



Certified Tech Developer

The Ultimate Degree

Exercício em Python

Objetivo

Neste exercício, propomos melhorar nosso jogo de “pedra, papel, tesoura, lagarto, Spock” para adicionar um sistema de pontuação que calcule o número médio de vitórias. Para isso, precisaremos ter nossa versão já funcionando. Você pode baixar o código em:

<https://github.com/nidiodolfini/digitalhouse/blob/main/Infra1/Aula9/pedrapapeltesouralagartospock.py>

Tendo nosso código já preparado, vamos modificá-lo.



Pedra, papel, tesoura, lagarto, Spock - Etapa I

Nesta primeira etapa, devemos definir as variáveis que utilizaremos para fazer nosso sistema de pontuação. Faremos isso fora do loop onde o programa funciona, logo após os módulos importados:

```
userScore = 0  
  
pcScore = 0  
  
totalScore = 0
```



Pedra, papel, tesoura, lagarto, Spock - Passo II

Em seguida, continuamos adicionando a função que vamos usar para calcular a porcentagem de vitórias. Vamos adicioná-lo antes do loop:

```
def percent():  
    if totalScore > 0:  
        x = ((totalScore - pcScore) / totalScore) * 100  
        return x  
    elif totalScore == 0:  
        x = 0  
        return x
```



Pedra, papel, tesoura, lagarto, Spock - Etapa III

Nesta etapa, teremos que adicionar uma opção ao menu principal de nossa aplicação e sua execução, na qual – como visto no exemplo – estamos chamando a função criada anteriormente:



```
print("6) Show Scores")

elif option == 6:

    print ("Pontuações: ")

    print ("Usuário: ", userScore)

    print ("Pc: ", pcScore)

    print ("Porcentagem de vitórias: ", percent(), "% " )

    continue
```



Pedra, Papel, Tesoura, Lagarto, Spock – Etapa IV

Estamos quase na linha de chegada! Nesta etapa, teremos que adicionar uma variável que soma um ponto a cada check, como visto no exemplo a seguir:

```
if choosePc == "pedra" and chooseUser == "papel":

    print("Você ganhou, papel embrulha pedra" )

    winUser = 1
```



Pedra, Papel, Tesoura, Lagarto, Spock - Etapa III

Se você chegou até aqui, parabéns! Pronto, só precisamos somar o cálculo das pontuações ao final das verificações. Faremos assim:

```
userScore = userScore + userWin  
pcScore = pcScore + pcWin  
totalScore = userScore + pcScore
```



Conclusões

Vamos analisar o código com a mesa de trabalho e debater:

- 1) Analisando a função, por que há uma condição adicional se totalScore tem valor 0?
- 2) Vamos investigar sobre persistência de dados, como podemos tornar nosso programa persistente para salvar os dados gerados pelo sistema de pontuação?
- 3) Podemos melhorar nosso código? Como e quais alterações você faria?