



**ANHANGUERA**

**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**MYKE ALEXSANDRO BUENO**

**LINGUAGEM DE PROGRAMAÇÃO**

**Cálculo de IMC**

**ARARAQUARA**

**2023**

## **1. INTRODUÇÃO**

Foi utilizado como ferramenta de desenvolvimento o Google Cloud Shell, que é uma máquina virtual que possui um armazenamento de 5 GB, e é executado no Google Cloud.

Tendo em vista que o Google Cloud Shell provisiona uma máquina virtual gratuitamente, utilizam de um sistema operacional gratuito e estável, como o Debian (Linux), na sua versão 11 (bullseye). Que por sua vez, já possui uma versão de python nativa.

Ferramenta utilizada - GOOGLE CLOUD SHELL

Sistema operacional - LINUX

Distro utilizada - DEBIAN 11 (BULLSEYE)

IDE utilizada - GNU NANO 5.4

Linguagem utilizada - PYTHON 3.9.2

## **2. OBJETIVOS**

Como objetivos, tive que instalar o sistema Google Cloud Shell, criar um programa que faça cálculo do IMC e, ao final criar um relatório da atividade.

## **3. MÉTODOS**

Primeiro, procurei entender o funcionamento do Google Cloud Shell e me ambientar com ele, pois utilizei o shell direto no browser. Em seguida procurei como é calculado o Índice de Massa Corporal (IMC), com a fórmula em mente dei início a criação do programa, visando o armazenamento desses dados em uma utilização futura.

Optei por criar funções, a fim de evitar repetição de código. Dei início com a função “main”, que requisita a entrada de dados como: nome(string), peso e altura como uma string, utilizei o replace, visando o “erro” de utilizarem “,” ao invés de “.” em seguida transformei essa string em (float). Com o peso e altura, realizei o cálculo do IMC e utilizei o resultado na função “classificacao\_peso”, que classifica em qual posição essa pessoa se encaixa, e com esse resultado fiz o registro desse cálculo de IMC. Antes do registro acontecer, ele tenta se conectar ao arquivo de texto “resultados\_IMC.txt”, caso falhe cria o arquivo e em seguida registra o IMC.

## 4. RESULTADOS

### 4.1. Programa: calculadora IMC.py

```
"IMC = PESO (KG) / ALTURA (M²)"

def main():
    """Funcao MAIN,
    Coleta informacoes como: nome, peso e altura,
    calcula o IMC, faz sua classificacao e o registra
    no arquivo .txt"""
    try:
        print("*** Calculadora de IMC ***")
        nome = str(input("Digite o nome do avaliado: ")).title()
        peso = convert_str_float("Digite o peso em Quilos (ex:70.8): ")
        altura = convert_str_float("Digite a altura em Metros (ex:1.7): ")
        imc = round(peso / altura**2, 2)
        classificacao = classificacao_peso(imc)
        registro_imc(nome, imc, peso, altura, classificacao)
        print("")
        print(f"[SUCESSO] Ao registrar o IMC!")
        print("")
        print(f"Resultado do IMC: {imc}kg/m2 | {classificacao}")
        print("")
    except:
        print("[ERROR] Ao tentar calcular IMC!")

def convert_str_float(mensagem):
    """
    Funcao usada em main(),
    Criada para evitar uma digitacao
    errada, como ',' ao inves de '.'
    """
    try:
        opc = str(input(f"{mensagem}")).replace(",", ".")
        return float(opc)
    except:
        print("[ERROR] Ao tentar converter de string para float")

def classificacao_peso(imc):
    # Endereco da tabela: "https://www.drrogermoura.com.br/images/artigos/tabela-imc.png"
    """
    Funcao usada em main(),
    intuito de fazer a classificacao de peso
    e retornar a classificacao dele
    """
    if imc <= 18.5:
        resultado = "Abaixo do peso"
    elif 18.6 <= imc <= 24.9:
        resultado = "Pesoal ideal"
    elif 25 <= imc <= 29.9:
        resultado = "Levemente acima do peso"
    elif 30 <= imc <= 34.9:
        resultado = "Obesidade grau I"
    elif 35 <= imc <= 39.9:
        resultado = "Obesidade grau II"
    else:
        resultado = "Obesidade grau III (Mórbida)"
    return resultado

def verif_arqui_result():
    """
    Funcao usada em registro_imc(),
    Tenta se conectar ao resultados_IMC.txt, caso falhe,
    cria esse arquivo com o cabecalho em formato csv
    """
    try:
        with open("resultados_IMC.txt", "r") as arquivo:
            return True
    except IOError:
        with open("resultados_IMC.txt", "a") as arquivo:
            arquivo.write("nome,imc,peso,altura,classificacao\n")
```

```
def registro_imc(nome="Anonimo",imc=0,peso=0,altura=0,classificacao=None):  
    """  
    Funcao usada em main(),  
    pega as informacoes como: nome, imc, peso, altura e classificacao e  
    as registra no arquivo .txt  
    """  
    try:  
        verif_arqui_result()  
        with open("resultados_IMC.txt", "a") as arquivo:  
            arquivo.write(f"{nome},{imc},{peso},{altura},{classificacao}\n")  
    except IOError:  
        print("[ERROR] Ao tentar registrar os dados (IMC) no arquivo 'resultados_IMC.txt' !")  
  
if __name__ == "__main__":  
    main()
```

#### 4.2. Execução: calculadora\_IMC.py

```
myke676@cloudshell:~ (portfolioimcpython) $ python3 calculadora_IMC.py
```

```
** Calculadora de IMC **
```

```
Digite o nome do avaliado:
```

```
** Calculadora de IMC **
```

```
Digite o nome do avaliado: Jorge
```

```
Digite o peso em Quilos (ex:70.8):
```

```
** Calculadora de IMC **
```

```
Digite o nome do avaliado: Jorge
```

```
Digite o peso em Quilos (ex:70.8): 89,8
```

```
Digite a altura em Metros (ex:1.7):
```

```
** Calculadora de IMC **
```

```
Digite o nome do avaliado: Jorge
```

```
Digite o peso em Quilos (ex:70.8): 89,8
```

```
Digite a altura em Metros (ex:1.7): 1.57
```

```
[SUCESSO] Ao registrar o IMC!
```

```
Resultado do IMC: 36.43kg/m2 | Obesidade grau II
```

#### 4.3 Execução: resultados\_IMC.txt

```
myke676@cloudshell:~/Portfolio/calculadora_IMC (portfolioimcpython) $ cat resultados_IMC.txt  
nome,imc,peso,altura,classificacao  
Jorge,36.43,89.8,1.57,Obesidade grau II
```

### 5. Conclusão

Vejo o intuito de aprender a usar ferramentas ágeis e diretas, onde somos “forçados” a buscar a entender algo “novo” e desempenhar em cima dele. Utilizar o Google Cloud, me trouxe uma experiência de ver um servidor estável, bem dimensionado e distribuído, onde temos um cenário de trabalhar em produção.

Python é uma linguagem que sua sintaxe é “fácil” de compreender e ler, acredito que esse seja o motivo da utilização dessa linguagem nesse primeiro momento.

Juntando Python, Google Cloud e mais o Linux, temos algo grandioso para ser desbravado e criado. Lembrando que o Google Cloud utiliza o sistema operacional Linux, que por sua vez, possui Python em sua forma nativa.