

# Documentação do processo de desenvolvimento

## Visão Geral do Projeto

O objetivo deste projeto é desenvolver uma Calculadora de IMC (Índice de Massa Corporal). A calculadora foi escolhida por ser de fácil desenvolvimento e utilidade prática para ajudar as pessoas a monitorar sua saúde.

## Passos do Desenvolvimento

**1. Primeiro passo:** Definir qual seria o projeto individual, determinei a **calculadora de IMC (Índice de massa corporal)**. Pois era de fácil desenvolvimento e útil.

## 2.Desenvolvendo funcionalidades

### 2.1.Desenvolvemos as seguintes funções:

\***calcular\_imc(peso, altura)**: Calcula o IMC utilizando a fórmula  $imc = peso / (altura * altura)$ , onde o peso e a altura são fornecidos pelo usuário.

\***classificar\_imc(imc)**: Classifica o IMC em diferentes categorias usando estruturas de controle (if, elif, else).

```
imc.py > ...
1  # Fórmula do cálculo do IMC
2  def calcular_imc(peso, altura):
3      imc = peso / (altura * altura)
4      return imc
5
6  # Classificação do IMC
7  def classificar_imc(imc):
8      if imc < 16:
9          return "Magreza grau 3º"
10     elif 16 <= imc < 16.99:
11         return "Magreza grau 2º"
12     elif 17 <= imc < 18.49:
13         return "Magreza grau 1º"
14     elif 18.50 <= imc < 24.99:
15         return "Normal"
16     elif 25 <= imc < 29.99:
17         return "Sobrepeso"
18     elif 30 <= imc < 34.99:
19         return "Sobrepeso 1º"
20     elif 35 <= imc < 39.99:
21         return "Sobrepeso 2º"
22     else:
23         return "Sobrepeso 3º"
24
```

### 3. Criação do Repositório no GitHub

Após desenvolver essas funções, criamos um repositório no GitHub e enviamos o arquivo `imc.py`. Esta será a primeira branch principal (main).

### 4. Desenvolvimento da interface gráfica

#### 4.1.Importação de bibliotecas

Primeiramente, importamos a biblioteca Tkinter, que possui um conjunto de ferramentas para a criação de interfaces gráficas, e importamos funções do `imc.py`, como `classificar_imc` e `calcular_imc`.

```
interface.py > ...
1  #Importando bibliotecas
2  import tkinter as tk
3  from tkinter import messagebox
4  from imc import classificar_imc, calcular_imc
```

#### 4.2.Construção da interface

Nesta parte temos todo o desenvolvimento de interface da calculadora, os textos que são representados por `label_ "nome_do_texto" = tk.Label`, os espaços que a introdução das informações que são representados por `entrada_ "nome_da_entrada" = tk.Entry`. Usamos o método `grid` para posicionar os elementos.

```
17  #Criação da janela
18  janela = tk.Tk()
19  janela.title("Calculadora IMC")
20
21  label_peso = tk.Label(janela, font=("", 15), text="Digite peso (Kg):")
22  label_peso.grid(column=0, row=0, padx=10, pady=10)
23
24  entrada_peso = tk.Entry(janela, font=("", 15))
25  entrada_peso.grid(column=1, row=0, padx=10, pady=10)
26
27  label_altura = tk.Label(janela, font=("", 15), text="Digite altura (m):")
28  label_altura.grid(column=0, row=1, padx=10, pady=10)
29
30  entrada_altura = tk.Entry(janela, font=("", 15))
31  entrada_altura.grid(column=1, row=1, padx=10, pady=10)
32
33  botao_p_calcular = tk.Button(janela, text="Calcular IMC", command=calcular)
34  botao_p_calcular.grid(columnspan=2, row=3, padx=10, pady=10)
35
36  janela.mainloop()
```

```

6  #Função que calcula
7  def calcular():
8      try:
9          peso = float(entrada_peso.get())
10         altura = float(entrada_altura.get())
11         imc = calcular_imc(peso, altura)
12         classificacao = classificar_imc(imc)
13         messagebox.showinfo("Resultado", f"Seu IMC é: {imc:.2f}\nClassificação: {classificacao}")
14     except ValueError:
15         messagebox.showerror("Erro", "Por favor, insira valores válidos para peso e altura.")

```

Aqui podemos criar mais uma branch no git chamada **interface** e enviar ao repositório os arquivos **interface.py** e **\_\_pycache\_\_**.

## 5.Elaboração do README.md

O arquivo **README.md** serve como a principal documentação para o seu projeto de programação no GitHub ou em outras plataformas de controle de versões. Ele deve incluir: Apresentação do Projeto, instruções de instalação, guia de Uso, objetivos, cronograma, licença e autor.

```

① README.md
1  TITULO DO PROJETO:  Calculadora de IMC([Índice de massa corporal]).
2
3
4  FUNCIONALIDADE:  O principal função deste projetos é, criar uma calculadora de IMC em uma interface simples, para informar o
5  usuario do seu estado fisico, em comparação ao ideal para sua saúde.
6
7
8  OBJETIVO:  O objetivo é poder informar o usuario de como esta sua saúde física, para que ele possa ter uma noção, de acordo com os critérios da OMS.
9
10
11  Cronograma:
12
13  01/12/2024  _____ Desenvolver backend, código base e suas funções de calcular e classificar. Desenvolver interface utilizando biblioteca Tkinter.
14  02/12/2024  _____ Criar repositório no GitHub, criar README.md, enviar o projeto para o Git.
15  03/12/2024  _____ Elaborar slides de apresentação, enviar link de acesso do repositório e slides no AVA.
16
17
18  COMO DEIXAR O CÓDIGO FUNCIONAL(INSTALAÇÃO):  Para poder utilizar o código e a interface, apenas é necessario fazer download dos arquivos, o arquivo (README) é
19  opcional, pois não tem relação direta com o funcionamento do código, após download feito deve ser aberto em um editor de programação, de preferência utilize o
20  (VsCode), por ser o mais completo.
21
22
23  USO:  Após a "instalação", para usa-lo basta abrir o arquivo (interface.py). após isso, basta clicar no botão similar ao play em apps de streaming e inserir as
24  informações necessarias, neste caso peso e altura, e clicar a no botão calcular.
25
26
27  LICENÇA:  Este é um programa simples, e poder ser modificado por outros desenvolvedores, pode ser usado em qualquer condição para testes, exercicios e
28  aprendizagem, e ser distribuido livremente.
29
30  AUTOR:  Everson Daniel
31
32  AGRADECIMENTO:  Agradeço a todos pelos utilização do programa.

```

## 6.GitHub

O GitHub nos ajuda no desenvolvimento pois nos fornece um controle de versões e histórico permitindo uma melhor noção do projeto, a branch (**main**), e a principal e existem outras **interface**, **README** são boas para organização, também em casos de projetos maiores onde possuem diversos desenvolvedores, o GitHub permite que todos trabalhem juntos em diferentes branches, gerenciando e resolvendo conflitos.