

## Turbinando o gerador de palavras



80%

ATIVIDADES  
3 de 7DISCORD  
ALURAFÓRUM DO  
CURSOVOLTAR  
PARA  
DASHBOARDMODO  
NOTURNO

17.3k xp



### Transcrição

Após algumas reflexões sobre o corretor, poderemos chegar em algumas conclusões; uma delas diz respeito à variável `palavra` que é a palavra “lógica”.

```
palavra = "lógica"
```

[COPIAR CÓDIGO](#)

Precisaremos corrigir este termo equivocado. Para isso, eliminaremos a letra “i” para obtermos o correto “lógica”. Nesta correção, realizamos uma única operação que deleta apenas um caractere.

Porém, às vezes poderemos ter um problema em que sejam necessárias mais operações para corrigirmos a palavra digitada equivocadamente.

Por exemplo, se escrevermos erroneamente “lóiigica”, precisaremos deletar o caractere “i” duas vezes, e o nosso corretor ainda não é capaz de fazer isso.

Logo, notaremos que o nosso corretor só corrige palavras que estão a **uma única distância** da correta. Então construiremos uma função que corrigirá as que estão a **duas distâncias** do resultado correto, e verificará qual será o retorno do novo corretor.



80%

ATIVIDADES  
3 de 7

DISCORD  
ALURA

FÓRUM DO  
CURSO

VOLTAR  
PARA  
DASHBOARD

MODOS  
NOTURNO



17.3k xp

a

Para isso, criaremos um novo gerador de palavras chamado `gerador_turbinado()` com `def`, o qual receberá as `palavras_geradas` do próprio `gerador_palavras` como parâmetro.

Esta função retornará os novos vocábulos corrigidos a duas distâncias da palavra digitada equivocadamente, ou seja, `novas_palavras`.

Pegaremos cada uma das palavras geradas pelo `gerador_palavras` com `for`, aplicando novamente o gerador para `palavra`.

No caso da palavra “lóiiigica”, o `gerador_palavras` conterà a palavra `'lóiigica'` como resultado. A partir daí, o `gerador_turbinado()` removerá o outro caractere “i” a mais, gerando a palavra correta `"lógica"`.

Então, para cada `palavra` das `palavras_geradas`, chamaremos a função `gerador_palavras()` novamente e colocaremos tudo dentro de `novas_palavras` que será igual a uma lista.

Não aplicaremos `.append()` no gerador, pois seu retorno será uma lista. Logo, a concatenaremos com `novas_palavras` usando `+=`.

Com isso, retornaremos as novas palavras corrigidas a duas distâncias da digitada equivocadamente.

Para testarmos, chamaremos a nova variável temporária `palavras_g` ao final do bloco, chamando o `gerador_turbinado()` que receberá as `palavras_geradas()` com a palavra digitada equivocadamente “lóiiigica”, ou seja, a `palavra`.

Agora queremos saber se o termo “lógica” está de fato em `palavras_g`. Caso o retorno seja verdadeiro, o algoritmo de gerador de palavras turbinado corrigirá este tipo de erro.



80%

ATIVIDADES  
3 de 7

DISCORD  
ALURA

FÓRUM DO  
CURSO

VOLTAR  
PARA  
DASHBOARD

MODO  
NOTURNO



17.3k xp

a

```
palavra = "lóiigica"
```

```
def gerador_turbinado(palavras_geradas):  
    novas_palavras = []  
    for palavra in palavras_geradas:  
        novas_palavras += gerador_palavras(palavra)  
    return novas_palavras
```

```
palavras_g = gerador_turbinado(gerador_palavras(palavra))  
"lógica" in palavras_g
```

COPIAR CÓDIGO

Rodando o código, teremos o resultado `True` .

Portanto, nosso `gerador_turbinado()` está corrigindo palavras com até duas de distância da correta.

Para descobrirmos o tamanho desta nova lista, aplicaremos `len()` das `palavras_g` na célula seguinte.

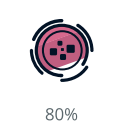
```
len(palavras_g)
```

COPIAR CÓDIGO

Como retorno, veremos que gerou `787396` variações de uma única palavra, ou seja teremos um volume extra de palavras geradas.

Sempre fazer a correção com esta grande quantidade de resultados talvez não faça sentido, pois demoraria muito.

A seguir, veremos uma maneira de eliminar as palavras que não existem dentro de `palavras_geradas` .



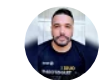
ATIVIDADES  
3 de 7

DISCORD  
ALURA

FÓRUM DO  
CURSO

VOLTAR  
PARA  
DASHBOARD

MODO  
NOTURNO



17.3k xp

a