**Título do Plano de Testes: Plano de Testes Unitário para o Sistema de Calculadora**

**Data de Criação**: 03/05/2024

**Autor:** Lucas Camargo de Lima

Ismael Duarte

**Objetivo:** Este plano de testes tem como objetivo definir as estratégias, processos e recursos necessários para a realização de testes unitários do Sistema de Calculadora.

**Escopo:** O escopo deste plano de testes inclui a verificação da funcionalidade de todas as funcionalidades do sistema Sistema de Calculadora.

**Ambiente de Teste:**

- Sistema Operacional: Windows 10

- Versão do Sistema: v1.0

**Data de Início:** 03/05/2024

**Data de Conclusão:** 03/05/2024

**Equipe de Teste:**

- Líder de Teste: Ismael Duarte

- Engenheiro de Teste 1: Lucas Camargo de Lima

**Tipos de Teste:**

- Teste Unitário

**Processo de Teste:**

1. Preparação do Ambiente de Teste

2. Preparação dos Casos de Teste

3. Execução dos Casos de Teste

**Ferramentas de Teste:**

- Ferramenta de Teste Unitário: Jasmine

**Aprovação:**

- Aprovação do Líder de Teste: Lucas C Lima

- Aprovação do Gerente de Projeto: Lucas C Lima

**Título do Caso de Teste: Verificar a funcionalidade calculadora**

**ID do Caso de Teste:** TC-001 – Somas Válidas

**Propósito:** Verificar se a função soma está fazendo o cálculo de forma correta.

**Pré-condições:**

- O sistema deve estar ativo e disponível.

**Passos de Execução:**

1. Escolher função soma.

2. Informar o Primeiro Número.

3. Informar o Segundo Número.

**Resultado Esperado:**

- Somar os números 1 e 2 e retornar 3.

**Desenho de teste Jasmine - Calculadora**

describe("Teste Calculadora", function () {

it("Somas válidas", function () {

expect(soma(1, 2)).toBe(3);

expect(soma(9, 9)).toBe(18);

});

it("Somas inválidas", function () {

expect(soma(1, 2)).not.toBe(5);

expect(soma(9, 9)).not.toBe(15);

});

it("Subtrações válidas", function () {

expect(subtracao(3, 2)).toBe(1);

expect(subtracao(9, 9)).toBe(0);

});

it("Multiplicações válidas", function () {

expect(multiplicacao(1, 2)).toBe(2);

expect(multiplicacao(9, 9)).toBe(81);

});

it("Divisões válidas", function () {

expect(divisao(2, 2)).toBe(1);

expect(divisao(18, 9)).toBe(2);

});

});