Alunos

Ana Beatriz de Araújo Farias
Bianca Henrique Rangel
Maria Luiza Teixeira Nascmento

Descrições

- Não atendeu ao requisito
- Atendeu ao requisito

Cards vermelhos correspondem ao teste do requisito não atendido
Cards verdes correspondem ao teste do requisito atendido

Teste caixa preta

Número de tentativas

Caso o usuário queira digitar — **Sim**

Casos de teste

- 1. Obedecer os limites (1 a 6);
- 2. Tratamento de erro ao digitar letras;
- 3. Tratamento de erro ao digitar espaços vazios (nenhuma entrada);
- 4. Tratamento de erro ao digitar símbolos.

1. Está aceitando valores fora do intervalo

Valor testado = 0 e 7

Resultado =

```
Digite a quantidade de tentativas
Digite a tamanho da palavra
[]
```

Esperado = Mensagem com feedback e a possibilidade do usuário digitar a quantidade de tentativas novamente.

2. Ao digitar letras e símbolos o código exibe erro

Valor testado = a e .

resultado =

```
Traceback (most recent call last):
  File "main.py", line 20, in <module>
    tentativas = int(input("Digite a quantidade de tentativas\n"))
                 ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
ValueError: invalid literal for int() with base 10: '.'
```

esperado = Mensagem com feedback e a possibilidade do usuário digitar a quantidade de tentativas novamente.

3. Um dos requisitos seria a possibilidade do usuário não digitar o número de tentativas

Valor testado =

Resultado =

```
Digite a quantidade de tentativas
Traceback (most recent call last):
  File "main.py", line 20, in <module>
    tentativas = int(input("Digite a quantidade de tentativas\n"))
                 ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
ValueError: invalid literal for int() with base 10: ''
[]
```

Esperado = Jogo continuaria e por padrão o número de tentativas seria 6.

Caso o usuário não queira digitar — **Não**

Casos de teste

- 1. Há possibilidade do usuário digitar o número de tentativas?
- 2. Por padrão o número de tentativas tem que ser 6.

Foi observado que não existe a possibilidade do usuário NÃO escolher o número de tentativas. Ou seja, ele precisa informar um valor de toda forma.

Tamanho da palavra

Casos de teste

- 1. Não é permitido uso de símbolos e Letras;
- 2. Espaço vazio (O usuário não informar nenhuma entrada);
- 3. Tamanho dentro do limite. (Mínimo 2).

1. Ao digitar letras e símbolos o código exie erro

Valor testado = x e [

Resultado =

```
Digite a quantidade de tentativas
Traceback (most recent call last):
  File "main.py", line 21, in <module>
    tamanho_palavra = int(input("Digite o tamanho da palavra\n"))
                      ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
ValueError: invalid literal for int() with base 10: '['
[]
```

Esperado = Mensagem com feedback e a possibilidade do usuário digitar o tamanho novamente.

2. Erro quando o usuário não digita nada, ou seja, não informa o tamanho

Valor testado =

Resultado =

```
Digite a quantidade de tentativas
Traceback (most recent call last):
  File "main.py", line 21, in <module>
    tamanho_palavra = int(input("Digite o tamanho da palavra\n"))
                      ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
ValueError: invalid literal for int() with base 10: ''
[]
```

Esperado = Caso o usuário por engano não informe o valor, exibir feedback e possibilitar que o tamanho seja informado novamente.

3. Erro ao digitar valores fora do intervalo

Valor testado = 0 e 24

Resultado =

```
Digite a quantidade de tentativas
Digite a tamanho da palavra
Traceback (most recent call last):
  File "main.py", line 24, in <module>
    lista = buscar_palavra(tamanho_palavra)
            ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "main.py", line 6, in buscar_palavra
    for i in range(len(aux)):
             ^^^^^^^^^^^^^^^
TypeError: object of type 'int' has no len()
[]
```

Esperado = Mensagem com feedback e a possibilidade do usuário informar o tamanho da palavra novamente.

Inserir a palavra

Casos de teste

- 1. Não é permitido numeração;
- 2. Não é permitido espaço vazio;
- 3. Palavras dentro dos limites do tamanho;

1. Atendeu corretamente o teste

Valor testado = 222

Resultado =

```
Digite a quantidade de tentativas
Digite a tamanho da palavra
Digite a palavra
Palavra inválida. Digite novamente
Digite a palavra
[]
```

Esperado = Mensagem com feedback e a possibilidade do usuário digitar a palavra novamente.

2. Atendeu corretamente o teste

Valor testado =

Resultado =

```
Digite a quantidade de tentativas
Digite a tamanho da palavra
Digite a palavra
Tamanho inválido. Tente novamente
Digite a palavra
[]
```

Esperado = Mensagem com feedback e a possibilidade do usuário digitar a palavra novamente.

3. Atendeu corretamente o teste

Valor testado =

Resultado =

```
Digite a quantidade de tentativas
Digite a tamanho da palavra
Digite a palavra
Tamanho inválido. Tente novamente
Digite a palavra
[]
```

Esperado = Mensagem com feedback e a possibilidade do usuário digitar a palavra novamente.

Validar palavra

Casos de teste

- 1. Sinalização da existência da letra na posição certa;
- 2. Sinalização da existência da letra na posição errada;
- 3. Mostrar notificação caso o usuário não consiga acertar a palavra;
- 4. Mostrar notificação caso o usuário acerte a palavra;
- 5. Sinalização da não existência da letra na palavra;
- 6. Aceitar a palavra mesmo sem acentuação;
- 7. Não é permitido palavra inexistente.
- 8. Ignorar formatação (Letra maiuscula ou minuscula)

1. Atendeu corretamente o teste

Valor testado = dei

Resultado =

```
Digite a palavra
dei
ei
```

Esperado = Letra marcada correspondendo a existencia e o lugar correto na palavra.

2. Atendeu corretamente o teste

Valor testado = mala

Resultado =

```
Digite a palavra
mala
mala
Digite a palavra
[]
```

Esperado = Letra marcada correspondendo o lugar errado na palavra.

3. Atendeu corretamente o teste

Valor testado = Digitar última tentativa

Resultado =

```
Perdeu
Que pena, você não conseguiu adivinhar :(
```

Esperado = Mensagem após a última tentativa informando que o usuário perdeu.

4. Deve ser mostrado uma menagem que o usuário venceu assim que ele acertar a palavra

Valor testado = arar

Resultado =

```
Digite a palavra
arar
arar
Digite a palavra
arar
Ganhou
Parabéns!!! Você acertou a palavra
```

Esperado = Mensagem ganhou na primeira vez que o usuário digitou a palavra correta.

5. Atendeu corretamente o teste

Valor testado = lua

Resultado =

```
Digite a palavra
lua
lua
Digite a palavra
[]
```

Esperado = Letra marcada de uma cor ou não marcada correspondendo que não existe na palvra.

6. Caso o usuário digite uma letra sem acentuação e ela contenha acentuação reconher a letra mesmo assim

Valor testado =

Resultado =

```
Digite a palavra
Barão
Ba ão
Digite a palavra
[]
Palavra inválida. Digite novamente
```

Esperado = Que a letra seja reconhecida na palavra mesmo sem sua pontuação.

7. Atendeu corretamente o teste

Valor testado = aaa

Resultado =

```
Digite a palavra
aaa
Palavra inválida. Digite novamente
Digite a palavra
[]
```

Esperado = Mensagem informando que a palavra é inválida

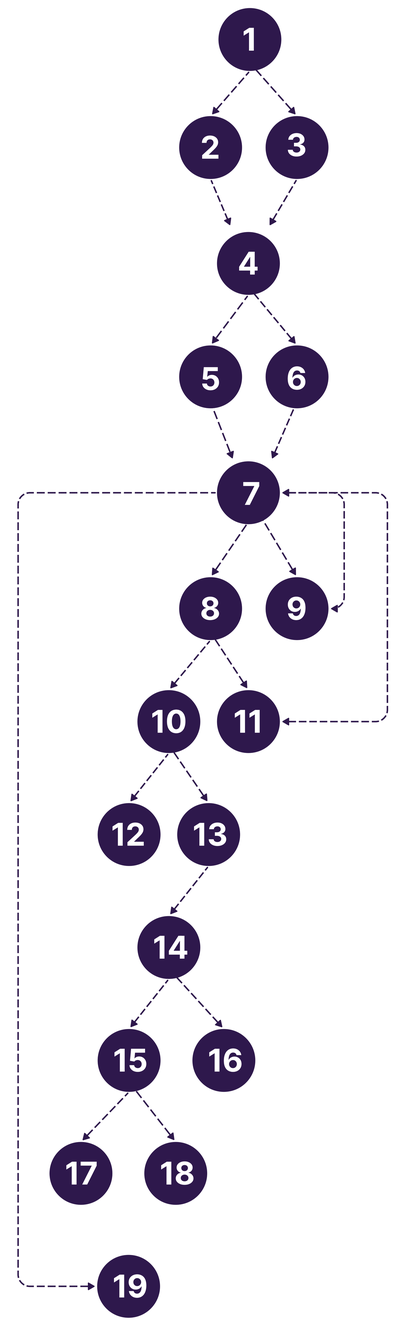
8. Reconhecer letra ou palavra indepente da sua formatação (maiuscula ou minsucula)

Valor testado = aarão

Resultado =

```
Digite a palavra
aarão
Palavra inválida. Digite novamente
Digite a palavra
Aarão
Aarão
```

Esperado = Que a palavra seja válida.

Código

```
def transformar_em_lista(palavra):
    lista = []
    for i in range(len(palavra)):
        lista.append(palavra[i])
    return lista

def buscar_palavra(tamanho):
    lista_palavras = []
    with open('palavras.txt','r',encoding="utf-8") as arq:
        palavras = arq.read()
        aux = palavras.split('\n')
        for i in range(len(aux)):
            if len(aux[i])==tamanho:
                lista_palavras.append(aux[i])
    return lista_palavras

def verificar_existencia(palavra):
    aux_arq = ""
    with open('palavras.txt','r',encoding="utf-8") as arq:
        palavras = arq.read()
        aux_arq = palavras.split('\n')
        if palavra  in aux_arq:
            return
        else:
            return False

def sortear_palavra(lista):
    lista_aux=[]
    palavra = random.choice(lista)
    for i in palavra:
        lista_aux.append(i)
    return lista_aux


#-------------------------------------------
#codigo

print(f.titulo)

print(f.info)

tentativas = int(input("Digite a quantidade de tentativas\n"))
tamanho_palavra = int(input("Digite o tamanho da palavra\n"))

lista = buscar_palavra(tamanho_palavra)

palavra = sortear_palavra(lista)

ganhou = False
c = 0

while c < tentativas:
    aux = input("Digite a palavra\n")
    aux_palavra = palavra.copy()
    if len(aux) == tamanho_palavra :
        if verificar_existencia(aux)==True:
            if ganhou == True:
                break
            else:
                tentativa_palavra = transformar_em_lista(aux)
                if tentativa_palavra == palavra:
                    ganhou = True
                for i in range(tamanho_palavra):
                    if tentativa_palavra[i] in aux_palavra:
                        if tentativa_palavra[i] == palavra[i]:
                            tentativa_palavra[i] = GREEN + tentativa_palavra[i] + RESET
                        else:
                            tentativa_palavra[i] = YELLOW + tentativa_palavra[i] + RESET
                        aux_palavra[i] = ""
                    else:
                        tentativa_palavra[i] = RED + tentativa_palavra[i] + RESET
                for i in tentativa_palavra:
                    print(i, end="")
                print()
                c+=1
        else:
            print("Palavra inválida. Digite novamente\n")
    else:
        print("Tamanho inválido. Tente novamente\n")


if ganhou == True:
    print(f.ganhou)
    print ("Parabéns!!! Você acertou a palavra\n")

else:
    print(f.perdeu)
    print("Que pena, você não conseguiu adivinhar :(\n")
```