



**UNIVERSIDADE FEDERAL DO CEARÁ - CAMPUS QUIXADÁ**  
**CURSO DE ENGENHARIA DE SOFTWARE**  
**DISCIPLINA: PROJETO INTEGRADO EM ENGENHARIA DE SOFTWARE II**  
**DOCENTE: PROFA. ANTONIA DIANA BRAGA NOGUEIRA**

**DOCUMENTO DE TESTES**

*nome da solução*

# MOTORA

*área temática*

Educação / Saúde

**Equipe:**

542155 – BRENO GONZAGA DE CARVALHO  
494296 – WESLEY LESSA PINHEIRO

**QUIXADÁ**  
**2024**

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>3</b>
<b>2 TIPOS DE TESTES.....</b>	<b>3</b>
<b>2.1 Testes Unitários.....</b>	<b>3</b>
<b>2.2 Testes de Usabilidade.....</b>	<b>3</b>
<b>3 METODOLOGIA DOS TESTES.....</b>	<b>4</b>
<b>3.1 Testes Unitários.....</b>	<b>4</b>
<b>3.2 Testes de Usabilidade.....</b>	<b>6</b>
3.2.1 Amostragem.....	6
3.2.2 Ambiente de teste.....	6
3.2.3 Lista de tarefas.....	6
3.2.4 Relatório de testes.....	7

# 1 INTRODUÇÃO

Este documento tem por finalidade descrever o conjunto de testes aplicados em meio ao processo de desenvolvimento da solução denominada Matora, versão Web, desenvolvida no íterim da disciplina de Projeto Integrado em Engenharia de Software II, ministrada pela Prof. Dra. Diana Braga.

A elaboração desta documentação se faz importante para que seja de conhecimento da equipe as especificidades relacionadas aos diferentes tipos de testes que serão conduzidos visando a melhoria da solução ora desenvolvida.

Este documento encontra-se dividido em três partes: esta breve Introdução; a descrição dos tipos de testes que serão realizados e a metodologia que será aplicada no contexto dos testes.

## 2 TIPOS DE TESTES

### 2.1 Testes Unitários

Os testes unitários visam garantir que funções ou métodos individuais funcionem conforme o esperado. Cada teste deve possuir seu foco em uma única unidade de funcionalidade, verificando tanto os casos de uso normais quanto os extremos.

### 2.2 Testes de Usabilidade

Os testes de usabilidade são projetados para avaliar a facilidade com que os usuários (Professores de Educação Física, no caso da solução) podem interagir com uma aplicação ou sistema, garantindo que a interface do usuário seja intuitiva, eficiente e agradável de usar. Por meio da execução desses testes, a equipe espera identificar problemas que podem não ser evidentes para os desenvolvedores, mas que podem afetar a experiência do usuário final.

Adiante, descreveremos as metodologias que foram utilizadas para o desenvolvimento e/ou condução dos testes unitários e de usabilidade pela equipe.

## 3 METODOLOGIA DOS TESTES

Nesta seção, descrever-se-á os procedimentos metodológicos definidos para cada um dos tipos de testes desenvolvidos para a aplicação.

### 3.1 Testes Unitários

Os testes unitários foram organizados seguindo um padrão de estrutura de diretórios, exemplificados abaixo:

Figura 1 - Exemplo de organização de estrutura de teste para o diretório com as classes moduladas de classificação dos testes físicos.

```
src/  
├── pages/  
│   ├── Avaliation/  
│   │   ├── Classification/  
├── tests/  
│   ├── pages/  
│   │   ├── Avaliation/  
│   │   │   ├── Classification/
```

Fonte: Elaboração própria dos autores.

Cada arquivo de teste contém uma série de casos que verificam as funcionalidades das respectivas funções. Os testes seguem a convenção *describe* para agrupar testes relacionados e *it* ou *test* para definir casos de teste específicos, conforme pressuposto para o JavaScript, linguagem na qual o sistema é baseado.

Na Figura 2 é apresentado um exemplo de teste unitário, a Classificação de IMC. Conforme visualizado, é descrita a função testada (*classificarIMC*) que recebe como parâmetros o valor do IMC (Índice de Massa Corporal), idade e sexo e retorna a classificação do indivíduo com base em tabelas predefinidas. O respectivo teste pode ser encontrado no arquivo “BMI.test.js”.

Figura 2 - Exemplo de teste unitário para função de classificação do IMC.

```
describe('Testes para a função classificarIMC', () => {

  test('Debe classificar como "Zona Saudável" para Rapazes com IMC abaixo do ponto de corte', () => {

    expect(classificarIMC(17.5, 10, 'Rapazes')).toBe('Zona Saudável');

  });

  test('Debe classificar como "Zona de Risco à Saúde" para Rapazes com IMC acima do ponto de corte', () => {

    expect(classificarIMC(24.5, 16, 'Rapazes')).toBe('Zona de Risco à Saúde');

  });

  test('Debe classificar como "Zona Saudável" para Moças com IMC abaixo do ponto de corte', () => {

    expect(classificarIMC(19.0, 12, 'Moças')).toBe('Zona Saudável');

  });

  test('Debe classificar como "Zona de Risco à Saúde" para Moças com IMC acima do ponto de corte', () => {

    expect(classificarIMC(25.0, 14, 'Moças')).toBe('Zona de Risco à Saúde');

  });

  test('Debe retornar mensagem de erro para idade fora do intervalo permitido', () => {

    expect(classificarIMC(20, 5, 'Rapazes')).toBe("Idade fora do intervalo permitido (6-17 anos)");

  });

});
```

Fonte: Elaboração própria dos autores.

Visando a otimização dos testes, a equipe adotou algumas boas práticas, quais sejam:

- **Isolamento dos Testes:** Cada teste deveria ser independente dos outros, permitindo que as falhas fossem rastreadas de forma precisa e o impacto pudesse ser limitado a casos específicos.

- **Cobertura de Casos Extremos:** Além dos casos comuns, os testes também cobriram, sempre que necessário, casos de borda, como valores mínimos e máximos possíveis para os parâmetros, principalmente a idade.
- **Comentários Descritivos:** Caso o teste possuísse uma lógica mais complexa, deveriam ser adicionados comentários explicando o motivo do teste ou descrevendo melhor o comportamento esperado.
- **Clareza nos Nomes dos Testes:** A equipe se certificou de que os nomes dos testes descrevessem claramente o que estava sendo testado e o comportamento esperado.

## 3.2 Testes de Usabilidade

### 3.2.1 Amostragem

Espera-se que participem do teste, no mínimo, 4 (quatro) participantes. Apesar de ser um teste de caráter menos formal, a equipe espera que o quantitativo estabelecido seja capaz de expor as principais deficiências de usabilidade da solução.

### 3.2.2 Ambiente de teste

Os testes serão realizados remotamente por meio da plataforma *Google Meet*. O avaliador utilizará de seu próprio equipamento, um computador *desktop*, com acesso à internet e com o software *StreamLabs OBS* instalado visando a gravação do teste. O usuário, por sua vez, utilizará do seu próprio dispositivo para execução do teste, que permitirá identificar variações comportamentais na solução com base na resposta a diferentes dispositivos e configurações, gerando insumos importantes para a equipe.

### 3.2.3 Lista de tarefas

O avaliador seguirá um roteiro de teste estruturado em uma lista de tarefas que contemplará todas as principais funcionalidades da solução desenvolvida. Para cada tarefa, o avaliador irá cronometrar o tempo necessário para a sua conclusão pelo participante, permitindo-se agregar informações comparativas para análise de performance. A lista com as tarefas que serão executadas está disposta abaixo:

Tarefa
Realizar cadastro.

Solicitar recuperação de senha.
Realizar login.
Cadastrar uma nova turma.
Editar uma turma.
Excluir uma turma.
Acessar uma turma.
Cadastrar um novo aluno.
Editar um aluno.
Excluir um aluno.
Acessar um aluno.
Adicionar resultados de testes.
Modificar resultados de testes.
Acessar as classificações dos testes.
Redigir um feedback.

### 3.2.4 Relatório de testes

Os dados obtidos por meio dos testes com os Professores de Educação Física serão apresentados em relatório específico, apresentado separadamente a esta documentação.