**Universidade Estadual de Campinas**

**Plano de Testes**

Tales Abdo dos Santos

**Limeira**

**2016**

|  |  |  |  |
| --- | --- | --- | --- |
| HISTÓRICO DE REVISÕES | | | |
| Revisão | Data | Descrição | Autor |
| 01 | 24/05/16 | Elaboração da primeira versão do documento. | Tales Abdo dos Santos |
| 02 | 31/05/16 | Mudança do Roteiro de Teste. | Tales Abdo dos Santos |
| 03 | 01/06/16 | Mudança do Roteiro de Teste. | Tales Abdo dos Santos |
| 04 | 06/06/16 | Estratégias e Introdução | Tales Abdo dos Santos |

**Sumário**

[**1.** **Introdução** 1](#_Toc453240549)

[**2.** **Roteiro de Teste** 2](#_Toc453240550)

[**3.** **Estratégias** 5](#_Toc453240551)

[3.1 Critérios de cobertura funcional 5](#_Toc453240552)

[3.2 Critérios de cobertura estrutural 5](#_Toc453240553)

[3.3 Teste unitário 5](#_Toc453240554)

[3.4 Teste de Integração 5](#_Toc453240555)

[3.5 Teste de Sistema 5](#_Toc453240556)

[3.6 Teste de Aceite 5](#_Toc453240557)

# **Introdução**

O software calculadora tem como objetivo fornecer operações matemáticas precisas, portanto é necessário saber a confiabilidade destes resultados. O objetivo dos testes é descobrir se os resultados fornecidos pelo software estão de acordo com os resultados reais. O software foi desenvolvido e testado na plataforma Dev C++, versão 5.11, Criado pela empresa Bloodsheed. O encarregado pelos testes é o programador Tales Abdo dos Santos.

# **Roteiro de Teste**

|  |  |
| --- | --- |
| ID: | 1 |
| Caso de teste: | Soma de 2 valores positivos. |
| Dados de teste: | 1.5 , +, 2.3 |
| Resultado esperado: | 3.8 |

|  |  |
| --- | --- |
| ID: | 2 |
| Caso de teste: | Soma de 1 valor positivo e 1 negativo. |
| Dados de teste: | (-1.5) , +, 2.3 |
| Resultado esperado: | 0.8 |

|  |  |
| --- | --- |
| ID: | 3 |
| Caso de teste: | Soma de 2 valores negativos. |
| Dados de teste: | (-1.5) , +, (-2.3) |
| Resultado esperado: | -3.8 |

|  |  |
| --- | --- |
| ID: | 4 |
| Caso de teste: | Subtração de 2 valores positivos. |
| Dados de teste: | 3.2 , -, 2.1 |
| Resultado esperado: | 1.1 |

|  |  |
| --- | --- |
| ID: | 5 |
| Caso de teste: | Subtração de 1 valor positivo e 1 negativo. |
| Dados de teste: | 4.5 , -, (-8.6) |
| Resultado esperado: | 13.1 |

|  |  |
| --- | --- |
| ID: | 6 |
| Caso de teste: | Subtração de 2 valores negativos. |
| Dados de teste: | (-13.1) , -, 5.8 |
| Resultado esperado: | -18.9 |

|  |  |
| --- | --- |
| ID: | 7 |
| Caso de teste: | Multiplicação de 2 valores positivos. |
| Dados de teste: | 2.4 , \*, 3.7 |
| Resultado esperado: | 8.88 |

|  |  |
| --- | --- |
| ID: | 8 |
| Caso de teste: | Multiplicação de 1 valor positivo e 1 negativo. |
| Dados de teste: | (-1.2) , \*, 2.6 |
| Resultado esperado: | -3.12 |

|  |  |
| --- | --- |
| ID: | 9 |
| Caso de teste: | Multiplicação de 2 valores negativos. |
| Dados de teste: | (-1.5) , \*, (-2.1) |
| Resultado esperado: | 3.15 |

|  |  |
| --- | --- |
| ID: | 10 |
| Caso de teste: | Divisão de 2 valores positivos. |
| Dados de teste: | 2.5 , /, 3.4 |
| Resultado esperado: | 0.735294 |

|  |  |
| --- | --- |
| ID: | 11 |
| Caso de teste: | Divisão de 1 valor positivo e 1 negativo. |
| Dados de teste: | (-2.4) , /, 1.7 |
| Resultado esperado: | 1.411764 |

|  |  |
| --- | --- |
| ID: | 12 |
| Caso de teste: | Divisão de 2 valores negativos. |
| Dados de teste: | (-1.25) , /, (-2.8) |
| Resultado esperado: | 3.5 |

|  |  |
| --- | --- |
| ID: | 13 |
| Caso de teste: | Potenciação. |
| Dados de teste: | 2 , ^, 3 |
| Resultado esperado: | 8 |

|  |  |
| --- | --- |
| ID: | 14 |
| Caso de teste: | Potenciação com números reais. |
| Dados de teste: | 2.1 , ^, 3.3 |
| Resultado esperado: | Limite ultrapassado |

|  |  |
| --- | --- |
| ID: | 15 |
| Caso de teste: | Caractere no número 1 |
| Dados de teste: | D |
| Resultado esperado: | Primeiro valor não aceito. Por favor, Digite o primeiro valor: |

|  |  |
| --- | --- |
| ID: | 16 |
| Caso de teste: | Caractere não válido na operação |
| Dados de teste: | 1, t |
| Resultado esperado: | Não foi digitado uma operação válida. Por favor, Digite a operação: |

|  |  |
| --- | --- |
| ID: | 17 |
| Caso de teste: | Caractere no número 2 |
| Dados de teste: | 1, +, e |
| Resultado esperado: | Segundo valor não aceito. Por favor, Digite o segundo valor: |

|  |  |
| --- | --- |
| ID: | 18 |
| Caso de teste: | Potenciação com número negativo. |
| Dados de teste: | -2 , ^, 3 |
| Resultado esperado: | Não é aceito números negativos na potenciação. |

|  |  |
| --- | --- |
| ID: | 19 |
| Caso de teste: | Potenciação com base ultrapassada. |
| Dados de teste: | 21 , ^, 3 |
| Resultado esperado: | Limite da base é 20. |

|  |  |
| --- | --- |
| ID: | 20 |
| Caso de teste: | Potenciação com expoente ultrapassado. |
| Dados de teste: | 2 , ^, 11 |
| Resultado esperado: | Limite do expoente é 10. |

# **Estratégias**

## 3.1 Critérios de cobertura funcional

Para a operação de potenciação foi abordado o critério de análise do valor limite, ou seja, analisar os extremos aceitos, pois esta operação possui limites em seus valores.

Para todas as outras operações, adição, subtração, multiplicação e divisão, foi usado particionamento de equivalência. Portanto foi usado combinações de números positivos, negativos e reais para testar todas estas operações.

Além disso, foi testado todos os valores que o usuário deve inserir com “Error Guessing”, ou seja, tanto nos dois números da conta tanto no símbolo da operação foi inserido valores não válidos, como letras e símbolos não aceitos.

## 3.2 Critérios de cobertura estrutural

Para testar corretamente o código foi avaliado e testado todos os caminhos possíveis que a execução pode tomar.

## 3.3 Teste unitário

Todas as funções foram testadas isoladamente, ou seja, foi verificado se cada função cumpre o seu papel.

Como critério para o teste unitário foi avaliado os possíveis valores que as variáveis “valor1” e “valor” podem receber. Foi testado as quatro operações básicas que estão no software com conjuntos de números positivos, negativos e a mistura dos dois. A operação de potência foi testada com análise de seus limites.

## 3.4 Teste de Integração

Todas as funções matemáticas possuem códigos próprios. Estes códigos se ligam com a função.

Para verificar se as integrações do sistema estão funcionando como deveriam foi inserido valores inválidos e valores válidos. No caso dos valores inválidos é exibida uma mensagem relatando erro e onde ele aconteceu. No caso de ultrapassagem do limite dos valores da potenciação é exibido um aviso ao usuário. Os testes procuram cobrir todas os caminhos possíveis no software.

## 3.5 Teste de Sistema

O software em si funciona conforme suas especificações e produz os resultados necessários.

## 3.6 Teste de Aceite

Foi colocado uma versão de teste nas máquinas em que o software será usado e tudo ocorreu sem problemas. O diretor Matheus Evers executou o programa e inseriu valores válidos e valores inválidos e para os dois tipos de valores os resultados foram os esperados.