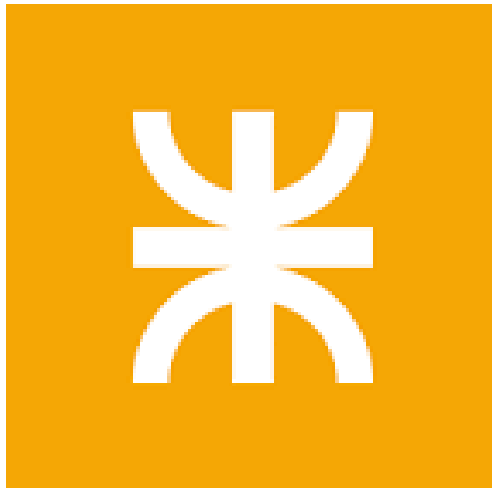


TRABAJO INTEGRADOR INDIVIDUAL

Algoritmos y estructuras de datos – 2023



FRVM - UTN

Ingeniería en Sistemas de Información

Docente:

- Ing. Simieli Paola.

Estudiante:

- Oliva Milagros Victoria

ÍNDICE

Nivel 1..... 3

 Pensamiento computacional..... 3

 Pseudocódigo: 3

 Código: 3

Nivel 2..... 4

 Pensamiento computacional..... 4

 Pseudocódigo: 4

 Código: 4

Nivel 3..... 5

 Pensamiento computacional..... 5

 Pseudocódigo: 5

 Código: 5

Nivel 1

El usuario va a ingresar un texto por teclado y debemos determinar cual es la palabra “feliz”.

Una palabra feliz se la considera aquella que tiene dos vocales iguales seguidas en dicha palabra. Por ejemplo: cooperativa, booleano.

Pensamiento computacional

- Para realizar esta consigna pienso en una función que me determine si la palabra es feliz o no,
- Para ello recorro el texto carácter por carácter, comparándolo con otro texto donde tengo guardadas las vocales, en caso de tener dos consecutivas y que sean vocales, la función se vuelve verdadera, si no, es falsa.

Pseudocódigo:

- Defino una variable que contenga las vocales.
- Empiezo un ciclo repetitivo de acuerdo a la cantidad de caracteres que tenga la palabra y evalúo las condiciones
 - Que sean dos vocales iguales y consecutivas.
- Si eso se cumple, la función retorna “true”, de lo contrario, retorna “false”.
- Por fuera de la función le pido al usuario que ingrese el texto y lo separo en palabras con la función “split()”, guardando cada una en un vector.
- Recorro el vector palabra por palabra y llamo a la función para que verifique si se cumple, en caso de cumplirse se imprime la palabra feliz; caso contrario se imprime un mensaje de que no se encontró palabra feliz.

Código:

```
def palabra_feliz(palabra):
    vocales = 'aeiouAEIOU'
    for i in range(len(palabra)-1):
        if palabra[i] == palabra[i+1] and palabra[i] in vocales:
            return True
    return False

texto = str(input('Ingrese su texto: '))
vector_palabras = texto.split()

for palabra in vector_palabras:
    if palabra_feliz(palabra):
        print('la palabra feliz es ', palabra)
    else:
        print('No es palabra feliz')
```

Nivel 2

Dada una matriz $m \times n$ la cual se la considera un diccionario de palabras se solicita encontrar en dicho diccionario cuantas palabras felices se encuentran en él.

Una palabra feliz se la considera aquella que tiene dos vocales consecutivas iguales en dicha palabra. Por ejemplo: cooperativa, booleano.

Pensamiento computacional

- Adapto la función creada en el nivel anterior a este nivel, trabajando con la matriz.
- Recorro la matriz palabra por palabra y verifico las condiciones, que sean dos vocales consecutivas e iguales

Pseudocódigo:

- Defino dos variables, una para las vocales y otra para inicializar en 0 el contador de palabras felices.
- Inicio un ciclo repetitivo for doble para recorrer la matriz y guardo en otra variable la palabra de esas coordenadas.
 - Para i en rango m
 - Para j en rango n
 - $Palabra = matriz[i][j]$
- Luego recorro dicha palabra carácter por carácter con otro ciclo for,
 - For carácter en $range(len(palabra))$
- Y evalúo las condiciones:
 - Si $palabra[carácter] = palabra[carácter+1]$ y $palabra[carácter]$ está en vocales
 - (incremento el contador) $cantidad_palabras_felices += 1$
- Fuera de la función le pido al usuario que me de las dimensiones de su matriz, la defino y con un ciclo for la recorro para que la complete.
- Por último, imprimo la cantidad de palabras felices llamando a la función y dándole como parámetro la matriz recién cargada.

Código:

```
def cant_esfeliz(matriz):
    vocales = 'aeiouAEIOU'
    cantidad_palabras = 0
    for i in range(m):
        for j in range(n):
            palabra = matriz[i][j]
            for caracter in range(len(palabra)-1):
                if palabra[caracter] == palabra[caracter+1] and palabra[caracter] in
vocales:
                    cantidad_palabras += 1
    return cantidad_palabras

m = int(input('Ingrese la cantidad de columnas:'))
n = int(input('ingrese la cantidad de filas: '))
```

```
matriz = [[str() for ind0 in range(n)] for ind1 in range(m)]

for i in range(m):
    for j in range(n):
        matriz[i][j] = str(input('Ingrese su palabra: '))

print ('La cantidad de palabras felices es: ', cant_esfeliz(matriz))
```

Nivel 3

Dado un archivo el cual contiene un diccionario de palabras, se solicita encontrar en ese diccionario la o las palabras felices, mostrándolo por pantalla y en otro archivo.

Pensamiento computacional

- Reutilizo la función del nivel 1 nuevamente, pero esta vez implementándola en un archivo.
- Mi pensamiento fue planteado para el caso que el **archivo este escrito línea por línea**.

Pseudocódigo:

- Utilizo la función antes creada para encontrar la palabra feliz.
- Creo una segunda función para poder encontrar la palabra en el archivo. En un vector guardo cada palabra feliz que encuentre mediante un ciclo for, leyendo LINEA POR LINEA.
- Luego guardo cada posición del vector en un nuevo archivo.
- Por ultimo imprimo el vector en forma de lista utilizando la función join.

Código:

```
#Dado un archivo el cual contiene un diccionario de palabras, se solicita encontrar en ese
#diccionario la o las palabras felices, mostrándolo por pantalla y en otro archivo.

def palabra_feliz(palabra): #utilizo la misma función del nivel 1 para encontrar la
palabra.
    vocales = 'aeiouAEIOUáéíóú'
    for i in range(len(palabra)-1):
        if palabra[i] == palabra[i+1] and palabra[i] in vocales:
            return True
    return False

def encontrar_palabras(archivo): #creo una nueva función para encontrar las palabras en un
archivo escrito línea por línea
    vector_felices= []
    for linea in archivo.readlines():
```

```
        if palabra_feliz(linea): #llamo a la función antes creadas y creo una lista con
las palabras felices
            vector_felices.append(linea)
    for i in range(len(vector_felices)):
        archivo2.write(vector_felices[i]) #guardo cada palabra feliz en un nuevo archivo.
    lista = ''.join(vector_felices)
    print(lista)

archivo1 = open('palabras.txt', 'r')
archivo2 = open('palabras_felices.txt', 'a+')

print(encontrar_palabras(archivo1))
archivo1.close()
archivo2.close()
```