**Título do Plano de Testes: Nome do Plano**

**Data de Criação**: 11/06/2024

**Autor:** Arthur Santos de Araújo

**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 (Soma, Subtração, Divisão e Multiplicação) e Sistema de Maioridade.

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

**Ambiente de Teste:**

- Sistema Operacional: Windows 11

- Versão do Sistema: 23H2

**Data de Início:** 11/06/2024

**Data de Conclusão:** 11/06/2024

**Equipe de Teste:**

- Líder de Teste: Arthur Santos de Araújo

- Engenheiro de Teste 1: Nome do Engenheiro 1

- Engenheiro de Teste 2: Nome do Engenheiro 2

**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: Arthur Santos de Araújo

- Aprovação do Gerente de Projeto: Tutor Rodrigo Jesus

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

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

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

**Pré-condições:**

- O sistema deve estar ativo e disponível.

**Passos de Execução:**

1. Escolher função multiplicação.

2. Informar o Primeiro Número.

3. Informar o Segundo Número.

**Resultado Esperado:**

- Multiplicar os números 2 e 3 e retornar 6.  
- Multiplicar os números 5 e 5 e retornar 25.

**Desenho de teste Jasmine**

        it("Multiplicação válidas", function(){

            expect(multiplicacao(2,3)).toBe(6);

            expect(multiplicacao(5,5)).toBe(25);

        });

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

**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.  
- Somar os números 9 e 9 e retornar 18.

**Desenho de teste Jasmine**

        it("Somas válidas", function(){

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

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

        });

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

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

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

**Pré-condições:**

- O sistema deve estar ativo e disponível.

**Passos de Execução:**

1. Escolher função subtração.

2. Informar o Primeiro Número.

3. Informar o Segundo Número.

**Resultado Esperado:**

- Subtração os números 1 e 2 e retornar 3.  
- Subtração os números 1 e 2 e retornar 3.

**Desenho de teste Jasmine**

        it("Subtrações válidas", function(){

            expect(subtracao(5,2)).toBe(3);

            expect(subtracao(10,5)).toBe(5);

        });

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

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

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

**Pré-condições:**

- O sistema deve estar ativo e disponível.

**Passos de Execução:**

1. Escolher função divisão.

2. Informar o Primeiro Número.

3. Informar o Segundo Número.

**Resultado Esperado:**

- Dividir os números 10 e 2 e retornar 5.  
- Dividir os números 20 e 4 e retornar 5.

**Desenho de teste Jasmine**

        it("Divisão válidas", function(){

            expect(divisao(10,2)).toBe(5);

            expect(divisao(20,4)).toBe(5);

        });

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

**ID do Caso de Teste:** TC-001 – Maioridade

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

**Pré-condições:**

- O sistema deve estar ativo e disponível.

**Passos de Execução:**

1. Escolher função maioridade.

2. Informar a Idade.

**Resultado Esperado:**

- Informar a idade de 18 anos e retornar ‘Maior Idade’.

**Desenho de teste Jasmine**

describe("Teste de validação - Maior Idade", function(){

    it("Validação Maior idade", function(){

    expect(maioridade(18)).toBe('Maior Idade');

    });

    });

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

**ID do Caso de Teste:** TC-001 – Maioridade

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

**Pré-condições:**

- O sistema deve estar ativo e disponível.

**Passos de Execução:**

1. Escolher função maioridade.

2. Informar a Idade.

**Resultado Esperado:**

- Informar a idade de 18 anos e retornar ‘Maior Idade’.  
- Informar a idade de 10 anos e retornar ‘Menor Idade’.

**Desenho de teste Jasmine**

describe("Teste de validação - Maior Idade", function(){

    it("Validação Maior idade", function(){

    expect(maioridade(18)).toBe('Maior Idade');

    });

    });

describe("Teste de validação - Menor de Idade", function(){

    it("Validação Menor Idade", function(){

    expect(maioridade(10)).toBe('Menor Idade');

    });

  });