

Sistema Inteligente para o Ensino-Aprendizagem de Expressões Algébricas

Title: Intelligent System for Teaching-Learning of Algebraic Expressions

Alex Junior Avila
Instituto Federal de Santa Catarina
alex_avila11@hotmail.com

EneDir Guimarães de Oliveira
Junior
Instituto Federal de Santa Catarina
guimaraes.junior94@gmail.com

Wilson Castello Branco Neto
Instituto Federal de Santa Catarina
wilson.castello@ifsc.edu.br

Ailton Durigon
Instituto Federal de Santa Catarina
ailton.durigon@ifsc.edu.br

Resumo

Os resultados do ENEM e exames vestibulares evidenciam a dificuldade dos alunos quanto ao domínio da matemática. Por outro lado, a tecnologia está se tornando uma importante aliada no aprendizado em sala de aula e está ganhando cada vez mais popularidade. Existe uma grande variedade de soluções disponíveis que visam auxiliar no processo de ensino-aprendizagem de matemática, sendo uma delas o software educacional IFMath. Entretanto, ele não possuía a funcionalidade para manipular expressões algébricas, um dos conteúdos mais complexos para serem ministrados. Diante disso, este trabalho teve como objetivo desenvolver uma ferramenta para a manipulação de expressões algébricas para ser integrada ao IFMath. O software proposto busca aproximar teoria e prática, visando desenvolver o cognitivo do aluno, e possibilitar aos professores uma nova forma de apresentar os conteúdos. Ele apresenta conceitos que proporcionam uma visão geral sobre o conteúdo e elabora a resolução passo a passo dos problemas propostos pelo usuário. Para isto, utiliza um compilador para validar a expressão algébrica e um sistema especialista, que adota o mecanismo de encadeamento progressivo, para executar as regras de produção que contêm o conhecimento sobre como resolver cada expressão. Além disso, apresenta uma explicação sobre cada uma das manipulações algébricas realizadas. Durante uma dinâmica, um grupo de professores avaliaram que o software possui os requisitos mínimos para ser utilizado como ferramenta de ensino-aprendizagem na sala de aula e em atividades extraclasse. 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: Ensino-Aprendizagem; Expressões Algébricas; Sistema Especialista; Compilador.

Abstract

The results of the ENEM and the vestibular exam show an attempt of the students in the field of mathematics. On the other hand, a technology is becoming an important ally in classroom learning and is gaining more and more popularity. There are a wide variety of solutions that aim to aid the teaching-learning process of mathematics, one of them being the educational software IFMath. However, it does not have a functionality to manipulate algebras. Therefore, this work aimed to develop a tool for the manipulation of algebras to be integrated with IFMath. The proposed software aims at relating theory and practice, in order to develop the student's cognitive, and to give teachers a new way of presenting the contents. It presents the concepts that provide an overview on the subject and elaborates a step by step solution of the problems proposed by the user. Thereunto, using a compiler to validate the expression and expert system which adopts the forward chaining to execute the production rules containing the knowledge of an expert to solve it. In addition, it provides an explanation of each of the algebraic manipulations

Cite as: Avila, A. J., de Oliveira Junior, E. G., Branco Neto, W. C. & Durigon, A. (2020). Intelligent System for Teaching-Learning of Algebraic Expressions (Sistema Inteligente para o Ensino-aprendizagem de Expressões Algébricas). Brazilian Journal of Computers in Education (Revista Brasileira de Informática na Educação - RBIE), 28, 359-388. DOI: 10.5753/RBIE.2020.28.0.359

performed. During an experiment, a group of teachers evaluate that the software has the minimum requirements for being used as a teaching and learning tool in the classroom and in extracurricular activities. The results indicate the classical theories of computation, such as compilers and expert systems, can be applied as a pedagogical tool in solving algebraic problems.

Keywords: Teaching-Learning; Algebraic Expressions; Expert System; Compiler.

1 Introdução

A matemática é considerada uma disciplina muito complexa pela maioria dos alunos. Segundo Fiorentini e Morin (2001), 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 acerto em testes de avaliação como vestibulares e o ENEM.

Conforme aponta Socas *et al.* (1996), o processo de ensino-aprendizagem da álgebra apresenta uma variedade de dificuldades, dentre as quais destacam-se o desenvolvimento cognitivo limitado dos alunos e a natureza complexa deste ramo da matemática. De acordo com Brum (2013), essas inúmeras dificuldades levam a uma falta de interesse pela matemática e ao seu consequente prejuízo em relação aos 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 à presença de coeficientes e incógnitas.

A dificuldade para o entendimento da álgebra por parte dos estudantes acarreta frequentes erros, destacando-se o uso incorreto da notação e de convenções algébricas necessárias para a resolução dos problemas (Ponte, 2005).

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 (Gramani & Scrich, 2013). 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.

Santos *et al.* (2018) incentivam a utilização de recursos tecnológicos no ensino da matemática. 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*, o qual propõe resoluções passo a passo de exercícios matemáticos apresentando sua fundamentação teórica. Entretanto, esta solução carece de um módulo relacionado à manipulação e solução de expressões algébricas, um importante e complexo conteúdo relativo à álgebra elementar que faz parte dos conteúdos matemáticos da Educação Básica.

Os sistemas especialistas, uma área de estudo da inteligência artificial, vem auxiliando pessoas na resolução de problemas que envolvem conhecimento, heurísticas e tomada de decisão em diferentes domínios do conhecimento (Khan, Amin & Rehman, 2011). Dentre eles, destaca-se a educação que, segundo Khanna, Kaushik e Barnela (2010), está sendo revolucionada pelos sistemas especialistas, pois são capazes de realizar: “(1) planejamento educacional e tomada de decisão, gerenciamento de registros de alunos, aconselhamento e programas de educação especial; (2) treinamento de professores, especificamente identificando necessidades de treinamento, usando instrução assistida por computador para ensinar conhecimentos e habilidades; e (3) sistemas tutores inteligentes que orientam os alunos de acordo com seus pontos fortes e fracos.

Diante do exposto, neste trabalho investiga-se a aplicação de técnicas de compiladores e inteligência artificial para a construção de um sistema capaz de realizar manipulações em expressões algébricas informadas dinamicamente pelo usuário. Aspectos relacionados às etapas

de análise léxica, análise sintática e geração de código intermediário, da área de compiladores, são utilizados para verificar a correção da expressão informada pelo usuário, bem como para transformá-la em uma estrutura que pode ser manipulada por um sistema especialista, que contém o conhecimento do professor sobre como a expressão deve ser resolvida.

Este sistema foi 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. Os resultados alcançados em um experimento realizado com professores de matemática das séries finais do ensino fundamental demonstram que o sistema pode contribuir ativamente como ferramenta pedagógica na resolução de problemas algébricos, pois permite 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 que pode atuar como um tutor em estudos extra classe.

Este artigo está organizado da seguinte maneira: após esta introdução, a seção 2 classifica metodologicamente o presente trabalho e descreve as atividades realizadas. Em seguida, a seção 3 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 4 e 5 apresentam o referencial teórico das áreas de compiladores e sistemas especialistas, os quais englobam as técnicas utilizadas para a implementação do sistema. A seção 6 descreve as técnicas utilizadas na implementação deste trabalho, enquanto a seção 7 apresenta o sistema resultante. Por fim, na seção 8 são tecidas algumas considerações.

2 Materiais e Métodos

Devido ao emprego de conhecimentos existentes a serem aplicados na prática para resolver um problema específico, esta pesquisa é definida como aplicada. Conforme aponta Gil (2010), a pesquisa aplicada 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. Marconi e Lakatos (2011) descrevem que a metodologia qualitativa se preocupa 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 Gil (2010), 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 Gil (2010), 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 características de cada uma, as etapas finais do mesmo podem ser consideradas um estudo de caso.

Para uma melhor organização, a metodologia para o desenvolvimento do presente projeto foi dividida em etapas. A primeira etapa compreendeu o levantamento bibliográfico para aprofundar os conhecimentos sobre compiladores e inteligência artificial, em especial sistemas especialistas, assim como sobre os conceitos e processos matemáticos envolvidos nas manipulações algébricas, sobretudo em equações e expressões algébricas. Concluindo a primeira etapa, foi efetuado um estudo de caso das principais soluções de manipulações algébricas disponíveis no mercado, com

o intuito de identificar características positivas em cada uma delas para serem incorporadas à solução que esta pesquisa se propôs a desenvolver.

A partir do conhecimento gerado na primeira etapa, a sequência dos trabalhos deu-se com o desenvolvimento do módulo de um compilador na linguagem *Java* baseando-se na teoria de compiladores e sem usar nenhuma ferramenta e/ou biblioteca de terceiros. O processo de compilação é dividido em seis etapas, sendo que somente as análises léxica e sintática e geração de código intermediário foram de interesse deste trabalho. Para isso, foram definidas expressões regulares para determinar os padrões de entrada válidos com o intuito de selecionar os *tokens* da expressão. Estas expressões regulares foram utilizadas na implementação de um analisador léxico, que tem a função de identificar símbolos inválidos das expressões algébricas, ou seja, *tokens* inválidos.

Em seguida, foi definida uma gramática livre de contexto LL para especificar as expressões válidas. Tal gramática foi necessária na implementação de um analisador sintático descendente preditivo, que tem como principal objetivo identificar erros de sintaxe em uma expressão algébrica.

Finalizando a etapa de desenvolvimento do compilador, um gerador de código intermediário foi implementado. Para isso, utilizou-se um esquema de tradução dirigido por sintaxe L-Atribuído, que requer a adição de ações semânticas nas produções da gramática com o intuito de gerar o código de três endereços, uma representação intermediária da expressão aritmética que pode ser facilmente manipulada computacionalmente.

Com os analisadores léxico, sintático e gerador de código intermediário desenvolvidos, iniciou-se a quinta etapa do projeto. Esta etapa correspondeu a modelagem e implementação de um sistema especialista (SE). Inicialmente, definiu-se junto a um especialista em matemática as regras para as manipulações algébricas que devem ser incorporadas ao sistema. Com as regras definidas, foi implementado o SE que efetua manipulações algébricas com base nas regras definidas previamente.

A última etapa foi destinada à integração do sistema especialista com o *IFMath* e à validação dos resultados obtidos. Para isso, foi criado um novo módulo no *back-end* do software. Esse novo módulo integrou o sistema criado à arquitetura do *IFMath*. Através deste módulo, o sistema especialista recebe as expressões algébricas informadas pelos usuários e retorna o resultado das manipulações das mesmas. A conclusão desta etapa consistiu na validação dos resultados obtidos junto a professores de matemática.

3 Educação e Tecnologia

Essa seção aborda o ensino da matemática e como a tecnologia pode auxiliar neste processo. Inicialmente, explica-se sobre o ensino da matemática e como a metodologia formal de ensino da matemática não se adapta a todos os perfis de alunos. Além disso, explana como o avanço da tecnologia, principalmente com a utilização de softwares educacionais no meio de ensino, vem trazendo novas formas de interação entre alunos e professores.

Em seguida, discute-se o conceito e a importância de softwares educacionais, realizando uma análise comparativa entre três software para o ensino da matemática. Por fim é explicado sobre o software educacional *IFMath*, um software gratuito para o ensino e aprendizagem da matemática, que se propõe a resolver problemas matemáticos explicando teoricamente cada passo necessário para resolver o problema.

3.1 Ensino da Matemática

As limitações da aprendizagem de matemática são um relato de diversos estudantes. Eles sentem dificuldades que acabam por gerar altos índices de reprovação. Conforme aponta França *et al.* (2007), nem mesmo todos os alunos aprovados conseguem efetivamente utilizar esse conhecimento de fundamental importância.

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 (Rodrigues, 2005). Muitas vezes, o estudo baseia-se na memorização do resumo dos conteúdos considerados importantes e suficientes pelo professor, assim como na repetição de exercícios. Fragoso (2001) define que essa metodologia formal e mecânica torna o ensino cansativo e desestimulante, visto que desenvolve um limitado raciocínio matemático nos alunos e possui uma serventia questionável no mundo real. Devido à complexidade para abordar problemas reais quando comparados aos problemas apresentados tradicionalmente em livros, a matemática nem sempre é trabalhada de forma a levar o aluno a associá-la com o cotidiano (Andrade, 2013).

Os problemas no ensino da matemática nas escolas estão diretamente relacionados com o cenário da educação no Brasil. De acordo com Druck (2003), uma situação dessa magnitude não surge repentinamente, ela é construída ao longo de décadas de ensino deficiente. Os reflexos dessa carência podem ser observados em avaliações como o PISA (*Programme for International Student Assessment*). Segundo o Ministério da Educação (MEC, 2016), em sua 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.

As transformações sociais implicam em mudanças na educação e nessa perspectiva surgiram tendências metodológicas que pretendem mudar o panorama da educação matemática. Zorzan (2007) e Maior e Trobia (2009) destacam como as principais tendências a etnomatemática, modelagem, história e filosofia, investigação matemática, resolução de problemas e a tecnologia.

A utilização de tecnologias, especialmente softwares, pode auxiliar no desenvolvimento do raciocínio matemático do estudante. As regras, técnicas e simbologias matemáticas devem ser sempre contempladas no ensino dessa disciplina. Entretanto, existem outros aspectos tão importantes quanto esses. De acordo com Vasconcelos (2000), 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.

Perius (2012) relata que 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.

3.2 Softwares Educacionais

De acordo com Oliveira, Casagrande e Galerani (2016), “as tecnologias se transformaram, adentrando-se ao contexto educacional e têm provocado mudanças consideráveis no processo de ensino e aprendizagem”. Neste contexto, aparece um novo modelo de educação, no qual ferramentas como giz, quadro e livros não são mais os únicos instrumentos utilizados para professores lecionarem. Tal modelo necessita do desenvolvimento de um conjunto de atividades didático-pedagógicas que apoie professores e alunos no aperfeiçoamento do processo de ensino-aprendizagem, a partir das tecnologias disponíveis na sala de aula e que os alunos trazem consigo (Slomski, Araujo, Camargo e Weffort, 2016).

Ao se referirem sobre as fases das tecnologias digitais em Educação Matemática, Borba, Gadanidis e Silva (2014) evidenciam 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 da velocidade na mesma. 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, Gravina e Santarosa (1998) afirmam 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 por meio da visualização, experimentação, interpretação e demonstração dos conteúdos. Ao informatizar uma sala de aula, torna-se possível a utilização de softwares educacionais, uma importante ferramenta tanto para professores como para alunos, no processo de ensino-aprendizagem.

Um software educacional pode ser definido como um sistema computacional interativo que objetiva propiciar o entendimento de conceitos específicos, como os matemáticos ou científicos (Gomes & Padovani, 2005).

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. Gladcheff, Zuffi e Silva (2001) 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. De acordo com Miranda e Blaudares (2007), os educadores precisam perceber que o processo de aquisição de conhecimento pode ser facilitado quando os estudantes interagem com a tecnologia no processo educativo.

Diversas iniciativas vêm sendo desenvolvidas com o intuito de auxiliar o processo de ensino-aprendizagem dos diferentes conteúdos da matemática por meio do uso de softwares educacionais. Dentre elas, pode-se citar o trabalho de Santos e Silva Junior (2016), que cria versões virtuais de jogos utilizados para o ensino da matemática, tais como o desafio com palitos de fósforo.

Outra perspectiva do uso de softwares educacionais para a aprendizagem de matemática pode ser vista no trabalho de Marczal *et al.* (2016), cujo objetivo consiste na construção de um método e uma ferramenta web que auxilia professores e estudantes a revisitarem os erros, possibilitando que o “erro cometido pelo aprendiz fosse utilizado como uma oportunidade de aprendizagem, isso por meio de mecanismos que permitissem remediação e retroação” (Marczal, *et al.*, 2016 p. 80).

Relacionado ao conteúdo de álgebra, um bom exemplo da utilização de softwares educacionais na sala de aula é demonstrado na pesquisa de Cataneo (2011), que relata o uso do software educacional *Geogebra* para auxiliar o professor na transposição didática dos conteúdos na disciplina de matemática no sétimo ano do ensino fundamental. A aplicação deste software como uma ferramenta auxiliar representa uma metodologia importante para o ensino-aprendizagem de matemática, pois os alunos mostraram uma melhor compreensão e interpretação diante do conceito matemático estudado.

A utilização de softwares educacionais representa uma nova metodologia para os professores, uma nova forma de trabalhar os conteúdos na sala de aula, onde os alunos podem interagir mais ativamente nas aulas (Uehara, Monteiro, Silva Mill e Fiscarelli, 2017). A figura do educador deixa de ser a pessoa que detém todo o conhecimento e se torna a figura do mediador que auxilia os estudantes em suas descobertas, diminuindo assim a distância entre docentes e discentes em sala de aula (Slomski, Araujo, Camargo e Weffort, 2016).

Existe uma vasta gama de softwares educacionais disponíveis na Internet, por esse motivo foi realizado este 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 software *IFMath*, foram selecionados softwares *online* de uso gratuito para análise, são eles: *WolframAlpha*, *Symbolab*, *GeoGebra* e *Path2Math*.

O *WolframAlpha* é uma enciclopédia virtual que aborda as mais diferentes disciplinas, entre elas a matemática (*WolframAlpha*, 2018). Entre os módulos oferecidos para resolver problemas matemáticos, existe o módulo algébrico. Um 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 não apresenta a resolução passo a passo da expressão em sua versão gratuita.

Por sua complexidade, o *WolframAlpha* é uma ferramenta mais adequada à utilização no ensino superior, onde o usuário já apresenta certa familiaridade com o formalismo matemático. Isso é averiguado no estudo de Rodrigues (2011), que aponta que o *WolframAlpha* pode desempenhar um papel importante na aprendizagem de certos conceitos de Cálculo Diferencial e Integral, melhorando sua compreensão. Diante disso, pode existir uma dificuldade de entendimento da linguagem matemática por parte dos alunos do ensino fundamental e médio devido às suas características, não chamando a atenção de professores.

O *Symbolab* é uma calculadora *online* que oferece suporte para diversos conteúdos da matemática, entre eles a álgebra (*Symbolab*, 2018). Assim como o *WolframAlpha*, destaca-se, por sua eficiência em reconhecer expressões algébricas. Entretanto, diferencia-se na forma de inserção dos dados, oferecendo um maior número de possibilidades de entradas. Ele também exibe o resultado final das expressões algébricas, tendo a possibilidade de visualizar todos os passos realizados para resolvê-las juntamente com o seu embasamento teórico. Além disto, também disponibiliza uma representação gráfica da expressão.

Assim como o *WolframAlpha*, devido à complexidade, o *Symbolab* é mais adequado para a utilização no ensino superior pois assemelha-se a uma calculadora científica e é necessário um conhecimento intermediário sobre símbolos e operações matemáticas para utilizá-lo de maneira efetiva. Como aponta Viegas (2017) o uso do software proporciona um melhor entendimento e aproveitamento dos alunos sobre o conteúdo estudado após a utilização no estudo de integrais duplas. Contudo, todos esses recursos, como integrais, derivadas e logaritmos, podem causar confusão nos alunos do ensino fundamental.

O *GeoGebra* é um software matemático que permite realizar construções geométricas, sendo uma ferramenta dinâmica e interativa de ensino (*GeoGebra*, 2018). 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, mais sim em sua representação gráfica.

Em sua utilização em sala de aula, Araujo (2017) afirma que ele pode facilitar a aprendizagem dos conteúdos matemáticos ministrados desde a Educação Básica até o Ensino Superior. De acordo com Oliveira *et al.* (2012), por ser um software livre, o *GeoGebra* vai ao encontro de novas estratégias de ensino e aprendizagem de conteúdos que a matemática abrange, permitindo aos professores e alunos a possibilidade de explorar, conjecturar, investigar tais conteúdos na construção do conhecimento matemático.

Outro trabalho relevante relacionado ao uso de softwares educacionais para o ensino de álgebra é o *Path2Math*, que segundo Jaques *et al.* (2013), é um Sistema Tutor Inteligente que visa assistir aos alunos no processo de aprendizagem de resolução de problemas algébricos, mais especificamente equações de primeiro grau. Ele é um sistema que permite que os alunos resolvam as equações passo a passo, fornecendo assistência ao aluno. Diversos trabalhos foram desenvolvidos com o objetivo de aprimorá-lo, melhorando os resultados gerados, tais como os apresentados por Jaques e Seffrin (2016) e Silva e Jaques (2017), que tratam da modelagem de

conhecimento do usuário com Redes Bayesianas e aplicam a Teoria da Carga Cognitiva para auxiliar os estudantes na resolução de problemas algébricos.

3.3 *IfMath*

O *IfMath* é uma aplicação *web* gratuita e em português que realiza a resolução de exercícios matemáticos (Durigon, 2018). 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 que preconizam Rocha e Campos (2008), 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.

Durigon (2018) ressaltam 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 nestes parâmetros. Isso possibilita ao usuário testar diferentes entradas, permitindo uma ampla interpretação sobre os conteúdos estudados.

O desenvolvimento do *IfMath* foi baseado na arquitetura padrão de projetos *web*, sendo dividido em lado cliente e servidor. O lado servidor disponibiliza serviços para o cliente e é responsável pelo processamento mais custoso, como por exemplo, os algoritmos de resolução de exercícios. Diante deste cenário, a linguagem de programação *Java* foi adotada, visto que possui um bom desempenho nos processamentos e manipulação de cálculos de pequena e média complexidade (Oracle, 2018). Para a padronização dos resultados gerados pelos algoritmos, foi utilizado *LaTeX* para facilitar ao leitor a visualização dos dados, principalmente das fórmulas matemáticas, ajudando-o a percorrer os resultados e encontrar informação com mais agilidade (LaTeX, 2018). O lado cliente realiza a interação entre usuário e aplicação, utilizando tecnologias padrões para criação de páginas web como *HTML5*, *CSS* e *JavaScript* (W3Schools, 2018).

Conforme aponta Durigon (2018), o *IfMath* contempla os conteúdos de números e operações, geometria, álgebra e funções, representando uma boa parcela dos conteúdos ministrados na disciplina de matemática da Educação Básica. Entretanto, esta aplicação não possuía um módulo para manipulações algébricas, um importante e complexo conteúdo da matemática, 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 vem contribuindo com a solução de problemas nas mais diversas áreas, inclusive no ensino de matemática. Um SE incorpora o conhecimento de um especialista humano de uma determinada área para resolver problemas específicos, permitindo, dessa forma, a resolução dos exercícios dentro da metodologia do *IfMath*. Entretanto, antes de resolvê-la, 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.

4 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 (Aho *et al.*, 2008). Inicialmente, os compiladores eram utilizados apenas 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.

Cooper e Torczon (2014) indicam que a estrutura de um compilador é 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* visa 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* (Singh *et al.*, 2008). A Figura 1 apresenta a estrutura de um compilador.

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 através de uma representação intermediária do tipo árvore. Utilizando a árvore de sintaxe, 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.

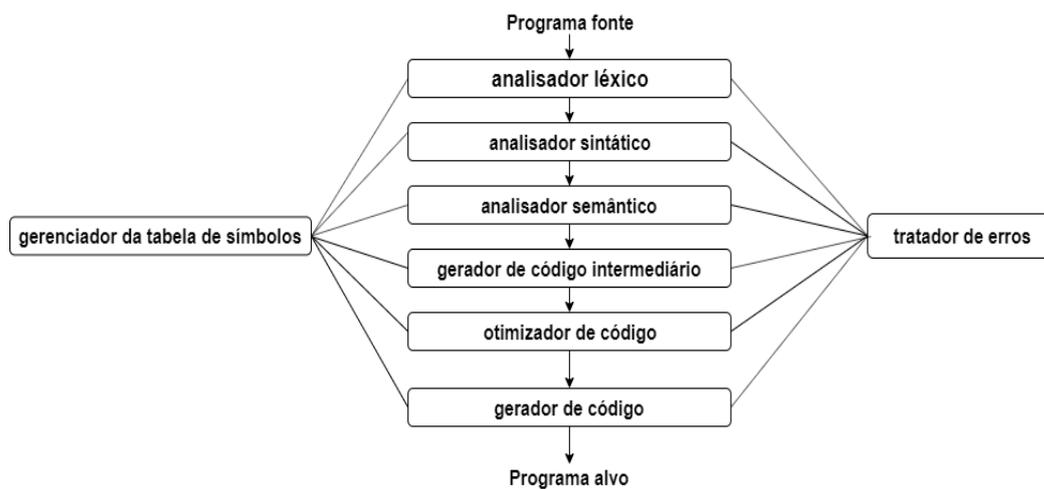


Figura 1: Etapas de um compilador.
FONTE: Aho *et al.* (1995).

Utilizando a representação intermediária do programa gerada pelas quatro primeiras etapas (*front-end*), o *back-end* realiza o mapeamento dessa representação em um conjunto de instruções da máquina alvo (Cooper e Torczon, 2014). 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. Somente a análise léxica, sintática e geração de código intermediário são de interesse deste trabalho, sendo descritas nas próximas sessões.

4.1 Análise Léxica

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 Aho *et al.* (2008), 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 para ser utilizado na análise semântica e geração de código. A Figura 2 exemplifica as transformações realizadas pela análise léxica.

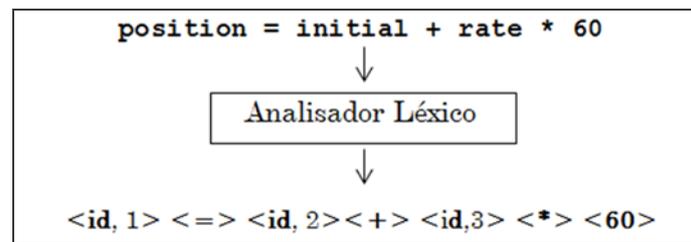


Figura 2 - Representação conceitual da análise léxica.
FONTE: Aho *et al.* (2008)

Singh *et al.* (2008) descrevem que um dos meios para a criação de um analisador léxico é a definição da lista dos *tokens* válidos através 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.

Cooper e Torczon (2014) afirmam que 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 ou não. Cada nó representa uma condição possível de ocorrer durante a análise de um lexema.

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 lexema que pertença a linguagem, este é adicionado à lista de *tokens*, 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.

4.2 Análise Sintática

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 (Aho *et al.*, 2008). Esse processo é apresentado na Figura 3.

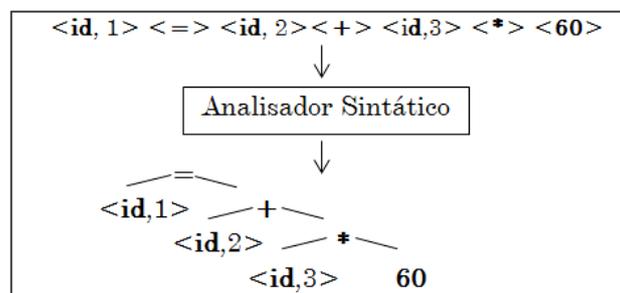


Figura 3 - Representação conceitual da análise sintática.
FONTE: Aho *et al.* (2008)

Para Singh *et al.* (2008), uma gramática livre de contexto (GFC) é um conjunto de símbolos terminais e não-terminais, um símbolo inicial, e produções que descrevem regras sobre 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 as 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 subclasses, dentre as quais destacam-se as LL e LR. Segundo José Neto (2016), 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. Uma gramática LL caracteriza-se por ser fatorada à esquerda, não ser ambígua e não possuir recursividade à esquerda. Já 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 derivações mais à direita. Este tipo de gramática não precisa ser fatorada e permite recursões à esquerda, mas também não pode ser ambígua (José Neto, 2016).

Segundo Aho *et al.* (2008), 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.

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. Aho *et al.* (2008) descreve que esta análise adota uma abordagem derivativa para construção da árvore. 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, até se chegar aos símbolos terminais existentes na lista de *tokens*.

Conforme aponta Singh *et al.* (2008), existem diferentes formas de implementar um analisador descendente, como o analisador sintático de descida recursiva e o analisador preditivo sem recursão. Neste trabalho, optou-se por utilizar o método de análise preditiva por ser o método mais eficiente de análise descendente. A analisador preditivo utiliza uma pilha para simular a árvore de derivação, produzindo uma derivação mais à esquerda, e uma tabela de análise, chamada de tabela M, para determinar a produção a ser utilizada para cada não terminal de acordo com o próximo símbolo a ser analisado, conforme ilustra a Figura 4 (Aho *et al.*, 2008). Por analisar o símbolo subsequente para decidir qual produção utilizar, é empregada uma classe de gramática LL chamada de LL(1).

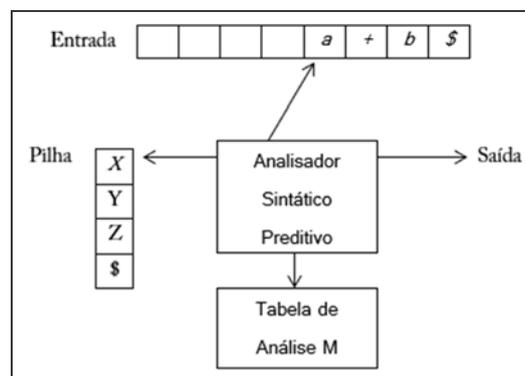


Figura 4 - Modelo de um analisador sintático preditivo dirigido por tabela.
FONTE: Aho et al. (2008)

4.3 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 Aho *et al.* (2008) apontam, atributos são agregados aos símbolos da GFC e seus valores são especificados pelas operações associadas às produções. Por ser executada

simultaneamente a análise sintática, essa técnica é útil para a verificação de tipos na análise semântica e para a geração de código intermediário.

De acordo com Aho *et al.* (2008), 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, conforme mostra a Figura 5. Um SDT é uma notação complementar para a SDD que incorporam fragmentos de código chamados de ações semânticas no corpo da produção.

$$\begin{array}{ll} \text{PRODUÇÃO} & \text{REGRA SEMÂNTICA} \\ E \rightarrow E_1 + T & E.code = E_1.code || T.code || '+' \end{array}$$

Figura 5 - Definição dirigida por sintaxe.
FONTE: Aho et al. (2008)

Singh *et al.* (2008) relatam a importância de trabalhar com uma árvore de derivação para avaliar o valor de todos os atributos. Ao gerar um grafo acíclico dirigido baseado nessa árvore, é possível determinar a ordem de avaliação para as instâncias dos atributos e identificar ciclos de dependência. Existem as definições S-Atribuídas e L-Atribuídas, que são classes de SDD que garantem uma ordem de avaliação sem ciclo de dependência (Aho *et al.*, 2008). A definição S-Atribuída é composta apenas de atributos sintetizados e é associada a análise ascendente e gramáticas LR, correspondendo assim, a um caminhamento pós-ordem na árvore. Por sua vez, a definição L-Atribuída é formada por atributos sintetizados do próprio símbolo, herdados 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 caminhamento pré-ordem.

Conforme definido anteriormente, um SDT é uma notação para especificar uma tradução que conecta fragmentos de código às produções de uma gramática. Aho *et al.* (2008) destacam que um SDT é como uma SDD, exceto pelo fato de que a ordem de avaliação das ações semânticas é especificada explicitamente. 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 (Aho *et al.*, 2008). 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 são executadas na ordem correta.

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 Aho *et al.* (2008), as ações semânticas, neste momento chamadas de registros-de-ação, buscam os atributos necessários a partir de posições conhecidas 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.

4.4 Geração de Código Intermediário

A geração de código intermediário é responsável por criar uma Representação Intermediária (IR), que corresponde a um código escrito em uma linguagem de alto nível, que seja entendível pelo *back-end*. Conforme Cooper e Torczon (2014), 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 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 6. Segundo Cooper e Torczon (2014), 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 6 - Conjunto de instruções de três endereços.
FONTE: Aho *et al.* (2008)

Ao realizar a tradução, as variáveis temporárias t_1 e t_2 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 Aho *et al.* (2008), desmembrar uma expressão aritmética com múltiplos operadores e estruturas de fluxo de controle aninhados torna o código de três endereços vantajoso para a otimização e geração de código objeto.

O código de três endereços é organizado a partir de dois principais conceitos: endereços e instruções. Os endereços podem ser especificados em três tipos básicos, sendo eles: nomes, constantes e temporários. Os nomes são os identificadores do programa fonte; as constantes podem ser numéricas e alfanuméricas; por fim, os temporários são variáveis criadas pelo compilador.

As instruções são classificadas em: acesso indexado e indireto, invocação de rotina, desvio e atribuição. As instruções de desvio realizam o pulo de uma linha de código x para uma linha de código y . As instruções de invocação de rotina são utilizadas para chamar funções e procedimentos. Por sua vez, as instruções de acesso indexado e indireto trabalham com o modo de endereçamento e acesso de valores em vetores, por exemplo. Por fim, as instruções de atribuição são utilizadas quando o resultado da operação é armazenado em uma variável especificada à esquerda do operador de atribuição.

De acordo com Cooper e Torczon (2014), 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. O Quadro 1 demonstra como a expressão $x + y * z$ deve ser representada em uma quádrupla.

Quadro 1 – Representação da quádrupla.

Operador	Arg.1	Arg.2	Resultado
*	y	z	T_1
+	x	T_1	T_2

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 inversa, a ser realizada sobre um termo quando este muda de posição em relação ao sinal de comparação e por ser indispensável resolver operações prioritárias antes das demais.

Após o processo de análise e geração do código intermediário da expressão algébrica, que é necessário para verificar se ela foi digitada corretamente e para transformá-la em um formato processável pelo computador, o sistema especialista utiliza esta representação para realizar as manipulações algébricas necessárias para a resolução do problema proposto.

5 Sistemas Especialistas

A inteligência artificial é definida por Luger e Subblefield (2008) como o ramo da computação preocupada com a automação de comportamento inteligente. Dentre os conceitos abordados pela inteligência artificial, os sistemas especialistas (SE) se destacam por reproduzir o conhecimento de um especialista humano em uma área limitada. Os SE são algoritmos que procuram atingir soluções de 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.

Os SE são normalmente projetados para resolver problemas relacionados às atividades de controle, diagnóstico, prescrição, monitoramento, interpretação, previsão, simulação, seleção, projeto, planejamento ou instrução (Durkin, 1994). 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, auxiliando assim no processo de ensino-aprendizagem da disciplina de matemática, este SE executa atividades de instrução.

Para que um sistema seja considerado especialista, Durkin (1994) afirma que este deve possuir ao menos os seguintes módulos: uma base de conhecimento, que armazena por meio de algum formalismo o conhecimento de um perito da área para ser utilizado durante as consultas ao sistema; uma memória de trabalho, que contém todos os fatos descobertos durante uma consulta específica; 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.

Tendo em vista a necessidade de explicar ao usuário os passos para resolver uma expressão algébrica, o mecanismo de explanação é uma característica adicional de extrema importância do SE projetado neste trabalho. Conforme aponta Durkin (1994), 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. Dentre as diferentes formas existentes para representação do conhecimento, destacam-se: lógica ou cálculo proposicional, lógica ou cálculo de predicados, ontologias, frames, redes semânticas e regras de produção. Para Durkin (1994), cada uma dessas técnicas enfatiza certas informações sobre o problema, enquanto ignora outras. Diante disso, cada uma trabalha com diferentes tipos de conhecimento, tornando a 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 (Durkin, 1994). 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 logicamente um antecedente ou premissa a uma consequência ou conclusão. Como exemplo, se o antecedente/premissa é verdadeiro, então a regra infere que a consequência/conclusão é verdadeira também.

As regras de produção são a base para o desenvolvimento de SE baseados em regras, que são definidos por Durkin (1994) como um programa de computador que processa informações de um problema específico baseado em regras que contém o conhecimento do domínio, usando mecanismos de inferência para inferir novas informações. A inferência em bases de conhecimento com regras de produção pode ser realizada por encadeamento progressivo e regressivo. No encadeamento regressivo define-se um objetivo e uma hipótese para ele. As regras que comprovam a hipótese tornam-se novos objetivos e perguntas são realizadas ao usuário quando são alcançadas premissas que não podem ser comprovadas por nenhuma regra. Esse processo é repetido até que a solução seja encontrada ou nenhuma hipótese possa ser avaliada. Por sua vez, o encaminhamento progressivo inicia o processo com fatos conhecidos, derivando novos fatos a partir do que é conhecido. O processo é repetido até que a solução seja encontrada ou nenhuma regra possa ser disparada.

Como pode ser visto na próxima seção, este trabalho utiliza regras de produção para representar o conhecimento do especialista na base de conhecimento e o algoritmo de encadeamento progressivo para realizar a inferência sobre elas.

6 Desenvolvimento do Sistema

O desenvolvimento de um módulo compilador na linguagem *Java* foi a primeira atividade de implementação realizada. Para isso, foram definidas as expressões regulares, mostradas na Figura 7, que determinam os padrões de entrada válidos. Estas expressões regulares foram utilizadas na implementação de um 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.

Token → Símbolo	Token → Símbolo	Token → Símbolo
L → { A-Z, a-z, _ }	= → =	{ → {
D → { 0-9 }	<> → <>	} → }
numero_natural → D(D)*	<= → <=	sen → sen
numero_decimal → D(D)*,D(D)*	< → <	cos → cos
id → L(D L)*	>= → >=	tg → tg
id_com_coeficiente → numero_natural id	> → >	raiz → raiz
+ → +	(→ (raizq → raizq
- → -) →)	log → log
* → *	[→ [log10 → log10
/ → /] →]	; → ;
^ → ^		

Figura 7 - Expressões regulares para expressões algébricas.

Em seguida, definiu-se uma gramática livre de contexto LL para especificar como formar expressões válidas. A Figura 8 exibe a gramática elaborada neste trabalho.

$$\begin{aligned}
E &\rightarrow TCT \\
C &\rightarrow = | \langle \rangle | \geq | \leq | < | > \\
T &\rightarrow MT' \\
T' &\rightarrow +MT' | -MT' | \epsilon \\
M &\rightarrow PM' \\
M' &\rightarrow *PM' | /PM' | \epsilon \\
P &\rightarrow OP' \\
P' &\rightarrow ^OP' | \epsilon \\
O &\rightarrow SF \\
S &\rightarrow + | - | \epsilon \\
F &\rightarrow FUNC | (T) | [T] | \{T\} | id | id_com_coeficiente | \\
&\quad \text{numero_natural} | \text{numero_decimal} \\
FUNC &\rightarrow \text{sen}(T) | \text{cos}(T) | \text{tg}(T) | \text{raizq}(T) | \text{raiz}(T; \text{numero_natural}) | \\
&\quad \text{log10}(T) | \text{log}(T; \text{numero_natural})
\end{aligned}$$

Figura 8 - Gramática livre de contexto LL para expressões algébricas.

Tal gramática foi utilizada na implementação de um analisador sintático descendente preditivo, que tem como principal objetivo averiguar a validade da sequência de *tokens* perante uma linguagem gerada pela GLC, 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. É importante destacar que a maneira como as produções e os operadores foram organizados na gramática mantém a precedência das operações ao gerar o código intermediário, pois operadores com maior precedência ficam em níveis mais profundos da árvore e são resolvidos antes dos demais.

Para finalizar o desenvolvimento do compilador, foi implementado um gerador de código intermediário. Para isso, foi utilizado um SDT L-Atribuído, que consiste na adição de ações semânticas nas produções da gramática. A Figura 9 exibe a gramática com códigos que identificam as ações semânticas (AS) inseridas no corpo das produções.

$ \begin{aligned} E &\rightarrow AS1 T_1 C AS2 T_2 AS3 \\ C &\rightarrow = AS4 \\ &\quad \langle \rangle AS4 \\ &\quad \geq AS4 \\ &\quad \leq AS4 \\ &\quad < AS4 \\ &\quad > AS4 \\ T &\rightarrow AS5 MT' AS6 \\ T' &\rightarrow + AS7 MT'_1 AS8 \\ &\quad -AS7 MT'_1 AS9 \\ &\quad \epsilon AS10 \\ M &\rightarrow AS11 PM' AS12 \\ M' &\rightarrow *AS13 PM'_1 AS14 \\ &\quad /AS13 PM'_1 AS15 \\ &\quad \epsilon AS16 \\ P &\rightarrow AS17 OP' AS18 \\ P' &\rightarrow ^AS19 OP'_1 AS20 \\ &\quad \epsilon AS21 \\ O &\rightarrow S AS22 F AS23 \end{aligned} $	$ \begin{aligned} S &\rightarrow + AS24 \\ &\quad - AS25 \\ &\quad \epsilon AS24 \\ F &\rightarrow AS26 FUNC AS27 \\ &\quad (AS28 T) AS29 \\ &\quad [AS28 T] AS29 \\ &\quad \{AS28 T\} AS29 \\ &\quad id AS30 \\ &\quad id_com_coeficiente AS30 \\ &\quad numero_natural AS30 \\ &\quad numero_decimal AS30 \\ FUNC &\rightarrow \text{sen} AS31 (T) AS32 \\ &\quad \text{cos} AS31 (T) AS32 \\ &\quad \text{tg} AS31 (T) AS32 \\ &\quad \text{raizq} AS31 (T AS33) AS34 \\ &\quad \text{raiz} AS31 (T ; \text{numero_natural} \\ &\quad AS35) AS34 \\ &\quad \text{log10} AS31 (T AS36) AS34 \\ &\quad \text{log} AS31 (T ; \text{numero_natural} \\ &\quad AS35) AS34 \end{aligned} $
---	---

Figura 9 - Gramática livre de contexto com ações semânticas.

A implementação de algumas ações semânticas é apresentada no Quadro 2. A AS6, por exemplo, cria uma nova operação do código de três endereços por meio de uma quádrupla expandida.

Quadro 2 - Exemplos de ações semânticas da gramática livre de contexto.

Código	Código-fonte
AS1	T_1 .position = 0; T_2 .level = 0;
AS3	E .param1 = T_1 .addr; E .comp = C .addr; E .param2 = T_2 .addr;
AS5	M .position = T .position; T' .position = T .position; M .level = T .level; T' .level = T .level;
AS6	if (T' .value) { T .addr = new Temp(); AddNewOperation(T' .op, M .addr, T' .addr, T .addr, T .position, T .level); } else { T .addr = M .addr; }

Os campos de nível e posição foram adicionados à quádrupla tradicional e são importantes para a resolução do problema pelo sistema especialista. A posição indica se o termo encontra-se à direita ou à esquerda do sinal de comparação, sendo relevante para indicar se a operação efetuada sobre o mesmo deve ser alterada para uma inversa quando mudar de posição. Por sua vez, o nível indica a precedência de operações contidas em parênteses, colchetes ou chaves. A ideia deste campo é baseada na teoria sobre profundidade da estrutura de dados árvore, que é definida, de forma simplificada, como a distância entre a raiz e o nodo em questão. Levando em consideração que a árvore sintática é uma árvore n-ária conceitual, tal conceito é útil para indicar que um determinado grupo de operações contido em um ramo da árvore possui precedência sobre as demais operações.

A AS3, apresentada na mesma figura 9 e descrita no quadro 2, é responsável por registrar o sinal de comparação, assim como, identificar a quádrupla inicial da expressão algébrica no lado esquerdo e direito do sinal de comparação. É importante destacar que 35 ações com propósitos semelhantes foram inseridas no corpo de todas as produções da gramática apresentada na Figura 9.

A última etapa do desenvolvimento correspondeu à modelagem, implementação e validação de um 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.

Uma visão geral do fluxo de funcionamento do sistema é apresentada pela Figura 10. O compilador recebe a expressão algébrica informada pelo usuário para validá-la, transformá-la em um código de três endereços e encaminhá-la para o SE apropriado. Cada conteúdo possui um módulo de SE exclusivo. O SE manipula o código de três endereços de acordo com as regras estipuladas e retorna o resultado das manipulações em notação *LaTeX*, para facilitar ao leitor a visualização dos dados. Para converter o código de três endereços em notação *LaTeX*, foi realizado

um caminhamento em ordem pelas quádruplas expandidas que compõe a expressão, tomando com raiz a operação inicial em cada lado do sinal de comparação

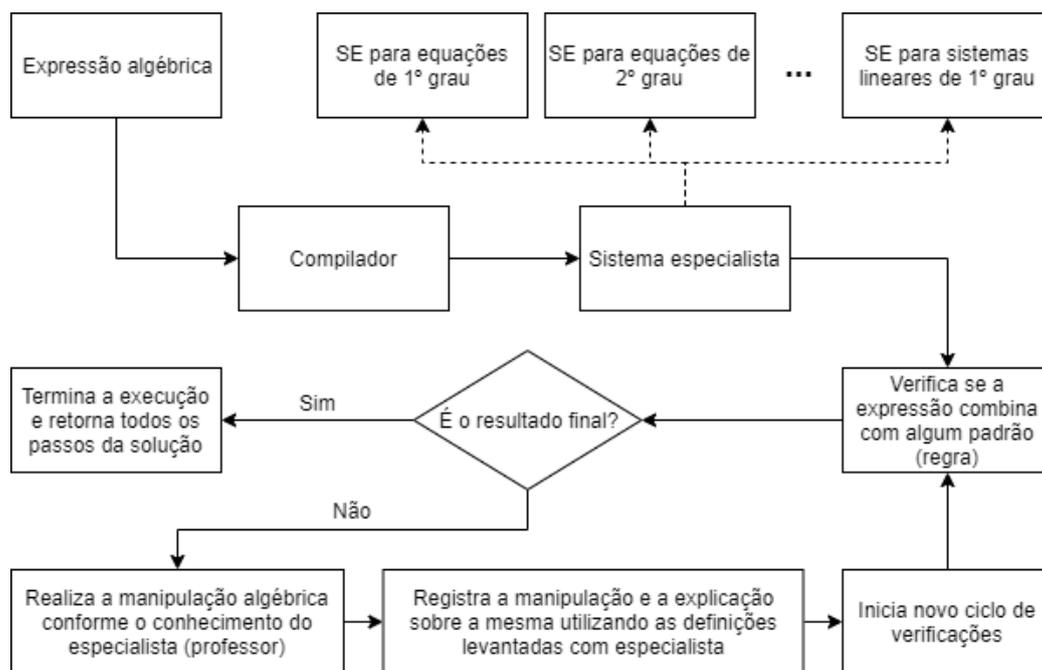


Figura 10 – Fluxo conceitual do sistema.

O SE correspondente analisa o código de três endereços gerado pelo compilador com o objetivo de identificar padrões que se encaixam nas regras definidas junto ao especialista. A inferência na base de conhecimento é realizada usando o encadeamento progressivo adaptado por questão de desempenho, pois, de acordo o conhecimento obtido junto ao especialista, é necessário reavaliar apenas algumas regras após uma manipulação nas expressões algébricas ensinadas no ensino fundamental e médio. 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. No Quadro 3, são exibidas as regras de produção utilizadas pelo SE para resolver equações do primeiro grau.

Quadro 3 - Regras de produção do SE de equações do 1º grau.

Regra	Padrão	Descrição
1 - Mínimo múltiplo comum	$T_1 = T_2$ $T_1 = ax / k$ $T_2 = b - j$	Calcula-se o MMC de todos os denominadores. Este número será o novo denominador dos termos da equação. Após deixar todos os membros sobre um mesmo denominador, este pode ser cancelados e as multiplicações são efetuadas.
2 - Multiplicação cruzada	$T_1 = T_2$ $T_1 = ax / k$ $T_2 = b / j$	Utiliza a propriedade fundamental das proporções.
3 - Distributiva	$T_2 = 0$ $T_1 = ax + b$ $T_2 = k * T_1$	Multiplica cada elemento e soma os resultados.
4 - Multiplicação	$T_1 = 0$ $T_1 = ax + T_2$ $T_2 = k * b$	Resolve multiplicações simples.

5 - Troca de sinais	$T_1 = 0$ $T_1 = ax - T_2$ $T_2 = \text{MINUS } b$	Aplica a regra de sinais em operações prioritárias, em duplas negações ou em somas com números negativos.
6 - Isolar termos semelhantes	$T_1 = 0$ $T_1 = ax + b$	Os termos semelhantes foram isolados. Ao trocar a posição de um termo perante a igualdade, é necessário mudar o seu sinal, aplicando a operação inversa.
7 - Somar termos	$T_1 = T_2$ $T_1 = ax + x$ $T_2 = b + j$	Soma dos termos semelhantes.
8 - Variável negativa	$T_1 = T_2$ $T_1 = \text{MINUS } ax$ $T_2 = b$	Multiplifica todos os termos da equação por -1 e as operações realizadas sobre os termos são invertidas para uma operação inversa.
9 - Variável com coeficiente	$T_1 = T_2$ $T_1 = ax$ $T_2 = b$	O coeficiente da variável foi movido para o outro lado da igualdade. Ele deixará de multiplicar a variável para dividir o termo independente.
10 - Fração simplificada	$x = T_1$ $T_1 = b / a$	Se a fração é uma divisão inteira ou existe um MDC entre o numerador e o denominador, então essa fração é simplificada.
11 - Resultado final	$x = k$ ou $x = T_1$ $T_1 = b / a$	Representa o resultado final simplificado.

Um exemplo do funcionamento do SE e de suas regras de produção sobre a equação do primeiro grau $2 * (x + 4) = 0$ é apresentada na Figura 11. O *IFMath* envia a equação ao compilador, que por sua vez executa a análise léxica, a análise sintática e a geração de código intermediário. Utilizando as expressões regulares apresentadas na Figura 7, os tokens identificados pelo analisador léxico nesta expressão são: 2, *, (, +, 4,), = e 0. Seguindo a gramática livre de contexto (Figura 8), o compilador realiza uma série de derivações nos ramos da árvore para validar sintaticamente a expressão, após a identificação de cada token pelo analisador léxico. Enquanto a estrutura da expressão é verificada pelo analisador sintático, as ações semânticas presentes no corpo das produções derivadas são executadas e o código de três endereços é gerado gradualmente. Ao concluir a análise sintática, como a expressão é válida, o compilador gerou o código de três endereços da equação e encaminhará o mesmo ao SE.

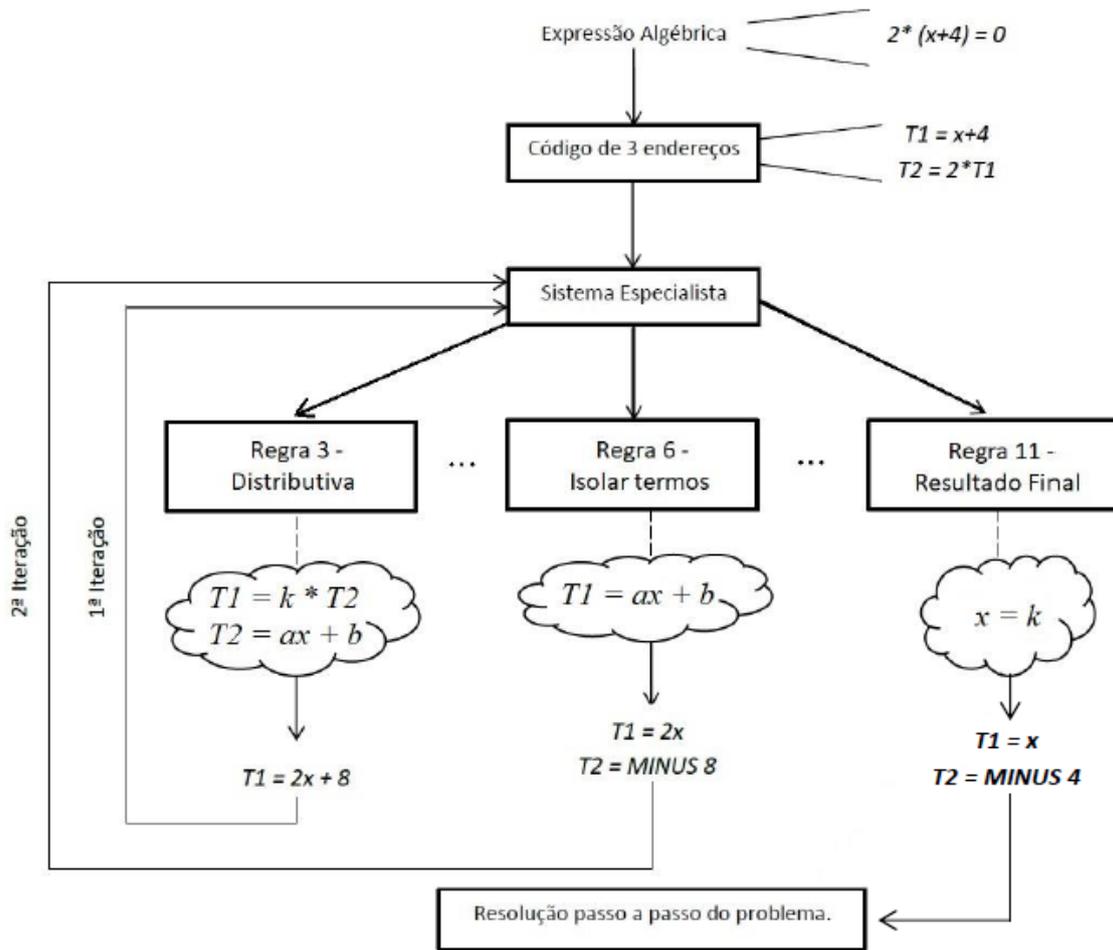


Figura 11 - Representação conceitual do SE e suas regras de produção.

Ao receber o código de três endereços apresentado na figura 11, o SE busca pelos padrões das regras do quadro 3. Primeiramente, ele identifica que o padrão da regra 3 está presente no código três endereços e realiza as transformações necessárias sobre o mesmo para gerar uma equação com apenas operações de adição e subtração, encaminhando-o para as próximas análises. O SE compara o código gerado com os padrões das regras subsequentes até chegar na regra 6, conforme o funcionamento do encadeamento progressivo explicado anteriormente.

Na regra 6, as variáveis e termos independentes são isoladas e somadas, em seguida, na regra 7. Na regra 9, o coeficiente da variável passa a dividir o termo independente, resultando em uma fração. A regra 11 verifica se o código gerado corresponde ao padrão representado no balão abaixo da regra e o assume como resposta final do problema, se possível, de maneira simplificada pela regra 10. Por fim, a sequência de passos realizados para chegar ao resultado final são retornados para o usuário.

A execução do SE para este exemplo é apresentada pelo Quadro 4. Destaca-se que todas as regras são avaliadas pelo mecanismo de inferência. Entretanto, elas só são disparadas e efetuam algum tipo de manipulação sobre o código de três endereços quando encontram o padrão esperado neste código. Devido a isso, as regras presentes no Quadro 4 cuja entrada e saída são iguais foram avaliadas mas não disparadas.

Quadro 4 – Execução das regras de produção do SE de equações do 1º grau.

Regra	Entrada	Saída	Código de três endereços resultante
1 - Mínimo múltiplo comum	$2 * (x + 4) = 0$	$2 * (x + 4) = 0$	T1: T2 = 0 T2: 2 * T3 T3: x + 4
2 - Multiplicação cruzada	$2 * (x + 4) = 0$	$2 * (x + 4) = 0$	T1: T2 = 0 T2: 2 * T3 T3: x + 4
3 - Distributiva	$2 * (x + 4) = 0$	$2x + 8 = 0$	T1: T2 = 0 T2: 2x + 8
4 - Multiplicação	$2x + 8 = 0$	$2x + 8 = 0$	T1: T2 = 0 T2: 2x + 8
5 - Troca de sinais	$2x + 8 = 0$	$2x + 8 = 0$	T1: T2 = 0 T2: 2x + 8
6 – Isolar termos semelhantes	$2x + 8 = 0$	$2x = -8$	T1: 2x = T2 T2: MINUS 8
7 - Somar termos	$2x = -8$	$2x = -8$	T1: 2x = T2 T2: MINUS 8
8 - Variável negativa	$2x = -8$	$2x = -8$	T1: 2x = T2 T2: MINUS 8
9 - Variável com coeficiente	$2x = -8$	$x = -\frac{8}{2}$	T1: x = T3 T2: MINUS 8 T3: T2 / 2
10 - Fração simplificada	$x = -\frac{8}{2}$	$x = -4$	T1: x = T2 T2: MINUS 4
11 – Resultado final	$x = -4$	$x = -4$	T1: x = T2 T2: MINUS 4

Com o compilador e o SE concluídos, estes foram integrados ao *IFMath*. Esta integração foi realizada em duas etapas. A primeira etapa consistiu no desenvolvimento das interfaces gráficas do conteúdo de equações de primeiro grau, levando em consideração as necessidades do usuário, assim como o padrão de comportamento e de *design* de interface do *IFMath*. É importante ressaltar a utilização do *framework Angular* para o desenvolvimento desta etapa, que auxiliou no desenvolvimento das interfaces gráficas e na implementação de comportamentos da aplicação. Por sua vez, a segunda etapa integrou a solução criada nesse projeto com o *back-end* do *IFMath*.

Como a linguagem de programação *Java* foi utilizada no *back-end* do *IFMath* e na solução proposta por este trabalho, apresentou-se a possibilidade de transformar o compilador e o SE implementados em uma biblioteca da linguagem *Java*, permitindo a importação e utilização desta no projeto do *back-end* do *IFMath* como uma dependência externa. Para realizar a comunicação entre a biblioteca e o *back-end* foi necessária a criação de uma estrutura padrão para envelopar as informações enviadas e retornadas pela biblioteca de uma maneira que possibilitou a utilização da mesma para outros conteúdos, permitindo assim, uma maior flexibilidade da solução proposta. Por fim, foi utilizado o *framework Spring Boot* no *back-end* do *IFMath* para o desenvolvimento de um *web service restful*, responsável por gerenciar a comunicação entre *back-end* e *front-end*.

7 Resultados e Discussões

A ferramenta proposta por este estudo tem como objetivo aproximar teoria e prática, visando o desenvolvimento cognitivo do aluno, e possibilitar aos professores uma nova forma de apresentar os conteúdos. Para isso, o sistema 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 12, 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 possível, levando em consideração o aspecto pedagógico. Além da teoria, são apresentados exemplos de resoluções de equações de 1º grau pré-definidas para permitir um vínculo entre a teoria e prática, possibilitando assim um melhor entendimento do conteúdo pelos estudantes.

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 identificadas junto ao professor de matemática. A Figura 13 mostra a interface com uma expressão digitada pelo usuário e resolução elaborada 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 é 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.

The screenshot shows the IFMath interface for the 'Equação do 1º Grau' topic. At the top, there is a navigation bar with 'IFMath', a logo, and 'Equação do 1º Grau' in the center, and 'Sobre' on the right. A left sidebar contains a 'Menu' section with 'Definição' and 'Resolução Dinâmica' options. The main content area is titled 'Equação' and contains the following text:

Equação
É toda sentença matemática que exprime uma relação de igualdade entre duas expressões algébricas e uma ou mais incógnitas (**variáveis ou valores desconhecidos**) que são designadas por letras. Dessa forma, toda equação possui: Sinal de igualdade; Primeiro membro (**antes do sinal de igualdade**) e segundo membro (**depois do sinal de igualdade**).

Equação do 1º Grau
É toda equação com uma incógnita (**geralmente "x"**) que pode ser escrita na forma $ax + b = 0$, onde **a** e **b** são números reais, e **a** $\neq 0$.

Como resolver uma equação do 1º grau?
O propósito de solucionar uma equação do 1º grau é descobrir o valor da incógnita, ou seja, encontrar o valor da incógnita que torna a igualdade verdadeira. Este valor encontrado chama-se **raiz**. Para esse fim, deve-se isolar os elementos desconhecidos em um dos lados do sinal de igual e os valores constantes do outro lado. Todavia, é importante observar que a mudança de posição desses elementos deve ser feita de forma que a igualdade continue sendo verdadeira. Quando um termo da equação mudar de lado do sinal de igual, devemos inverter a operação. Desta forma, se tiver multiplicando, passará dividindo, se tiver somando, passará subtraindo e assim por diante.

Below the text, there are two examples:

Exemplo 1
 $3x - 4 = x + 6$ Solução da Equação

Exemplo 2
 $3(x - 4) = 7x$ Solução da Equação

Figura 12 – Interface de conceitos sobre tema.

The screenshot shows the IFMath software interface for solving a first-degree equation. The title bar reads "Equação do 1º Grau". The main area displays the following steps:

Equação: $2 \cdot (3 + x) - x = 5 - \frac{x}{2}$

$2 * (3 + x) - x = 5 - \frac{x}{2}$

$\frac{2 * 2 * (3 + x)}{2} - \frac{2 * x}{2} = \frac{2 * 5}{2} - \frac{1 * x}{2}$

$2 * 2 * (3 + x) - 2x = 10 - 1x$

$4 * (3 + x) - 2x = 10 - 1x$

$12 + 4x - 2x = 10 - 1x$

$4x - 2x + 1x = -12 + 10$

$3x = -2$

$x = \frac{-2}{3}$

$x = -\frac{2}{3}$

Figura 13 - Interface para resolução de equações de 1º grau.

Como pode ser visto na Figura 14, 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 de um SE. O SE identifica a necessidade de usar uma determinada manipulação e insere as explicações referentes necessárias sobre a mesma. Por padrão, esses detalhes não são apresentados em um primeiro momento (Figura 13), 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.

↓

$$2 * (3 + x) - x = 5 - \frac{x}{2}$$

Equação inicial.

↓

$$\frac{2 * 2 * (3 + x)}{2} - \frac{2 * x}{2} = \frac{2 * 5}{2} - \frac{1 * x}{2}$$

Calcula-se o MMC de todos os denominadores (lembre-se que quando um termo está em uma fração, seu denominador é 1). Este número será o novo denominador de todos os termos da equação. Lembre-se que quando trocamos o denominador de uma fração, o numerador precisa sofrer uma correção. Este processo é conhecido como "dividir pelo de baixo, multiplicar pelo de cima".

↓

$$2 * 2 * (3 + x) - 2x = 10 - 1x$$

Após deixar todos membros sobre um mesmo denominador, os denominadores podem ser cancelados e as multiplicações são efetuadas

↓

$$4 * (3 + x) - 2x = 10 - 1x$$

Resolução da multiplicação.

↓

$$12 + 4x - 2x = 10 - 1x$$

Utilizando a propriedade distributiva, o elemento externo multiplicou cada elemento da operação interna

↓

$$4x - 2x + 1x = -12 + 10$$

Os termos semelhantes foram isolados. As incógnitas foram movidas para um lado da igualdade e as constantes foram movidas para o outro lado. Ao trocar a posição de um termo perante a igualdade, aplica-se a operação inversa.

⋮

Figura 14 - Interface para resolução de equações de 1º grau com a apresentação de explicações.

Após a conclusão da implementação do sistema e realização dos testes preliminares, realizou-se uma avaliação do mesmo junto a professores de matemática. Foi optado por realizar a primeira avaliação apenas com os professores visto que o objetivo foi validar se o software estava funcionando de acordo com a metodologia adotada em sala de aula. Esta avaliação foi realizada por onze professores de matemática da rede pública de educação do município de Lages, SC. Estes profissionais ministram aulas em turmas do sexto ao nono ano e estavam na faixa etária entre 25 e 50 anos. Este evento ocorreu em um dos laboratórios de informática do Instituto Federal de Santa Catarina (IFSC) campus Lages no dia 24 de outubro de 2018.

A dinâmica da avaliação iniciou-se com a apresentação do problema, objetivo e metodologia adotada na elaboração deste trabalho. Em seguida, os professores resolveram uma série de equações do 1º grau previamente selecionadas, da mesma forma que cada um normalmente resolve em sala de aula. Esta resolução foi manual e escrita no quadro. A cada equação resolvida, a solução proposta pelo software era apresentada e uma discussão fazendo o comparativo entre ambas as resoluções era iniciada. Tal técnica foi adotada para não induzir os professores a seguirem a metodologia proposta pelo software, gerando assim diferenças entre as soluções para serem debatidas. Após resolver todas as equações, os professores puderam testar o software com as equações que cada um julgava interessante, gerando assim novas discussões

sobre as diferenças nas metodologias adotadas. Por fim, um questionário de avaliação, de forma eletrônica e sem identificação, foi aplicado para registrar as impressões destes profissionais sobre o software.

Este questionário era composto de três categorias de perguntas, sendo elas: aspectos de usabilidade e adequação da linguagem e notação matemática; nível de aprofundamento e interpretação do conteúdo; e avaliação da utilidade do software como ferramenta de apoio ao processo de ensino-aprendizagem. Na Figura 15, são apresentados os gráficos com as respostas dos professores referentes a usabilidade e adequação da linguagem e notação matemática. O gráfico 1 indica que os professores consideraram o padrão de interfaces adotado neste módulo próximo do ideal para ser utilizado pelos estudantes como uma ferramenta de estudos. Os gráficos 2 e 3 demonstram que a linguagem, vocabulário e notação matemática adotada atendem às expectativas.

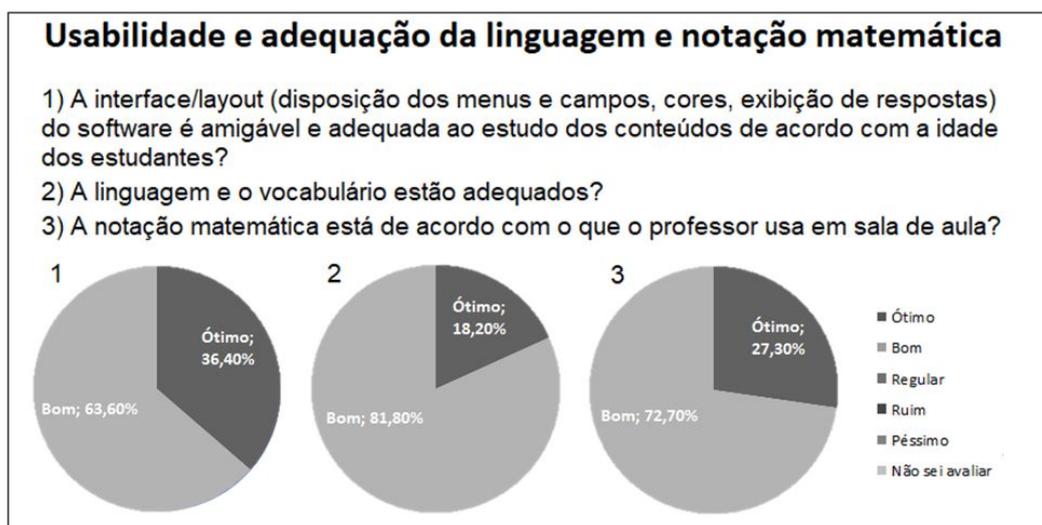


Figura 15 - Gráficos referentes à usabilidade e adequação da linguagem e notação matemática.

A Figura 16 apresenta as respostas dos professores com relação a interpretação, entendimento e nível de aprofundamento dos conteúdos. O gráfico 1 aponta que os professores consideram, de forma unânime, que o nível de aprofundamento do conteúdo está adequado para estudantes do ensino fundamental. Já o gráfico 2 sugere que as informações apresentadas e a facilidade para leitura e interpretação estão satisfatórias. Por fim, o gráfico 3 mostra que a maioria dos professores não acrescentaria mais nenhuma informação para ser exibida pelo software.



Figura 16 – Gráficos referentes à interpretação, entendimento e nível de aprofundamento dos conteúdos.

Durante a avaliação dos professores, alguns erros na solução foram identificados. Estes erros podem ser categorizados em gramaticais, notação matemática e erros de resolução. Os erros gramaticais eram, em sua grande maioria, referentes a pequenos erros de português ao longo do texto da seção de definição assim como nas explicações das resoluções. Por sua vez, os erros de notação matemática apontados eram relacionados às diferenças entre os termos adotados nas explicações geradas pelo software e os utilizados em sala pelo professor. Nessa categoria, muitos dos ajustes visam adequar a teoria apresentada para alunos do ensino fundamental. Por fim, os erros de resoluções representam os erros mais críticos e ocorreram em cenários não previstos pelo SE, o que acabou gerando uma mensagem de exceção ou uma resposta incorreta. A identificação de todos estes erros ajudou na aprimoração do software para trabalhar com as equações do primeiro grau que normalmente são abordadas no ensino fundamental, aumentando assim a confiabilidade do mesmo.

Os padrões ergonômicos adotados no desenvolvimento das interfaces gráficas colaboraram para um melhor entendimento do funcionamento do software. Além disso, na seção de definições, são apresentados exemplos de soluções geradas pelo software, o que colabora para o aprendizado do usuário de uma maneira mais rápida. Diante disso, os professores relataram que não sentiram a necessidade de um menu de ajuda.

Os professores consideraram que o sistema desenvolvido pode contribuir positivamente no ensino-aprendizagem da álgebra, conforme mostra a Figura 17. O gráfico 1 mostra que todos os professores acreditam que o software atingiu os requisitos mínimos para ser utilizado na sala de aula. Entretanto, quando é considerada a possibilidade de ser utilizado como uma ferramenta de estudo extraclasse, os professores julgaram que o software tem maior probabilidade de atingir seu objetivo, visto que pode servir como um tutor aos discentes em um ambiente que os docentes não estão presentes para auxiliar. Essa informação é mostrada no gráfico 2.

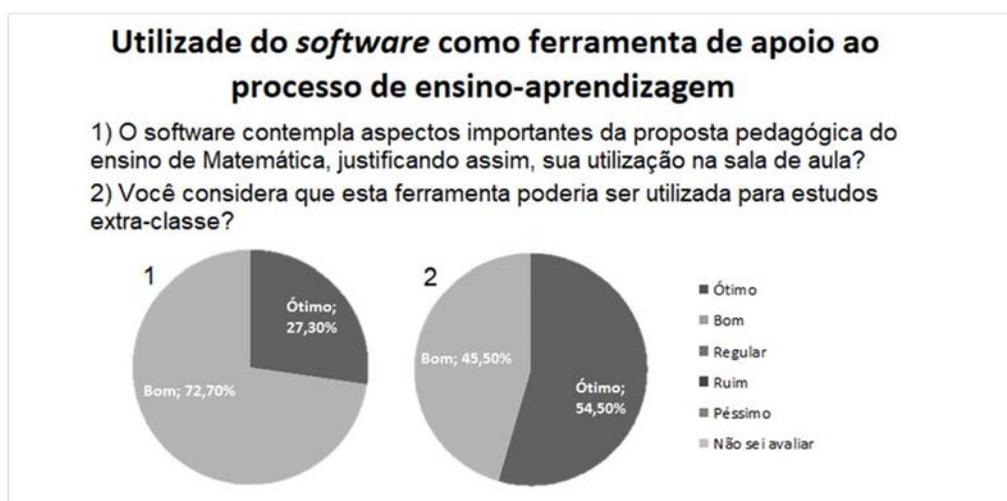


Figura 17 – Gráficos sobre a avaliação da utilidade do software como ferramenta de apoio ao ensino-aprendizagem.

8 Considerações Finais

As expressões algébricas constituem uma área essencial à compreensão da matemática, pois permitem aos alunos uma visão mais ampla ao deixar de realizar apenas operações com números, passando a trabalhar com variáveis representadas por letras. A partir do desenvolvimento e incorporação do módulo algébrico ao *IFMath*, o usuário tem à disposição um recurso que possibilita a análise e interpretação de equações de primeiro grau de maneira dinâmica, ou seja, ele pode inserir qualquer equação de seu interesse no software. 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.

Os resultados alcançados por este estudo indicam que teorias clássicas da computação, como compiladores e sistemas especialistas, podem contribuir ativamente para o processo de ensino-aprendizagem de expressões algébricas. Elas possibilitam a construção de ferramentas que permitem ao docente apresentar o conteúdo de uma forma mais interessante e criativa, atraindo assim a atenção do discente, que, por sua vez, tem à disposição uma ferramenta que pode atuar como um tutor em estudos extraclasse. Isso foi demonstrado pelas respostas dos professores no questionário, que avaliaram que o software possui os requisitos mínimos para ser utilizado como ferramenta de ensino-aprendizagem, tanto na sala de aula quanto, principalmente, em atividades extraclasse.

É importante ressaltar que este experimento ainda não é capaz de demonstrar os benefícios que o sistema pode trazer aos alunos, pois ele teve como objetivo apenas verificar a adequação do sistema às expectativas dos docentes da área de matemática. Este experimento considerou três aspectos diferentes: usabilidade e adequação da linguagem e notação matemática; nível de aprofundamento e interpretação do conteúdo; e avaliação da utilidade do software como ferramenta de apoio ao processo de ensino-aprendizagem. Os resultados positivos deste experimento indicam que o trabalho deve ser continuado, com a realização de experimentos com alunos para: (1) avaliar os aspectos de usabilidade e adequação citados acima sob a ótica dos alunos; (2) mensurar se ele realmente é capaz de melhorar a aprendizagem dos alunos, por meio de testes quantitativos ou qualitativos.

Além de novos testes com professores de matemática, pedagogos e do experimento com alunos, 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. Para isso, pode-se utilizar um algoritmo de busca para identificar todas as manipulações possíveis em cada passo da resolução. 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. Aproveitando o mesmo algoritmo de busca, outra funcionalidade que pode ser desenvolvida é a possibilidade do estudante submeter sua resposta para o sistema realizar a correção da mesma. Além disso, os conteúdos de equações do segundo grau, sistemas de equações, fatoração, expressões algébricas e produtos notáveis serão incorporados ao sistema.

Referências

- Aho, A. V.; Sethi, R., & Ullman, J. D. (1995) *Compiladores: princípios, técnicas e ferramentas* (1st ed). Rio de Janeiro: LTC.
- Aho, A. V.; Sethi, R., & Ullman, J. D. (2008) *Compiladores: princípios, técnicas e ferramentas* (2nd ed). Rio de Janeiro: Pearson Prentice Hall.
- Andrade, C. C. (2013). *O ensino da matemática para o cotidiano*. (Trabalho de Conclusão de Curso de Especialização não publicado). Universidade Tecnológica Federal do Paraná, Medianeira. [\[GS Search\]](#)
- Araújo, J. J. (2007). *Atividades exploratórias de Álgebra e Geometria com a utilização do software GeoGebra para a formação continuada de Professores de Matemática do Ensino Fundamental*. Dissertação (Trabalho de Conclusão de Mestrado Profissionalizante em Educação Matemática não publicado). Universidade Federal de Ouro Preto. [\[GS Search\]](#)

- Borba, M. C.; Gadanidis, G., & Silva, R. R. S. (2014). *Fases das tecnologias digitais em Educação Matemática: sala de aula e Internet em movimento*. Belo Horizonte: Autêntica. [GS Search]
- Brum, L. D. (2013). *Análise de erros cometidos por alunos de 8º ano do ensino fundamental em conteúdos de álgebra*. (Trabalho de Conclusão de Mestrado Profissionalizante em Ensino de Física e Matemática não publicado). Centro Universitário Franciscano de Santa Maria. [GS Search]
- Cataneo, V. I. (2011). *O uso do software GeoGebra como ferramenta que pode facilitar o processo de ensino aprendizagem da matemática no ensino fundamental séries finais*. (Monografia não publicada) Centro Universitário Barriga Verde - UNIBAVE. Retrieved from: <http://www.uniedu.sed.sc.gov.br/wp-content/uploads/2013/10/Vanessa-Isabel-Cataneo.pdf>.
- Cooper, K. D., & Torczon, L. (2014). *Construindo Compiladores*. (2nd ed). Rio de Janeiro: Elsevier. Tradução: Daniel Vieira.
- Druck, S. (2003). A crise no Ensino de Matemática no Brasil. *Revista do Professor de Matemática*, 52. Rio de Janeiro. [GS Search]
- Durkin, J. (1994). *Expert systems: design and development*. Englewood Cliffs: Prentice Hall.
- Durigon, A., Avila, A. J., Girardi, G., Madruga, I. C. T., Oliveira Junior, E. G., & Souza, W. P. (2018). *IFMATH – um aplicativo web direcionado ao estudo de matemática da educação básica. II EDUPALA - Congresso Internacional de Conhecimentos Pertinentes para a Educação na América Latina*. Lages, Brasil. Retrieved from: <https://edupala.com.br/downloads/anais2018.pdf>
- Fiorentini, D., & Morin, M. Â. (2001). *Por trás da porta, que matemática acontece?* Campinas, SP: FE/Unicamp – Cemmpem.
- Fragoso, W. C. (2001). O medo da matemática. *Revista do centro de educação*. Santa Maria. doi: [10.5902/19846444](https://doi.org/10.5902/19846444).
- França, K. V.; Santos, J. A., & Santos, L. S. B. (2007). *Dificuldades na aprendizagem da Matemática*. (Tese de licenciatura em Matemática não publicada) Centro Universitário Adventista de São Paulo, Campus de São Paulo. [GS Search]
- Geogebra. (2018). *Software Matemático*. Florida Atlantic University. Retrieved from: <https://www.geogebra.org/>.
- Gil, A. C. (2010). *Como Elaborar Projetos de Pesquisa*. (5th ed). São Paulo: Atlas.
- Gladcheff, A. P.; Zuffi, E. M., & Silva, M. (2011). Um Instrumento para Avaliação da Qualidade de Softwares Educacionais de Matemática para o Ensino Fundamental. In *Anais do XXI Congresso da Sociedade Brasileira de Computação*. [GS Search]
- Gomes, A. S., & Padovani, S. (2005). Usabilidade no ciclo de desenvolvimento de software educativo. *Simpósio Brasileiro de Informática na Educação - SBIE*. Juiz de Fora (MG). [GS Search]
- Gramani, M. C. N., & Scrich, C. R.. (2013). Influência do desempenho educacional na escolha da profissão. *Cadernos de Pesquisa*, 42, 868-883. doi: [10.1590/S0100-15742012000300012](https://doi.org/10.1590/S0100-15742012000300012). [GS Search]
- Gravina, M. A., & Santarosa, L. M. (1998). A aprendizagem da matemática em ambientes informatizados. *IV Congresso RIBIE*. Brasília. [GS Search].
- Jaques, P. A., Seffrin, H., Rubi, G., Morais, F. D., Ghilardi, C., Bittencourt, I. I., & Isotani, S. (2013). Rule-based expert systems to support step-by-step guidance in algebraic problem

- solving: The case of the tutor PAT2Math. *Expert Systems with Applications*, 40(14), 5456-5465. doi: [10.1016/j.eswa.2013.04.004](https://doi.org/10.1016/j.eswa.2013.04.004). [\[GS Search\]](#)
- José Neto, J. (2016). *Introdução à compilação* (2nd ed). Rio de Janeiro: Elsevier.
- LaTeX (2018). Projeto *LaTeX*. Retrieved from <https://www.latex-project.org/>.
- Khan, A. R., & Amin, H. U., & Rehman, Z. U. (2011). Application of Expert System with Fuzzy Logic in Teachers' Performance Evaluation. *International Journal of Advanced Computer Science and Applications*, 2(2), 51-57. doi: [10.14569/IJACSA.2011.020210](https://doi.org/10.14569/IJACSA.2011.020210). [\[GS Search\]](#)
- Khanna, S., & Kaushik, A., & Barnela, M. (2011). Expert systems advances in education. *National Conference on Computational Instrumentation*. Chandigarh, India. [\[GS Search\]](#)
- Luger, G. L., & Stubblefield, W. A. (2008). *Artificial Intelligence: Structures and Strategies for Complex Problem Solving* (6th ed). Benjamin Cummings.
- Maior, L., & Trobia, J. (2009). Tendências metodológicas de ensino-aprendizagem em educação matemática: resolução de problemas - um caminho. Ponta Grossa. [\[GS Search\]](#)
- Marconi, M. A., & Lakatos, E. M. (2011). *Metodologia Científica* (5th ed). São Paulo: Atlas.
- Marczal, D., Direne, A. I., Pimentel, A. R., Maschio, E., & Borille, A. C. (2016). Metodologia e Software Educacional para a Investigação e Remediação de Erros Conceituais em Matemática. *Revista Brasileira de Informática na Educação*, 24(02), 77. doi: [10.5753/rbie.2016.24.02.77](https://doi.org/10.5753/rbie.2016.24.02.77). [\[GS Search\]](#)
- Miranda, D. F., & Blaudares, J. B. (2007). *Informatização no ensino de matemática: investindo no ambiente de aprendizagem*. Campinas: Zetetiké. [\[GS Search\]](#)
- MEC (2016). *Resultado do Pisa de 2015 é tragédia para o futuro dos jovens brasileiros, afirma ministro*. Retrieved from: <http://portal.mec.gov.br/component/content/article?id=42741/>
- Oliveira, J., & Casagrande, N. M., & Galerani, L. D. J. (2016). A evolução tecnológica e sua influência na educação. *Revista Interface Tecnológica*, 13(1), 23-38. [\[GS Search\]](#)
- Oliveira, J. B.; Santana, A. M., & Reali, G. A. (2012). O uso de *tablets* e o *GeoGebra* como ferramentas auxiliaadoras no ensino da matemática. *Conferência Latinoamericana de GeoGebra*. Montevideo, Uruguai. [\[GS Search\]](#)
- Oracle (2018). Linguagem *Java*. Retrieved from: <https://www.oracle.com/java/technologies/index.html>.
- Perius, A. A. B. (2012). *A tecnologia aliada ao ensino de matemática*. (Trabalho de conclusão de curso de especialista em mídias na educação). Universidade Federal do Rio Grande do Sul (UFRGS). Porto Alegre. [\[GS Search\]](#)
- Ponte, J. P. (2005). Álgebra no currículo escolar. *Educação e Matemática*. Retrieved from: <https://pt.scribd.com/document/201429534/Algebra-no-curriculo-escolar-2>.
- Rocha, A. R.; Campos, G. H. B. (2008). Avaliação da qualidade de software educacional. *Em Aberto*, 12, 53. Brasília. [\[GS Search\]](#)
- Rodrigues, J. A. (2011). *Wolfram Alpha: uma nova visão da Matemática*. 6ª Conferência Ibérica de Sistemas e Tecnologias de Informação. Lisboa. [\[GS Search\]](#)
- Rodrigues, L. L. (2005). *A Matemática ensinada na escola e a sua relação com o cotidiano*. Brasília: UCB.
- Santos, W. O., & Silva Junior, C. G. (2016). Virtualização de Jogos Educativos: Uma Experiência no Ensino de Matemática. *Revista Brasileira de Informática na Educação*, 24(02), 108. doi: [10.5753/rbie.2016.24.02.108](https://doi.org/10.5753/rbie.2016.24.02.108). [\[GS Search\]](#)

- Santos, G. P.; Silva, S. L., & Santos, G. M. (2018). A Metodologia do Ensino da Matemática: Softwares e Jogos. *Revista Científica Multidisciplinar Núcleo do Conhecimento*. São Paulo. Retrieved from: <https://www.nucleodoconhecimento.com.br/matematica/matematica-softwares-e-jogos>.
- Seffrin, H. M., & Jaques, P. A. (2016). Modelagem do Conhecimento Algébrico dos Estudantes com Redes Bayesianas Dinâmicas. *Revista Brasileira de Informática na Educação*, 24(02), 54. doi: [10.5753/rbie.2016.24.02.54](https://doi.org/10.5753/rbie.2016.24.02.54). [GS Search]
- Silva, B. F., Jaques, P. A. (2017). Assistindo os estudantes na resolução de problemas algébricos: uma proposta baseada na Teoria da Carga Cognitiva. *Revista Brasileira de Informática na Educação*, 25(3), 47-79. doi: [10.5753/rbie.2017.25.3.47](https://doi.org/10.5753/rbie.2017.25.3.47). [GS Search]
- Singh, R.; Sharma, V., & Varshney, M. (2008). *Design and Implementation of Compiler* (1st ed). Nova Deli: New Age International.
- Slomski, V. G., & Araujo, A. M. P., & Camargo, A. S. S., & Weffort, E. F. J. (2016). Tecnologias e mediação pedagógica na educação superior à distância. *Journal of Information Systems and Technology Management*. 13(1), 131-150. doi: [10.4301/S1807-17752016000100007](https://doi.org/10.4301/S1807-17752016000100007). [GS Search]
- Socas, M. M.; Camacho M.; Palarea M., & Hernández J. (1996). *Iniciación al algebra*. Madrid: Ed Síntesis.
- Symbolab (2018). *Software Matemático*. EqsQuest. Retrieved from: <https://www.symbolab.com/>.
- Uehara, F. M., & Monteiro M. I., & Silva Mill, D. R., & Fiscarelli, S. H. (2017). *The use of learning objects in elementary school: a bibliometric study*. *Revista Ibero-americana de Estudos em Educação*. 12, 1539-1553. doi: [10.21723/riaee.v12.n.esp.2.10308](https://doi.org/10.21723/riaee.v12.n.esp.2.10308). [GS Search]
- Vasconcelos, C. C. (2000). *Ensino-aprendizagem da matemática: velhos problemas, novos desafios*. Lisboa: Editora Instituto Politécnico de Viseu.
- Viegas, E. C. (2017). *O uso do Symbolab e MalMath em dispositivos móveis: uma ferramenta prática para o cálculo de integrais duplas*. (Trabalho Monográfico não publicado apresentado à Coordenação do Curso de Licenciatura em Matemática). Universidade Federal da Paraíba, Rio Tinto. [GS Search]
- Wolframalpha (2018). *Software Matemático*. Wolfram Alpha LLC. Retrieved: March 16, 2018, from: <http://wolframalpha.com/>.
- W3Schools (2018). *Tecnologias Web*. Retrieved from: <https://www.w3schools.com/>.
- Zorzan, A. S. L. (2007). Ensino-Aprendizagem: algumas tendências na educação matemática. *Revista Ciências Humanas*. Frederico Westphalen. [GS Search]