

Agosto de 2023

UNOPAR

LINGUAGEM DE PROGRAMAÇÃO

CÁLCULO DE IMC - PYTHON



Aluno: David Rodrigues da Silva (3521884802)

Professora: Vanessa Matias Leite

Tutora: Luana Gomes de Souza

Introdução

Este relatório da aula prática de linguagem de programação mostra como foi feito um algoritmo para calcular o IMC (Índice de Massa Corpórea), usando a linguagem de programação Python e a IDE (Ambiente de Desenvolvimento Integrado) Google Cloud Shell Editor.

Python

Python é uma linguagem de programação de tipagem dinâmica, interpretada e com uma sintaxe simples, foi criada pelo Holandês Guido Van Rossum, que começou o desenvolvimento da mesma em dezembro de 1989 e teve sua primeira versão pública em fevereiro de 1991, o Python 0.9.0.

O motivo por trás da criação do Python era desenvolver uma linguagem de programação que fosse fácil de ler, escrever e manter. Guido van Rossum desejava criar uma linguagem que tivesse uma sintaxe clara e simples, favorecendo a legibilidade do código, o que é expresso no lema do Python: "Readability counts" (A legibilidade conta).

Características:

- **Sintaxe Limpa e Legível:** O Python enfatiza a legibilidade do código através do uso de uma sintaxe simples e coesa, utilizando espaços em branco para definir a estrutura do código em vez de chaves ou delimitadores.
- **Tipagem Dinâmica:** As variáveis em Python não precisam ser explicitamente declaradas com um tipo, e seu tipo pode mudar durante a execução do programa.
- **Gerenciamento de Memória Automático:** O Python possui um coletor de lixo que automaticamente gerencia a alocação e desalocação de memória, tornando mais fácil a gestão de recursos.
- **Ampla Biblioteca Padrão:** O Python possui uma extensa biblioteca padrão que oferece módulos e funções para várias tarefas, desde manipulação de strings e arquivos até acesso à rede e análise de dados.
- **Orientação a Objetos:** O Python suporta programação orientada a objetos (POO) e permite a criação de classes e objetos, facilitando a organização e reutilização de código.
- **Interpretada:** Python é uma linguagem interpretada, o que significa que os programas são executados por um interpretador em vez de serem compilados.

Há outras características da linguagem que não foram citadas.

Introdução

Google Cloud Shell Editor

O Google Cloud Shell Editor é uma ferramenta integrada ao Google Cloud Shell que oferece um ambiente de desenvolvimento online baseado em navegador. O Google Cloud Shell é uma máquina virtual temporária que é provisionada na nuvem da Google para fornecer um ambiente de linha de comando para interagir com os serviços e recursos do Google Cloud Platform (GCP).

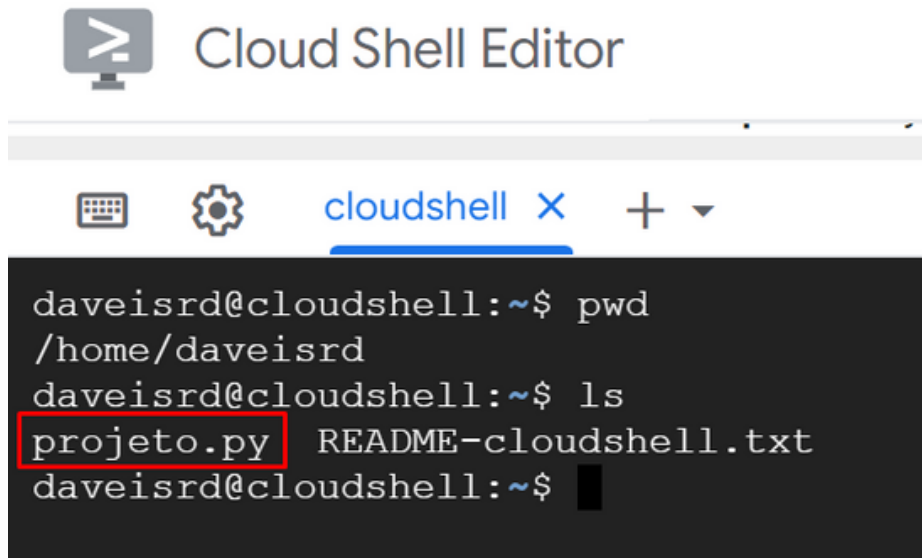
Características:

- **Ambiente Integrado:** O Google Cloud Shell Editor está integrado ao ambiente do Google Cloud Shell, permitindo que você acesse facilmente os recursos do GCP enquanto desenvolve e edita seu código.
- **Editor de Código:** O editor oferece uma interface de edição de código com suporte a várias linguagens de programação populares, como Python, JavaScript, Java, Go e muito mais.
- **Terminal Integrado:** Além do editor de código, o Google Cloud Shell Editor inclui um terminal de linha de comando totalmente funcional, permitindo que você execute comandos diretamente no ambiente do Google Cloud.
- **Acesso aos Serviços do Google Cloud:** Você pode interagir diretamente com serviços e recursos do Google Cloud Platform, como armazenamento de objetos, bancos de dados e serviços de computação, sem sair do ambiente de desenvolvimento.
- **Armazenamento de Arquivos:** Os arquivos que você cria e edita no Google Cloud Shell Editor são armazenados temporariamente, permitindo que você os acesse e sincronize facilmente com repositórios do Google Cloud.
- **Personalização do Ambiente:** Você pode personalizar o ambiente do Google Cloud Shell Editor, instalando ferramentas e bibliotecas adicionais, de acordo com suas necessidades de desenvolvimento.

Há outras características da ferramenta que não foram citadas.

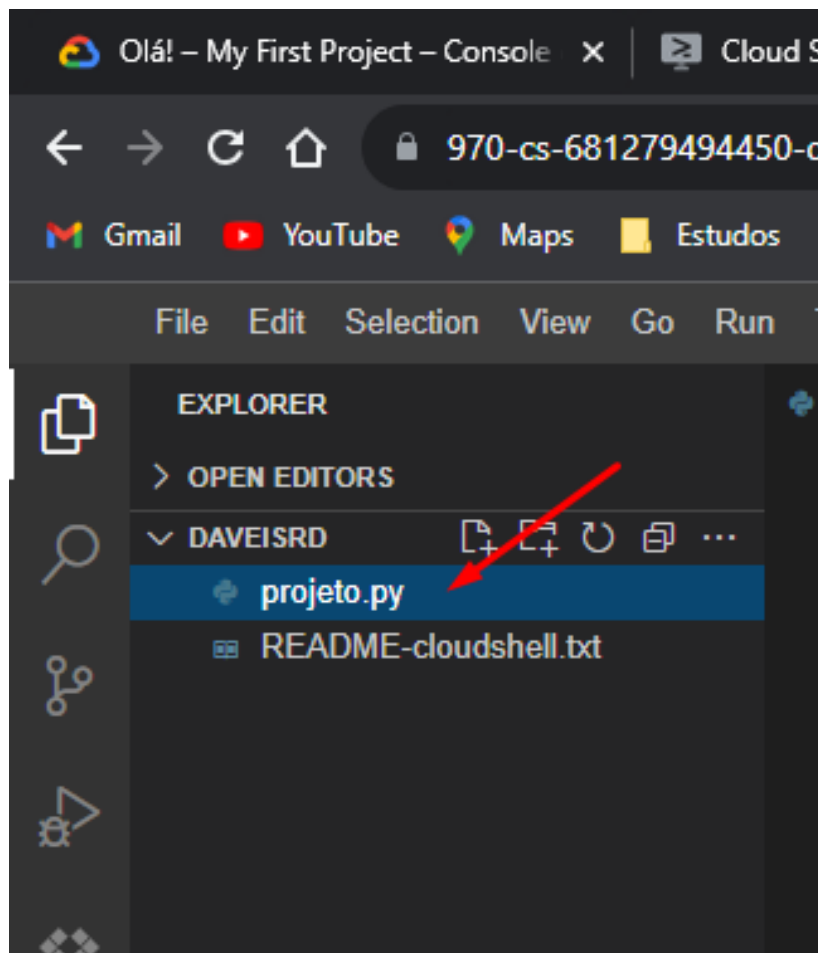
Métodos

Foi criado um módulo chamado "projeto.py" no ambiente virtual do GCP e acessado via Google Cloud Shell Editor para facilitar a edição do código.



The screenshot shows the Cloud Shell Editor interface. At the top, there's a title bar with a terminal icon and the text "Cloud Shell Editor". Below the title bar, there's a toolbar with a keyboard icon, a settings gear, a tab labeled "cloudshell" with a close button, and a plus sign. The main area is a terminal window with a dark background. The prompt is "daveisrd@cloudshell:~\$". The first command is "pwd", and the output is "/home/daveisrd". The second command is "ls", and the output is "projeto.py README-cloudshell.txt". The "projeto.py" file name is highlighted with a red rectangle.

```
daveisrd@cloudshell:~$ pwd
/home/daveisrd
daveisrd@cloudshell:~$ ls
projeto.py README-cloudshell.txt
daveisrd@cloudshell:~$
```



Métodos

Lógica do programa:

O cálculo do IMC é feito da seguinte forma:

O peso é dividido pelo quadrado da altura, essa expressão em Python fica da seguinte forma -> `peso / (altura ** 2)`

Então, primeiro o usuário deve informar qual o peso dele, e logo em seguida, qual a altura para que seja possível fazer esse cálculo.

Então, foi criada uma variável chamada `altura`, e outra chamada `peso`, cada uma delas armazenará o valor que a função `input` retornar do usuário enquanto o programa estiver ativo:

```
print("### CALCULADORA DE IMC ###") # Mostra ao usuário a proposta do programa

while True: # Criando um laço de repetição
    try: # Tratamento de erros com try
        altura = float(input('Digite sua altura (em metros) - ( e use "." no lugar da ","): '))
        peso = float(input('Digite o seu peso (em kg) - (e use "." no lugar da ","): '))
```

Esse valor que é armazenado na variável, inicialmente é do tipo "texto", ou string, sendo impossível fazer cálculo entre números e strings, então foi feito o type coercion (Conversão de Tipos) usando a classe "`float()`", assim o valor que for digitado pelo usuário, passará a ser ponto flutuante, ou seja, possui casas decimais, e tendo a propriedade de fazer cálculos.

Foi criada uma variável chamada `imc` que contém a expressão do cálculo -> `peso / (altura ** 2)`

E usado a função `print()` formatada exibindo a altura informada, o peso, e o resultado do cálculo do IMC:

```
altura = float(input('Digite sua altura (em metros) - ( e use "." no lugar da ","): '))
peso = float(input('Digite o seu peso (em kg) - (e use "." no lugar da ","): '))

imc = peso / (altura ** 2)
print(f'Você tem {altura}m de altura, {peso}kg de peso e seu IMC é de: {imc:.2f}')
```

Somente com essas pequenas linhas de código, já seria possível finalizar o projeto com o objetivo solicitado inicialmente, porém foi resolvido implementar o código com algumas funcionalidades extras e tratamento de erros:

Métodos

Implementações:

- Classificação do IMC usando estruturas condicionais (if, elif, else);
- Tratamento de erros usando try e except;
- Estrutura de laços de repetição (Impedir que o programa feche sem que o usuário queira) usando while, continue e break.

Classificação de IMC

```
if imc < 18.5:
    print('Seu estado de classificação é "Magreza"')
elif imc >= 18.5 and imc < 25:
    print('Seu estado de classificação é "Normal"')
elif imc >= 25 and imc < 30:
    print('Seu estado de classificação é "Sobrepeso"')
elif imc >= 30 and imc < 35:
    print('Seu estado de classificação é "Obesidade grau I"')
elif imc >= 35 and imc < 40:
    print('Seu estado de classificação é "Obesidade grau II"')
elif imc > 40:
    print('Seu estado de classificação é "Obesidade grau III"')
```

Tratamento de Erros

```
projeto.py > ...
5 while True: # Criando um laço de repetição
6     try: # Tratamento de erros com try
7         altura = float(input('Digite sua altura (em metros) - (e use "." no lugar da ","): '))
8         peso = float(input('Digite o seu peso (em kg) - (e use "." no lugar da ","): '))
9
10        imc = peso / (altura ** 2)
11        print(f'Você tem {altura}m de altura, {peso}kg de peso e seu IMC é de: {imc:.2f}')
12
13        if imc < 18.5:
14            print('Seu estado de classificação é "Magreza"')
15        elif imc >= 18.5 and imc < 25:
16            print('Seu estado de classificação é "Normal"')
17        elif imc >= 25 and imc < 30:
18            print('Seu estado de classificação é "Sobrepeso"')
19        elif imc >= 30 and imc < 35:
20            print('Seu estado de classificação é "Obesidade grau I"')
21        elif imc >= 35 and imc < 40:
22            print('Seu estado de classificação é "Obesidade grau II"')
23        elif imc > 40:
24            print('Seu estado de classificação é "Obesidade grau III"')
25
26        novo_calculo = input('Deseja fazer um novo cálculo de IMC? [S]im ou [N]ão: ').lower().startswith('s')
27        if novo_calculo is True:
28            continue
29        else:
30            break
31
32    except ValueError: # Se ocorrer algum tipo de erro de valor, então vai exibir as seguintes opções:
33        print('O valor digitado é inválido, digite somente números e utilize "." ao invés de ","')
34        tentar_novamente = input('Deseja tentar novamente? [S]im ou [N]ão: ').lower().startswith('s')
35        if tentar_novamente is True:
36            continue
37        else:
38            break
```

Métodos

Laços de repetições

```

5 while True: # Criando um laço de repetição
6     try: # Tratamento de erros com try
7         altura = float(input('Digite sua altura (em metros) - ( e use "." no lugar da ","): '))
8         peso = float(input('Digite o seu peso (em kg) - (e use "." no lugar da ","): '))
9
10        imc = peso / (altura ** 2)
11        print(f'Você tem {altura}m de altura, {peso}kg de peso e seu IMC é de: {imc:.2f}')
12
13        if imc < 18.5:
14            print('Seu estado de classificação é "Magreza"')
15        elif imc >= 18.5 and imc < 25:
16            print('Seu estado de classificação é "Normal"')
17        elif imc >= 25 and imc < 30:
18            print('Seu estado de classificação é "Sobrepeso"')
19        elif imc >= 30 and imc < 35:
20            print('Seu estado de classificação é "Obesidade grau I"')
21        elif imc >= 35 and imc < 40:
22            print('Seu estado de classificação é "Obesidade grau II"')
23        elif imc > 40:
24            print('Seu estado de classificação é "Obesidade grau III"')
25
26        novo_calculo = input('Deseja fazer um novo cálculo de IMC? [S]im ou [N]ão: ').lower().startswith('s')
27        if novo_calculo is True:
28            continue
29        else:
30            break
31
32    except ValueError: # Se ocorrer algum tipo de erro de valor, então vai exibir as seguintes opções:
33        print('O valor digitado é inválido, digite somente números e utilize "." ao invés de ",')
34        tentar_novamente = input('Deseja tentar novamente? [S]im ou [N]ão: ').lower().startswith('s')
35        if tentar_novamente is True:
36            continue
37        else:
38            break

```

Dessa forma, caso o usuário insira um valor incorreto, por exemplo um texto onde deveria ser número, o programa vai exibir uma mensagem de erro "O valor digitado é inválido, digite somente números e utilize '.' ao invés de ','" e logo em seguida perguntará "Deseja tentar novamente? [S]im ou [N]ão". Se o valor da resposta começar com "S", então vai ser ativado o "continue", fazendo com que o início do while volte a processar.

Resultados

Programa sendo interpretado:

```
Python Python x
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to promising-flash-395922.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
rojeto.pycloudshell:~ (promising-flash-395922)$ /usr/bin/python /home/daveisrd/pr
### CALCULADORA DE IMC ###
Digite sua altura (em metros) - ( e use "." no lugar da ","): 
```

```
rojeto.pycloudshell:~ (promising-flash-395922)$ /usr/bin/python /home/daveisrd/pr
### CALCULADORA DE IMC ###
Digite sua altura (em metros) - ( e use "." no lugar da ","): 1.70
Digite o seu peso (em kg) - (e use "." no lugar da ","): 
```

```
rojeto.pycloudshell:~ (promising-flash-395922)$ /usr/bin/python /home/daveisrd/pr
### CALCULADORA DE IMC ###
Digite sua altura (em metros) - ( e use "." no lugar da ","): 1.70
Digite o seu peso (em kg) - (e use "." no lugar da ","): 62
Você tem 1.7m de altura, 62.0kg de peso e seu IMC é de: 21.45
Seu estado de classificação é "Normal"
Deseja fazer um novo cálculo de IMC? [S]im ou [N]ão: 
```

```
rojeto.pycloudshell:~ (promising-flash-395922)$ /usr/bin/python /home/
### CALCULADORA DE IMC ###
Digite sua altura (em metros) - ( e use "." no lugar da ","): 1.70
Digite o seu peso (em kg) - (e use "." no lugar da ","): 62
Você tem 1.7m de altura, 62.0kg de peso e seu IMC é de: 21.45
Seu estado de classificação é "Normal"
Deseja fazer um novo cálculo de IMC? [S]im ou [N]ão: sim
Digite sua altura (em metros) - ( e use "." no lugar da ","): 
```

```
Você tem 1.7m de altura, 62.0kg de peso e seu IMC é de: 21.45
Seu estado de classificação é "Normal"
Deseja fazer um novo cálculo de IMC? [S]im ou [N]ão: não
daveisrd@cloudshell:~ (promising-flash-395922)$ 
```

```
### CALCULADORA DE IMC ###
Digite sua altura (em metros) - ( e use "." no lugar da ","): teste
O valor digitado é inválido, digite somente números e utilize "." ao invés de ","
Deseja tentar novamente? [S]im ou [N]ão: 
```


Conclusão

É notável a impressionante potência da linguagem Python, que se revela capaz de resolver expressões matemáticas, estruturas de repetição (loops), operadores lógicos, construções condicionais e muito mais com apenas algumas linhas de código. Além de sua eficácia, Python apresenta uma sintaxe intuitiva e de fácil compreensão.

O Google Cloud Shell Editor, por sua vez, demonstrou ser uma ferramenta extraordinariamente conveniente. A eliminação da necessidade de instalar o Python em minha máquina pessoal foi notável, uma vez que a configuração já estava disponível no ambiente Google Cloud. Essa solução também ressalta a portabilidade do código, permitindo acesso a partir de qualquer parte do mundo, desde que haja conexão à internet.

Acesse ao código completo em: <https://github.com/EaeDave/IMCCalc>

