

Framework Java de cálculo de problemas matemáticos utilizando métodos numéricos.

Rubens Silva Lima¹

¹Instituto Federal Catarinense-IFC
rubenssilva905@gmail.com

Abstract. *This work reports the development of a framework for solving mathematical problems applying certain numerical methods related to the specific problem, specifies a didactic way the framework facilitates the generation of tables and graphs in a simple and intuitive way..*

Key-words: *Framework; Google Chars; numerical methods, Java.*

Resumo. *Este trabalho relata o desenvolvimento de um framework para resolução de problemas matemáticos aplicando certos métodos numéricos referente ao problema específico, visando obter uma forma didática o framework facilita a geração de tabelas e gráficos de forma simples e intuitiva.*

Palavras-chave: *Framework; Google Charts; Métodos numéricos, Java.*

1. Introdução

A matemática está presente em várias áreas de nossa vida, mesmo parecendo muito complicada para muita gente precisamos dela para diversas atividades, alguns mais que outros, matemáticos enfrentam diversos cálculos diariamente e visando auxiliar quem utiliza esses métodos de forma manual, para quem esteja aprendendo e deseja acompanhar como funciona os métodos numéricos a cada passo, esse trabalho mostra a construção de uma aplicação para facilitar os cálculos e poder visualizar de forma simplificada e didática como cada método funciona com sua particularidade, utilizando todas as boas maneiras de programação e aplicando todos os conceitos de SOLID.

1.1. Objetivos

O principal objetivo deste trabalho é desenvolver um framework que seja funcional e ofereça maior produtividade da resolução de sistemas lineares e funções polinomial e não polinomial. Gerar gráficos que seja possível visualizar as funções e entender os seus comportamentos, e tabelas para entender o cálculo realizado por cada método e perceber a aproximação a cada iteração.

1.2. Justificativa

Realizar cálculo numérico manualmente, ter que montar um algoritmo sempre que necessário encontrar o resultado e gerar uma tabela específica para cada base de dados novos são tarefas que demandam tempo e tem uma complexidade elevada, esse

trabalho visa trazer ao desenvolvedor ou usuário uma ferramenta que facilite esses passos.

2. Framework Desenvolvido

Nesta seção abordamos as características e todo o processo de concepção do framework desenvolvido.

2.1. Ferramentas utilizadas

2.1.1. Eclipse

O IDE utilizado no desenvolvimento foi o eclipse 2021-03 uma IDE completa destacam-se algumas funcionalidades no Eclipse, para o desenvolvimento em Java, que são:

- Auto-complete;
- Organização de imports;
- Auto-formatação de código;
- Compilação imediata;
- Geração de trechos de códigos;
- Geração de JavaDOC;
- Integração com JUnit.

O Eclipse é uma ferramenta FREE e Open-Source podendo ser baixado através do site <http://www.eclipse.org>.

2.1.2. Github

O controlador de versão utilizado para salvar o projeto no repositório e ter controle de cada etapa do desenvolvimento algumas funcionalidades do GitHub que foram utilizados

- Git Commit;
- Git Push;
- Git Ignore.

O github é uma plataforma Free que pertence a Microsoft podendo ser acessado pelo site [GitHub.com](https://github.com) ou via console.

2.2. Padrões e Características Empregadas

Neste projeto foram utilizados vários conceitos e boas práticas do mundo de desenvolvimento de softwares com base no conceito SOLID. Todos os conceitos foram aplicados visando garantir uma boa qualidade do código, bem como o emprego de padrões de desenvolvimento, facilitando assim, o estudo do framework por outros desenvolvedores. Abordando ainda alguns conceitos e características empregadas no framework, mostrando também onde e como foram empregados.

2.2.1 Template Method

Esse padrão foi utilizado para abstrair a complexidade dos algoritmos tanto de cálculo como a geração de gráficos e tabelas, fazendo com que o desenvolvedor foque na questão da preparação de dados e interpretação dos elementos gerados.

3. Funcionamento do framework

Ao entender a necessidade do cálculo que é preciso ser feito, o desenvolvedor ou usuário deve criar uma Interface do método escolhido e passar para a classe de execução de métodos CalculaZeroFuncao (figura 1), e a partir daí escolher se deseja encontrar o x dentro de um intervalo escolhido, se deseja gerar o relatório ou a tabela com os passos feito pelo método.

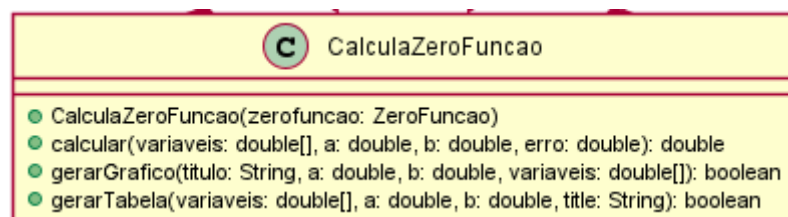


Figura 1 – Classe CalculaZeroFuncao

3.1. Métodos Implementados

Foram escolhidos alguns principais métodos de cálculo de zero de função para serem implementados, mas modelo de forma que o framework em trabalhos futuros possa receber novos sem alterações dos antigos.

3.1.1. Bhaskara

O bhaskara é o primeiro método que se aprende na escola para encontrar zero da função, apesar de não ser um método iterativo e não ser aberto a cálculo de funções de graus diferentes de 2 foi implementado pela questão da didática e comparação de tempo e poder computacional gasto para o cálculo.

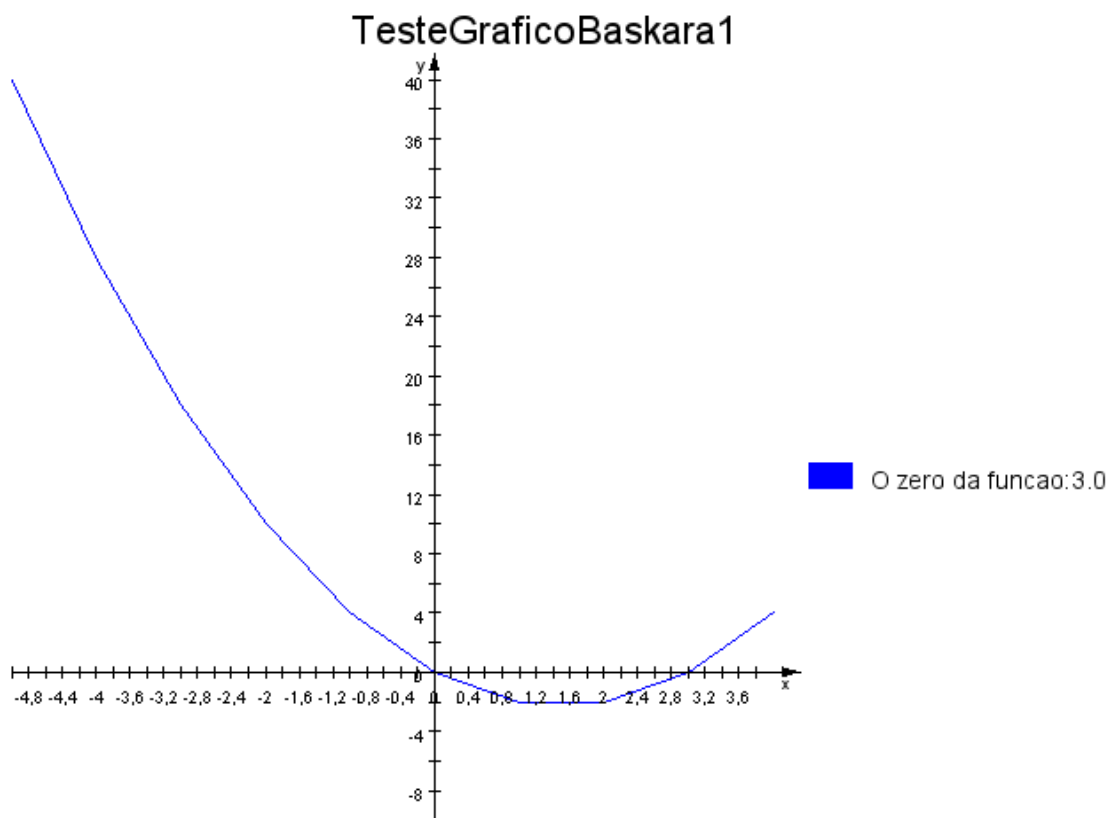


Figura 2 – Exemplo de gráfico gerado pelo método de bhaskara

Na figura dois é um exemplo de gráfico gerado pelo método de Bhaskara para a função $F(x) = x^2 - 3x$.

Delta	x1	x2
9	0	6
9	0	3

Figura 3 – Exemplo de tabela gerada pelo método de bhaskara

Na figura três é um exemplo de uma tabela gerada pelo método de Bhaskara para a função $F(x) = x^2 - 3x$, mostrando o valor de delta, e os passos até encontrar os valores finais de x.

3.1.2. Newton Raphson

Diferente do método de bhaskara o método de Newton-Raphson é um método iterativo, ele usa uma aproximação inicial e a partir disso calculasse valores das retas tangente para chegar ao valor mais próximo desejado, com critérios de parada definindo o número de iterações, para se encontrar o x que será usado numa próxima iteração se utiliza da fórmula $x_n = x_{n-1} - \left(\frac{f(x_{n-1})}{f'(x_{n-1})}\right)$ até se encontrar uma raiz que satisfaça o critério de parada, que seriam o número de iteração ou comparação erro estipulado com $|x_n - x_{n-1}|$, ou seja o módulo dos x anterior com o x da interação atual.

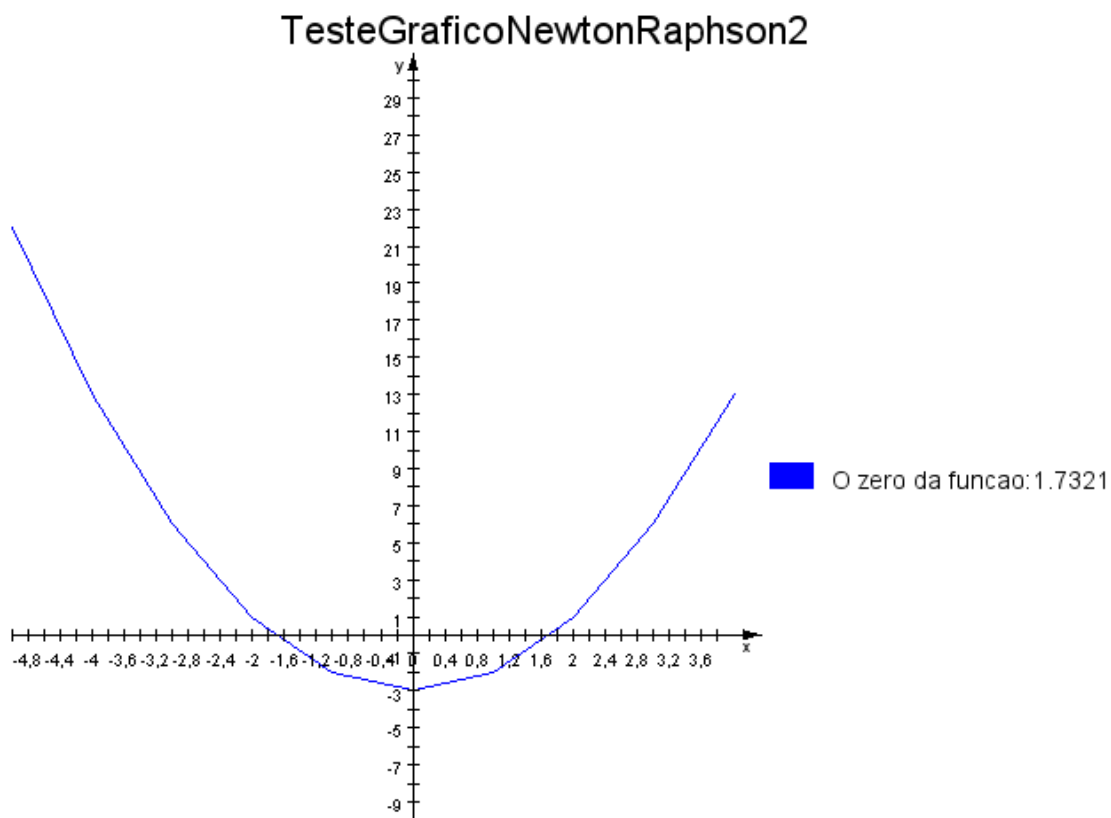


Figura 4 – Exemplo de gráfico gerado pelo método de bhaskara

Na figura quatro é um exemplo de gráfico gerado pelo método de Newton Raphson para a função $F(x) = x^2 - x$.

n	xn	f(xn)	derivada(xn)	xn+1	erro
1	-1	-4	-3	-2.333	7.111
2	-2.333	7.111	-13.667	-1.813	1.083
3	-1.813	1.083	-9.504	-1.699	0.052
4	-1.699	0.052	-8.592	-1.693	0

Figura 5 – Exemplo de tabela gerado pelo método de Newton Raphson.

Na figura cinco é um exemplo de uma tabela gerada pelo método de Newton Raphson para a função $F(x) = x^2 - x$, mostrando os valores de cada iteração, o valor relativo ao x, o valor de F(x), a derivada o valor do próximo x e erro naquela iteração.

3.1.3. Método das secantes

O método das secantes é uma variação do método de Newton Raphson, ele evita de se calcular a derivada a cada iteração sua fórmula é $x_{n+1} = x_n - (x_n - x_{n-1}) * f(x_n) / (f(x_n) - f(x_{n-1}))$, como visto na fórmula é necessário ter dois valores iniciais para partir para o cálculo do próximo x. O método das secantes pode ser mais rápido que o da bissecção, porém mais lento que o de Newton Raphson.

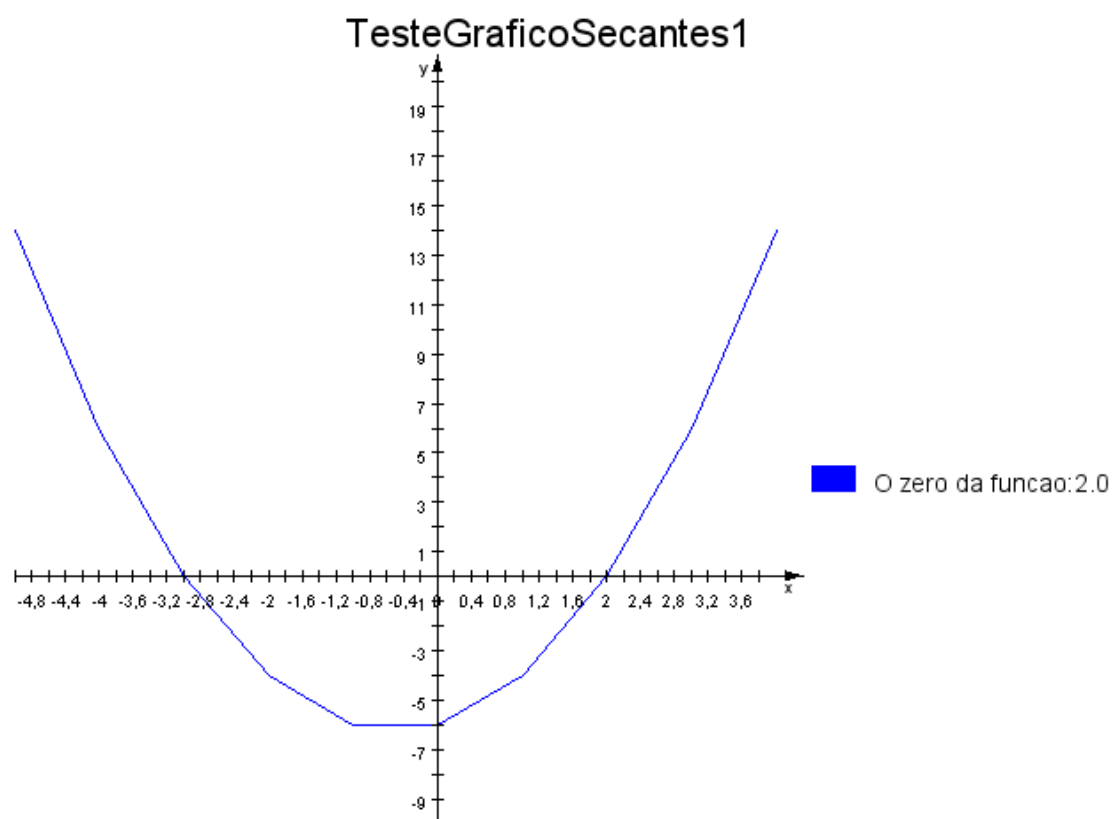


Figura 6 – Exemplo de gráfico gerado pelo método das Secantes.

Na figura seis é um exemplo de gráfico gerado pelo método das secantes para a função $F(x) = x^2 + x - 6$ com intervalos um e dois.

n	xn	f(xn)	erro
1	1	-4	0
2	2	0	1
3	2	0	0

Figura 7 – Exemplo de tabela gerado pelo método das Secantes.

A figura sete é um exemplo de uma tabela gerada pelo método das secantes para a função $F(x) = x^2 + x - 6$, mostrando os valores de cada iteração, o valor relativo ao x , o valor de $F(x)$, o valor do próximo x e erro naquela iteração.

3.1.4. Método da bisseção

O método da bisseção utiliza o valor médio de um intervalo dado para calcular o x e substituindo o lado mais longe do zero da função.

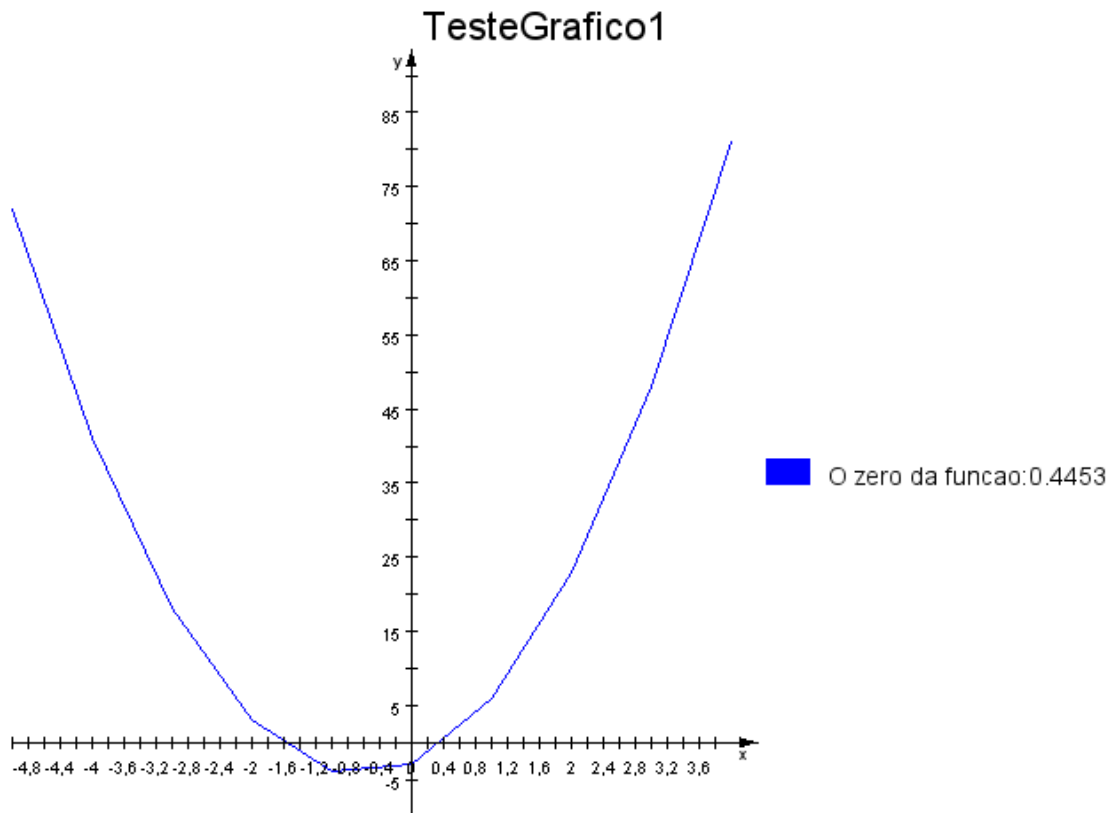


Figura 8 – Exemplo de gráfico gerado pelo método da bisseção.

Na figura oito é um exemplo de gráfico gerado pelo método da bisseção para a função $F(x) = 4x^2 + 5x - 3$ com intervalo passado de $[-1,1]$.

n	a	xn	b	f(a)	f(xn)	f(b)	erro
1	-1	0	1	-4	-3	6	3
2	0	0.5	1	-3	0.5	6	0.5
3	0	0.25	0.5	-3	-1.5	0.5	1.5
4	0.25	0.375	0.5	-1.5	-0.563	0.5	0.563
5	0.375	0.438	0.5	-0.563	-0.047	0.5	0.047
6	0.438	0.469	0.5	-0.047	0.223	0.5	0.223
7	0.438	0.453	0.469	-0.047	0.087	0.223	0.087
8	0.438	0.445	0.453	-0.047	0.02	0.087	0.02
9	0.438	0.441	0.445	-0.047	-0.014	0.02	0.014
10	0.441	0.443	0.445	-0.014	0.003	0.02	0.003
11	0.441	0.442	0.443	-0.014	-0.005	0.003	0.005
12	0.442	0.443	0.443	-0.005	-0.001	0.003	0.001

Figura 9 – Exemplo de tabela gerado pelo método da bisseção.

A figura nove é um exemplo de uma tabela gerada pelo método das secantes para a função $F(x) = 4x^2 + 5x - 3$, mostrando os valores de cada iteração, valor do inferior do intervalo, o valor relativo ao x, o valor superior do intervalo, o valor de F(a), o valor de F(x), o valor de F(b) e o erro naquela iteração.

4. Conclusão

A abordagem de frameworks contribui significativamente para a reutilização e facilitando o desenvolvimento de software. Frameworks é uma abordagem que está sendo cada vez mais utilizada, com o objetivo de diminuir o tempo e esforços no desenvolvimento de artefatos de software, e fazendo que o acoplamento no sistema fique baixo facilitando sua manutenção. Uma vez utilizado as boas práticas de programação, os padrões de projeto de maneira eficiente e com uma boa documentação do framework, o framework pode ser bastante útil e utilizado no mercado.

Referências

Burden, Richar L.; Fairies, J. Douglas **Numerical Analysis** 9. ed.