

N704-PROGRAMAÇÃO FUNCIONAL

ÍNDICE DE MASSA CORPORAL

Fabiano da Silva Tavares / 2222829

2024

ÍNDICE

Autor	-03
1. Introdução	-03
2. Visão Geral da Arquitetura	-03
3. Requisitos de Sistema	-04
4. Casos de teste	-11
5. Implantação e operações	-15
6. Código Fonte	-16
Bibliografia	-16

AUTOR

Fabiano da Silva Tavares, matrícula: 2222829 – Curso Analise e desenvolvimento de sistema

1. INTRODUÇÃO

Documentação referente a criação de página web para calculo do índice de massa corporal, com o objetivo de direcionar os trabalhos de desenvolvimento e conclusão do artefato.

Atividade requerida para conclusão da Disciplina Programação Funcional, cursada na Universidade Federal de Fortaleza no segundo semestre de 2024.

2. REQUISITOS DE SISTEMA

Como esclareceu adequadamente o site Até o Momento (2024), a diferença entre requisito funcional e regra de negócio, conceitualmente falando, é que o requisito funcional refere-se à o que o sistema deverá fazer, enquanto a Regra de Negócio refere-se a como o sistema deverá fazer. Do ponto de vista do negócio (negócio do cliente para o qual o sistema está sendo feito), ambos são necessidades (requisito funcional e Regra de Negócio), mas cada uma com um foco diferente. [1]

Por sua vez, segundo o site Estratégia (2024), os requisitos Funcionais especificam “o que o sistema deve fazer”, ou seja, as funcionalidades do sistema. Os requisitos Não Funcionais: Especificam “como o sistema deve operar”, ou seja, abrange aspectos de qualidade e desempenho, restrições e limitações, e aspectos técnicos. [2]

1.1 Regras de Negócio

RN001 – Cálculo do índice de massa corporal

O sistema realiza o cálculo do IMC que consiste na divisão do peso dividido pela altura elevada ao quadrado do usuário, que fornecerá as duas variáveis necessárias para o cálculo.

RN002 – Classificação do IMC em faixa de risco

O sistema fará a classificação do resultado IMC por risco, compreendida em seis faixas, e comunicará como sinal de alerta ao usuário.

- $< 18,5$: Abaixo do peso;
- $\geq 18,5$ e ≤ 25 : Peso ideal;
- > 25 e ≤ 30 : Sobrepeso;
- > 30 e ≤ 35 : Obesidade moderada;
- > 35 e ≤ 40 : Obesidade severa;
- > 40 : Obesidade mórbida;

1.2 Requisitos funcionais

RF001 – Formulário: Cálculo IMC

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

A página inicial do aplicativo apresenta o formulário denominado Cálculo IMC com os campos Peso em Kg e Altura em metros a ser preenchido pelo usuário.

O cálculo é realizado após o preenchimento destes campos seguido do click no botão Calcular.

A captura dos dados é feito pelo html.

```

61     <div className="input-box">
62       <label for="weight">
63         Peso em Kg
64       </label>
65       <div className="input-field">
66         <i class="fa-solid fa-weight-hanging"></i>
67         <input type="number" step=".1" id="weight" required />
68         <span>
69           Kg
70         </span>
71       </div>
72     </div>
73

```

E repassado ao Java Script por este trecho

```

12     const weight = document.getElementById('weight').value;
13     const height = document.getElementById('height').value;
14

```

A função responsável por esta função está neste trecho do código-fonte do arquivo indes.jsx da pasta home.

```

14
15     const bmi = (weight / (height **2)).toFixed(2);

```

Realizamos tentativa de substituir o calculo do IMC (Const bmi) por uma função **Closure**, que funcionou aos testes isoladamente, mas teve êxito ao ser integrada a totalidade do projeto. Mas o desafio foi cumprido na fórmula Home, conforme explicaremos nos Requisitos não funcionais.

```

function calculo() {
  let Height = height**2;
  function imc () {
    (weight/Height).toFixed(2);
  } return mci
}
const bmi = calculo();
bmi();

```

RF002 – Resultado do IMC com alerta e orientações

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Após as envio das informações, o formulário apresenta o resultado do cálculo e a classificação deste resultado, com sinal de alerta.

Ainda na pasta home e arquivo index.jsx, consta o código responsável pela classificação e o alerta.

O resultado é visualizado na página por este trecho do código, que tem ligação com este trecho do html:

JavaScript

```
17   const value = document.getElementById('value');
18   let description = '';
```

Html

```
96   <span id="value"></span>
97   <span>Seu IMC</span>
98 </div>
```

A Função **Lambda** If e Else faz a classificação do resultado nas seis faixas previstas para IMC. A função retorna com uma mensagem (em description) conforme cada faixa. Para a faixa entre 18.5 e 25, é retirada a cor vermelha, Denominada Attention no código.

```

25 // Função Lambda
26
27 (weight == "" || weight == 0) ? (
28   alert("Peso inválido!")
29 ) : (height == "" || height == 0) ? (
30   alert("Altura inválida!")
31 ) : (bmi < 18.5) ? (
32   description = "Cuidado! Você está abaixo do peso!"
33 ) : (bmi >= 18.5 && bmi <= 25) ? (
34   description = "Você está no peso ideal!" &&
35   value.classList.remove('attention') &&
36   value.classList.add('normal')
37 ) : (bmi > 25 && bmi <= 30) ? (
38   description = "Cuidado! Você está com sobrepeso!"
39 ) : (bmi > 30 && bmi <= 35) ? (
40   description = "Cuidado! Você está com obesidade moderada!"
41 ) : (bmi > 35 && bmi <= 40) ? (
42   description = "Cuidado! Você está com obesidade severa!"
43 ) : (description = "Cuidado! Você está com obesidade morbida!")
44 value.textContent = bmi.replace('.', ',');
45 document.getElementById('description').textContent = description;
46 }

```

Além disso, o aplicativo disponibiliza (torna visível) um link que direciona o usuário a uma página com informações sobre o tema, através deste trecho do código-fonte.

```

21
22 document.getElementById('infos').classList.remove('hidden');
23

```

RF003 – Validação dos dados

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

O formulário emitirá uma janela de alerta para valores vazios ou zerados para qualquer dos dois campos. Este trecho JavaScript está inserido na função Lambda If Else citada anteriormente.

```

25 // Função Lambda
26
27 (weight == "" || weight == 0) ? (
28   alert("Peso inválido!")
29 ) : (height == "" || height == 0) ? (
30   alert("Altura inválida!")
31 ) : (bmi < 18.5) ? (
32   description = "Cuidado! Você está abaixo do peso!"

```

1.3 Requisitos não funcionais: Requisitos de Produto, Requisitos Organizacionais e Requisitos Externos

RNF001 – Tipo de Plataforma

O aplicativo foi desenvolvido em Html, JavaScript e CSS e tem por principal destinatário o acesso Web.

No arquivo style.css da pasta home, definimos configurações para compatibilidade com Mobile.

```

178
179 @media screen and (max-width: 1024px) {
180   #img {
181     display: none;
182   }
183 }
184
185 @media screen and (max-width: 500px) {
186   #container {
187     width: 100%;
188     height: 100%;
189   }
190   #calculator{
191     width: 100%;
192     height: 100%;
193     border-radius: 0;
194     justify-content: center;
195   }
196 }
197

```


RNF002 – Funcionamento da Home

A interface está no arquivo index.jsx, que compreende a linguagem JavaScript e Html, com os requisitos funcionais da aplicação

A função Home é uma função **Closure** por ser chamada no arquivo Main.js da pasta src e apresentar o resultado da função interna Calculate.

```
6 // Função Closure
7
8 function Home() {
9
10
11   function calculate() {
12     event.preventDefault();
13
14     const weight = document.getElementById('weight').value;
15     const height = document.getElementById('height').value;
16
```

```
frontend > src > main.jsx
You, 2 hours ago | 1 author (You)
1 import { StrictMode } from 'react'
2 import { createRoot } from 'react-dom/client'
3 import Home from './pages/home'
4 import './index.css'
5
6 createRoot(document.getElementById('root')).render(
7   <StrictMode>
8     <Home />
9   </StrictMode>,
10 )
11
```

RNF003 – Tecnologias

As tecnologias utilizadas para construção do Software: JavaScript, Html, CSS e o Framework React.

RNF004 – Aspectos legais

Os dados pessoais cadastrados devem ser tratado conforme determina a lei LGPD (Lei 13.709 - Proteção de dados). [3]

Por não haver identificação de usuários e solicitação de login, não aplicamos recursos adicionais para proteção de dados dos clientes.

Requisitos do sistema

RS001 Formulário: Cálculo IMC

Referência	[RF001 – Formulário: Cálculo IMC]
Sumário	Módulo para calcular o Índice de Massa Corporal
Pré-condições	Usuário informar seu peso e altura válidos
Atores	Usuários
Descrição	Sessão Home apresenta formulário com campos para preenchimento de peso e altura do usuário. Após o preenchimento com dados válidos, o usuário pressiona o botão Calcular
Alternativas	Indisponível
Exceção	Dados zerados e nulos

RS002 Resultado do IMC com alerta e orientações

Referência	[RF002 – Resultado do IMC com alerta e orientações] [RF001 – Formulário: Cálculo IMC]
Sumário	Tornar visível Resultado IMC, alerta e orientações
Pré-condições	Pressionar o botão
Atores	Usuários
Descrição	Após pressionar o botão calcular, o aplicativo torna visível Resultado IMC, alerta e orientações no formulário de Cálculo do IMC.
Alternativas	Indisponível
Exceção	Em caso de valores informados serem inválidos, o sistema apresentará alerta.

RS003 Validação dos dados

Referência	[RF003 – Validação dos dados] [RF001 – Formulário: Cálculo IMC]
Sumário	Checagem dos valores informados pelo cliente

Pré-condições	Crítica aparecerá para o usuário apenas se o botão Calcular for acionado.
Atores	Usuários
Descrição	Checagem dos valores informados pelo cliente para peso e altura. Em caso de valores zerados, negativos e vazios o sistema emitirá o pop Alert informando o item incorreto.
Alternativas	Indisponível
Exceção	Valores irreais não são detectados pela ferramenta, como altura superior a 2 metros e peso superior a 200 quilos.

4. Casos de teste

Como cita o site Monitoratec (2024) [4], há cinco alternativas para o teste do software:

1. Caixa Branca: teste estrutural que submete o código-fonte do programa à análise;
2. Caixa Preta: teste funcional que avalia as funcionalidades – Visão cliente;
3. Caixa Cinza: Utiliza simultaneamente os testes caixa Branca e Preta;
4. Regressão: testa todas as versões ou protótipos de um sistema ;
5. Teste não funcionais: não se relacionam diretamente com as funções e respostas do sistema;

A simplicidade estrutural do software desenvolvido não requer técnicas sofisticadas de teste. Para seu desenvolvimento, o teste predominante foi Caixa preta, por priorizar o funcionamento do artefato, que é o objetivo primeiro do projeto.

Para inserção das funções Lambda e Clusore, utilizamos a técnica Caixa Cinza, pois a arquitetura do código foi revisada diversas vezes (Caixa Branca) concorrentemente com testes de funcionalidade (Caixa Preta).

Ratifico também que, dada as características do software, não é eficiente automação de testes ou uso de ferramentas com esse objetivo. [5]

Destaco os tipos de testes elencados pelo site Atlassian (2024) [5]:

1. Testes de unidade: testar métodos e funções individuais de classes, componentes ou módulos usados pelo software.
2. Testes de integração: verificam se diferentes módulos ou serviços usados pelo seu aplicativo funcionam bem juntos.
3. Testes funcionais: verificam a saída de uma ação e não verificam os estados intermediários do sistema ao executar essa ação.

4. Testes de ponta a ponta: replica o comportamento de um usuário com o software em um ambiente de aplicativo completo.

5. Testes de aceitação: testes formais executados para verificar se um sistema atende aos requisitos de negócios e foca em replicar os comportamentos do usuário.

6. Testes de desempenho: avaliam o desempenho de um sistema sob uma carga de trabalho específica.

7. Teste de fumaça (smoke test): são testes básicos que verificam a funcionalidade básica do aplicativo.

Considerando essa classificação, utilizamos Teste de funcionais para o codificação do projeto.

Para as funções Lambda e Clusore (função If/Else e Calulo IMC), utilizamos a técnica Teste de unidade e Teste de integração.

Caso de Teste

Projeto: ÍNDICE DE MASSA CORPORAL
Plano de Teste: Validação Geral
Suíte de Teste: Validação do Módulo

Caso de Teste:	1	Resumo:	Checar a precisão do calculo do IMC		
Prioridade:	Alta	Status:	Aprovado	Executor:	Fabiano
Data da criação:	18/09/24				
Pré-condição:	Rodar o aplicativo e abrir a página				
Passos:	1. Informar valor valido no campo Peso 2. Informar valor valido no campo Altura 3. Apertar o botão Calcular				
Resultado Esperado:	O valor do campo Seu IMC ser compatível com o cálculo IMC				
Observações:	Sem observações				
Comentários:	Sem comentários				

Caso de Teste

Projeto: ÍNDICE DE MASSA CORPORAL

Plano de Teste: Validação Geral

Suíte de Teste: Validação do Módulo

Caso de Teste:	1	Resumo:	Checar as mensagens de classificação do resultado IMC		
Prioridade:	Alta	Status:	Aprovado	Executor:	Fabiano
Data da criação:	18/09/24				
Pré-condição:	Rodar o aplicativo e abrir a página				
Passos:	1. Informar vazio no campo Peso 2. Preencher Altura válida 3. Apertar o botão Calcular 4. Repetir o processo inserindo acrescentando 5 kg ao primeiro peso informado e mantendo a mesma altura inicial				
Resultado Esperado:	A mensagem de classificação ser compatível com o resultado do calculo do IMC				
Observações:	Sem observações				
Comentários:	Sem comentários				

Caso de Teste

Projeto: ÍNDICE DE MASSA CORPORAL

Plano de Teste: Validação Geral

Suíte de Teste: Validação do Módulo

Caso de Teste:	2	Resumo:	Checar a inserção de peso inválido		
Prioridade:	Média	Status:	Aprovado	Executor:	Fabiano
Data da criação:	18/09/24				
Pré-condição:	Rodar o aplicativo e abrir a página				
Passos:	<ol style="list-style-type: none">1. Informar vazio no campo Peso2. Informar valor valido no campo Altura3. Apertar o botão Calcular4. Repetir o processo inserindo zero no campo Peso				
Resultado Esperado:	Apresentar alerta informando o item inválido				
Observações:	Sem observações				
Comentários:	Sem comentários				

Caso de Teste

Projeto: ÍNDICE DE MASSA CORPORAL

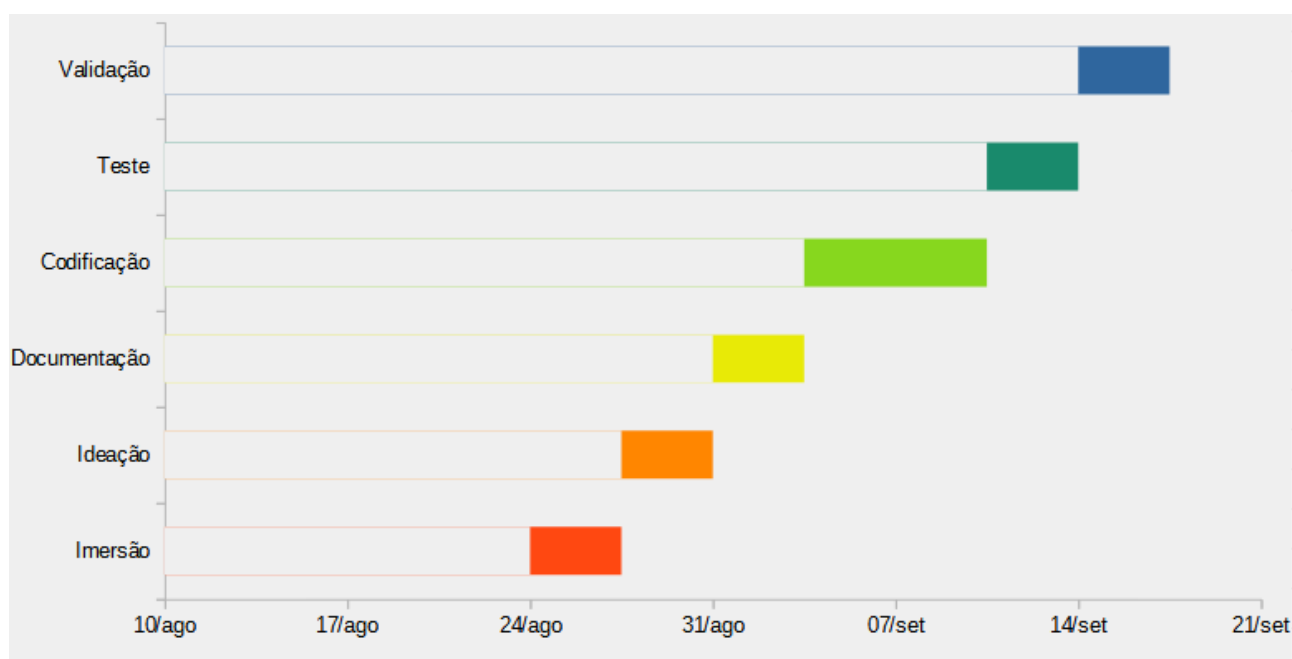
Plano de Teste: Validação Geral

Suíte de Teste: Validação do Módulo

Caso de Teste:	3	Resumo:	Checar a inserção de altura inválida		
Prioridade:	Média	Status:	Aprovado	Executor:	Fabiano
Data da criação:	18/09/24				
Pré-condição:	Rodar o aplicativo e abrir a página				
Passos:	<ol style="list-style-type: none">1. Informar vazio no campo Altura2. Informar valor valido no campo Peso3. Apertar o botão Calcular4. Repetir o processo inserindo zero no campo Altura				
Resultado Esperado:	Apresentar alerta informando o item inválido				
Observações:	Sem observações				
Comentários:	Sem comentários				

5. Implantação e Operações

Não houve formação de grupo e atividade foi realizada individualmente. Não havendo distribuição de tarefas, informamos apenas o cronograma de cumprimento das etapas de entrega do artefato.



FONTE: Etapas de entrega do projeto - Diagrama de GANT - Autor, 2024

5. Código fonte

O projeto está salvo na plataforma de controle de versões Github neste endereço:

<https://github.com/TavaresFabiano/N704>

BIBLIOGRAFIA

[1] ESTRATÉGIA, 2024. Requisitos Funcionais e Não Funcionais para o EBSEH (TI).

Disponível em <<https://www.estrategiaconcursos.com.br/blog/requisitos-funcionais-nao-funcionais-ebserh-ti/#:~:text=Requisitos%20Funcionais%3A%20Especificam%20%E2%80%9Co%20que,e%20limita%C3%A7%C3%B5es%2C%20e%20aspectos%20t%C3%A9cnicos.>>. Acessado em 18/09/24 às 20h22

[2] ATÉ O MOMENTO, 2024. Diferença de Requisito Funcional e Regra de Negócio.

Disponível em <<https://www.ateomomento.com.br/requisito-funcional-x-regra-de-negocio-quem-e-quem/>>. Acessado em 18/09/24 às 20h35

[3] PLANALTO, 2024. LEI Nº 13.709, DE 14 DE AGOSTO DE 2018. Disponível em <https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm >. Acessado em 16/09/24 às 19h10

[4] MONITORATEC, 2024. 5 tipos de metodologia de testes de software para você conhecer já. Disponível em <<https://www.monitoratec.com.br/blog/metodologia-de-testes-de-software/>>. Acessado em 18/09/24 às 22h39

[5] ATlassian, 2024. Teste manual vs. automatizado. Disponível em <<https://www.atlassian.com/br/continuous-delivery/software-testing/types-of-software-testing>>. Acessado em 18/09/24 às 22h52