

Relatório Final

Calculadora Funcional

Augusto Samuel Modesto

Engenharia de Software

augusto.modesto@live.com

Jonathan Henrique Maia de Moraes

Engenharia de Software

arkye@hotmail.com.br

Matheus Herlan dos Santos Ferraz

Engenharia de Software

matheus.herlan@gmail.com

Rudmar Rodrigues Campos Junior

Engenharia de Software

rudmar\_junior@hotmail.com

Disciplina de Paradigmas de Programação

Professora Dra. Milene Serrano

Gama - DF, Abril de 2014

UnB - Campus Universitário Gama

Área Especial de Indústria Projeção A - UnB/Brasília, Setor Leste. CEP: 72.444-240

GAMA - DF

Professora Dra. Milene Serrano Abril, 2014

Relatório Final

Augusto Samuel Modesto

Jonathan Henrique Maia de Moraes

Matheus Herlan dos Santos Ferraz

Rudmar Rodrigues Campos Junior

Engenharia de Software, Universidade de Brasília - Faculdade do Gama, Gama/DF  
Engenharia de Software, Universidade de Brasília - Faculdade do Gama, Gama/DF

Engenharia de Software, Universidade de Brasília - Faculdade do Gama, Gama/DF  
Engenharia de Software, Universidade de Brasília - Faculdade do Gama, Gama/DF

**Resumo**. Esse relatório é referente ao trabalho desenvolvido em Haskell, uma linguagem funcional, da disciplina de Paradigmas de Programação. O mesmo apresenta uma modelagem dividida em Módulos funcionais, cada um com funções específicas daquele determinado módulo, configuração do repositório GitHub®, código fonte referente à implementação, imagens do protótipo gerado e o relatório final dos resultados obtidos. Este projeto trata de um software com funções matemáticas desenvolvidas em linguagem funcional, o nome do produto final será Calculadora.

**Palavras-chave**: Calculadora, Software, Haskell, Funções, Linguagem Funcional, Módulos.

SUMÁRIO

[1 INTRODUÇÃO 4](#_Toc386392240)

[2 ANALISE INSTITUCIONAL 5](#_Toc386392241)

[2.1 Público Alvo 5](#_Toc386392242)

[2.2 Descrição das Regas de Negócio 5](#_Toc386392243)

[3 ABRANGÊNCIA DO SISTEMA PROPOSTO 6](#_Toc386392244)

[3.1 Objetivo Geral 6](#_Toc386392245)

[3.2 Objetivos Específicos 6](#_Toc386392246)

[3.3 Requisitos não funcionais 6](#_Toc386392247)

[3.4 Metodologia e Tecnologias Utilizadas 7](#_Toc386392248)

[4 DOCUMENTAÇÃO DO PROJETO 8](#_Toc386392249)

[4.1 Arquitetura do Sistema 8](#_Toc386392250)

[5 IMPLEMENTAÇÃO DO SISTEMA 10](#_Toc386392251)

[6 CONTROLE DE VERSÃO 11](#_Toc386392252)

[7 CONSIDERAÇÕES FINAIS 11](#_Toc386392253)

[8 REFERÊNCIAS BIBLIOGRÁFICAS 13](#_Toc386392254)

# INTRODUÇÃO

Nesse relatório técnico, procuramos apresentar os resultados obtidos até o momento, destacando-se a conclusão parcial do trabalho solicitado na disciplina de Paradigmas de Programação, UnB/FGA.

O relatório está organizado em partes específicas. Inicialmente, tem-se uma breve descrição do sistema adotado para a realização do trabalho, assim como a abstração realizada para obtenção dos módulos. Em seguida, faz-se a apresentação de uma modelagem baseada na arquitetura do sistema.

Após a apresentação da descrição e modelagem, são disponibilizadas imagens da implementação de algumas partes do sistema, bem como comentários referentes aos atributos e relacionamentos estabelecidos.

Além dos aspectos citados acima, são disponibilizados os links para os repositórios GitHub e também, as referências bibliográficas ao final.

# ANALISE INSTITUCIONAL

## Público Alvo

O desenvolvimento da Calculadora visa atender a um publico especifico, aqueles que necessitam de uma ferramenta com resultados rápidos e ágeis de funções mais especificas na área da matemática, tais como o calculo do MDC e do MMC, regressão linear, entre outros.

## Descrição das Regas de Negócio

A Calculadora deverá realizar além das operações básicas, tais como somar, subtrair, multiplicar e dividir, determinar valores como cálculos de funções mais específicas como MDC e MMC de uma série de números que serão fornecidas pelo usuário.

Vale ressaltar que o escopo do desenvolvimento do projeto é limitado as funções pré-estabelecidas, em outras palavras, a calcula somente realizara calculo dento do que fora definido em seu desenvolvimento.

# ABRANGÊNCIA DO SISTEMA PROPOSTO

## Objetivo Geral

Fornecer uma ferramenta que possibilite a realização de cálculos gerais e específicos da matemática avançada.

## Objetivos Específicos

A calculadora deverá realizar os seguintes cálculos para solução do software:

* Soma, Subtração, Multiplicação e Divisão;
* Calculo do MDC e MMC;
* Regressão linear;

## Requisitos não funcionais

Deverá apresentar uma interface no prompt de comandos de interação fácil e objetiva, visando facilitar a usabilidade do público-alvo.

## Metodologia e Tecnologias Utilizadas

Atestado nas aulas de Paradigmas de Programação, Haskell, uma das linguagem que mais se aproxima de C em relação a desempenho, é uma linguagem de programação orientada ao paradigma funcional emergindo na década de 90 nomeada em homenagem ao lógico Haskell Curry. O estilo de sua tipagem é forte, estática e inferida[[1]](#footnote-1) sendo influenciadora de diversas linguagens como Python.

A linguagem conta ainda com extensa bibliografia com diversão funções já definidas. Além disso, a linguagem Haskell possui um forte embasamento no paradigma sequencia, com suporte a funções recursivas, tipos de dados, entre outras características. A combinação das diversas características permite que Haskell construa funções simples no qual, em outras linguagens, poderia ter um teor complexo. Isso porque tudo o objetivo de Haskell está voltado na construção modular de função.

Para a interface gráfica serão desenvolvidos menus cuja exibição está agregada a cada função do projeto. Ou seja, serão diversas telas construídas com caracteres especiais, que serão apresentadas no decorrer das escolas das funções.

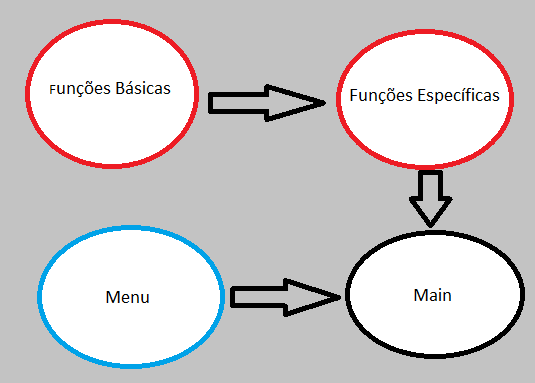
O objetivo principal das telas será ORIENTAR o usuário a como utilizar o aplicativo, isto é, além de haver uma breve descrição do calculo, haverá mini tutorial em cada menu, mostrando como utilizar determinado função do programa. A seguir serão apresentadas as telas de algumas das principais funções do aplicativo.

# DOCUMENTAÇÃO DO PROJETO

## Arquitetura do Sistema

Todo o desenvolvimento do projeto foi estruturado em módulos, isto é, em pequenos blocos com funções especificas daquele bloco. Logo, é possível aumentar a coesão uma vez que modulando o programa a compreensão e manutenção se tornam mais simples e objetiva e diminui o acoplamento por ter definições dos locais de trabalho (módulos) bem definidos.

A imagem abaixo retrata como são divididos e organizados cada módulo da calculadora:



*Figura 01: Diagrama de Blocos Funcionais.*

No bloco **funções** **básicas** estão disponíveis as funções mais simples da matemática, são ela:

- Soma;

- Subtração;

- Multiplicação;

- Divisão;

- Exponencial;

- Fatorial;

- Calcular quantidade de elementos em uma lista.

No bloco **funções específicas** estão disponíveis funções mais complexo, no qual exigem algum tipo de conhecimento prévio, são elas:

- Calculo de MMC;

- Calculo de MDC;

- Calculo do PPR;

- Calculo de função quadráticas.

No bloco **menu** estão os menus onde são feitas as interações com cada função matemáticas, além da parte visual do aplicativo

E no bloco **main** estão alocados todos os blocos. É na main que são disparados as chamadas das funções de cada bloco para operar o desenvolvimento do projeto.

# IMPLEMENTAÇÃO DO SISTEMA

# CONTROLE DE VERSÃO

O controle de versão é uma importante ferramenta no desenvolvimento de um software. A eficácia do controle de versão de software é comprovada por fazer parte das exigências para melhorias do processo de desenvolvimento de certificações tais como [CMMI](http://pt.wikipedia.org/wiki/CMMI) e [SPICE](http://pt.wikipedia.org/wiki/SPICE).

O repositório com o histórico de modificações no sistema implementado está disponível no endereço contendo todas as informações necessárias: [https://github.com/MatheusFerraz/Paradigmas---Funcional](https://github.com/tatielen/projetoOO).

A imagem a seguir retrata o estado final do repositório, isto é, ao final do desenvolvimento o congelamento do repositório é ativado permanecendo da seguinte maneira:

|  |
| --- |
|  |

*Figura 02: Total de Commit ao final do projeto.*

# CONSIDERAÇÕES FINAIS

As dificuldades foram concretizadas durante a construção dos códigos, uma vez que, emergi como um novo paradigma de programação para os participantes deste grupo. No entanto, apesar das dificuldades na elaboração dos códigos, nas interações entre os módulos, a produção deste projeto nos permitiu um melhor grande aprendizado na linguagem, além do aprimoramento no que diz respeito ao nível de abstração na programação em geral.

# REFERÊNCIAS BIBLIOGRÁFICAS

Wikipédia. **Haskell (Linguagem de Programação).** Disponível em: <http://www.portugal-a-programar.pt/topic/958-haskell-alguns-trabalhos/>. Acessado em: 18 de abril de 2014.

Aprendendo Haskell. **Aprender Haskell será um grande bem para você**. Disponível em: <http://haskell.tailorfontela.com.br/chapters>. Acessado em: 19 de abril de 2014.

RAUBER, A.D. B**. Programação Funcional com a linguagem Haskell**. Departamento de Informática da Universidade Federal do Paraná. Disponível em: <http://www.inf.ufpr.br/andrey/ci062/ProgramacaoHaskell.pdf>. Acessado em: 22 de abril de 2014.

LIMA, M. A. V. **Introdução a Linguagem Funcional Haskell**. Faculdade de Ciências da Computação da Universidade federal de Uberlândia. Disponível em: <http://www.facom.ufu.br/~madriana/PF/hs.pdf>. Acessado em: 22 de abril de 2014.

1. Wikipédia. Disponível em: <http://pt.wikipedia.org/wiki/Haskell\_(linguagem\_de\_programa%C3%A7% C3%A3o)>. Acessado em 25 de abril de 2014. [↑](#footnote-ref-1)