

equipe: Rafael Gustavo Reinert (apenas 1)

Apresente um documento textual descrevendo as variáveis de entrada e saída, perguntas e regras

atenção: isto é um trabalho de universidade

Introdução

meu trabalho consiste na automatização de uma tarefa, um canal de youtube chamado [personali testes](#), o qual produz vídeos na qual o usuário precisa somar pontos de acordo com perguntas de um teste de personalidade, cada vídeo possui 1 teste

a parte do teste a qual foi automatizada se trata da soma desses pontos e o resultado final, apenas sendo necessário responder as perguntas que aparecem, pois no video original era necessario somar, escolher e armazenar os pontos para no final do vídeo você encontrasse o resultado final compatível com sua pontuação

selecionei e codifiquei 4 testes

as variáveis de entrada do programa consiste nas respostas, que podem ser letras "a","b","c","d" ou "(s/n)" (sim ou não) e também uma variável numérica de porcentagem de 0 a 100, são usadas essas entradas como respostas às perguntas

não está exatamente igual ao teste, pois todos os testes usam apenas opções entre letras, aqui inserimos a variável numérica para o usuário informar com mais precisão em algumas questões

as regras são simples:

```
if res == "a":  
    soma += 10
```

```
elif res == "b":  
    soma += 20  
elif res == "c":  
    soma += 30  
elif res == "d":  
    soma += 40
```

porém tal modelo foi inserido pelo uso de funções

```
i = validar(valores_permitidos, res)  
if i == True:  
    soma+= soma_resposta(valores_permitidos,[10,20,30],res)  
else:  
    print("fora do escopo, repetindo")
```

conforme cada resposta se soma uma quantidade de pontos, mas não é apenas essa estrutura que é usada, também é usada estruturas com respostas numéricas

```
print("5- as pessoas costumam elogiar suas habilidades artisticas?  
(s/n)")  
res = str(input())  
res = res.lower()  
if res == "s":  
    print("de 0 a 100, o quanto ocorre o elogio?")  
    q = float(input("quantos %:"))  
    g = q / 3.31  
    soma += g  
else:  
    soma += 10
```

neste exemplo ocorre o uso do percentual, sendo a regra diferente, caso seja informado “não” passa direto sem a avaliação percentual, porém caso contrário ocorre a avaliação, já que não precisaria caso fosse negativa

perguntas:

as perguntas no começo do programa se referem a qual teste voce deseja fazer

```
print("2- teste de qual minion voce é \n")
print("3- teste de voce seria um bom lider? \n")
print("4- teste voce é uma pessoa dramática? \n")
print("5- teste voce é popular")
```

selecionando uma opção aparecem as perguntas dos testes com as opções de resposta

```
print("3-seu celular toca 1h da manhã, o'que você acha ser o objetivo da ligação?" )
print("A) alguém precisa da minha ajuda")
print("B) alguém morreu")
print("C) sei lá")
```

as perguntas são em série e correspondentes ao teste original de cada vídeo, contendo o link em cada teste para acesso

O programa está disponível em

<https://replit.com/@RafaelGustavo4/sistemaespecialistapersonalite>

modelo usado

O modelo usado para inferir a personalidade foi o somatório de pontos por respostas, no exemplo do teste de drama aparecem perguntas sobre sua reação a diferentes situações, algumas reações são menos ou mais dramáticas, para cada opção de reação há um somatório diferente (10,20,30) os somatórios menores são os menos dramáticos e os maiores são os mais dramáticos, a partir dessas somas em cada pergunta se cria um somatório final que irá definir sua personalidade no sistema, pessoas mais dramáticas selecionam as reações com

maior pontuação, somando assim mais na variável, e vice versa, após as perguntas, a partir da soma final será dado o resultado

testes

quão dramático
qual sua liderança
qual minion você seria
quão popular

cada teste possui perguntas e opções relacionadas ao teste de um aspecto da personalidade, e de acordo com as respostas o sistema define um resultado após 10 questões respondidas

regras de produção

foram usadas 3 técnicas de resposta com cada uma delas contendo 1 modelo de regra de produção,

1º opções limitadas

```
i=False
while i == False:
    print("4- em uma mensagem em grupo todo mundo tenta planejar algo, voce: ")
    print("A) tomo a frente e vejo algo que funcionaria para todos")
    print("B) digo quando posso e deixo que eles marquem")
    print("C) deixo que planejem e apareço no dia combinado")

    res = str(input())
    res = res.lower()
    i = validar(valores_permitidos, res)
    if i == True:
        soma+= soma_resposta(valores_permitidos,[10,20,30],res)
    else:
        print("fora do escopo, repetindo")
```

```
//função validar
```

```
def validar(permitidos, valor_inserido):  
    if valor_inserido in permitidos:  
        return True  
    else:  
        return False
```

```
//função soma resposta
```

```
def soma_resposta(opcoes, somatorio, letra):  
    if len(somatorio) != len(opcoes):  
        print("erro: listas de somatorio e opcoes diferem")  
    else:  
        i = 0  
        while i < len(opcoes):  
            if opcoes[i] == letra:  
                qual = opcoes[i]  
                break  
            else:  
                pass  
            i += 1  
  
        sele = ord(qual) - 97  
        return somatorio[sele]
```

explicação:

a primeira função recebe uma lista com as possíveis opções [a,b,c] e o outro argumento é a resposta inserida, servindo para validar se a resposta está dentro do escopo de opções, a primeira regra de produção serve para validar isso, pois caso retorne verdadeiro (o primeiro argumento está contido no segundo elemento) segue para o processo, senão repete pelo while i == False

a segunda função soma_resposta recebe as opções de resposta, o somatório e a opção selecionada, percorre a lista de opções para encontrar a posição, caso não esteja na posição continua percorrendo, e caso esteja a variável qual salva o valor

a ultima parte transforma letra em número para que assim selecione do somatório a opção correta para somar na variável

2° tipo

```
print("5- as pessoas costumam elogiar suas habilidades artisticas?  
(s/n)")
```

```
res = str(input())  
res = res.lower()  
if res == "s":  
    print("de 0 a 100, o quanto ocorre o elogio?")  
    q = float(input("quantos %:"))  
    g = q / 3.225  
    soma += g  
else:  
    soma += 10
```

neste código caso a opção seja verdadeira ele pede para avaliar de 0 a 100 o quanto ocorre o elogio, caso contrário soma apenas 10 correspondendo com a opção de não receber elogio do vídeo

este trecho diferem do vídeo pois se trata de uma seleção mais precisa do usuário, pois algumas vezes a opção que mais corresponde não está dentro das alternativas, principalmente quando envolve questões mais quantitativas que qualitativas

3° tipo

```
print("1- de 0 a 100, você se preocupa com sua aparência?")  
q = float(input("quantos %:"))  
g = q / 3.225
```

```
soma += 10
#print(soma)
soma += g
```

semelhante a anterior, porém sem a opção do “não”, porém a seleção do 0 corresponde a dizer “não”

regras de produção em geral:

as regras de produção da seleção de cada opção estão no código, porém de uma forma diferente

isso:

```
soma = 0
from tecnicas import validar
from tecnicas import soma_resposta
valores_permitidos = ["a", "b", "c", "d"]

i = False
while i == False:
    print("3- como é o seu círculo de amigos?")
    print("A) Amigos?, mmmmm")
    print("B) bem grande, mas eu não confio em ninguém")
    print("C) grupo pequeno, mas bem próximo")
    print("D) grande, mas poucos são íntimos")

    res = str(input("opção: "))
    res = res.lower()
    i = validar(valores_permitidos, res)
    if i == True:
        soma += soma_resposta(valores_permitidos, [10, 20, 30, 40], res)
    else:
        print("fora do escopo, repetindo")
```

equivale a:

```
i = False
```

```
while i == False:
```

```
    print("3- como é o seu circulo de amigos?")
```

```
    print("A) Amigos?, mmmmm")
```

```
    print("B) bem grande, mas eu não confio em ninguém")
```

```
    print("C) grupo pequeno, mas bem proximo")
```

```
    print("D) grande, mas poucos são intimos")
```

```
res = str(input("opção: "))
```

```
res = res.lower()
```

```
if res == "a":
```

```
    soma+=10
```

```
    i=True
```

```
elif res== "b":
```

```
    soma+=20
```

```
    i=True
```

```
elif res == "c":
```

```
    soma+=30
```

```
    i=True
```

```
elif res == "d":
```

```
    soma+=40
```

```
    i=True
```

```
else:
```

```
    print("fora do escopo, repetindo")
```

essa é a principal regra de produção que foi inserida diferentemente através das funções validar e soma_resposta, pois era muito maçante ter que inserir tanto código durante a programação, técnica aprendida em algoritmo e programação de computadores 2

apesar disso, outros trechos de código possuem regras de produção explícitas, como as de resposta numérica, onde não foi implicitado através de funções

variáveis de entrada e de saída:

entrada:

validar(valores_permitidos, res)

as variáveis que entram são a lista de valores permitidos e a resposta

saída:

True/False

caso o res esteja contido em valores_permitidos retorna True, caso contrário retorna False

entrada:

soma_resposta(valores_permitidos, [10, 30, 20], res)

a entrada possui os valores permitidos de resposta ex: ["a","b","c"] , o somatório de cada valores e a resposta

saída:

a saída é algum dos somatórios [10,20,30], podendo assim ser 10, 20 ou 30, pois é uma função de seleção de lista, retornando um inteiro

finalização

a criação do trabalho apesar de ter a facilidade de já ter o conhecimento disponível ainda sim houve dificuldade principalmente com as variáveis numéricas, pois seus valores no momento de somar

ultrapassaram significativamente a opção com maior pontuação do vídeo, apesar disso houve um aprendizado de sistema especialista considerável