

# Sistema Especialista para o Ensino-Aprendizagem de Expressões Algébricas

**Alex Junior Avila**

Instituto Federal de Educação  
Ciência e Tecnologia de Santa  
Catarina – Lages, Brasil  
[alex\\_avila11@hotmail.com](mailto:alex_avila11@hotmail.com)

**EneDir Guimarães de Oliveira  
Junior**

Instituto Federal de Educação  
Ciência e Tecnologia de Santa  
Catarina – Lages, Brasil  
[guimaraes.junior94@gmail.com](mailto:guimaraes.junior94@gmail.com)

**Wilson Castello Branco Neto**

Instituto Federal de Educação  
Ciência e Tecnologia de Santa  
Catarina  
Lages, Brasil  
[wilson.castello@ifsc.edu.br](mailto:wilson.castello@ifsc.edu.br)

**Ailton Durigon**

Instituto Federal de Educação  
Ciência e Tecnologia de Santa  
Catarina  
Lages, Brasil  
[ailtondurigon@yahoo.com.br](mailto:ailtondurigon@yahoo.com.br)

## ABSTRACT

The results of the ENEM, vestibular exams e other evaluations evidence a students' great difficulty related to understanding of mathematics which represents a continuous challenge for students and teachers. There is a wide variety of solutions available to help in the teaching-learning process of mathematics, one of which is IFMath. This paper presents a tool for handling algebraic expressions to be integrated with IFMath. The proposed software aims at approximating theory and practice and providing to teachers a new way of presenting the contents. It presents concepts that provide an overview of the content and elaborates the step-by-step resolution of the problems proposed by the user. In addition, it presents an explanation of each of the algebraic manipulations performed. Results show that classical theories of computing, such as compilers and expert systems, can contribute actively as a pedagogical tool by solving algebraic problems.

## RESUMO

Os resultados do ENEM, vestibulares e outras avaliações evidenciam uma grande dificuldade dos alunos quanto ao domínio da matemática, o que representa um contínuo desafio para alunos e professores. Existe uma grande variedade de soluções cujo objetivo é auxiliar o ensino-aprendizagem de matemática, sendo uma delas o IFMath.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org) or Publications Dept., ACM, Inc., fax +1 (212) 869-0481

Este trabalho tem como objetivo desenvolver uma ferramenta para a manipulação de expressões algébricas para ser integrado ao IFMath. O software proposto visa aproximar teoria e prática e possibilitar aos professores uma nova forma de apresentar os conteúdos. Para isso, o sistema apresenta conceitos que proporcionam uma visão geral sobre o conteúdo e elabora a resolução passo a passo dos problemas propostos pelo próprio usuário. Ainda, apresenta uma explicação sobre cada uma das manipulações algébricas realizadas. Os resultados alcançados evidenciam que teorias clássicas da computação, como compiladores e sistemas especialistas, podem contribuir ativamente como ferramenta pedagógica na resolução de problemas algébricos.

## Palavras-chave

Álgebra. Ensino-aprendizagem. Compilador. Sistema especialista.

## ACM Classification Keywords

K.3.1 Computer Uses in Education.

## 1. INTRODUÇÃO

A matemática é, em geral, considerada uma disciplina muito complexa pelos alunos. Segundo [8], inúmeras vezes, a aversão a esta disciplina está relacionada ao estudo da álgebra, que se destaca como um dos conteúdos da matemática elementar com menores índices de acertos em exames como vestibulares e o ENEM.

Conforme aponta [26], o processo de ensino-aprendizagem da álgebra apresenta uma grande variedade de dificuldades, dentre elas destacam-se o desenvolvimento cognitivo limitado dos alunos e a natureza complexa deste ramo da matemática. De acordo com [4], essas inúmeras dificuldades também podem ser justificadas pela falta de interesse pela matemática e dos conhecimentos necessários

para uma correta interpretação e compreensão de linguagens e simbologias. Uma expressão algébrica normalmente amedronta os alunos que a consideram de difícil e confusa resolução devido a presença de coeficientes e incógnitas.

A dificuldade para o entendimento da álgebra por parte dos estudantes acarreta em frequentes erros, destacando-se o uso incorreto da notação e de convenções algébricas necessárias para a resolução dos problemas [22].

A busca pelo aperfeiçoamento de métodos e técnicas que visam um melhor aproveitamento dos conteúdos ministrados em sala de aula tem sido o objeto de estudo de muitos pesquisadores [13]. Aproveitando o desenvolvimento das tecnologias de informação e comunicação, surge a possibilidade de uma integração mais efetiva da mesma com a educação facilitando dessa forma o processo de ensino-aprendizagem e promovendo aulas mais dinâmicas e eficazes. Na internet, existe uma grande quantidade de softwares matemáticos disponíveis para utilização, cada um com uma metodologia específica para atingir um objetivo comum a todos: auxiliar no processo de ensino-aprendizagem.

Um destes softwares é o *IFMath*, desenvolvido no Instituto Federal de Santa Catarina, *campus* Lages, que propõe resoluções passo a passo de exercícios matemáticos apresentando sua fundamentação teórica. Entretanto, esta solução carece de um módulo relacionado a manipulações e solução de expressões algébricas, um importante e complexo conteúdo relativo à álgebra elementar e que faz parte dos conteúdos matemáticos da Educação Básica.

Os sistemas especialistas, uma área de estudo da inteligência artificial, vem contribuindo com a educação, ofertando mais uma ferramenta para auxiliar na aprendizagem dos alunos. Estes sistemas são alimentados com o conhecimento de docentes para serem utilizados na resolução de problemas nas mais diversas áreas, inclusive matemática.

Diante do exposto, este trabalho tem como objetivo central criar um sistema especialista que realiza manipulações em expressões algébricas informadas dinamicamente pelo usuário para ser integrado ao *IFMath*, com o intuito de auxiliar professores e alunos no processo de ensino-aprendizagem de matemática, apresentando ao usuário cada passo da resolução dos problemas, juntamente com o embasamento teórico para o entendimento da resposta.

Este artigo está organizado da seguinte maneira: após esta introdução, a seção 2 apresenta o resultado do levantamento bibliográfico realizado sobre o ensino de matemática e softwares educacionais, sendo finalizada com a descrição de alguns trabalhos relacionados. As seções 3 e 4 apresentam o referencial teórico das áreas de compiladores e sistemas especialistas, os quais englobam as técnicas utilizadas para a implementação do sistema. A seção 5 descreve as atividades realizadas e as técnicas utilizadas,

enquanto a seção 6 apresenta o sistema resultante. Por fim, na seção 7 são tecidas algumas considerações.

## 2. EDUCAÇÃO E TECNOLOGIA

### 2.1. ENSINO DA MATEMÁTICA

As limitações da aprendizagem de matemática são um relato de diversos estudantes. Eles enfrentam dificuldades que acabam por gerar altos índices de reprovação. Conforme aponta [9], nem mesmo os alunos aprovados conseguem efetivamente utilizar esse conhecimento de fundamental importância.

Os problemas no ensino da matemática nas escolas estão diretamente relacionados com o cenário da educação no Brasil. De acordo com [6], uma situação dessa magnitude não surge repentinamente, ela é construída ao longo de décadas de ensino deficiente. O ensino da matemática no século XX e início do século XXI trabalha o formalismo das regras, das fórmulas e dos algoritmos, bem como a complexidade dos cálculos com seu caráter rígido e disciplinador, levando à exatidão e precisão dos resultados [24]. Muitas vezes, o estudo baseia-se na memorização do resumo dos conteúdos considerados importantes e suficientes pelo professor, assim como na repetição de exercícios.

Os reflexos dessa carência podem ser observados em avaliações como o PISA (*Programme for International Student Assessment*). Em sua última realização em 2015, o teste constatou que sete em cada dez alunos brasileiros, com idade entre 15 e 16 anos, estão abaixo do nível básico de conhecimento matemático [18].

As transformações sociais implicam em mudanças na educação e nessa perspectiva surgiram tendências metodológicas que visam mudar o panorama da educação matemática. Os trabalhos [16] e [28] destacam como as principais tendências a etnomatemática, tecnologia, modelagem, história e filosofia, investigação matemática e resolução de problemas.

Através da utilização de tecnologias, especialmente softwares, o estudo dos conceitos aplicados à realidade pode ter um maior significado no desenvolvimento do raciocínio matemático do estudante. De acordo com [27], o desenvolvimento da tecnologia justifica e proporciona outros meios para que a ênfase no ensino incida nos aspectos mais conceituais da matemática em detrimento dos seus aspectos mais mecânicos.

Conforme [21], a incorporação dos recursos obtidos pelos avanços tecnológicos na prática pedagógica dos professores pode facilitar o entendimento do aluno, proporcionando uma aprendizagem mais significativa e atraente.

### 2.2. TRABALHOS RELACIONADOS

A influência das tecnologias mudou o cotidiano do professor em sala de aula. Neste contexto, aparece um novo

modelo de educação, no qual ferramentas tradicionais como giz, quadro e livros não são mais os únicos instrumentos utilizados para professores lecionarem.

Ao se referirem sobre as fases das tecnologias digitais em Educação Matemática, em [3] evidencia-se que a utilização de tecnologias móveis como laptop, telefones celulares ou tablets tem se popularizado nos últimos anos devido ao advento da internet e ao aumento de sua velocidade. Vários estudantes utilizam a internet em sala de aula a partir de seus telefones para acessar plataformas como *google*, *dropbox* e *onedrive*. Outros ainda utilizam as câmeras para registrar momentos dessa aula com fotos e vídeos, para lhes auxiliar mais tarde. Neste contexto, em [14] afirma-se que a informatização do ambiente escolar pode agilizar o processo de compreensão do conhecimento ao instigar a capacidade cognitiva do aluno, auxiliando na superação dos obstáculos através da visualização, experimentação, interpretação e demonstração dos conteúdos.

Um software educacional pode ser definido como um sistema computacional interativo que objetiva propiciar o entendimento de conceitos específicos, como os matemáticos ou científicos [12].

Com a tecnologia cada vez mais presente em sala de aula, o uso de software como uma ferramenta de auxílio torna-se cada vez mais comum. Alguns autores [11] ressaltam que, devido ao seu dinamismo, o uso dos softwares é um importante aliado no desenvolvimento cognitivo de cada aluno ao adaptar-se a distintos ritmos de aprendizagem. Por consequência, os professores têm a possibilidade de mudar a sua metodologia em sala de aula, adaptando o uso da tecnologia para desenvolver novas atividades.

Existe uma vasta gama de softwares educacionais disponíveis na internet, por esse motivo foi realizado um estudo de caso, com o propósito de identificar os principais aspectos que um software educacional para o ensino de álgebra deve apresentar. Tomando como base as características do *IFMath*, foram selecionados três softwares online de uso gratuito para realizar o presente estudo de caso, conforme segue:

- *WolframAlpha*: É uma enciclopédia virtual que aborda as mais diferentes disciplinas, entre elas a matemática. Entre os módulos oferecidos para resolver problemas matemáticos, existe o módulo algébrico, cujo fator de destaque é sua eficiência em reconhecer automaticamente as expressões algébricas informadas pelo usuário. Ele apresenta o resultado final das expressões algébricas, juntamente com uma representação gráfica necessária, porém ele não apresenta a resolução passo a passo da expressão, em sua versão gratuita.
- *Symbolab*: Trata-se de uma calculadora online que oferece suporte a diversos conteúdos matemáticos, entre eles a álgebra. Assim como o *WolframAlpha*, destaca-se por sua eficiência em reconhecer

expressões algébricas. Exibe o resultado final das expressões algébricas, oferecendo ao usuário a possibilidade de visualizar todos os passos realizados para resolvê-las, juntamente como o seu embasamento teórico. Ele também disponibiliza uma representação gráfica da expressão.

- *GeoGebra*: É um software matemático que permite realizar construções geométricas, sendo uma ferramenta dinâmica e interativa de ensino. Ele se destaca pela capacidade de representar graficamente expressões algébricas de forma dinâmica tornando possível o acesso a outras informações como o valor das raízes e alguns dados adicionais. Diferente dos softwares apresentados anteriormente, o *GeoGebra* não é focado na resolução das expressões algébricas, mas sim em sua representação gráfica.

### 2.3. IFMATH

O *IFMath* é uma aplicação web gratuita e em português que realiza a resolução de exercícios matemáticos. Ele apresenta os resultados detalhados, o embasamento teórico que justifica a resposta, conceitos, curiosidades e propriedades sobre o tema, bem como a visualização gráfica necessária para interpretação do problema, observando o preconizado em [23], sobre a avaliação da qualidade de um software educacional. Diante disso, pode auxiliar tanto o docente no ensino, bem como o discente no processo de aprendizagem.

É importante ressaltar que a aplicação é acessível por qualquer dispositivo que possua acesso à Internet e um navegador. Além disso, o sistema permite que o usuário informe os dados de entrada para os algoritmos, realizando o processamento das respostas de forma dinâmica, com base neste parâmetro. Isso possibilita ao usuário testar diferentes entradas, permitindo uma ampla interpretação sobre os conteúdos estudados.

O *IFMath* contempla números e operações, geometria, álgebra e funções, representando uma boa parcela dos conteúdos abordados na disciplina de matemática da Educação Básica. Entretanto, esta aplicação não possuía um módulo para manipulações algébricas, um importante e complexo conteúdo, que é o objeto deste trabalho.

Diante da proposta do *IFMath* em resolver exercícios informados pelo usuário, apresentando a solução passo a passo e a fundamentação teórica, torna-se necessário trabalhar com expressões algébricas de maneira dinâmica. Para isso, a utilização de um Sistema Especialista (SE) é uma das soluções aplicáveis, visto que eles vêm contribuindo com a solução de problemas nas mais diversas áreas, inclusive no ensino de matemática. Entretanto, antes disto é necessário averiguar a validade sintática e compreender de uma maneira computacionalmente processável a estrutura da expressão. Para isso, a aplicação das técnicas de compiladores, especialmente as análises léxica e sintática e a geração de código intermediário, se faz necessária.

### 3. COMPILADORES

Os compiladores são ferramentas que, de forma simplificada, realizam a leitura de um programa escrito em uma linguagem (linguagem fonte) e a traduzem para uma outra linguagem (linguagem alvo) mantendo a semântica original [2]. Inicialmente, os compiladores eram utilizados para traduzir códigos em linguagens de alto nível para linguagens de baixo nível, realizando otimizações importantes para resolver problemas de limitações de hardware. Entretanto, seus conceitos passaram a ser aplicados na superação de obstáculos em diferentes áreas, como na construção de tradutores automáticos e mineração de dados.

De acordo com [5], a estrutura de um compilador é normalmente dividida em duas partes, o *front end* e o *back end*, e cada uma delas possui etapas com funções e objetivos específicos. O *front end* objetiva a compreensão da linguagem-fonte, garantindo que o programa esteja bem formado, para transformá-lo em uma representação intermediária que possa ser utilizada pelo *back end* [25]. Utilizando padrões definidos por expressões regulares, a primeira etapa consiste na análise léxica, que subdivide o programa em partes chamadas de *tokens*. A segunda etapa, a análise sintática, analisa todos os *tokens* a partir de uma gramática livre de contexto a fim de encontrar erros por meio da representação do código por uma árvore de sintaxe. Utilizando esta árvore, o analisador semântico verifica a consistência semântica do programa e reúne informações necessárias para as próximas etapas. A geração de código intermediário, que representa a quarta etapa do processo de compilação, gera uma representação explícita de baixo nível do programa para uma máquina abstrata.

Utilizando a representação intermediária do programa gerada pelo *front end*, o *back end* realiza o mapeamento dessa representação em um conjunto de instruções da máquina alvo [5]. Na quinta etapa, ocorre a otimização deste código com o intuito de produzir um código na linguagem alvo que utilize menos recursos. Por fim, na sexta etapa, é realizada a geração do código na linguagem alvo, como mostra a figura 1.

Somente a análise léxica, sintática e geração de código intermediário são de interesse deste trabalho, sendo detalhadas nas próximas sessões.

#### 3.1. ANALISADOR LÉXICO

A análise léxica consiste na leitura e análise das cadeias de caracteres do código, com o intuito de identificar a lista de símbolos válidos com base em padrões definidos. Segundo [2], é realizada a leitura do fluxo de caracteres que compõem o programa fonte, agrupando-os em sequências significativas, chamadas de *lexemas*. Para cada *lexema* reconhecido, é produzido uma saída chamada *token* no formato nome e valor, para ser utilizado nas fases subsequentes. Quando necessário, o valor de cada *token* é inserido na tabela de símbolos, que por sua vez, é utilizada na análise semântica e na geração de código.

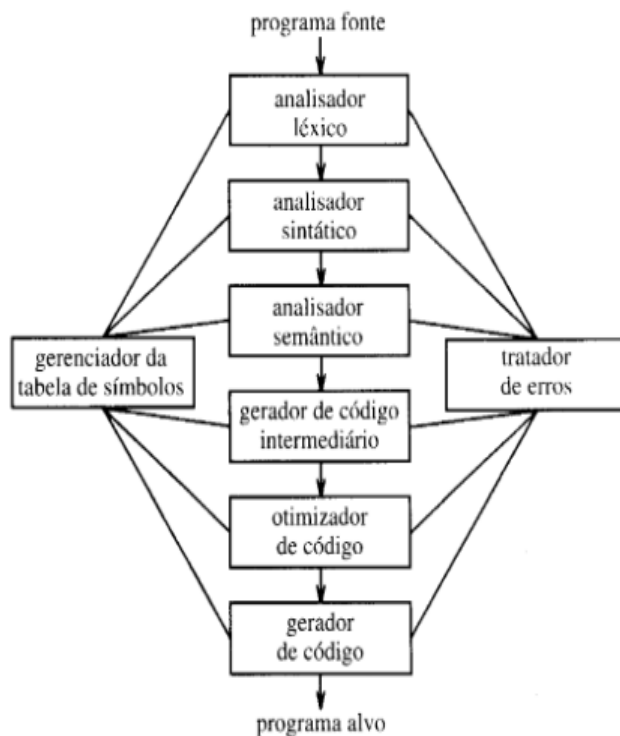


Figura 1 - Etapas de um compilador extraído de [1]

De acordo com [25], um dos meios para a criação de um analisador léxico é a definição da lista dos *tokens* válidos por meio de expressões regulares (RE). Uma RE é a especificação de um padrão entre um conjunto finito de símbolos chamado de alfabeto, construída com operadores de união, concatenação e fechamento. Uma única RE forma uma cadeia ou palavra e um conjunto de palavras representa uma linguagem.

A implementação de um analisador léxico é realizada ao converter uma linguagem descrita por RE em um autômato finito (FA). Esse FA, composto por estados (nós) e arestas, é um reconhecedor da linguagem, que confirma se uma determinada cadeia pertence à mesma. Cada nó representa uma condição possível de ocorrer durante a análise de um *lexema* [5].

O compilador realiza a leitura dos caracteres do programa. O FA inicia a análise de um *lexema* com um caractere e aumenta o tamanho do mesmo à medida que nenhum padrão é encontrado. Ao encontrar um padrão que pertença à linguagem para o *lexema*, um *token* é gerado e adicionado à lista de *tokens*. Por fim, seus caracteres são removidos da cadeia de caracteres do programa e um novo ciclo se inicia. Esse processo se encerra quando todo programa for analisado ou um *lexema* não for reconhecido pelo FA.

#### 3.2. ANALISADOR SINTÁTICO

A análise sintática é responsável por verificar se a sequência de *tokens* identificados na análise léxica representa um programa ou estrutura sintaticamente válida. Para isso, o analisador cria conceitualmente uma árvore de

derivação ou sintaxe para averiguar a validade da sequência de *tokens* perante uma linguagem gerada por uma gramática livre de contexto [2].

Para [25], uma gramática livre de contexto (GFC) é um conjunto de símbolos inicial, terminais, não-terminais e produções que descrevem regras de como formar uma sentença. Uma sentença é uma cadeia de símbolos que podem ser derivadas das regras de uma gramática. O símbolo inicial é o ponto de partida para efetuar derivações. Terminais são as palavras que ocorrem em uma sentença, enquanto os não-terminais representam variáveis sintáticas utilizadas pelas produções. Por fim, cada regra em uma GFC é chamada produção.

As GFC possuem diversas subclasses, dentre as quais destacam-se as LL e LR. As gramáticas LL permitem a construção de reconhecedores determinísticos que analisam a cadeia de entrada da esquerda para a direita, realizando derivações mais à esquerda. Elas caracterizam-se por serem fatoradas à esquerda, não serem ambíguas e não possuírem recursividade à esquerda. Uma gramática LR é mais geral que uma LL e permite a construção de analisadores que fazem a análise de uma cadeia de entrada da esquerda para a direita, realizando reduções mais à direita. Esta gramática também não pode ser ambígua [19].

Segundo [2], um analisador sintático geralmente é implementado baseado nas estratégias ascendente ou descendente. Utilizando uma gramática LR, a análise sintática ascendente (*bottom-up*) realiza a construção da árvore de derivação a partir das folhas em direção à raiz, ou seja, da parte de baixo para o topo. Para isso, conforme [5], são realizadas reduções de uma cadeia para o símbolo inicial da gramática. Em cada estágio da redução, uma regra de produção é substituída pelo não-terminal correspondente.

Por sua vez, o analisador descendente (*top-down*) constrói a árvore de derivação da raiz em direção às folhas, criando os nós da árvore em pré-ordem. Esta análise adota uma abordagem derivativa para construção da árvore de derivação. Utilizando uma gramática LL, inicia-se a análise a partir do símbolo inicial da gramática e a cada passo é realizada a substituição de um não-terminal pelo corpo de uma de suas produções. Esse processo é chamado de derivação [2].

Conforme apontam [25], existem inúmeras formas de implementar um analisador descendente, como o analisador sintático de descida recursiva e preditivo sem recursão. Neste trabalho, optou-se por utilizar o método de análise preditiva por ser o método mais eficiente de análise descendente. O analisador preditivo utiliza uma pilha para simular a árvore de derivação, produzindo uma derivação mais à esquerda, e uma tabela de análise, chamada de *M*, para determinar a produção a ser utilizada para cada não-terminal de acordo com o próximo símbolo a ser analisado.

Por analisar o símbolo subsequente, é empregada uma classe de gramática LL chamada de LL(1).

### 3.3. TRADUÇÃO DIRIGIDA POR SINTAXE

A tradução dirigida pela sintaxe consiste, fundamentalmente, na adição de operações conhecidas como regras ou ações semânticas nas produções de uma gramática livre de contexto. Além disto, conforme apontado em [2], atributos são agregados aos símbolos da GFC e seus valores são especificados pelas operações associadas às produções. Por ser executada simultaneamente à análise sintática, essa técnica é útil para a verificação de tipos na análise semântica e para a geração de código intermediário.

De acordo [2], a tradução dirigida pela sintaxe pode ser implementada por definições dirigidas por sintaxe (SDD) ou esquemas de tradução dirigido por sintaxe (SDT). Uma SDD é uma GFC com o acréscimo de atributos e regras que estabelecem dependências entre os atributos. Enquanto que um SDT é uma notação complementar para a SDD que incorporam fragmentos de código, chamados de ações semânticas, no corpo da produção.

Segundo [5], os atributos dividem-se em sintetizados e herdados. Um atributo sintetizado é definido totalmente a partir do valor de atributos do próprio nó, seus filhos e constantes, enquanto que o herdado é definido através do valor dos demais atributos do próprio nó e de seus irmãos ou pai. Existem as definições S-Atribuídas e L-Atribuídas, que são classes de SDDs que garantem uma ordem de avaliação sem ciclo de dependência [2]. A definição S-Atribuída é composta apenas de atributos sintetizados e é associada a análise ascendente e gramáticas LR, correspondendo assim, a um caminho pós-ordem na árvore. Por sua vez, a definição L-Atribuída é formada por atributos sintetizados do próprio símbolo, ou herdados e/ou sintetizados associados ao pai ou à irmãos localizados à esquerda do símbolo. Este tipo de definição é associada à análise descendente e gramáticas LL, correspondendo a um caminho pré-ordem na árvore.

Conforme definido anteriormente, um SDT é uma notação para especificar uma tradução que conecta fragmentos de código às produções de uma gramática. Normalmente, o SDT é implementado durante a fase de análise sintática, sem construir de fato uma árvore de derivação. Para converter uma SDD L-Atribuída em SDT, é necessário incorporar as ações que avaliam os atributos herdados para um determinado não-terminal, imediatamente antes da ocorrência do mesmo no corpo da produção e colocar as ações que avaliam atributos sintetizados da cabeça de uma produção no fim do corpo da mesma [2]. Um SDT L-Atribuído percorre a árvore sintática em pré-ordem, processando os atributos herdados durante a descida e, posteriormente, os sintetizados durante a subida. A pilha sintática é utilizada para armazenar, além dos símbolos terminais e não terminais, as ações semânticas que serão executadas na ordem desejada e adequada.

Este trabalho utiliza um SDT L-Atribuído para gerar o código de três endereços, uma representação intermediária das expressões algébricas analisadas pelo compilador. Por utilizar um analisador sintático descendente preditivo, é necessário manter os atributos dos símbolos da gramática na pilha sintática para se implementar um SDT em uma gramática LL. Segundo [2], as ações semânticas, neste momento chamadas de registros-de-ação, buscam os atributos necessários a partir de posições conhecida na pilha. Os atributos de cada símbolo estão em uma estrutura chamada de registro-de-sintetizados e ficam abaixo na pilha de todos os símbolos derivados a partir deste. Ao utilizar a estratégia de estender a pilha sintática para conter registro-de-ação e registro-de-sintetizados, é garantido que os atributos necessários para a execução de uma ação semântica estejam disponíveis até a execução da mesma.

### 3.4. GERADOR DE CÓDIGO INTERMEDIÁRIO

A geração de código intermediário é responsável por gerar uma IR (Representação Intermediária) que representa um código escrito em uma linguagem de alto nível em uma estrutura entendível para o *back end*. Conforme [5], todo compilador utiliza alguma forma de IR para modelar o código que está sendo analisado, traduzido e otimizado. Uma forma usual de representar uma IR é através do código de três endereços.

O nome *três endereços* está relacionado à especificação de uma instrução, que deve ter no máximo um operador do lado direito de uma atribuição e três variáveis, duas para os operandos e uma para o resultado. Nenhuma expressão constituída com vários operadores é permitida. Desta forma, a expressão  $x + y * z$  deve ser traduzida para o conjunto de instruções de três endereços mostrado na Figura 2. Segundo [4], o código de três endereços é atraente por vários motivos, destacando-se seu tamanho razoavelmente compacto e por não oferecer operações destrutivas.

$$\begin{aligned}
 t_1 &= y * z \\
 t_2 &= x + t_1
 \end{aligned}$$

Figura 2 - Conjunto de instruções de três endereços extraído de [2].

Ao realizar a tradução, as variáveis temporárias t1 e t2 são criadas pelo compilador para guardar os valores das operações aritméticas. A utilização dessas variáveis para valores intermediários permite que o código de três endereços seja facilmente rearranjado. Para [2], desmembrar uma expressão aritmética com múltiplos operadores e estruturas de fluxo de controle aninhados torna o código de três endereços vantajoso para a geração e otimização de código objeto.

De acordo com [5], o código de três endereços é frequentemente implementado como um conjunto de quádruplas que possui quatro campos: um operador, dois

operandos (ou origem) e um destino. A Figura 3 demonstra como a expressão  $x + y * z$  deve ser representada em uma quádrupla.

Este trabalho utiliza uma IR que implementa o conceito do código de três endereços, para representar instruções de atribuição através de uma quádrupla expandida. Foram adicionados dois campos à quádrupla, um para indicar se o termo da expressão algébrica se encontra à esquerda ou à direita do sinal de comparação e outro para indicar o nível de precedência de operações contidas em parênteses, colchetes ou chaves. Esta modificação é justificada devido à necessidade de trocar a operação para uma inversamente proporcional, a ser realizada sobre um termo quando este muda de posição em relação ao sinal de comparação e por ser indispensável resolver operações de maior prioridade antes das demais.

Operador	Arg. 1	Arg. 2	Resultado
*	y	z	T <sub>1</sub>
+	x	T <sub>1</sub>	T <sub>2</sub>

Figura 3 - Representação da quádrupla.

Após o processo de análise e geração do código intermediário da expressão algébrica, que é representado por meio das quádruplas expandidas e necessário para transformar a expressão em um formato processável pelo computador, o sistema especialista utiliza esta representação para realizar as manipulações algébricas necessárias para a resolução do problema proposto.

## 4. SISTEMAS ESPECIALISTAS

A inteligência artificial é definida por [15] como o ramo da computação preocupada com a automação de comportamento inteligente. Dentre as técnicas existentes dentro da inteligência artificial, os sistemas especialistas (SE) se destacam por reproduzir o conhecimento de um especialista humano em uma área específica e bem delimitada.

Os SE são algoritmos que visam encontrar soluções para determinados problemas do mesmo modo que especialistas humanos, se estiverem sob as mesmas condições, usando o conhecimento representado de forma explícita para resolver problemas difíceis em domínios restritos. Segundo [20], a construção de um SE que incorpora o conhecimento humano especializado sobre uma área usando fatos e regras é uma das aplicações de maior sucesso das técnicas de raciocínio da inteligência artificial.

Os SE são normalmente projetados para resolver problemas relacionados às atividades de controle, diagnóstico, prescrição, monitoramento, interpretação, previsão, simulação, seleção, projeto, planejamento ou instrução [7]. Por inferir a resolução de uma expressão algébrica

informada, o SE elaborado neste trabalho realiza atividades de interpretação. Além disso, ao explicar teoricamente a resposta ao usuário, assim auxiliando no processo de ensino-aprendizagem da disciplina de matemática, este SE executa atividades de instrução.

Para que um sistema seja considerado especialista, [7] afirma que este deve possuir ao menos os seguintes módulos: uma base de conhecimento, que armazena por meio de um formalismo o conhecimento de um perito da área para ser utilizado em todas as consultas ao sistema; uma memória de trabalho, que contém todos os fatos descobertos durante uma consulta; e um mecanismo de inferência para associar os fatos presentes na memória de trabalho com as regras armazenadas na base de conhecimento para concluir novos fatos. A Figura 4 representa como esses módulos se relacionam entre si e com o usuário.

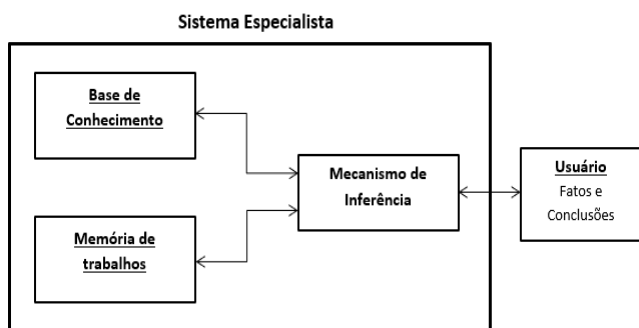


Figura 4 - Sistema especialista baseado em conhecimento  
Fonte: Adaptado de [7]

Tendo em vista a necessidade de explicar ao usuário os passos para resolver uma expressão, o mecanismo de explanação é um módulo adicional de extrema importância para o SE projetado neste trabalho. Conforme aponta [7], esse mecanismo é responsável por explicar ao usuário o raciocínio que está sendo executado pelo SE, justificando o motivo das conclusões do sistema. Através da utilização deste mecanismo, é possível explicar a resolução das expressões. Prover essas explicações sobre um domínio não trivial é essencial para o usuário entender como funciona o processo de resolução de um determinado problema.

O conhecimento do especialista humano pode ser representado de várias maneiras processáveis pelo computador chamadas de formalismo. Entre os inúmeros formalismos existentes, destacam-se: lógica ou cálculo proposicional, lógica ou cálculo de predicados, ontologias, frames, redes semânticas e regras de produção. Para [7], cada uma dessas técnicas enfatiza certas informações sobre o problema, enquanto ignora outras. Diante disso, cada uma trabalha com diferentes tipos de conhecimento, fazendo da escolha do formalismo mais adequado para um problema essencial para projetar uma estrutura eficaz na busca pela solução.

As regras de produção (RP) são estruturas comuns e eficientes para resolver problemas quando o SE deve gerar novas informações a partir de inferências sobre fatos

conhecidos [7]. As RP são uma forma de conhecimento estrutural, pois associam um fato com uma ação, descrevendo assim como resolver um problema. A estrutura de uma RP associa-a logicamente com um antecedente ou premissa e uma consequência ou conclusão. Como exemplo, se o antecedente/premissa é verdadeiro, então a regra infere que a consequência/conclusão é verdadeira também. Por suas características, este é o formalismo escolhido para representar o conhecimento neste trabalho.

## 5. MATERIAS E METODOS

Por empregar conhecimentos existentes para serem aplicados na prática a fim de resolver um problema específico, esta pesquisa é definida como aplicada, que de acordo com [10], é voltada à aquisição de conhecimentos para aplicação em uma situação específica.

A presente pesquisa é considerada qualitativa, pois se interpreta fenômenos e atribui-se significados sem requerer de técnicas e métodos estatísticos. Para [17], esta metodologia preocupa-se em analisar e interpretar aspectos mais profundos, descrevendo a complexidade do comportamento de forma indutiva pelo próprio autor.

Os procedimentos técnicos adotados neste trabalho classificam-se como uma pesquisa bibliográfica e estudo de caso. Segundo [10], uma pesquisa bibliográfica é o estudo sistematizado desenvolvido com base em materiais já publicados. Sendo assim, por utilizar o conhecimento elaborado por outros autores e interligando-os com a parte prática do estudo nas suas fases iniciais, esta pesquisa é considerada uma pesquisa bibliográfica. Ainda para [10], o estudo de caso é um profundo e exaustivo estudo de um ou poucos objetos, de maneira que permita seu conhecimento amplo e detalhado. Portanto, como este trabalho se propõe a estudar algumas soluções disponíveis no mercado a fim de identificar as qualidades de cada uma, esta etapa do mesmo pode ser considerada um estudo de caso.

Para uma melhor organização, a metodologia para o desenvolvimento do presente trabalho foi dividida em etapas. O início deu-se com o desenvolvimento do módulo do compilador na linguagem Java. Para isso, foram definidas as expressões regulares, mostradas na Figura 5, que determinam os padrões de entrada válidos.

$L \rightarrow \{A-Z, a-z, \_ \}$	$= \rightarrow =$	$\{ \rightarrow \{$
$D \rightarrow \{0-9 \}$	$\langle \rightarrow \langle$	$\} \rightarrow \}$
$id \rightarrow L(D L)^*$	$\leq \rightarrow \leq$	$sen \rightarrow sen$
$id\_com\_coeficiente \rightarrow numero\_natural\ id$	$\lt \rightarrow \lt$	$cos \rightarrow cos$
$numero\_natural \rightarrow D(D)^*$	$\gt; \rightarrow \gt;$	$tg \rightarrow tg$
$numero\_decimal \rightarrow D(D)^*, D(D)^*$	$\> \rightarrow \>$	$raiz \rightarrow raiz$
$+ \rightarrow +$	$( \rightarrow ($	$raizq \rightarrow raizq$
$- \rightarrow -$	$) \rightarrow )$	$log \rightarrow log$
$* \rightarrow *$	$[ \rightarrow [$	$log10 \rightarrow log10$
$/ \rightarrow /$	$] \rightarrow ]$	$;$
$\wedge \rightarrow \wedge$		

Figura 5 - Expressões regulares para expressões algébricas.

Estas expressões regulares foram utilizadas na implementação do analisador léxico, que tem a função de identificar símbolos válidos das expressões algébricas através da leitura e análise das cadeias de caracteres que a compõe. Para cada símbolo válido identificado é gerada uma saída chamada de *token*. Os símbolos &, # e @ são exemplos de caracteres inválidos identificados pela análise léxica.

Em seguida, foi definida uma gramática livre de contexto LL para especificar como formar expressões válidas. A Figura 6 exibe a gramática elaborada neste trabalho.

```

E -> TCT
C -> = | < > | > = | < = | < >
T -> MT'
T' -> + MT' | - MT' | ε
M -> PM'
M' -> * PM' | / PM' | ε
P -> OP'
P' -> ^ OP' | ε
O -> SF
S -> + | - | ε
F -> FUNC | ( T ) | [ T ] | { T } | id | id_com_coeficiente |
      numero_natural | numero_decimal
FUNC -> sen ( T ) | cos ( T ) | tg ( T ) | raizq ( T ) | raiz ( T ; numero_natural ) |
      log10 ( T ) | log ( T ; numero_natural )

```

**Figura 6 - Gramática livre de contexto LL para expressões algébricas.**

Tal gramática foi utilizada na implementação do analisador sintático descendente preditivo, que tem como principal objetivo averiguar a validade da sequência de *tokens* perante uma linguagem gerada pela gramática livre de contexto, ou seja, verificar se correspondem a uma expressão algébrica válida. Nesta etapa, por exemplo, são identificados parênteses sem fechamento, a ausência de operadores e operandos em uma operação, dentre outras inconsistências.

Para finalizar o desenvolvimento do compilador, foi implementado o gerador de código intermediário. Para isso, foi utilizado um esquema de tradução dirigido por sintaxe L-Atribuído, que consiste na adição de ações semânticas nas produções da gramática. A Figura 7 exibe uma parte da gramática com ações semânticas inseridas no corpo das produções.

A *AS6* apresentada na figura 7 cria uma nova operação do código de três endereços por meio de uma quádrupla expandida. Por sua vez, a *AS3* é responsável por registrar o sinal de comparação, assim como, registrar a quádrupla inicial da expressão algébrica no lado esquerdo e direito do sinal de comparação. As demais ações semânticas exibidas

na Figura 7 realizam a deslocamento dos valores referentes à posição e ao nível para os níveis mais profundos da árvore de derivação. Esses valores são utilizados na criação das quádruplas e são importantes para a resolução do problema pelo sistema especialista. É importante destacar que outras ações foram inseridas no corpo de todas as produções da gramática exibida na Figura 6.

*E -> AS1 T<sub>1</sub> CAS2 T<sub>2</sub> AS3*

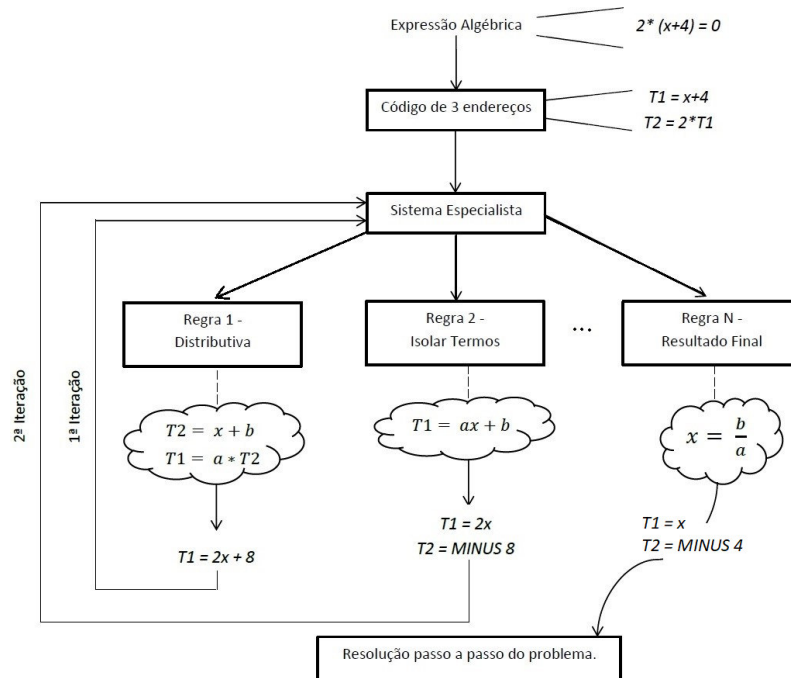
*T -> AS5 M T' AS6*

Ações semânticas (AS)	
Código	Código-fonte
<i>AS1</i>	<i>T<sub>1</sub>.position = 0;</i> <i>T<sub>1</sub>.level = 0;</i>
<i>AS2</i>	<i>T<sub>2</sub>.position = 1;</i> <i>T<sub>2</sub>.level = 0;</i>
<i>AS3</i>	<i>E.param1 = T<sub>1</sub>.addr;</i> <i>E.comp = C.addr;</i> <i>E.param2 = T<sub>2</sub>.addr;</i>
<i>AS5</i>	<i>M.position = T.position;</i> <i>T'.position = T.position;</i> <i>M.level = T.level;</i> <i>T'.level = T.level;</i>
<i>AS6</i>	<i>if ( T'.value ) {</i> <i>    T.addr = new Temp();</i> <i>    AddNewOperation( T'.op, M.addr, T'.addr,</i> <i>                    T.addr, T.posicao, T.level );</i> <i>    }</i> <i>else {</i> <i>        T.addr = M.addr;</i> <i>    }</i>

**Figura 7 - Gramática livre de contexto com ações semânticas.**

A última etapa correspondeu à modelagem, implementação e validação do SE. Inicialmente, foi definido junto a um especialista em matemática as regras para as manipulações algébricas a serem incorporadas ao sistema. Essas regras representam o passo a passo para a resolução de um problema que um professor normalmente ensina aos alunos. Elas são fundamentais para atingir o propósito deste trabalho, pois através de sua utilização, torna-se possível vincular a teoria pertinente ao assunto com cada passo da resolução

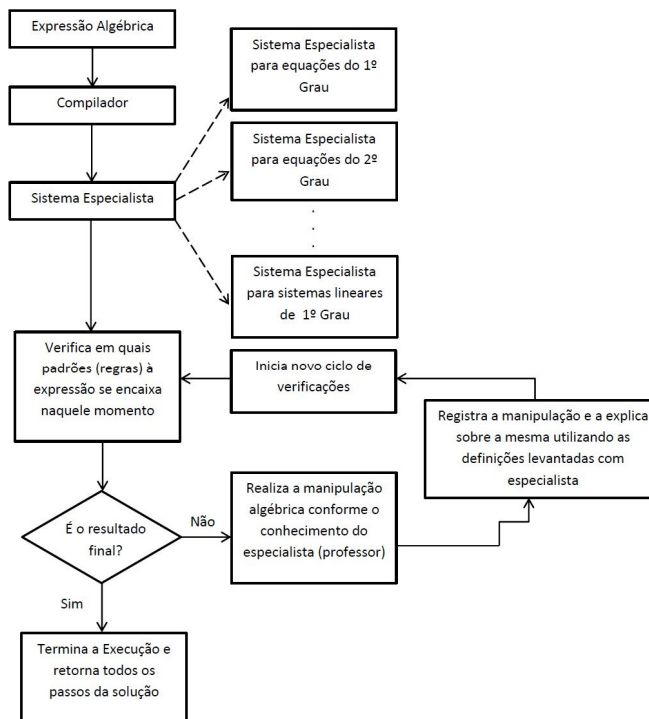
O SE analisa o código de três endereços gerado pelo compilador com o objetivo de identificar padrões que se encaixam nas regras definidas junto ao especialista. Ao encontrar uma combinação, é realizada uma série de transformações sobre o código, gerando uma representação da resolução ensinada pelos professores aos alunos. Esse processo é iterativo, sendo repetido até o SE encontrar a solução do problema. A representação conceitual do funcionamento do SE e de suas regras de produção é apresentada na Figura 8.



**Figura 8 - Representação conceitual do SE e suas regras de produção.**

Em seguida, realizou-se a implementação do SE que efetua manipulações algébricas com base nas regras definidas previamente. O SE recebe as expressões algébricas informadas pelos usuários, após serem verificadas e transformadas pelo compilador, e retorna o resultado das

manipulações das mesmas em notação LaTeX, para facilitar ao leitor a visualização dos dados, principalmente das fórmulas matemáticas. Uma visão geral do fluxo de funcionamento é apresentada pela Figura 9.



**Figura 9 – Fluxo conceitual do sistema**

## 6. RESULTADOS E DISCUSSÃO

A ferramenta proposta por este estudo busca aproximar teoria e prática, visando o desenvolvimento cognitivo do aluno, e possibilitar aos professores uma nova forma de apresentar os conteúdos. Para isso, o desenvolvimento foi dividido em duas partes, uma conceitual e outra voltada às manipulações algébricas. Tal abordagem, igualmente utilizada nos demais módulos do *IFMath*, gerou uma padronização de funcionamento e interfaces de todo o sistema.

A parte conceitual apresenta o embasamento teórico necessário sobre o tema. Como mostra a Figura 10, esses conceitos encontram-se no menu *Definição* e proporcionam uma visão geral sobre o conteúdo, servindo como base para a construção do conhecimento do aluno. A teoria necessária para elaboração deste material foi obtida em livros de matemática utilizados no ensino fundamental e médio. Este processo foi acompanhado por um professor de matemática, visando contemplar todos os conceitos necessários e organizá-los da melhor maneira, levando em consideração o aspecto pedagógico.



Figura 10 – Interface de conceitos sobre tema.

Por sua vez, o manipulador algébrico é responsável por elaborar a resolução passo a passo dos problemas propostos pelo usuário tomando como base as regras levantadas com um professor de matemática. Ele utiliza conceitos de compiladores para a validação e geração de uma representação intermediária da expressão algébrica que será manipulada pelo sistema especialista. A Figura 11 mostra a resolução de um problema pelo manipulador algébrico

desenvolvido neste trabalho. Ressalta-se que no campo de edição existente na parte superior da interface, o usuário pode digitar qualquer expressão algébrica de seu interesse. Ao clicar no botão *Calcular*, a expressão será verificada pelo compilador e, se estiver correta, sua representação intermediária é repassada ao sistema especialista, o qual realiza as manipulações adequadas e apresenta o resultado.

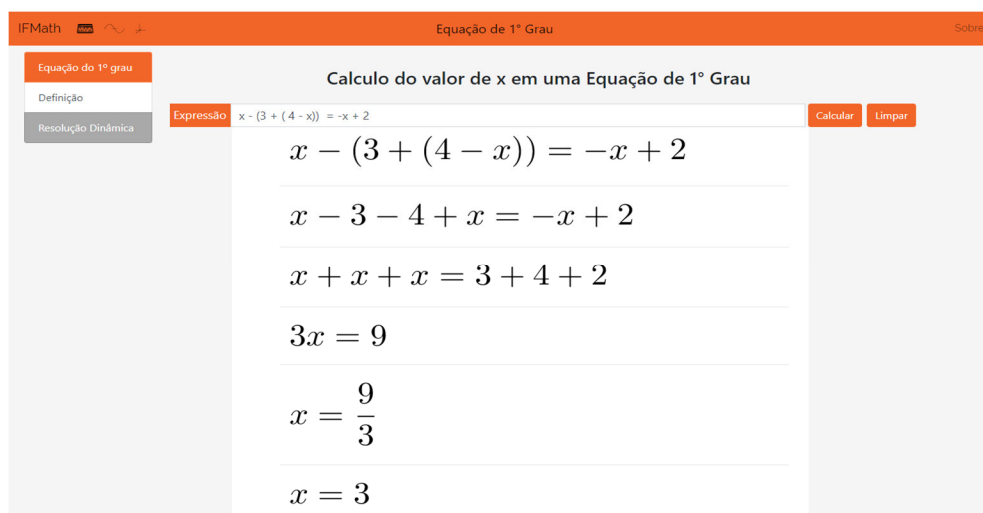


Figura 11 - Interface para resolução de equações de 1º grau

Como pode ser visto na figura 12, uma explicação sobre cada uma das manipulações algébricas realizadas é disponibilizada pela ferramenta. Essas informações foram obtidas junto a um professor de matemática e incorporadas ao conhecimento do sistema especialista. Por padrão, esses detalhes não são apresentados em primeiro momento (figura 11), podendo ser exibidos ou omitidos de acordo com o interesse do usuário. O objetivo é complementar a teoria apresentada na interface de definição, possibilitando que o usuário visualize quais conceitos foram aplicados em cada passo da resolução do problema, aumentando assim o entendimento do mesmo sobre o assunto como um todo.

$$x - (3 + (4 - x)) = -x + 2$$

Equação inicial.

$$x - 3 - 4 + x = -x + 2$$

Aplicação da regra de sinais em operações prioritárias, em duplas negações ou em somas de números negativos.

$$x + x + x = 3 + 4 + 2$$

Os termos semelhantes foram isolados. As variáveis foram movidas para a esquerda da igualdade e as constantes foram movidas para a direita. Quando é necessário mudar a posição de um termo perante a igualdade, a operação realizada sobre este é invertida para uma operação inversamente proporcional. Neste caso, quando o termo mudou de posição, + passou a ser - e - passou a ser +.

$$3x = 9$$

Soma dos termos semelhantes.

**Figura 12 - Interface para resolução de equações de 1º grau com a apresentação de explicações**

## 7. CONSIDERAÇÕES FINAIS.

A tecnologia, quando devidamente empregada na educação, passa a ser uma poderosa aliada no processo de ensino-aprendizagem ao ampliar as possibilidades na maneira de abordar os conteúdos. Os resultados alcançados por este estudo evidenciam que teorias clássicas da computação, como compiladores e sistemas especialistas, podem contribuir ativamente como ferramenta pedagógica na resolução de problemas algébricos ao permitir que o docente apresente o conteúdo de uma forma mais interessante e criativa, atraindo assim a atenção do discente, que, por sua vez, tem à disposição uma ferramenta para estudos extra classe.

A partir do desenvolvimento e incorporação do módulo algébrico no *IFMath*, o usuário tem a disposição um recurso que possibilita a análise e interpretação de equações de primeiro grau de maneira dinâmica, ou seja, o usuário pode inserir qualquer equação de seu interesse no *software*. Este conteúdo constitui uma área essencial à compreensão da matemática, pois permite aos alunos uma visão mais ampla

ao deixar de realizar apenas contas com números, passando a trabalhar com variáveis representadas por letras. Além disso, cada manipulação algébrica realizada para se alcançar a resposta é justificada através de um embasamento teórico, permitindo assim o desenvolvimento cognitivo do aluno sobre o conteúdo abordado.

Nas próximas etapas deste estudo, pretende-se acrescentar ao software mais interatividade com o usuário, permitindo que este selecione a manipulação algébrica desejada em cada passo da resolução dos problemas propostos. Tal funcionalidade possibilitará uma visão clara das consequências de uma manipulação sobre a quantidade de passos necessários para resolver o exercício ao comparar a resposta elaborada pelo usuário com a melhor resposta de acordo com as regras levantadas junto a um especialista. Além disso, os conteúdos de equações polinomiais, sistemas de equações, fatoração, expressões algébricas e produtos notáveis serão incorporados ao sistema. Por fim, com o intuito de aumentar a qualidade da ferramenta, serão realizados mais testes do *software* com outros professores de matemática da educação básica e também junto aos alunos, a fim de validá-lo.

## REFERÊNCIAS

1. AHO, A. V.; SETHI, R.; ULLMAN, J. D. Compiladores: princípios, técnicas e ferramentas. 1ª ed.. Tradução: Daniel de Ariosto Pinto. Rio de Janeiro: LTC, 1995.
2. AHO, A. V.; SETHI, R.; ULLMAN, J. D. Compiladores: princípios, técnicas e ferramentas. 2ª ed.. Rio de Janeiro: Pearson Prentice Hall, 2008.
3. BORBA, M. C; GADANIDIS, G.; SILVA, R. Fases das tecnologias digitais em Educação Matemática: sala de aula e internet em movimento. Belo Horizonte: Autêntica, 2014.
4. BRUM, L. D.. Análise de erros cometidos por alunos de 8º ano do ensino fundamental em conteúdos de álgebra. Curso de Mestrado Profissionalizante em Ensino de Física e Matemática. Centro Universitário Franciscano de Santa Maria. Rio Grande do Sul, 2013.
5. COOPER, K. D.; TORCZON, L. Construindo Compiladores. 2.ed. Rio de Janeiro: Elsevier, 2014. Tradução: Daniel Vieira.
6. DRUCK, S. A crise no Ensino de Matemática no Brasil. Revista do Professor de Matemática, Vol. 52. Rio de Janeiro, 2003.
7. DURKIN, J.. Expert systems: design and development. Englewood Cliffs: Prentice Hall, 1994.
8. FIORENTINI, D.; MORIN, M. Â.. Por trás da porta, que matemática acontece? Campinas, SP: FE/Unicamp – Cempem, 2001.
9. FRANÇA, K. V., SANTOS, J. A., SANTOS, L. S. B.. Dificuldades na aprendizagem da Matemática. Tese de

- licenciatura em Matemática Centro Universitário Adventista de São Paulo, Campus de São Paulo, 2007.
10. GIL, A. C.. Como Elaborar Projetos de Pesquisa. 5ª ed. São Paulo: Atlas, 2010.
  11. GLADCHEFF A. P.; ZUFFI, E.M.; SILVA, M.. Um Instrumento para Avaliação da Qualidade de Softwares Educacionais de Matemática para o Ensino Fundamental. Anais do XXI Congresso da Sociedade Brasileira de Computação. Fortaleza, 2001.
  12. GOMES, A. S.; PADOVANI, S.. Usabilidade no ciclo de desenvolvimento de software educativo. Simpósio Brasileiro de Informática na Educação SBIE2005, Juiz de Fora, 2005.
  13. GRAMANI, M. C. N.; SCRICH, C. R.. Influência do desempenho educacional na escolha da profissão. Cadernos de Pesquisa, v. 42, n. 147, p. 868-883, 2013.
  14. GRAVINA, M. A.; SANTAROSA, L. M.. A aprendizagem da matemática em ambientes informatizados. IV Congresso RIBIE, Brasília 1998.
  15. LUGER, G. L., STUBBLEFIELD, W. A., Artificial Intelligence: Structures and Strategies for Complex Problem Solving. 6ª ed. Benjamin Cummings, 2008.
  16. MAIOR, L.; TROBIA, J.. Tendências metodológicas de ensino-aprendizagem em educação matemática: resolução de problemas - um caminho. Ponta Grossa, 2009.
  17. MARCONI, M. A.; LAKATOS, E. M.. Metodologia Científica. 5ªed. São Paulo: Atlas,2011.
  18. MEC. Resultado do Pisa de 2015 é tragédia para o futuro dos jovens brasileiros, afirma ministro. 2016.
  19. NETO, J. J.. Introdução à compilação. 2ª ed.. Rio de Janeiro: Elsevier, 2016.
  20. NILSSON, N.. Artificial intelligence: a new synthesis. Burlington: Morgan Kaufmann Publishers, 1998.
  21. PERIUS, A. A. B. A tecnologia aliada ao ensino de matemática. Trabalho de conclusão de curso de especialista em mídias na educação para Universidade Federal do Rio Grande do Sul (UFRGS). Porto Alegre, 2012.
  22. PONTE, J.P. Álgebra no currículo escolar. Educação em Matemática – Revista da Associação de Professores de Matemática, Lisboa, n. 85, Nov./Dez., 2005.
  23. ROCHA, A. R.; CAMPOS, G. H.. Avaliação da qualidade de software educacional. 2008.
  24. RODRIGUES, L. L. A Matemática ensinada na escola e a sua relação com o cotidiano. Brasília: UCB, 2005.
  25. SINGH, R.; SHARMA, V.; VARSHNEY, M.. Design and Implementation of Compiler. 1ª ed.. Nova Deli: New Age International, 2008.
  26. SOCAS, M. M.; CAMACHO M.; PALAREA M.; HERNÁNDEZ J.; Iniciación al algebra. Madrid: Ed Síntesis, 1996.
  27. VASCONCELOS, C. C. Ensino-aprendizagem da matemática: velhos problemas, novos desafios. Lisboa: Editora Instituto Politécnico de Viseu, 2000.
  28. ZORZAN, A. S. L.. Ensino-Aprendizagem: algumas tendências na educação matemática. Revista Ciências Humanas. URI - Frederico Westphalen, 2007.