Histórico de Revisão

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versão | Descrição | Autor |
| 10/05/2016 | 0.1 | Início do Documento | Luis Fernando |

Introdução

Este documento mostra os testes de software realizados no programa “Calculadora de Bhaskara”, que tem por objetivo calcular os valores de x (x1 e x2) que resolvem a equação. O programa foi desenvolvido pelo Thiago Alves, usando a linguagem C++ em ambiente DOS. Será aplicado diferentes tipos de testes, visando encontrar possíveis falhas, assim pode-se melhorar no funcionamento do programa após serem corrigidas.

Roteiro de Teste

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Caso de Teste | Dados de Teste | Resultado Esperado | Resultado Real | Status |
| 01 | Calcular Bhaskara | A = 1 ; B = -4; C = -5; | X’ = 5 ;  X’’ = -1; | X’ = 5;  X’’ = 0; | Resultado Incompatível |
| 02 | Verificar Erro | A = 4; B = 8; C= 6; | Mensagem de Erro | Mensagem de Erro | Resultado Compatível |
| 03 | Calcular Delta = 0 | A = 1; B = 2; C = 1; | X’ = -1;  X’’ = -1 | X’ = -1;  X’’ = 0; | Resultado Incompatível |
| 04 | Usando Zero no Calculo | A = 1; B = 0; C = -9 | X’ = 3  X’’ = -3 | X’ = 3;  X’’ = 0; | Resultado Incompatível |
| 05 | Usando Dois Zeros | A = 179; B = 0; C = 0; | X’ = 0;  X’’ = 0; | X’ = 0;  X’’ = 0; | Resultado Compatível |
| 06 | Obter Ajuda | Menu Ajuda | Mensagem de Ajuda | Mensagem de Ajuda | Resultado Compatível |
| 07 | Entrada de caracteres | A = ‘a’, B = ‘b’, C = ‘c’ | Mensagem de Erro | Loop Infinito | Resultado Incompatível |

Estratégias

Para realização dos testes foram utilizadas algumas técnicas.

* A princípio foi utilizado o **teste unitário**, sendo analisado separadamente classe por classe:  
  Delta; ValorX1X2; Mensagem e pôr fim a main.
* Após isso foi feito o **teste de integração**, dessa vez considerando a comunicação de uma classe para outra classe. Se tratando de um programa simples, somente a main se comunicava com todas as outras classes já citadas. Dento da integração foi feito um **teste híbrido**, considerando as técnicas **Bottom-Up e Top-Down**, ou seja, primeiro o ponto de partida foram as classes mais especificas, sendo analisadas até chegar do main. Depois o início do teste foi o main, sendo visto as outras classes quando são chamadas.
* Um outro teste utilizado, foi o conhecido como **ACEITE Beta**. Foi dado o programa para o cliente testar em seu próprio ambiente, assim, ele pode verificar as funcionalidades, e dizer se o programa atende o que foi pedido sem detectar falhas.
* Uma estratégia funcional foi adotada, a **“Error Guessing”**, onde baseado na intuição do testador, foi feito testes com valores que tendem a dar problemas.