PLANO DE TESTE

Histórico de Revisão

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versão | Descrição | Autor |
| 01/06/2016 | 0.1 | • Formatação. | Guilherme Bornia |
| 03/06/2016 | 0.2 | • Preenchimento. | Guilherme Bornia |

Sumário

[1. Introdução 4](#_Toc452747445)

[2. Roteiro de Teste 5](#_Toc452747446)

[2.1. Teste 1: Opções do Usuário 5](#_Toc452747447)

[2.2. Teste 2: Limite de Caracteres 5](#_Toc452747448)

[2.3. Teste 3: Deixando Maiúscula 5](#_Toc452747449)

[2.4. Teste 4: Retirando os Espaços 6](#_Toc452747450)

[2.5. Teste 5: Deixando Maiúscula se For a Primeira Letra 6](#_Toc452747451)

[2.6. Teste 6: Main 6](#_Toc452747452)

[3. Estratégias 7](#_Toc452747453)

# Introdução

Neste documento serão realizados testes para garantir a qualidade do software, ou seja, “o grau no qual um conjunto de características inerentes satisfaz requisitos” (NBR ISSO 9000:2005). O objetivo é verificar se existe algum defeito que impede o perfeito funcionamento do programa.

Além disso, também será descrita a metodologia do teste e em que ela será baseada, bem como as entradas de dados utilizadas para verificar se o resultado está sendo o esperado.

# Roteiro de Teste

## Teste 1: Opções do Usuário

|  |  |
| --- | --- |
| **ID:** | RR001 |
| **Caso de Teste:** | Opções do Usuário |
| **Dados de Teste:** | {1, 2, -1, 3, a} |
| **Resultado Esperado:** | Se o usuário entrar com as opções válidas 1 ou 2 o programa deve continuar sua execução normalmente ou encerrar-se, respectivamente. Se as opções -1 ou “a” forem digitadas (números negativos ou caracteres) o programa não deve prosseguir ou encerrar-se. Este deve continuar repetindo a tela de opções até que o usuário entre com uma opção válida. |

## Teste 2: Limite de Caracteres

|  |  |
| --- | --- |
| **ID:** | RR002 |
| **Caso de Teste:** | Limite de Caracteres |
| **Dados de Teste:** | {“Era uma vez ... e viveram felizes para sempre.” (508 caracteres) } |
| **Resultado Esperado:** | Como o limite de caracteres é de 500 então o texto, após ser processado, ficaria assim: “Era uma vez ... e viveram felizes para”. |

## Teste 3: Deixando Maiúscula

|  |  |
| --- | --- |
| **ID:** | RR003 |
| **Caso de Teste:** | Deixando Maiúscula |
| **Dados de Teste:** | {“teste para Ver se Esta funcionando\_1. ”} |
| **Resultado Esperado:** | {“teste Para Ver Se Esta Funcionando\_1. ”} |

## Teste 4: Retirando os Espaços

|  |  |
| --- | --- |
| **ID:** | RR004 |
| **Caso de Teste:** | Retirando os Espaços |
| **Dados de Teste:** | {“teste para Ver se Esta funcionando\_1. ”} |
| **Resultado Esperado:** | {“testeparaVerseEstafuncionando\_1. ”} |

## Teste 5: Deixando Maiúscula se For a Primeira Letra

|  |  |
| --- | --- |
| **ID:** | RR005 |
| **Caso de Teste:** | Deixando Maiúscula se For a Primeira Letra |
| **Dados de Teste:** | {“teste\_1. ”} |
| **Resultado Esperado:** | {“Teste\_1. ”} |

## Teste 6: Main

|  |  |
| --- | --- |
| **ID:** | RR006 |
| **Caso de Teste:** | Main |
| **Dados de Teste:** | {“teste para Ver se Esta funcionando\_1. ”} |
| **Resultado Esperado:** | {“TesteParaVerSeEstaFuncionando. ”} |

# Estratégias

Para descobrir a presença de defeitos foi utilizado o critério de Error-Guessing onde, por meio da experiência e intuição, os analistas supõem tipos prováveis de erros.

Supôs-se que o tratamento de caracteres não havia sido implementado e, por isso, letras foram inseridas, bem como caracteres especiais como o “\_” (underline), em menus onde só os números se mostravam válidos.

E também, para verificar se o software está satisfazendo os requisitos, foram digitados textos com e sem espaços além letras maiúsculas e minúsculas tanto logo no início quanto no final.