Relatório de Teste Funcional – Aplicação Triângulo

# 1. Contexto do Teste

Objetivo: Verificar a funcionalidade que classifica triângulos com base nos lados informados, retornando os tipos: equilátero, isósceles ou escaleno.

Além disso, validar a regra matemática que garante que os lados formem um triângulo válido, onde o comprimento de qualquer lado deve ser menor que a soma dos outros dois. Caso contrário, a aplicação deve exibir uma mensagem de erro.

# Ambiente de Testes

|  |  |
| --- | --- |
| Componentes | Detalhes |
| Ferramenta | Robot Framework 7.2 SeleniumLibrary 6.7.1 |
| Ambiente | *http://www.vanilton.net/triangulo/* |
| Endpoints Testados | *http://www.vanilton.net/triangulo/* |

Abordagem: Automação de testes funcionais com testes parametrizados (*Test Template*) para cobrir classificações válidas e validações de entradas inválidas.

# 2. Escopo dos Testes Automatizados

Testes de Classificação Válida

* Triângulo Equilátero: lados iguais (exemplo: 5, 5, 5)
* Triângulo Isósceles: dois lados iguais (exemplo: 5, 5, 3 e variações)
* Triângulo Escaleno: todos os lados diferentes (exemplo: 4, 5, 6)

Testes de Validação de Erro para Entradas Inválidas

* Casos em que a soma de dois lados é igual ou menor que o terceiro (exemplo: 1, 2, 3; 5, 5, 10)
* Lados com valor zero ou negativo
* Entradas não numéricas (exemplo: 'a', 2, 3)

# 3. Cenários de Teste Executados

## Tabela de Cenários

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cenário | Entradas | Resultado Esperado | Resultado Obtido | Status |
| Triângulo Equilátero | 5, 5, 5 | Retorna 'equilátero' | Retornou 'equilátero' | ✅ Passou |
| Triângulo Isósceles (2 lados iguais) | 5, 5, 3 | Retorna 'isósceles' | Retornou 'isósceles' | ✅ Passou |
| Triângulo Isósceles (lados invertidos) | 3, 5, 5 | Retorna 'isósceles' | Retornou 'isósceles' | ✅ Passou |
| Triângulo Escaleno | 4, 5, 6 | Retorna 'escaleno' | Retornou 'escaleno' | ✅ Passou |
| Soma dois lados igual ao terceiro | 1, 2, 3 | Exibe mensagem de erro | Não exibiu mensagem de erro | ❌ Falhou |
| Lado maior que soma dos outros dois | 10, 2, 3 | Exibe mensagem de erro | Não exibiu mensagem de erro | ❌ Falhou |
| Lado igual soma outros dois | 5, 5, 10 | Exibe mensagem de erro | Não exibiu mensagem de erro | ❌ Falhou |
| Todos lados zero | 0, 0, 0 | Exibe mensagem de erro | Não exibiu mensagem de erro | ❌ Falhou |
| Lado negativo | -1, 4, 5 | Exibe mensagem de erro | Não exibiu mensagem de erro | ❌ Falhou |
| Entrada não numérica | a, 2, 3 | Exibe mensagem de erro | Não exibiu mensagem de erro | ❌ Falhou |

# 4. Evidências

* Capturas de tela foram realizadas para todos os testes (disponíveis sob demanda).
* Relatórios gerados automaticamente pelo Robot Framework em formato HTML com logs e screenshots.
* Código fonte dos testes automatizados com testes parametrizados para cobertura completa.

# 5. Observações Importantes

* A aplicação não exibe mensagem de erro adequada quando os lados informados não satisfazem a condição para formar um triângulo válido.
* O comportamento esperado é que o sistema alerte o usuário (ex: mensagem "Erro: não forma triângulo válido" ou equivalente).
* Por conta disso, os testes automatizados de validação para entradas inválidas falharam.
* Os testes de classificação para triângulos válidos passaram conforme esperado.

# 6. Recomendação de Correção para a Aplicação

* Implementar uma validação no front-end ou back-end para verificar a regra do triângulo (lado < soma dos outros dois).
* Apresentar mensagem de erro clara e visível quando a condição não for satisfeita.
* Validar também entradas não numéricas, zeros e valores negativos com feedback adequado ao usuário.

# 7. Conclusão

O sistema atende corretamente a classificação dos triângulos válidos, porém falta o tratamento e feedback de erros para dados inválidos, o que é essencial para garantir a robustez da aplicação e a experiência do usuário.

O teste automatizado detectou essa falha, cumprindo seu papel de garantir qualidade e apontar problemas.