web Search</i>	Textual + Gráfico	<i>MathML, TeX</i> e outras	Menus expansíveis contendo botões, para cada tipo de termo
<i>Whelp</i>	Textual	Linguagem do <i>Coq</i>	Botões auxiliares
<i>ActiveMath</i>	Textual ou Gráfico	<i>OpenMath</i>	Menu com abas contendo botões, para cada tipo de termo
<i>Mahdex</i>	Textual + Gráfico	<i>MathML</i>	Botões expansíveis contendo sub-menus

Alguns pontos devem ser levados em consideração a respeito das ferramentas e suas interfaces, como:

- *Math Web Search*
 - a expressão não é montada da forma como é feita a leitura. Por exemplo, para montar uma expressão do tipo: $\sum_{i=1}^n \prod_{j=1}^i \frac{1}{x_j}$, o usuário começaria clicando no símbolo da fração, para depois clicar no produtório e em seguida no somatório. Se após produzir uma equação semelhante à usada acima, o usuário clicar por exemplo, no botão $a+b$, a ferramenta entenderá que toda a expressão produzida é o a , acrescentando apenas $+b$ na frente da expressão;
 - enquanto produz a expressão, o usuário seleciona uma linguagem de entrada. A expressão é então construída de acordo com a linguagem selecionada, sendo possível a alteração da expressão na forma textual da linguagem. Contudo, o usuário pode não estar familiarizado com nenhuma das opções apresentadas. Não fica clara qual a diferença e o escopo da busca entre as linguagens;

- a mesma expressão pode ser encontrada em múltiplos menus da ferramenta, como por exemplo o somatório, que está presente na aba *Arithmetic* e na aba *arith1*.
- *Whelp*
 - A falta de uma interface gráfica pode dificultar na formulação da expressão;
 - Edição textual de expressões complexas pode se tornar uma tarefa difícil;
 - A apresentação das expressões ambíguas é confusa;
 - O fato do *Whelp* também não estar funcionando pode desencorajar novos usuários.
- *ActiveMath*
 - Necessidade de um cadastro para utilização da ferramenta;
 - Uso de janela *popup* para a tela de busca;
 - Necessidade de um *plugin Java* para utilização do editor gráfico de equações.
- *Mathdex*
 - Não está disponível ao público;
 - Dois tipos de métodos de entrada numa mesma tela;
 - Botões contendo submenus, que por sua vez contém termos

4

Interface para Ferramenta de Busca Matemática

Este capítulo apresenta a proposta desta monografia. É organizado da seguinte forma: A Seção 4.1 apresenta uma breve descrição da modelagem orientada a objetos. Na Seção 4.2 são descritas informações sobre a UML e, por fim, é apresentada na Seção 4.3 a modelagem do projeto através de diagramas da UML.

4.1 Modelagem orientada a objetos

Atualmente o paradigma da orientação a objetos é amplamente utilizado no desenvolvimento de sistemas de software das mais diversas áreas. Um paradigma é uma forma de abordar um problema (Bezerra, 2007). Logo, o paradigma da orientação a objetos é uma forma de abordar os aspectos de um sistema de modo que sejam visíveis as suas interações aos olhos da equipe de desenvolvedores, facilitando no processo de desenvolvimento do software. A visão deste paradigma quanto ao sistema de software é de agentes interconectados chamados objetos, sendo que cada um deles responsável por realizar tarefas específicas. É através destas interações entre os objetos que uma tarefa computacional é realizada.

A seguir têm-se alguns conceitos importantes com relação à orientação a objetos, definidos em Bezerra (2007):

- **Classe:** é uma descrição dos atributos e serviços comuns a um grupo de objetos. Sendo assim, pode-se entender uma classe como sendo um molde a partir do qual objetos são construídos. Ainda sobre terminologia, diz-se que um objeto é uma instância de uma classe. É importante notar que uma classe é uma abstração das características de um grupo de coisas do mundo real.

Na maioria das vezes, as coisas do mundo real são muito complexas para que todas as suas características sejam representadas em uma classe. Além disso, para fins de modelagem de um sistema, somente um subconjunto de características pode ser relevante. Portanto, uma classe representa uma abstração das características relevantes do mundo real.

- **Mensagens:** são estímulos enviados de um objeto a outro, requisitando que ele realize alguma operação, ou seja, é pela troca de mensagens que os objetos executam tarefas dentro do sistema ao qual estão inseridos.
- **Abstrações:** é qualquer modelo o qual permite gerenciar a complexidade e concentrar a atenção nas características essenciais de um objeto, dependendo do contexto inserido.
- **Encapsulamento:** é uma forma de restringir o acesso ao comportamento interno de um objeto. Um objeto que precise da colaboração de outro objeto para realizar alguma tarefa simplesmente envia uma mensagem a este último. Através do encapsulamento, a única coisa que um objeto precisa saber para pedir a colaboração de outro objeto é conhecer a sua interface.
- **Polimorfismo:** indica a capacidade de abstrair várias implementações diferentes em uma única interface. A abstração é aplicada neste contexto já que um objeto pode enviar a mesma mensagem para objetos semelhantes, mas que implementam a sua interface de formas diferentes.
- **Herança:** pode ser vista como um nível de abstração acima da encontrada entre classes e objetos. Neste caso, classes semelhantes são agrupadas em hierarquias. Cada nível de uma hierarquia pode ser visto como um nível de abstração. Esse mecanismo facilita o compartilhamento de comportamento comum entre um conjunto de classes semelhantes.

4.2 Linguagem de Modelagem Unificada (UML)

A *UML* é uma linguagem visual para modelar sistemas orientados a objetos. Através dos elementos gráficos definidos nesta linguagem podem-se construir diagramas que representam diversas perspectivas de um sistema. Cada elemento gráfico possui uma sintaxe (isto é, uma forma predeterminada de desenhar o elemento) e uma semântica que define o que significa o elemento e para que ele deva ser utilizado.

A *UML* é independente tanto de linguagens de programação quanto de processos de desenvolvimento. Esse é um fator importante para a utilização da *linguagem*, pois diferentes sistemas de software requerem diferentes abordagens de desenvolvimento.

No desenvolvimento de um sistema pode-se analisar o sistema em diversas perspectivas, denominadas visões. Os autores da *UML* sugerem que um sistema pode ser descrito por cinco visões interdependentes desse sistema (Booch et al., 1999). Cada visão enfatiza aspectos diferentes do sistema. Dependendo das características e complexidade do sistema, nem todas as visões são necessárias, podendo então destacar as seguintes:

- Visão de Casos de Uso: descreve o sistema de um ponto de vista externo como um conjunto de interações entre o sistema e os agentes externos ao sistema. Esta visão é criada inicialmente e direciona o desenvolvimento das outras visões do sistema.
- Visão de Projeto: enfatiza as características do sistema que dão suporte tanto estrutural quanto comportamental, às funcionalidades externamente visíveis do sistema.

No processo de desenvolvimento utilizando a *UML* como linguagem de suporte à modelagem são criados documentos, estes sendo textuais ou gráficos, denominados artefatos. Os artefatos gráficos produzidos durante o

desenvolvimento são definidos através da utilização dos diagramas da *UML*, apresentados na figura 4.1.

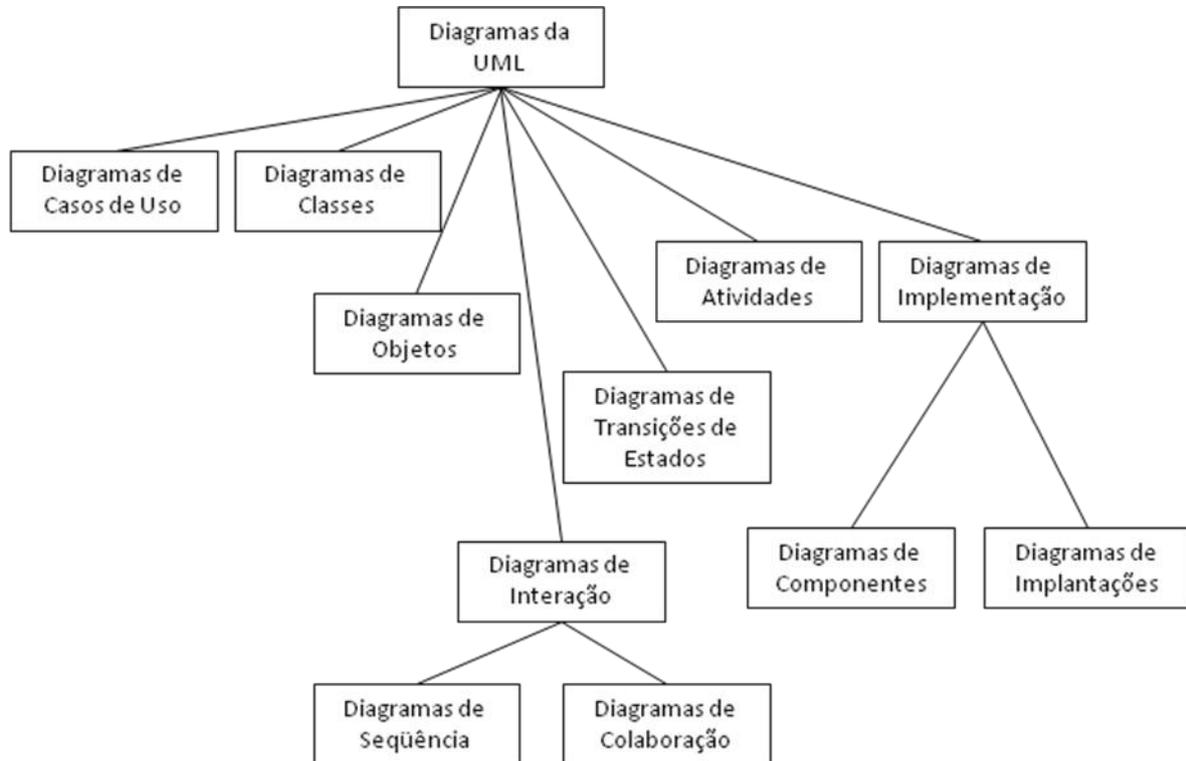


Figura 4.1 - Diagramas definidos pela *UML* (Bezerra, 2007).

Neste trabalho foram utilizados alguns destes diagramas para a modelagem da interface proposta.

4.3 Projeto

4.3.1 Escopo do projeto

Em matemática, uma equação é uma sentença aberta expressa por uma igualdade envolvendo expressões matemáticas. Expressão matemática é uma combinação de números, operadores, símbolos gráficos (como colchetes e parênteses) e variáveis

livres ou ligadas agrupadas de forma significativa de modo a permitir a verificação de valores, formas, meios ou fins.

As equações normalmente propõem um problema sobre sua validade e são geralmente compostas por incógnitas e coeficientes. Os coeficientes são entidades matemáticas conhecidas (números reais, números inteiros, conjuntos, funções entre outros). Resolver a equação, ou seja, o problema por ela proposto, consiste em determinar quais são os elementos de um determinado conjunto (o das possíveis soluções) que tornam a equação verdadeira.

Alguns exemplos de equações:

- Equações polinomiais de uma variável são equações desta forma $f(x) = 0$, onde $f(x)$ é um polinômio em x ;
- Equações podem ter por incógnita uma função, por exemplo:

Determinar as possíveis funções contínuas $f : R \rightarrow R$ tais que:

$$\forall x \forall y, f(x + y) = f(x) + f(y)$$

ou ainda:

$$f(f(x)) = e^x;$$

- Equações diferenciais possuem uma função como uma incógnita e envolvem derivadas desta função:

$$f'(x) = f(x);$$

- Equivalentemente, equações integrais possuem como incógnita uma função e envolvem integrais:

$$\int_0^t f(s) ds = 2f(t) - 1;$$

- Sistemas de equações são duas ou mais expressões que devem ser resolvidas simultaneamente:

$$\begin{cases} x - y = 7 \\ xy = 30 \end{cases}$$

A partir do estudo destas informações foi possível obter os artefatos para a modelagem da interface proposta, que por sua vez será apresentada a seguir.

4.3.2 Modelagem

Foram propostos alguns diagramas após a análise de diferentes expressões matemáticas e interfaces das ferramentas de busca apresentadas, além de programas de montagem de equações, como o *Microsoft Equation*. Estes diagramas serão apresentados em visões, sendo eles de caso de uso, classe e de sequência.

4.3.2.1 Visão de Interface

Em um primeiro momento, é apresentada ao usuário a possibilidade de edição de expressões matemáticas através de elementos gráficos e exclusão de uma expressão, caso já tenha formulado uma. Os elementos escolhidos são enviados para um *container* e então o usuário pode submeter a expressão para enviar esta informação para busca, caso a interface esteja acoplada a um sistema de busca. O caso de uso é apresentado na figura 4.2.



Figura 4.2 - Caso de uso da interface.

Uma classe principal chamada Interface instancia objetos das classes TagLib, BarraBotao (o *container* referido no parágrafo anterior) e Menu. A classe Interface possui um método para consultar *tags* em *MathML* encontradas na TagLib, um método para atribuir ou recuperar o objeto do tipo BarraBotao atual e um método para recuperar a barra de botões inicial da interface. O diagrama de classe é apresentado na figura 4.3.

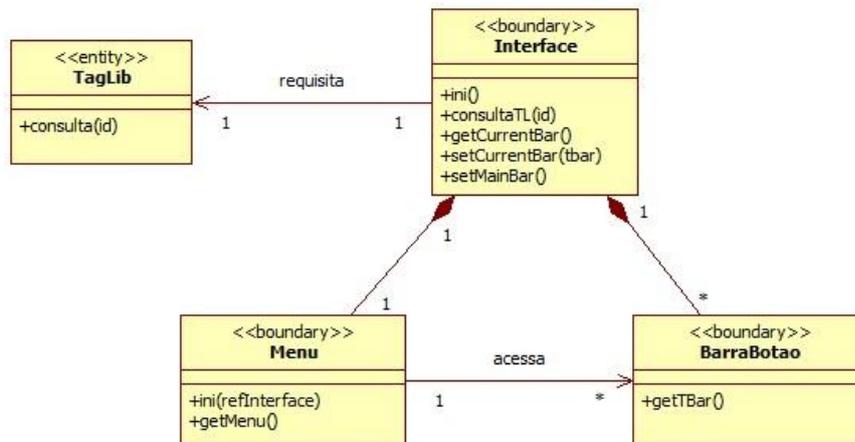


Figura 4.3 - Diagrama de classe referente a visão de Interface.

Após instanciados os objetos dos quais a classe Interface fará uso, caso o usuário submeta uma expressão para busca, o objeto Interface requisita o código *MathML* do objeto BarraBotao, para que então esta informação seja enviada para o sistema de busca acoplado à interface. A troca de mensagens dentre os métodos das classes é descrito no diagrama de seqüência apresentado na figura 4.4.

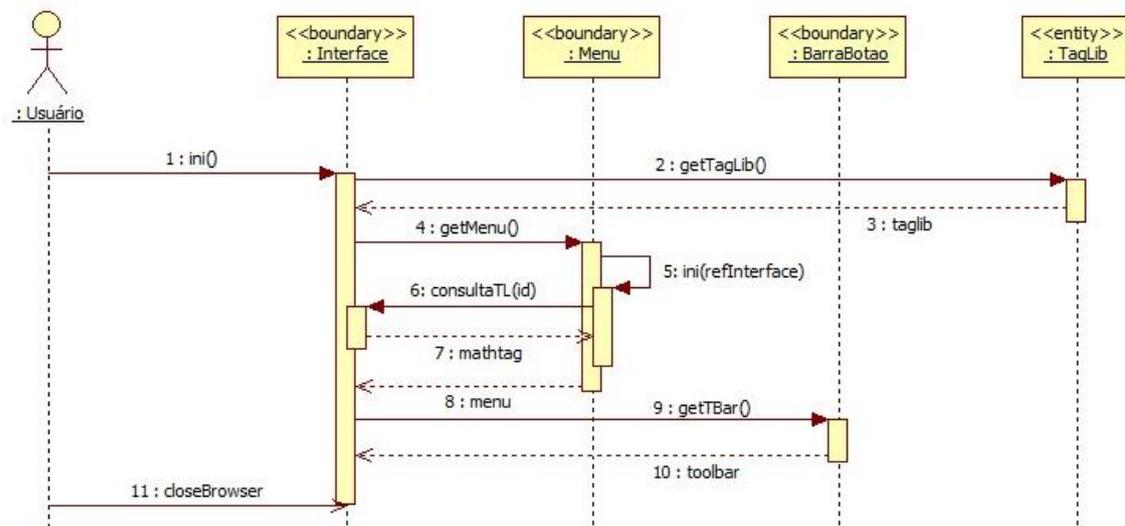


Figura 4.4 - Diagrama de seqüência da visão de Interface.

4.3.2.2 Visão de Menu

O usuário pode editar a sua expressão através de um menu contendo funções, símbolos e entrada de valores, sendo estes numéricos ou variáveis. Ele escolhe e edita estes elementos graficamente, de acordo com o que lhe é pedido em cada caso, podendo ainda adicionar, remover e alterar novos elementos. O caso de uso para esta visão é mostrado na figura 4.5.

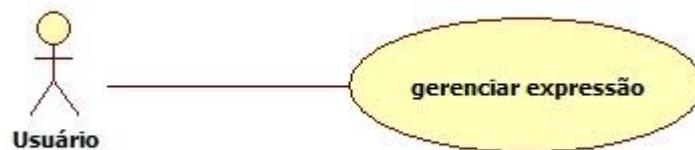


Figura 4.5 - Caso de uso para a visão Menu.

A classe Menu é composta pelas classes Funcoes, Simbolos e Valor. As classes Funcoes e Simbolos são compostas pela classe SubMenu, sendo esta definida de acordo com o tipo escolhido na instanciação de um objeto da mesma. O diagrama de classes referente à visão Menu é apresentado na figura 4.6.

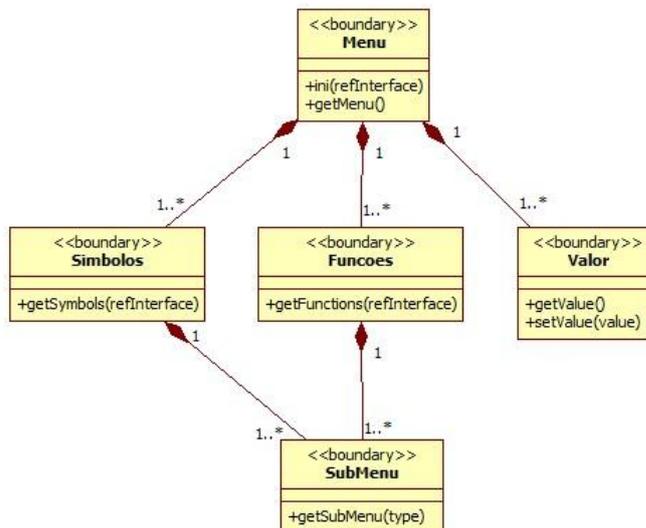


Figura 4.6 - Diagrama de classe para a visão de Menu.

O objeto da classe Menu é instanciado tendo como atributo a referência da classe principal e esta referência então é passada para os demais objetos instanciados, que no caso são os submenus para cada tipo de elemento matemático. O diagrama de sequência para a visão Menu é mostrado na figura 4.7.

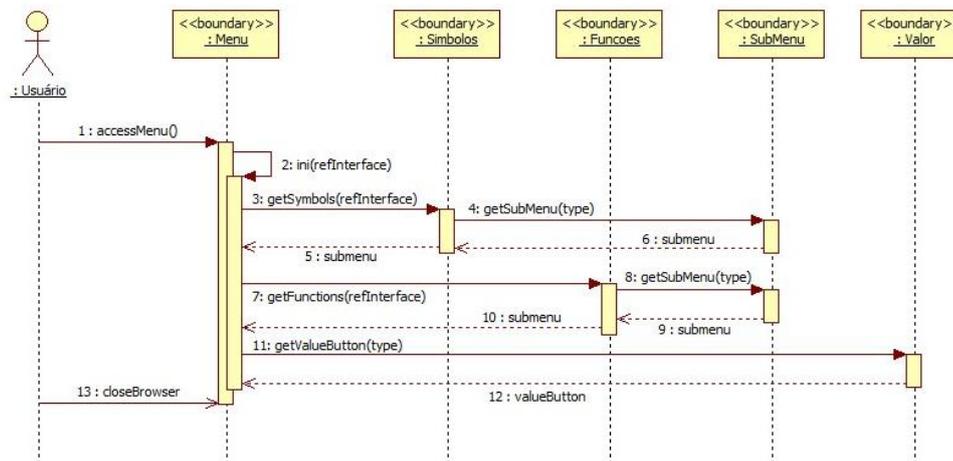


Figura 4.7 - Diagrama de sequência para a visão de Menu.

4.3.2.3 Visão Símbolo

O usuário escolhe um símbolo. Os tipos de símbolos disponíveis são: operadores, funções trigonométricas, relacionais, setas, teoria dos conjuntos e lógicos. O elemento contendo o símbolo escolhido é então passado para o contêiner correspondente. O caso de uso desta visão é mostrado na figura 4.8.

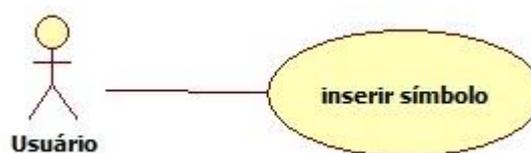


Figura 4.8 - Caso de uso da visão Símbolo.

A classe Simbolos é composta por SubMenus, estes sendo definidos no momento da instanciação, através do atributo *type*. Definido o tipo de submenu, são instanciados objetos da classe BtSimbolo, que por sua vez contém os atributos *title* e *mathtag*. O *title* definirá a representação gráfica do símbolo ao elemento,

enquanto que a *mathtag* detém a *tag* MathML para posteriormente ser agregada a outras *tags*, dependendo da expressão matemática formulada pelo usuário. O diagrama de classe para a visão Símbolo é mostrado na figura 4.9.

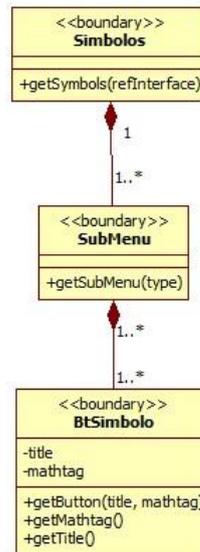


Figura 4.9 - Diagrama de classe da visão Símbolo.

A classe SubMenu faz uma chamada no método *consultaTL()* da classe principal para obter uma *tag MathML* da TagLib, passando como parâmetro o *id* da *tag*. Obtida a *mathtag*, um objeto do tipo BtSimbolo é instanciado passando como parâmetro o título do elemento e a *mathtag*. Uma nova chamada é feita para a classe principal obtendo-se o contêiner atual para inserção do elemento criado. O objeto BtSimbolo é então retornado para o objeto SubMenu. O diagrama de sequência desta visão é mostrado na figura 4.10.

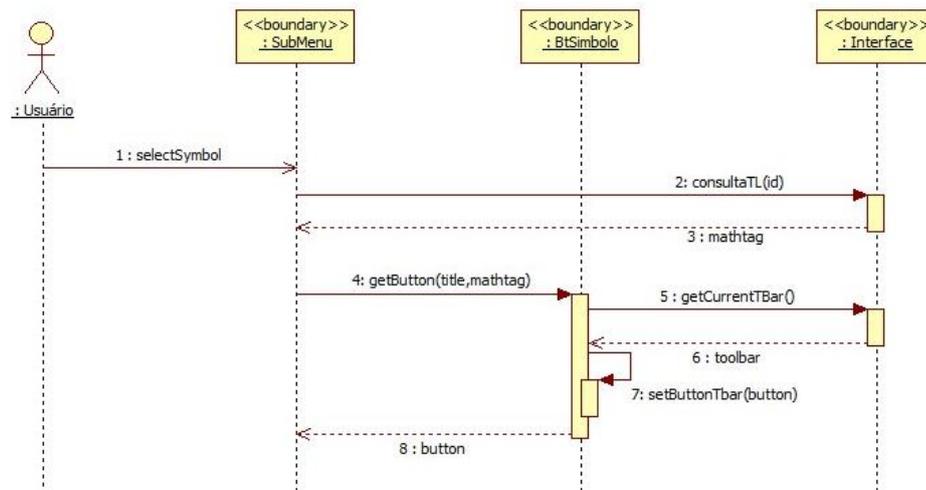


Figura 4.10 - Diagrama de seqüência da visão Símbolo.

4.3.2.4 Visão Função

O usuário escolhe uma função. Os tipos de funções estão subdivididos entre os grupos: 1-elemento, 2-elementos, 3-elementos, 4-elementos e matriz. O elemento contendo a função escolhida é então passada para o contêiner correspondente. O caso de uso desta visão é mostrado na figura 4.11.

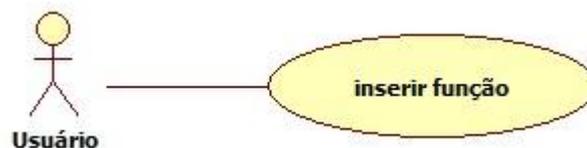


Figura 4.11 - Caso de uso da visão Função.

Da mesma forma que ocorra na classe Simbolos, a classe Funcao é composta por SubMenu's, estes sendo definidos no momento da instanciação, através do atributo *type*. Os tipos *E1*, *E2*, *E3*, *E4* e *Matrix* têm estes nomes devido ao fato da quantidade de elementos envolvidos na edição de uma função, como exemplo da função fatorial que possui um elemento, enquanto que na função limite possui três elementos. Após a criação do submenu, são instanciados objetos da classe BtFuncao, que por sua vez contém os atributos *title*, *tagStart* e *tagEnd*. O *title* definirá a representação gráfica da função ao elemento, a *tagStart* possui a *tag*

MathML inicial e a *tagEnd* a *tag MathML* final, sendo estas delimitadoras do escopo da função escolhida. O diagrama de classe para a visão Função é mostrado na figura 4.12.

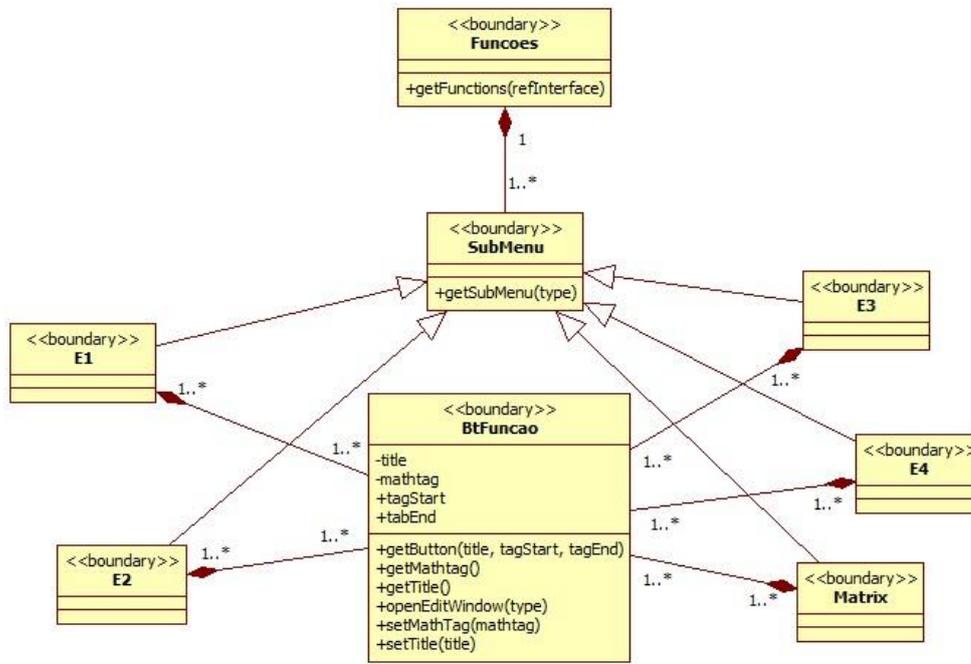


Figura 4.12 - Diagrama de classe da visão Função.

A classe *SubMenu* faz uma chamada no método *consultaTL()* da classe *Interface* para obter uma *tag MathML* da *TagLib*, passando como parâmetro o *id* da *tag*. Obtidas as *mathtag's*, um objeto do tipo *BtFuncao* é instanciado passando como parâmetro o título do elemento, a *tagStart* e a *tagEnd*. Uma nova chamada é feita para a classe principal obtendo-se o contêiner atual para inserção do elemento criado. O objeto *BtFuncao* é então retornado para o objeto *SubMenu*. O diagrama de sequência desta visão é mostrado na figura 4.13.

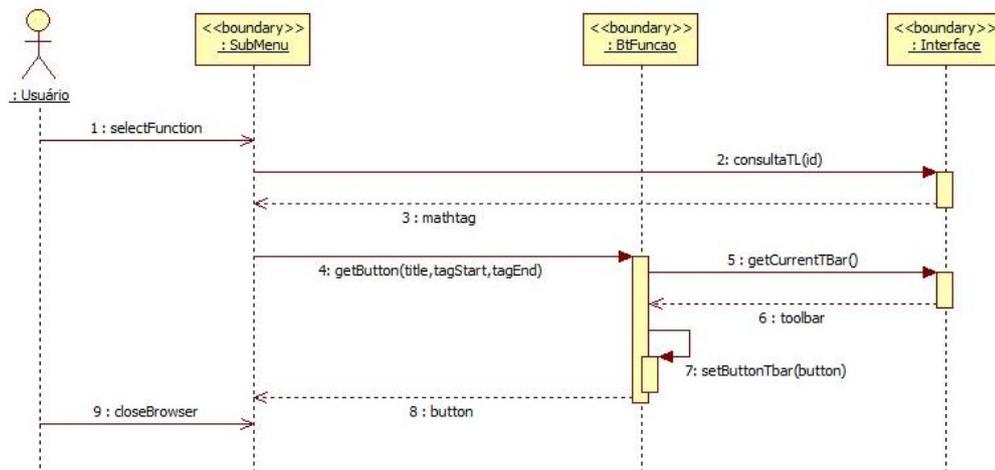


Figura 4.13 - Diagrama de sequência da visão Função.

4.3.2.5 Visão de edição de uma função

Após adicionar um elemento do tipo função, o usuário então aciona a edição deste elemento, podendo adicionar, remover e alterar novos elementos matemáticos, como é mostrado no caso de uso da figura 4.14.

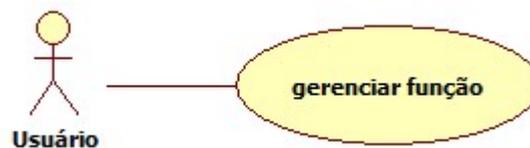


Figura 4.14 - Caso de uso da visão de edição de uma função.

O objeto da classe BtFuncao requisita um objeto da classe JanelaEdicaoFunc, passando como atributo a sua referência. A classe JanelaEdicaoFunc é composta por objetos da classe Elemento. Cada objeto Elemento faz uma chamada de um novo objeto da classe JanelaToolbar, gerando assim um novo contêiner para inserção de novos elementos, resultando num processo recursivo de edição de novas funções dentro de uma função. O diagrama de classe desta visão é mostrado na figura 4.15.

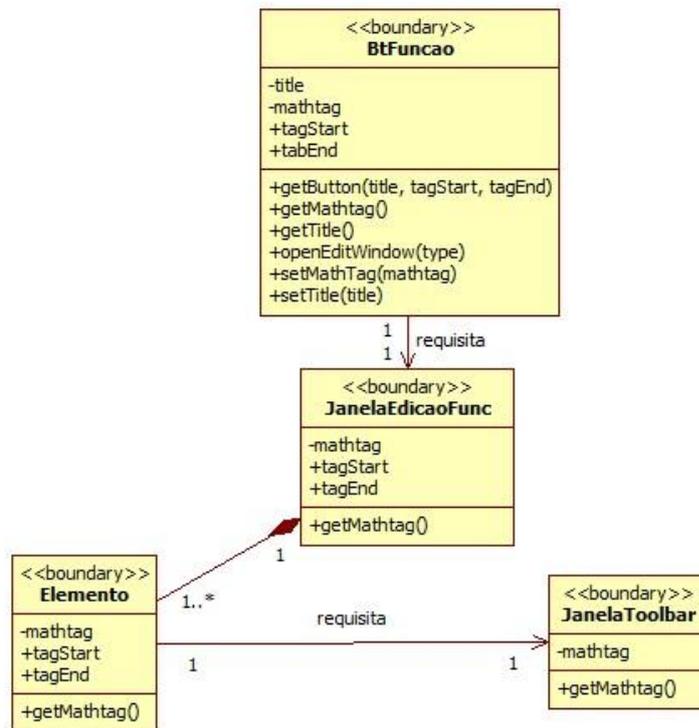


Figura 4.15 - Diagrama de classe da visão de edição de uma função.

Através da troca de mensagens entre os objetos envolvidos no processo de edição da função que são obtidas as *tags MathML* parciais e, a medida que o usuário vai confirmando as entradas de elementos matemáticos, ao término da edição é obtido um código *MathML* estruturado e completo. O diagrama de sequência desta visão é apresentado na figura 4.16.

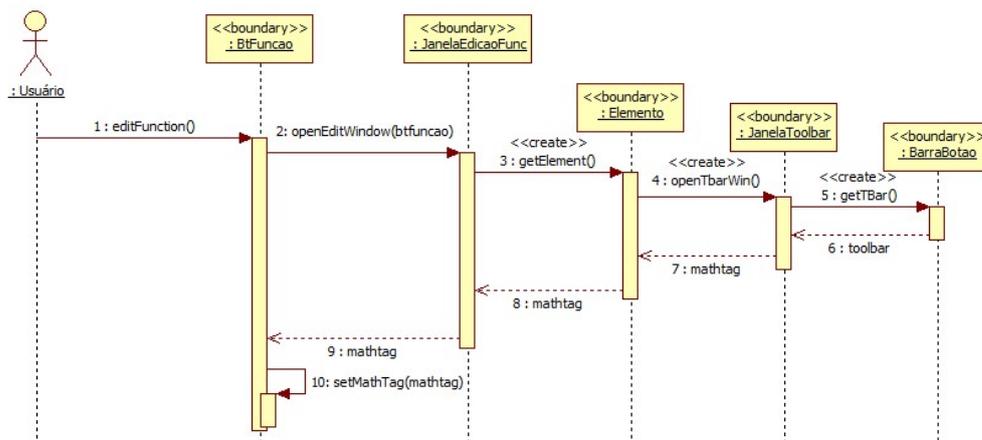


Figura 4.16 - Diagrama de sequência da visão de edição de uma função.

5

Resultados

Este capítulo apresenta na Seção 5.1 considerações a cerca do termo desenvolvimento web, na Seção 5.2 tem-se a apresentação do framework ExtJS. Na Seção 5.3 é analisada uma estrutura proposta para o tratamento de tags MathML, logo em seguida é mostrado na Seção 5.4 como foi o desenvolvimento da interface analisando os pontos negativos das interfaces apresentadas no Capítulo 3. Finalmente, são feitas considerações finais na Seção 5.5 a respeito da aplicabilidade da interface desenvolvida a um sistema de busca matemática.

5.1 Considerações iniciais

Desenvolvimento web é o termo utilizado para descrever o desenvolvimento de sites, na Internet ou numa intranet. Normalmente está associado à programação e marcação, configuração e trabalho realizado por de trás dos sites, mas também pode ser usado para se referir ao projeto visual das páginas. Estas podem variar desde simples páginas estáticas a aplicações ricas, comércios eletrônicos ou redes sociais.

As páginas web estão cada vez mais atrativas contendo elementos dinâmicos, imagens bem trabalhadas, sons, animações, agregando os conceitos de usabilidade com o planejamento da interface, garantindo que o usuário final atinja seus objetivos de forma agradável e intuitiva. Tudo isso só foi possível com o aparecimento de novas tecnologias como, por exemplo, os *frameworks* para *Javascript*, animações em *Flash* e CSS avançado, resultando na chamada *Web 2.0*.

5.2 ExtJS

O *ExtJS* é um *framework* de código livre criado por *Jack Slocum*, disponibilizado sob a licença *LGPL*, ou seja, as aplicações criadas utilizando este *framework* também necessitam ser de código livre. No entanto pode ser comprada uma licença para o desenvolvimento de aplicativos comerciais. O *ExtJS* é utilizado na construção de aplicativos web interativos baseados em *AJAX*, *DHTML* e *DOM*.

Criado originalmente como uma extensão do *Yahoo! User Interface (YUI)* chamava-se *yui-ext* e funcionava somente em conjunto com o *YUI*, que era a base. Com o desenvolvimento do *yui-ext*, foram incorporados diversos adaptadores que funcionavam de ponte entre o *yui-ext* e outros *frameworks* de *javascript*, como o *Prototype* e *jQuery*. Ao ser lançada a versão 1.0 final do *yui-ext*, este passou a ser chamado *ExtJS* e não era mais uma extensão do *YUI*, funcionando sem a base desta. Na versão 1.1.1 o *ExtJS* passou a ser oficialmente suportado pelos diversos navegadores disponíveis na *web*.

Atualmente o *ExtJS* está em sua versão 3.2.1 e além de suportar os *frameworks* já citados anteriormente através de adaptadores, ele pode ser trabalhado juntamente com o *Adobe AIR* e o *Google Web Toolkit (GWT)*. Além disso, no site do *ExtJS* existe um ambiente de desenvolvimento (*IDE*) chamado *Ext Designer*, onde os componentes são manipulados graficamente tornando o processo de construção de interfaces mais ágil. No entanto o *Ext Designer* não é livre, podendo ser utilizado por um período de avaliação.

Com o uso deste *framework* foi possível obter uma interface robusta e dinâmica, devido à sua facilidade ao se trabalhar com orientação a objetos, além de temas ricos em detalhes através de arquivos *CSS* disponíveis na própria biblioteca.

Com a utilização de janelas, botões, painéis e barras de ferramentas tornou-se possível a criação de uma interface mais atrativa, com opções de carregar e soltar itens em outros componentes, atualização de partes da página através do *AJAX*, além do fato deste *framework* ser suportado por muitos navegadores atuais.

As demais informações sobre este *framework* podem ser encontradas no livro *Learning ExtJS* (Frederick, 2008).

5.3 TagLib

A TagLib foi uma classe criada para prover *tags* do *MathML* para as demais classes do projeto. Escrita em *Javascript*, ela possui um vetor contendo as informações da linguagem matemática *MathML* e um método para efetuar a busca da *tag* desejada através de um *id*. Assim, caso haja a necessidade de alteração de algum destes códigos, não será preciso atualizar todas as demais classes que utilizem as *tags* *MathML* para composição de expressões matemáticas.

Alguns elementos matemáticos diferenciam-se uns dos outros pela quantidade de *tags* *MathML* utilizadas nas suas codificações. É o caso, por exemplo, de um operador, um número e uma matriz. No operador há somente uma *tag* correspondente para sua composição já que este é definido, enquanto que no número existem duas *tags* (abertura e fechamento), já que o valor dentre estas será variado de acordo com a necessidade do usuário. O mesmo ocorre com a matriz só que segue um esquema diferente, pois precisa de *tags* de abertura e fechamento para delimitar linhas, colunas e a própria matriz. Estas distinções podem ser vistas nos códigos extraídos da TagLib, apresentados na Figura 5.1.

```
//Matriz
['129', '<mfenced open="[" close="]"><mtable>'], //Inicio matriz
['130', '<mtr>'], //Inicio linha
['131', '<mttd>'], //Inicio elemento
['132', '</mttd>'], //Fim elemento
['133', '</mtr>'], //Fim linha
['134', '</mtable></mfenced>'], //Fim matriz

//Operadores
//Adicao
['8', '<mo> \u002B </mo>'],

//Numero Real
['6', '<mn>'],
['7', '</mn>'],
```

Figura 5.1 - Exemplo da diferença de *tags* usadas na composição de operadores, números e matrizes.

No momento em que o usuário clica em um elemento contido em um botão, este consulta a(s) tag(s) da TagLib referente(s) ao elemento requerido. Para a visão do usuário é apresentado o elemento renderizado, já que os códigos de abertura e fechamento do *MathML* são adicionados à *tag* já existente. Já a informação da *tag* é guardada em um atributo *mathtag* contido no botão. A Figura 5.2 apresenta um exemplo desta operação de consulta à TagLib.

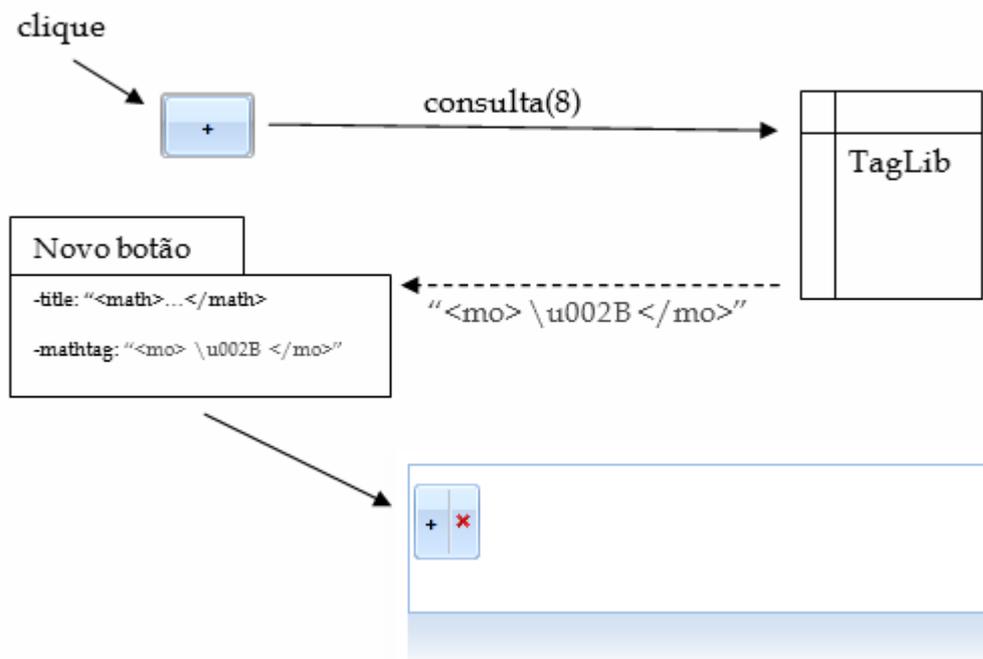


Figura 5.2 - Consulta à TagLib para criação de um novo botão para o operador "soma".

5.4 Desenvolvimento

Com a análise e testes feitos nas interfaces apresentadas no Capítulo 3, alguns aspectos foram listados para que fossem corrigidos com a interface proposta por este trabalho. Estes pontos analisados são:

- Na *Whelp* ocorre uma dependência de linguagem matemática na entrada;

- Na *Math Web Search* há a necessidade de que o usuário escolha qual linguagem usada para efetuar a busca, além do fato da montagem de expressões ter que obedecer a uma ordem padrão;
- Para pesquisas com expressões na *ActiveMath*, é necessário que o usuário tenha um *plugin Java* pré-instalado e a máquina virtual desta em pleno funcionamento;

5.4.1 Interface resultante

A proposta do trabalho é de se obter uma interface que ofereça os elementos matemáticos ao usuário e que este seja capaz de montar expressões matemáticas graficamente, sem a necessidade de conhecer alguma linguagem matemática, ao contrário da interface proposta pela *Whelp* que tem como entrada expressões usando uma linguagem matemática padrão, de forma textual. Sendo assim, foi criado um menu ao topo da página contemplando elementos matemáticos, que por sua vez estão contidos em painéis separados de acordo com as áreas de aplicação, sendo estes agrupados em abas do tipo Símbolos, Funções ou Letras Gregas, além dos botões para variáveis ou números. A língua utilizada para os menus e demais informações é o inglês devido a sua ampla difusão mundial.

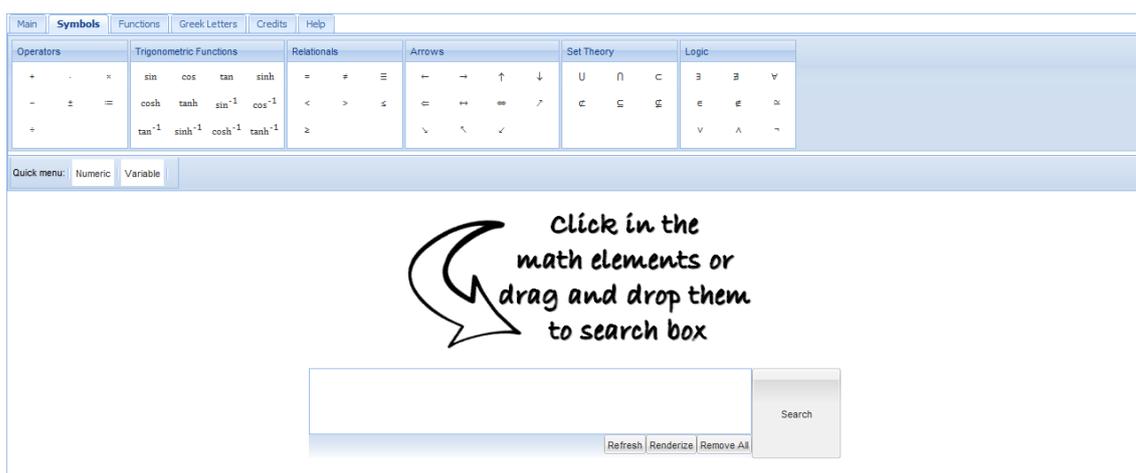


Figura 5.3 - Tela da interface.

O usuário pode clicar nos elementos matemáticos ou carregá-los para a barra de busca. Os elementos estão dispostos em botões, sendo que não são editáveis os de tipos Simbólicos ou Letras Gregas. À medida que o usuário vai clicando ou carregando, a barra de busca vai inserindo os botões escolhidos. Na *Math Web Search* o usuário fica preso a ordem dos elementos inseridos enquanto que nesta interface caso ele queira mudar a ordem dos elementos dispostos na barra de busca, poderá fazê-lo carregando o botão para a posição que achar melhor.

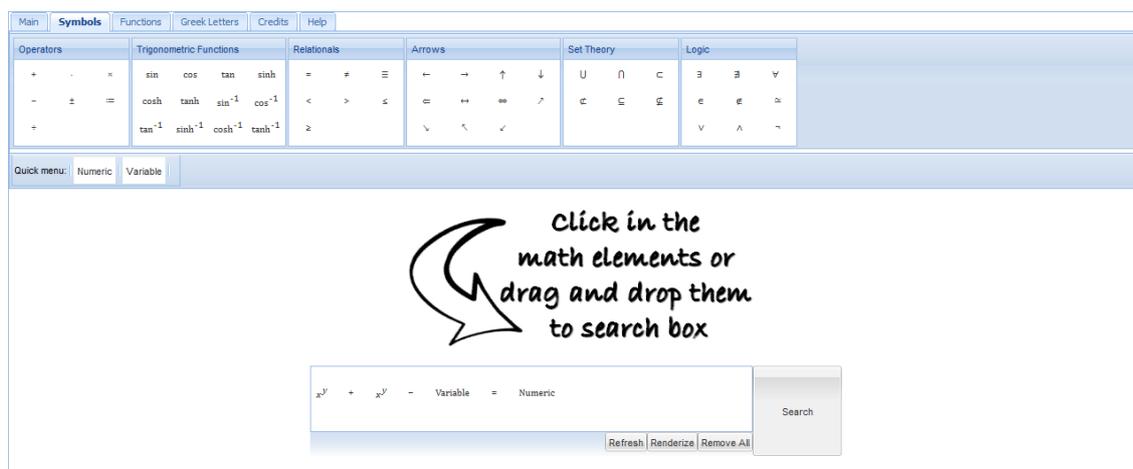


Figura 5.4 - Elementos matemáticos inseridos na barra de busca.

Ao se clicar num elemento do tipo variável ou numérico, uma janela do tipo *input* é apresentada ao usuário para que se entre com algum valor, como é mostrado na Figura 5.5. Caso não queira aquele tipo de elemento, o usuário pode excluí-lo através de um “xis” vermelho do lado direito do botão selecionado.

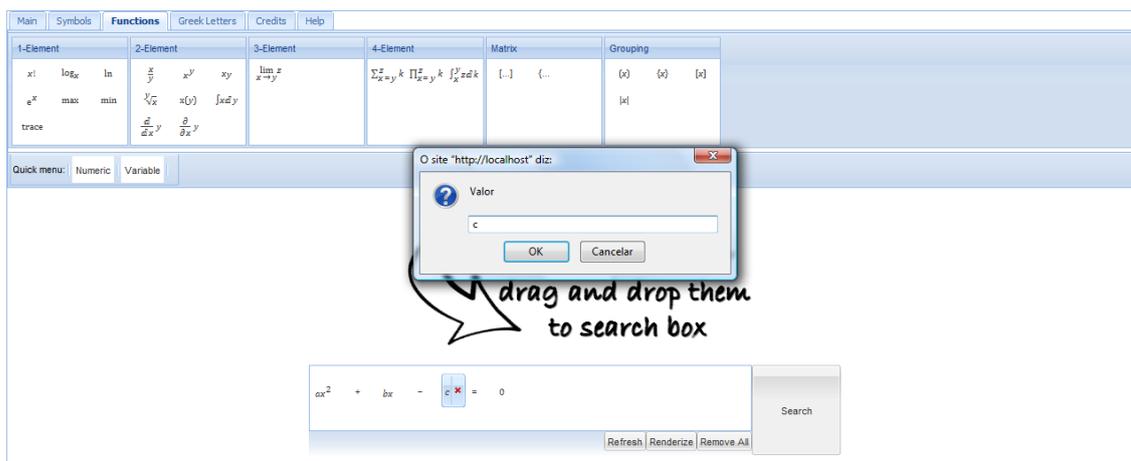


Figura 5.5 - Inserção de um valor do tipo variável.

Caso queira editar uma função, é apresentada uma janela contendo os botões para a área da função a ser editada. Por exemplo, para editar o expoente de uma potenciação, o usuário deve clicar no botão referente ao expoente. Nesse momento será então exibida uma janela contendo uma barra do mesmo tipo da barra de busca, sendo possível a inserção de qualquer elemento nesse campo (seja um número ou mesmo uma subexpressão).

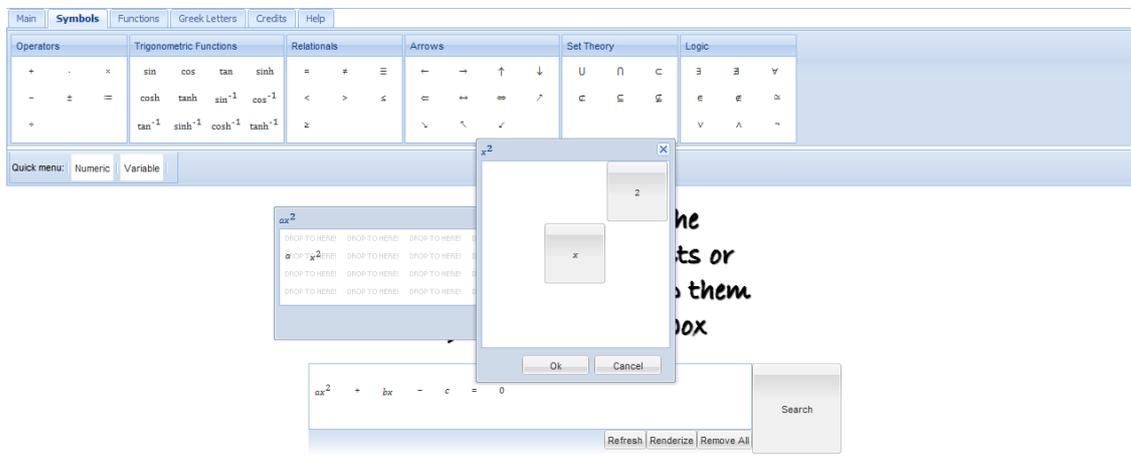


Figura 5.6 - Edição de uma função, que no caso tem-se o exemplo de uma potenciação.

Os painéis encontrados em cada aba do menu podem também ser arrastados para dentro da página, deixando a critério do usuário escolher quais serão necessários para formular a sua expressão matemática. A facilidade de se

arrastar os painéis mais usados para a área de trabalho da ferramenta minimiza a necessidade de ficar trocando de aba do menu a todo momento, ajudando na usabilidade. A Figura 5.7 exibe essa possibilidade.

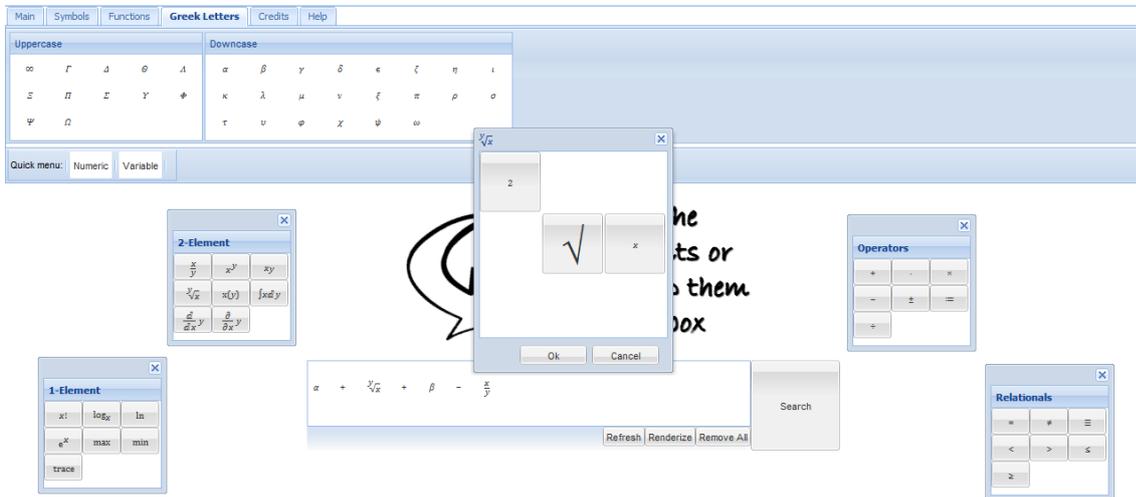


Figura 5.7 - Painéis separados dos submenus.

Em qualquer momento da edição o usuário pode ver a expressão matemática renderizada (a partir do código *MathML* obtido da junção das *tags* de cada elemento adicionado) através do botão *renderize* encontrado no rodapé da barra de busca, como é mostrado na Figura 5.8.

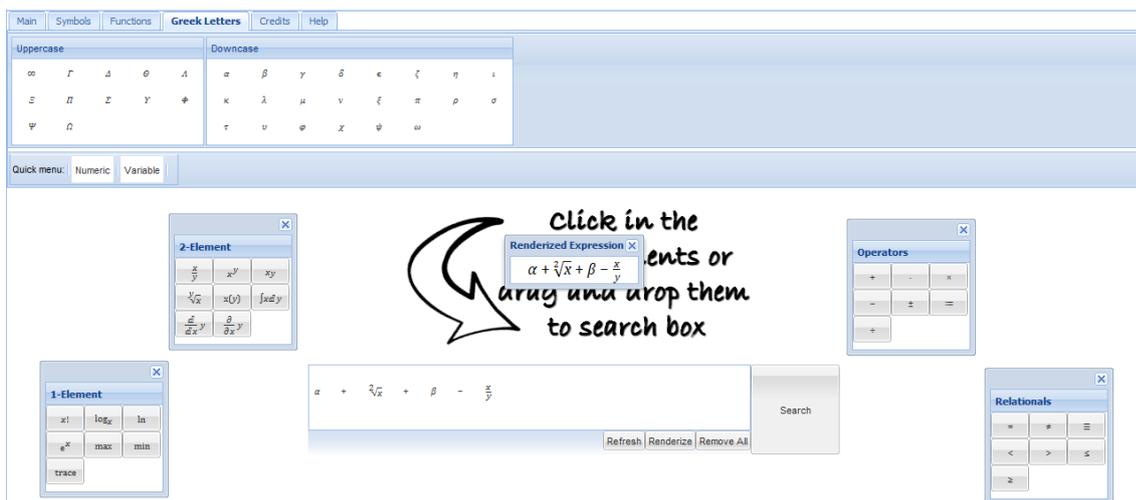


Figura 5.8 - Expressão matemática renderizada a partir do código *MathML* da mesma.

Cada elemento adicionado à barra de busca pode ser visualizado em um tamanho maior ao passar o mouse por cima do botão correspondente. O que é mostrado ao usuário é uma renderização do código *MathML* contido naquele elemento. Esta funcionalidade é demonstrada na Figura 5.9, onde há a renderização de uma matriz 2×2 .

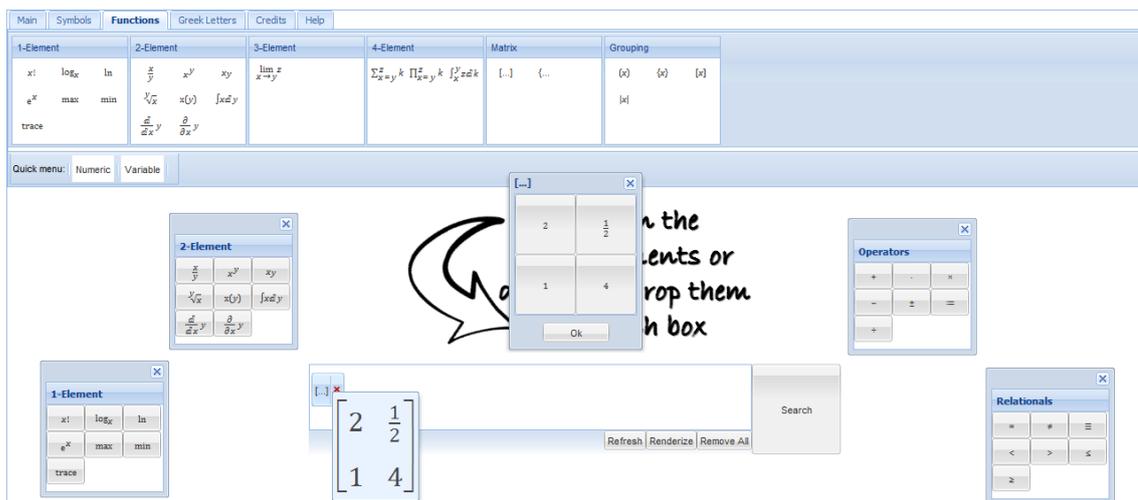


Figura 5.9 - Visualização aumentada do elemento editado.

A título de representação, ao se clicar no botão de busca é apresentada o código *MathML* resultante de toda a formulação da expressão matemática. Este código será então passado para o servidor, caso a interface esteja acoplada a um, para ser efetuada a busca da equação nos conteúdos encontrados em bibliotecas matemáticas digitais. Ou seja, fica a critério do sistema de busca fazer as devidas conversões para outras linguagens, caso seja necessário, ao contrário do que acontece com a interface da *Math Web Search*, onde o usuário deve escolher qual linguagem matemática utilizar na busca. A Figura 5.10 apresenta um exemplo de código *MathML* para uma expressão matemática envolvendo um somatório e uma radiciação.

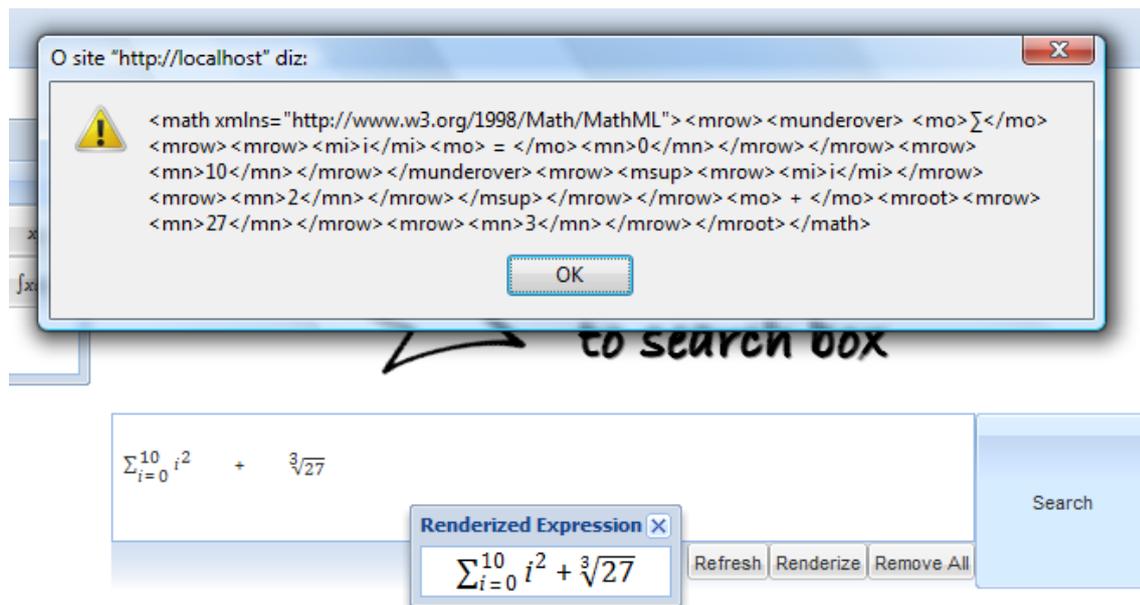


Figura 5.10 - Código MathML resultante da expressão formulada.

Na interface proposta pela *ActiveMath* há a necessidade de um *plugin Java* para navegadores, além do fato da máquina virtual *Java* estar em pleno funcionamento no sistema operacional. A interface obtida neste trabalho, por ser totalmente desenvolvida em *xHTML* e um *framework* de *Javascript*, não há a necessidade de que o usuário tenha *plugins* específicos para ter acesso às funcionalidades da interface, exceto *plugins* de suporte à linguagem *MathML* para navegadores não compatíveis. Por este motivo, a interface tornou-se genérica tanto no ponto de vista de navegadores de internet quanto no sistema operacional utilizado pelo usuário.

5.5 Considerações finais

A interface genérica obtida deste trabalho pode ser acoplada a um sistema de busca, agregando outras linguagens e banco de dados, incrementando as funcionalidades da interface. Esta possibilidade é apresentada na Figura 5.11, onde

preenche exatamente o quesito 1 da estrutura de um sistema de busca, sendo este proposto na qualificação de Gonzaga (2010).

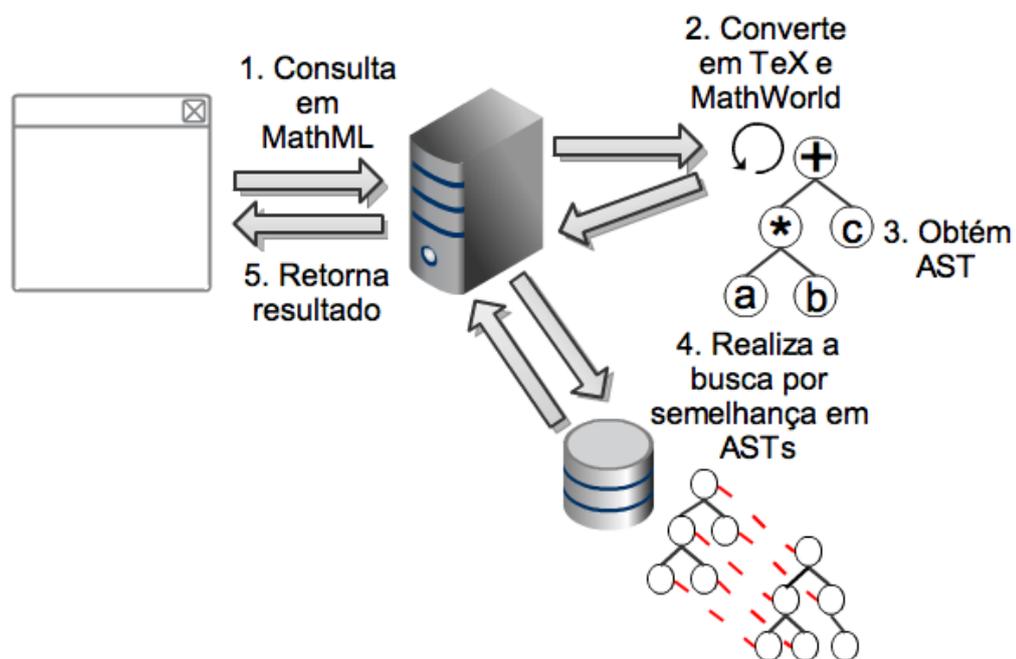


Figura 5.11 - Estrutura de um sistema de busca matemática (Gonzaga, 2010).

6

Conclusões

Este capítulo apresenta as conclusões desta monografia, retomando alguns pontos já abordados em capítulos passados.

A recuperação e busca de conteúdo matemático tem recebido atenção em estudos recentes. Ao se comparar as funcionalidades apresentadas nos protótipos desenvolvidos para a busca matemática com as apresentadas em ferramentas de busca textual (como por exemplo o *Google*), pode-se observar serviços oferecidos pelas ferramentas de busca textual, e que a médio ou longo prazo poderiam ser oferecidos de forma semelhante pelas ferramentas de busca matemática. Um ponto importante para este tipo específico de busca é a interface da ferramenta e como é o seu método de entrada apresentado ao usuário, seja textual ou gráfico.

A interface para formulação de equações em ferramentas de busca matemática proposta por este trabalho foi concluída satisfatoriamente. Através dos diagramas de caso de uso e apresentação das funcionalidades da interface pode-se entender como esta se comporta, quais opções disponíveis para a edição das expressões matemáticas, como trabalha com os elementos visuais, cumprindo com a meta de se obter ao término da formulação da expressão um código em *MathML* que poderá ser enviado à ferramenta de busca acoplada à interface desenvolvida.

7

Trabalhos Futuros

Este capítulo apresenta trabalhos futuros desta monografia, tais como testes de usabilidade, compatibilidade de navegadores e tratamento de usuários avançados.

O trabalho apresentado é parte de um projeto de desenvolvimento de uma ferramenta de busca de conteúdo matemático. Além das questões relacionadas ao desenvolvimento da ferramenta em si, fora do escopo do presente trabalho, questões relacionadas à Interface podem ainda ser melhoradas. Dentre as quais, destacam-se:

- Realização de testes de usabilidade com usuários, utilizando de técnicas de análise em Interface Homem-Computador. Tais análises permitirão observar pontos em que a ferramenta pode ser melhorada, como por exemplo, deixá-la mais intuitiva, e permitir ao usuário o desenvolvimento de expressões em menos tempo;
- A questão da compatibilidade e desempenho com navegadores também precisa ser testada. O desenvolvimento da ferramenta para diferentes ambientes e dispositivos (como por exemplo celulares) está ainda em aberto;
- Fazer a ferramenta independente de conhecimento de linguagem é um ponto positivo, pois permite o uso da mesma por qualquer usuário que possua uma expressão matemática. Outra vertente, no entanto, pode ser ainda explorada, visando atender especificamente a usuários mais avançados. Nesses casos, poderá ser desenvolvido um ambiente que permita a entrada de expressão diretamente em linguagens matemáticas, como a *TeX* e a *MathML*. Ao permitir a entrada da expressão em linguagem,

o usuário poderá então simplesmente copiar e colar a saída exportada por *softwares* matemáticos na interface desenvolvida.

8

Referências Bibliográficas

- Abramowitz, M. & Stegun, I. A. (1964). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. National Bureau of Standards Applied Mathematics Series, U.S. Government Printing Office, Washington, D.C.
- Asperti, A., Padovani, L., Coen, C. S., & Schena, I. (2001). *HELM and the Semantic Math-Web*. In Proceedings of the 14th international Conference on theorem Proving in Higher Order Logics (September 03 - 06, 2001). R. J. Boulton and P. B. Jackson, Eds. Lecture Notes In Computer Science, vol. 2152. Springer-Verlag, London, pp. 59-74.
- Asperti, A., Guidi, F., Coen, C. S., Tassi, E. & Zacchiroli, S. (2006). *A Content Based Mathematical Search Engine: Whelp*. Types for Proofs and Programs, pp. 17-32, SpringerLink.
- Asperti, A., Guidi, F., Coen, C. S., Tassi, E. & Zacchiroli, S. (2007). *User Interaction with the Matita Proof Assistant*. Journal of Automated Reasoning, Special Issue on User Interfaces for Theorem Proving, Springer Netherlands, ISSN 0168-7433, pp. 109-139.
- Baeza-Yates, R. A. (2000). *Teaching Algorithms*. In: ACM SIGACT News 26, v. 4, pp. 51-59.
- Bertot, Y. & Casteran, P. (2004). *Interactive Theorem Proving and Program Development*. SpringerVerlag.
- Bezerra, E. (2007). *Princípios de Análise e Projeto de Sistemas com Uml*. Rio de Janeiro: Elsevier, pp. 4 - 17.
- Booch, G., Rumbaugh, J. & Jacobson, I. (1999). *The Unified Modeling Language User Guide*. Addison-Wesley.
- Brin, S. & Page, L. (1998). *The Anatomy of a Large-Scale Hypertextual Web Search Engine*. Seventh International World-Wide Web Conference (WWW 1998), April 14-18, 1998, Brisbane, Australia.

- Capes (2010). *O Portal Brasileiro da Informação Científica*. Disponível em: <<http://www.periodicos.capes.gov.br/portugues/index.jsp>>. Acesso em: 10 jun. 2010.
- Caprotti, O., Carlisle, D. P. & Cohen, A. M. (2002). *The OpenMath Standard*. The OpenMath Consortium.
- Connexions (2010). *Connexions*. Disponível em: <<http://cnx.org/aboutus/>>. Acesso em: 10 jun. 2010.
- Costa, H. A. X., Santos, R. P. & Werner, C. M. L. (2010). *Uma Análise do Processo de Ensino e Aprendizagem de Engenharia de Software: Desafios e Soluções no Contexto Brasileiro*. In: Proceedings of the XI International Conference on Engineering and Technology Education, Ilhéus, Brasil, pp. 367-371.
- Fallows, D. (2004). *The Internet and Daily Life*. Pew Internet and American Life Project, August 11, 2004. Disponível em: <http://www.pewinternet.org/~media/Files/Reports/2004/PIP_Internet_and_Daily_Life.pdf>. Acessado em: 10 jun. 2010.
- Frederick, S., Ramsay, C. & Blades, S. C. (2008). *Learning Ext JS*. Packt Publishing, Birmingham, UK. ISBN-13: 978-1847195142, 299 pages. November 28, 2008.
- Gonzaga, F. B. (2010). *Recuperação e Busca de Conteúdo Matemático*. COPPE/UFRJ, Rio de Janeiro.
- Haden, P. & Mann, S. (2003). *The Trouble with Teaching Programming*. In: Proceedings of the 16th Annual NACCQ, Palmerston North, New Zealand.
- Kesselman, M. & Watsen, S. B. (2005). *Google Scholar and libraries: point/counterpoint*. Reference Services Review, Vol. 33 No. 4, pp. 380-7.
- Knuth, D. E. (1985). *The TeXbook*. Addison Wesley.
- Kohlhase, M. & Sucas, I. (2006). *A search engine for mathematical formulae*. Proceedings of Artificial Intelligence and Symbolic Computation, AISC'2006, number 4120 in LNAI, pp. 241-253. Springer Verlag.
- Kruger, H. (2004). *Searching mathematics with zentralblatt math: Overview and outlook*. Enhancing the Searching of Mathematics, IMA "Hot Topics" Workshop, U.S.A.
- Kukich, K. (1992). *Techniques for automatically correcting words in text*. ACM Computing Surveys (CSUR), v.24 n.4, pp. 377-439, Dec. 1992.

- Libbrecht, P. & Melis, E. (2006). *Methods to access and retrieve mathematical content in activemath*. In ICMS, volume 4151 of Lecture Notes in Computer Science, pp. 331-342. Springer.
- Manning, C. D., Raghavan, P. & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, Online Edition.
- Melis, E., Gogvadze, G., Libbrecht, P. & Ullrich, C. (2009). *ActiveMath, a Learning Platform With Semantic Web Features*. *Ontologies and Semantic Web for e-Learning*. Dicheva, D. & Greer, J. (eds).
- Miner, R. & Munavalli, R. (2007). *An Approach to Mathematical Search Through Query Formulation and Data Normalization*. *Towards Mechanized Mathematical Assistants*. SpringerLink, pp. 342-355.
- Neves, M. F. & Coello, J. M. A. (2006). *OntoRevPro: Uma Ontologia sobre Revisão de Programas para o Aprendizado Colaborativo de Programação Java*. In: *Anais do XVII Simpósio Brasileiro de Informática na Educação, Brasília, Brasil*, pp. 569-578.
- Nist (2010). *Digital Library of Mathematical Functions*. Disponível em: <<http://dlmf.nist.gov/>>. Acesso em: 10 jun. 2010.
- Pereira, M. N. F. (1995). *Bibliotecas Virtuais: realidade, possibilidade ou alvo de sonho*. *Ciência da Informação*, vol. 24, número 1.
- Ramalho, R. A. S., Vidotti, S. A. B. G. & Fujita, M. S. L. (2005). *Bibliotecas Digitais na era da Web Semântica: reflexões no âmbito da gestão de conteúdos informacionais*. *Simpósio Internacional de Bibliotecas Digitais, 3, 28 nov. - 02 dez. 2005, São Paulo*. *Anais do 3º Simpósio Internacional de Bibliotecas Digitais*. São Paulo: Universidade de São Paulo: Universidade Estadual Paulista.
- Tobar, C. M., Rosa, J. L. G., Coelho, J. M. A. & Pannain, R. (2001). *Uma Arquitetura de Ambiente Colaborativo para o Aprendizado de Programação*. In: *Anais do XII Simpósio Brasileiro de Informática na Educação, Vitória, Brasil*, pp. 21-23.
- Tolhurst, W. A., Pike, M. A. & Blinton, K. A. (1994). *Using the Internet; special edition*. Que Corporation, p. 29.
- W3Ca (2010). *MathML - Mathematical Markup Language*. Disponível em: <<http://www.w3.org/math/>>. Acesso em: 10 jun. 2010.

- W3Cb (2010). *XML - Extensible Markup Language*. Disponível em: <<http://www.w3.org/xml/>>. Acesso em: 10 jun. 2010.
- W3Cc (2010). *MathML3 - Mathematical Markup Language*. Disponível em: <<http://www.w3.org/TR/MathML3/>>. Acesso em: 10 jun. 2010.
- W3Schools (2010). *XML - How to use*. Disponível em: <http://www.w3schools.com/xml/xml_tree.asp>. Acesso em: 10 jun. 2010.
- Warner, S. (2005). *The arXiv: Fourteen Years of Open Access Scholarly Communication*. Free Culture and the Digital Library Symposium Proceedings.
- Weisstein, E. W. (2007). *Making MathWorld*. The Mathematica Journal 10, pp. 474-488.
- Wikipedia (2010). *Wikipedia, a enciclopédia livre*. Disponível em: <<http://pt.wikipedia.org/wiki/Wikipedia:Sobre>>. Acesso em: 10 jun. 2010.
- Wolfram (2010). *Wolfram Functions site*. Disponível em: <<http://functions.wolfram.com/About/>>. Acesso em: 10 jun. 2010.
- Youssef, A. (2006). *Roles of Math Search in Mathematics*. 5th International Conference on Mathematical Knowledge Management, pp. 2-16.
- Zhao, J., Kan, M. & Theng, Y. L. (2008). *Math Information Retrieval: User Requirements and Prototype Implementation*. JCDL'08, Pennsylvania.