



SOFTWARE DE SIMPLIFICAÇÃO E RESOLUÇÃO DE EXPRESSÕES

ALGÉBRICAS

Discente: Antonio Raphael de Arruda Basso

Docente Orientador: Odahyr Cavalini Júnior

INTRODUÇÃO

Há uma série de ferramentas que têm como propósito tornar o conhecimento sistemático computável. Alguns *softwares* são mais voltados para conhecimento algébrico, como o *Derive* ou *TI-Nspire da Texas Instruments*, outros vão além, tentando tornar acessível todo o conhecimento sistemático computável fornecendo respostas para consultas factuais, como o *Wolframalpha*. Devido ao caráter complexo dessas ferramentas, muitas são pagas, ou os recursos oferecidos são extremamente limitados. Outro problema dessas ferramentas está no suporte, seja por questões de internacionalização ou pela descontinuidade da ferramenta. Além disso, os algoritmos usados dessas ferramentas não está disponível para reconhecimento.

OBJETIVO

Este trabalho tem como objetivo desenvolver um software de conhecimento computacional matemático de código aberto, cujo objetivo principal é analisar, interpretar, simplificar e resolver de forma iterativa expressões algébricas com base em heurísticas simples, utilizando para isso conceitos das disciplinas de exatas do curso de Ciência da Computação, uma vez que as ferramentas disponíveis são proprietárias. O software fornece uma forma simples de se estendê-lo utilizando linguagens embarcadas, não sendo necessária a recompilação do código fonte do projeto.

METODOLOGIA

Para o desenvolvimento do software, foi utilizada a plataforma Java, cuja estrutura foi definida como um multiprojeto do Gradle/Maven. Tendo como base o design tático e estratégico do Domain Driven Design, Test Driven Development, Padrões de Projetos (como Inversion of Control, Composite, Visitor e Model-View-Controller) e teorias de compilação, o projeto foi dividido em dois módulos: *core* (que contêm o domínio da aplicação), onde foram definidas a gramática do ANTLR para análise de expressões e os pacotes com as classes que compõe o núcleo do software; e *web* (implementação web da interface do usuário utilizando o SpringBoot para o gerenciamento das heurísticas que serão armazenadas em banco de dados, e que serão carregadas e interpretadas em tempo de execução por conta da interoperabilidade entre as linguagens Java e Groovy).

RESULTADOS

Como o princípio adotado para o desenvolvimento da aplicação foi o TDD, a maior parte dos problemas encontrados foram previamente resolvidos no decorrer do projeto utilizando-se testes automatizados. Além de testes automatizados, foram realizados testes para a validação da aplicação no cadastro e edição de heurísticas, bem como na simplificação e resolução de uma expressão algébrica extensa contendo diversos fatores (figura 1).

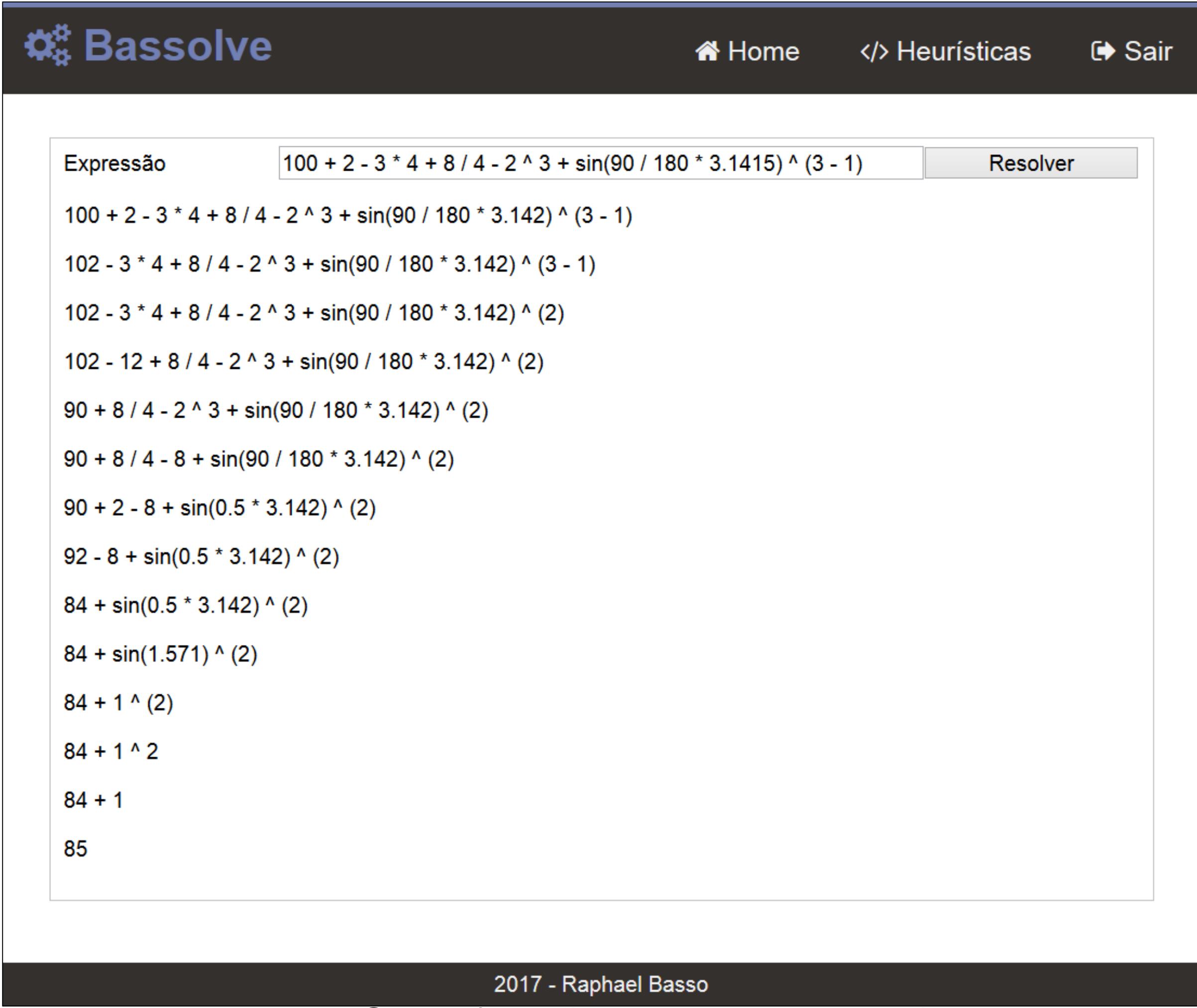


Figura 1: Simplificação de uma expressão.

CONCLUSÃO

Os resultados obtidos foram bastante satisfatórios pois além de atender ao que foi proposto, o *software* permite um elevado grau de extensão com melhorias e adição de novos recursos. Apesar do projeto ter tratado apenas de simplificação de expressões algébricas numéricas, é possível adicionar suporte a simplificação de polinômios, derivadas, integrais, etc.

REFERÊNCIAS

1. ANTLR. ANTLR 4 documentation. Disponível em: <<https://github.com/antlr/antlr4/blob/4.6/doc/index.md>>. Acesso em: 26 mar. 2017.
2. APACHE GROOVY. Integrating Groovy into applications. Disponível em: <<http://groovy-lang.org/integrating.html>>. Acesso em: 20 mar. 2017.
3. BECK, Kent. TDD: desenvolvimento guiado por testes. Porto Alegre: Bookman, 2010.
4. EVANS, Eric. Domain-driven design: atacando as complexidades no coração do software. Rio de Janeiro: Alta Books, 2010.
5. FOWLER, Martin. Inversion of control containers and the dependency injection pattern. Disponível em <<https://martinfowler.com/articles/injection.html>>. Acesso em 20 mai. 2017.
6. GAMMA, Erich; HELM, Richard; JOHNSON, Ralph; VLISSIDES, John. Padrões de projetos: soluções reutilizáveis de software orientado a objetos. Porto Alegre: Bookman, 2000.
7. JAVA. Obtenha informações sobre a tecnologia java. Disponível em: <https://www.java.com/pt_BR/about/>. Acesso em: 22 mai. 2017.
8. LOUDEN, Kenneth C. Compiladores: princípios e práticas. São Paulo: Pioneira Thomson Learning, 2004.