<Exercicio 1 Teste caixa preta>

Test Case outline

Version <x.0>

<dd/mm/yyyy>

**Histórico de Versões**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version #** | **Implemented**  **By** | **Revision**  **Date** | **Approved**  **By** | **Approval**  **Date** | **Reason** |
| 1.0 | *<Miguel Tavares>* | *<18/11/2023>* | *<name>* | *<mm/dd/yy>* | draft |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Tabela de Conteúdos

[1. Introdução 4](#_Toc24302914)

[1.1. Identificador do documento 4](#_Toc24302915)

[1.2. Âmbito 4](#_Toc24302916)

[1.3. References 4](#_Toc24302917)

[1.4 Glossário 4](#_Toc24302918)

[2. Detalhes 4](#_Toc24302919)

[2.1. Identificador do caso de teste 4](#_Toc24302920)

[2.2. Objetivo ou requisito/funcionalidade associado 4](#_Toc24302921)

[2.3. Inputs 4](#_Toc24302922)

[2.6. Estado inicial ou pré-condições 4](#_Toc24302923)

[2.5. Resultados esperados 4](#_Toc24302924)

[2.6. Procedimentos especiais ou STEPS 4](#_Toc24302925)

[2.7. Dependência de outros casos 4](#_Toc24302926)

# 1. Introdução

## 1.1. Identificador do documento

[Fornecer um identificador único para este documento. Este identificador único será usado por outros documentos para referenciar este de forma unívoca.]

## 1.2. Âmbito

## 1.3. References

[Fazer referência a outros documentos importantes para uma melhor compreensão deste. Exemplo: documento de requisitos, Master Test Plan, …]

## 1.4 Glossário

# 2. Detalhes

[cada uma das seções abaixo deverá ser repetida para cada test case. Em alternativa poderá usar-se uma tabela. Uma das principais diferenças entre as seções 2.3 e 2.4 do documento Test Case Design e o test case em si, está diretamente relacionado com os dados, ou seja, este documento pretende apresentar os dados de acordo com as descrições definidas no âmbito do documento test case design]

## 2.1. Identificador do caso de teste

## 2.2. Objetivo ou requisito/funcionalidade associado

## 2.3. Inputs

## 2.6. Estado inicial ou pré-condições

## 2.5. Resultados esperados

## 2.6. Procedimentos especiais ou STEPS

## 2.7. Dependência de outros casos

Alternativamente poderá ser considerado a apresentação de cada caso no formato de tabela, facilitando a sua leitura e monitorização e revisão dos mesmos.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | TC001 | **Dependências** |  | |
|  |  |  |  | |
|  | | | | |
| **Pré-condições / estado inicial** | | | | |
| Nenhum triângulo foi criado | | | | |
| **Procedimento / Passos do *user* para esta função** | | | | |
| 1.Criar um triângulo  2.  (...) | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valores do Inputs** | **Resultados Esperados** | **Pass/Fail/Untested** |
| Verificar se é possível | Lados do triângulo | 3, 4, 5 | Espera-se que seja retornado true |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | TC002 | **Dependências** |  | |
|  |  |  |  | |
|  | | | | |
| **Pré-condições / estado inicial** | | | | |
| Nenhum triângulo foi criado | | | | |
| **Procedimento / Passos do *user* para esta função** | | | | |
| 1.Criar um triângulo  2.  (...) | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valores do Inputs** | **Resultados Esperados** | **Pass/Fail/Untested** |
| Verificar se é possível | Lados do triângulo | 20, 0, 10 | Deve ser retornado true |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | TC003 | **Dependências** |  | |
|  |  |  |  | |
|  | | | | |
| **Pré-condições / estado inicial** | | | | |
| Nenhum triângulo foi criado | | | | |
| **Procedimento / Passos do *user* para esta função** | | | | |
| 1.Criar um triângulo  2.  (...) | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valores do Inputs** | **Resultados Esperados** | **Pass/Fail/Untested** |
| Verificar se é escaleno | Lados do triângulo | 3, 5, 4 | Deve ser retornado true, uma vez que o triângulo é escaleno |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | TC004 | **Dependências** |  | |
|  |  |  |  | |
|  | | | | |
| **Pré-condições / estado inicial** | | | | |
| Nenhum triângulo foi criado | | | | |
| **Procedimento / Passos do *user* para esta função** | | | | |
| 1.Criar um triângulo  2.  (...) | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valores do Inputs** | **Resultados Esperados** | **Pass/Fail/Untested** |
| Verificar se o triângulo é retângulo | Lados do triângulo | 3, 5, 4 | Deve ser retornado true, uma vez que o triângulo é retângulo e satizfaz o teorema de pitágoras |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | TC005 | **Dependências** |  | |
|  |  |  |  | |
|  | | | | |
| **Pré-condições / estado inicial** | | | | |
| Nenhum triângulo foi criado | | | | |
| **Procedimento / Passos do *user* para esta função** | | | | |
| 1.Criar um triângulo  2.  (...) | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valores do Inputs** | **Resultados Esperados** | **Pass/Fail/Untested** |
| Verificar se o triângulo é retângulo | Lados do triângulo | 1, 2, 3 | Deve ser retornado false, uma vez que o triângulo não é retângulo |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | TC006 | **Dependências** |  | |
|  |  |  |  | |
|  | | | | |
| **Pré-condições / estado inicial** | | | | |
| Nenhum triângulo foi criado | | | | |
| **Procedimento / Passos do *user* para esta função** | | | | |
| 1.Criar um triângulo  2.  (...) | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valores do Inputs** | **Resultados Esperados** | **Pass/Fail/Untested** |
| Verificar se o triângulo é equilátero | Lados do triângulo | 5, 5, 5 | Deve ser retornado true, uma vez que o triângulo é equilátero |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | TC007 | **Dependências** |  | |
|  |  |  |  | |
|  | | | | |
| **Pré-condições / estado inicial** | | | | |
| Nenhum triângulo foi criado | | | | |
| **Procedimento / Passos do *user* para esta função** | | | | |
| 1.Criar um triângulo  2.  (...) | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valores do Inputs** | **Resultados Esperados** | **Pass/Fail/Untested** |
| Verificar se o triângulo é equilátero | Lados do triângulo | 3, 4, 5 | Deve ser retornado false, uma vez que o triângulo não é equilátero |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | TC008 | **Dependências** |  | |
|  |  |  |  | |
|  | | | | |
| **Pré-condições / estado inicial** | | | | |
| Nenhum triângulo foi criado | | | | |
| **Procedimento / Passos do *user* para esta função** | | | | |
| 1.Criar um triângulo  2.  (...) | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valores do Inputs** | **Resultados Esperados** | **Pass/Fail/Untested** |
| Verificar se o triângulo é isósceles | Lados do triângulo | 4, 4, 5 | Deve ser retornado true, uma vez que o triângulo é isósceles |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | TC009 | **Dependências** |  | |
|  |  |  |  | |
|  | | | | |
| **Pré-condições / estado inicial** | | | | |
| Nenhum triângulo foi criado | | | | |
| **Procedimento / Passos do *user* para esta função** | | | | |
| 1.Criar um triângulo  2.  (...) | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valores do Inputs** | **Resultados Esperados** | **Pass/Fail/Untested** |
| Verificar se o triângulo é isósceles | Lados do triângulo | 3, 4, 5 | Deve ser retornado false, uma vez que o triângulo não é isósceles |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | TC011 | **Dependências** |  | |
|  |  |  |  | |
|  | | | | |
| **Pré-condições / estado inicial** | | | | |
| Nenhum triângulo foi criado | | | | |
| **Procedimento / Passos do *user* para esta função** | | | | |
| 1.Criar um triângulo  2.  (...) | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valores do Inputs** | **Resultados Esperados** | **Pass/Fail/Untested** |
| Classificar um triângulo | Lados do triângulo | 9, 3, 3 | Deve ser retornado “isosceles”, uma vez que o triângulo é isósceles |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | TC012 | **Dependências** |  | |
|  |  |  |  | |
|  | | | | |
| **Pré-condições / estado inicial** | | | | |
| Nenhum triângulo foi criado | | | | |
| **Procedimento / Passos do *user* para esta função** | | | | |
| 1.Criar um triângulo  2.  (...) | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valores do Inputs** | **Resultados Esperados** | **Pass/Fail/Untested** |
| Classificar um triângulo | Lados do triângulo | 3, 4, 5 | Deve ser retornado “escaleno”, uma vez que o triângulo é escaleno |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | TC013 | **Dependências** |  | |
|  |  |  |  | |
|  | | | | |
| **Pré-condições / estado inicial** | | | | |
| Nenhum triângulo foi criado | | | | |
| **Procedimento / Passos do *user* para esta função** | | | | |
| 1.Criar um triângulo  2.  (...) | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valores do Inputs** | **Resultados Esperados** | **Pass/Fail/Untested** |
| Classificar um triângulo | Lados do triângulo | 3, 3, 4 | Deve ser retornado “isosceles”, uma vez que o triângulo é isósceles |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | TC014 | **Dependências** |  | |
|  |  |  |  | |
|  | | | | |
| **Pré-condições / estado inicial** | | | | |
| Nenhum triângulo foi criado | | | | |
| **Procedimento / Passos do *user* para esta função** | | | | |
| 1.Criar um triângulo  2.  (...) | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valores do Inputs** | **Resultados Esperados** | **Pass/Fail/Untested** |
| Classificar um triângulo | Lados do triângulo | 7, 24, 25 | Deve ser retornado “scalene”, uma vez que o triângulo é escaleno |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | TC015 | **Dependências** |  | |
|  |  |  |  | |
|  | | | | |
| **Pré-condições / estado inicial** | | | | |
| Nenhum triângulo foi criado | | | | |
| **Procedimento / Passos do *user* para esta função** | | | | |
| 1.Criar um triângulo  2.  (...) | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valores do Inputs** | **Resultados Esperados** | **Pass/Fail/Untested** |
| Classificar um triângulo | Lados do triângulo | 3, 4, 33 | Deve ser retornado “impossible”, uma vez que o triângulo é impossível |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | TC016 | **Dependências** |  | |
|  |  |  |  | |
|  | | | | |
| **Pré-condições / estado inicial** | | | | |
| Nenhum triângulo foi criado | | | | |
| **Procedimento / Passos do *user* para esta função** | | | | |
| 1.Criar um triângulo  2.  (...) | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valores do Inputs** | **Resultados Esperados** | **Pass/Fail/Untested** |
| Classificar um triângulo | Lados do triângulo | 3, 4, 33 | Deve ser retornado “impossible”, uma vez que o triângulo é impossível |  |