

Guía de Ejercicios Prácticos - Ficha 11

Sitio: [Universidad Virtual UTN FRC](https://uv.frc.utn.edu.ar)
Curso: Algoritmos y Estructuras de Datos (2020)
Libro: Guía de Ejercicios Prácticos - Ficha 11

Imprimido por: Luciana Lisette Montarce
Día: lunes, 23 de noviembre de 2020, 21:22



Descripción

Esta guía contiene enunciados de algunos ejercicios para aplicar los conceptos de programación en **Python** que se analizan en la **Ficha 11**. Los alumnos no deben subir nada al aula virtual: la guía se propone como fuente de ejercicios generales. Se sugiere intentar resolver cada uno de estos problemas ya sea trabajando solos o en grupos de estudio, y cuando las soluciones se publiquen, controlar lo hecho con las sugerencias propuestas por sus docentes. Utilice el foro del curso para plantear dudas y consultas, que cualquier alumno puede intentar responder.



Tabla de contenidos

1. Problema 1

1.1. Solucion

2. Mayusculas

2.1. Solucion

3. Ejercicio lalelilolu

3.1. Solución

3.2. Solución sin menu

4. Parcial 1 [1k01/05/14] 2017

4.1. general.py

4.2. principal.py

5. Parcial 1 [1k06] 2017

5.1. Solución

6. Parcial 2 [1k03] 2018

6.1. Solución

7. Parcial 2 [1k10] 2018

7.1. Solución



1. Problema 1

Se solicita procesar un texto caracter a caracter. Las palabras del texto se separan con espacios en blanco y el fin del texto se indica con un punto.

A partir del texto se pide:

- a. Determinar la cantidad de palabras que comienzan con la expresión 'SI'.
- b. Determinar la cantidad de palabras que terminan con vocal y tienen una cantidad impar de letras.
- c. Determinar la cantidad de palabras que tienen sólo una vocal.
- d. Determinar la cantidad de palabras que comienzan y terminan con la misma letra.
- e. Determinar la cantidad de palabras que contienen la expresión 'CC'.
- f. Determinar el porcentaje que representan las palabras del punto b sobre el total de palabras.
- g. Determinar la longitud de la palabra más corta.
- h. Determinar el promedio de letras por palabra.



1.1. Solucion



```
__author__ = 'Catedra de Algoritmos y Estructuras de Datos'

# determina si car es una vocal en mayuscula...
def is_vocal(car):
    if car in 'AEIOU':
        return True

    return False

# calcula el porcentaje y el promedio...
def calcular(cp, cv, ac):
    pc = 0
    pm = 0
    if cp > 0:
        pc = cv * 100 / cp
        pm = ac / cp

    return pc, pm

# funcion principal del programa...
def test():
    print("PROCESADOR DE TEXTO\n")

    # inicializamos variables
    ss = ssi = scc = False
    cvocal = clet = cc = cpal = cif = 0
    csi = cvi = clv = ccc = acu = 0
    primera = ultima = None
    menor = 0

    # carga del texto y conversión a mayuscula...
    texto = input("Ingrese el texto a procesar (finalice con un punto): ")
    texto = texto.upper()

    # procesamiento del texto...
    for car in texto:
        if car != " " and car != ".":
            clet += 1

        # detector SI
        if car == "S" and clet == 1:
            ss = True

        else:
            if car == 'I' and ss:
                ssi = True
            else:
                ss = False

        # detector vocales
        if is_vocal(car):
            cvocal += 1

        # inicio - fin
        if clet == 1:
            primera = car

        ultima = car

    # detector cc
    if car == "C":
```

```
        cc += 1
        if cc == 2:
            scc = True

    else:
        cc = 0

# fin de palabra
else:
    if clet > 0:
        cpal += 1

        if cpal == 1:
            menor = clet

        elif clet < menor:
            menor = clet

        if ssi:
            csi += 1

        if cvocal == 1:
            clv += 1

        if is_vocal(ultima) and clet % 2 == 1:
            cvi += 1

        if scc:
            ccc += 1

    acu += clet

    if primera == ultima:
        cif += 1

    clet = cc = cvocal = 0
    ss = ssi = scc = False

# calculamos promedios y porcentajes
porc, prom = calcular(cpal, cvi, acu)

# visualización de resultados...
print("Cantidad de palabras que comienzan con la expresión \"SI\":", csi)
print("Cantidad que termina en vocal y con cantidad impar de letras:", cvi)
print("Cantidad con una única vocal:", clv)
print("Cantidad que comienza y termina con la misma letra:", cif)
print("Cantidad que contiene la expresión \"CC\":", ccc)
print("Porcentaje terminada en vocal y total impar de letras:", porc, "%")
print("Longitud de la palabra más corta:", menor)
print("Promedio de letras por palabra:", prom)

# script principal...
test()
```

2. Mayusculas

Desarrollar un programa que permita ingresar un texto por teclado, con palabras separadas por un espacio y terminado en punto. En base al texto ingresado, determinar:

- Cantidad de palabras que empiezan con mayúscula.
- Cantidad de números del 0 al 9 en todo el texto.
- Cantidad de palabras que tienen más de una e.
- Promedio de letras por palabra, para las palabras de longitud impar.

Ejemplo: Me llamo Elena y naci el 10 de febrero de 1990.

Palabras que empiezan con mayúscula: 2

Numeros del 0 al 9 en todo el texto: 6

Palabras que tienen más de una e: 2

Promedio de letras por palabra, para las palabras de longitud impar: 4.5



2.1. Solucion



```
__author__ = 'Catedra de Algoritmos y Estructuras de Datos'

def calcular_promedio (suma, total):
    if total != 0:
        promedio = round(suma / total, 2)
    else:
        promedio = 0
    return promedio

def es_mayuscula(letra):
    if letra >= 'A' and letra <= 'Z':
        return True
    else:
        return False

def es_numero(letra):
    numeros = '0123456789'
    if letra in numeros:
        return True
    else:
        return False

def test():
    print('Análisis de Texto')
    print('*' * 80)

    #Inicialización
    letras_pal = 0
    palabras_mayusc = 0
    cant_numeros = 0
    cant_e = 0
    palabras_e = 0
    palabras_impar = 0
    letras_impar = 0

    #Carga de datos y proceso
    texto = input('Ingrese el texto a analizar, separando las palabras con un espacio y terminando con punto: ')

    for letra in texto:
        if letra==' ' or letra==',.':
            #Tiene más de una e?
            if cant_e > 1:
                palabras_e += 1
            #Longitud impar? Acumular letras y contar
            if letras_pal % 2 != 0:
                palabras_impar += 1
                letras_impar += letras_pal
            #Reiniciar los indicadores de palabra
            letras_pal = 0
            cant_e = 0
        else:
            #Contar letras de la palabra
            letras_pal += 1
            #Empieza con mayúscula?
            if letras_pal==1 and es_mayuscula(letra):
                palabras_mayusc += 1
            #Contar si es un número
            if es_numero(letra):
                cant_numeros += 1
            #Contar letras e
            if letra == 'e' or letra == 'E':
                cant_e += 1

    #Resultados
    print('*' * 80)
```

```
print('Palabras que empiezan con mayúscula:',palabras_mayusc)
print('Numeros del 0 al 9 en todo el texto:',cant_numeros)
print('Palabras que tienen más de una e:',palabras_e)
promedio = calcular_promedio(letras_impar,palabras_impar)
print('Promedio de letras por palabra, para las palabras de longitud impar:',promedio)

# Script principal
test()
```



3. Ejercicio lalelilolu

Escribir un programa guiado por menú de opciones que permita cargar un texto por teclado, el texto termina con '.' y las palabras se separan por espacios.

Luego a partir del texto cargado informar:

1. Promedio general de letras por palabra del texto.
2. Cantidad de palabras terminadas en vocal
3. Orden de la palabra más larga del texto.
4. Cantidad de palabras que contengan al menos una vez la sílaba l+vocal (es decir la, le, li, lo o lu).



3.1. Solución



```
__author__ = 'Catedra de Algoritmos y Estructuras de Datos'

def es_vocal(letra):
    vocales = ('a', 'e', 'i', 'o', 'u')
    return letra in vocales

def leer_texto():
    texto = input('Ingrese el texto (termina con "." y las palabras se separan por espacio)\n')
    while(texto[-1] != '.'):
        print('Error: el texto debe terminar con ".")')
        texto = input('Ingrese el texto (termina con "." y las palabras se separan por espacio)\n')
    texto = texto.lower()
    return texto

def leer_entero_en_rango(lim_inferior, lim_superior, mensaje):
    resp = int(input(mensaje))
    while(resp < lim_inferior or resp > lim_superior):
        print('Error: el numero debe estar entre ' + str(lim_inferior) + ' y ' + str(lim_superior) + ', vuelva a intentarlo.')
        resp = int(input(mensaje))

    return resp

def mostrar_menu():
    print('* * 40')
    print('      Procesador de Textos')
    print('      Opciones:')
    print('* * 40')
    print(' 1 - Cargar el Texto')
    print(' 2 - Mostrar el promedio de letras por palabra del texto')
    print(' 3 - Cantidad de palabras terminadas en vocal')
    print(' 4 - Orden de la palabra más larga')
    print(' 5 - Cantidad de palabras que contienen 1 + <vocal>')
    print(' 6 - Salir')

def pausa():
    input('\n\nPresione enter para continuar...\n')

def test():

    #Iniciación de contadores y acumuladores Generales
    texto_cargado = False
    acumulador_letras = 0
    contador_palabras = 0
    cont_pal_term_vocal = 0
    cont_pal_l_vocal = 0
    orden_mas_larga = 0
    letras_mas_larga = 0

    opcion = 0
    while (opcion != 6):
        mostrar_menu()
        opcion = leer_entero_en_rango(1, 6, 'Ingrese su opción: ')

        if (opcion == 1): # Cargar el texto

            # Inicializaciones
            contador_letras = 0
            ultima_letra = ' '
            vino_l = False
            vino_l_vocal = False

            texto = leer_texto()
```

```

for car in texto:
    if (car != ' ' and car != '.'): # Por verdadero estoy procesando las letras dentro de una palabra
        # Cuento las letras de la palabra
        contador_letras += 1
        # Me guardo cada letra para tener la última cuando la palabra termine
        ultima_letra = car

        # Detecto si vino l + vocal
        if (car == 'l'):
            vino_l = True
        else:
            if (vino_l and es_vocal(car)):
                vino_l_vocal = True
            vino_l = False
    else: # Por el falso esto al final de una palabra (o entre 2 palabras, parado en el espacio)
        # Cuento las palabras
        contador_palabras += 1

        # Acumulo las letras para tener el total de letras para el promedio
        acumulador_letras += contador_letras

        # Cuentos las palabras con la última letra vocal
        if (es_vocal(ultima_letra)):
            cont_pal_term_vocal += 1

        # Chequeo si la palabra anterior fue más larga y guardo los datos si así fuera
        if (contador_palabras == 1 or contador_letras > letras_mas_larga):
            letras_mas_larga = contador_letras
            orden_mas_larga = contador_palabras

        #Cuento las palabras con l + vocal
        if (vino_l_vocal):
            cont_pal_l_vocal += 1

        # Vuelvo al inicio los contadores y banderas para la próxima palabra
        contador_letras = 0
        vino_l = False
        vino_l_vocal = False

print('\n\nTexto procesado correctamente...\n\n')
texto_cargado = True
pausa()

elif(opcion == 6): # Chau
    print('Gracias por usar el Procesador de Textos!!!')
    print('Fin!.')
else:
    if (texto_cargado):
        if (opcion == 2): # Mostrar promedio
            if (contador_palabras > 0):
                promedio = acumulador_letras / contador_palabras
            else:
                promedio = 0
            print('El promedio de letras por palabra fue:', promedio)
            pausa()
        elif (opcion == 3): # Mostrar terminadas en vocal
            print('La cantidad de palabras terminadas en vocal fue:', cont_pal_term_vocal)
            pausa()
        elif (opcion == 4): # Mostrar terminadas en vocal
            print('La ', orden_mas_larga, '° fue la palabra más larga, y tuvo ', letras_mas_larga, ' letras', sep='')
            pausa()
        elif (opcion == 5): # Mostrar terminadas en vocal
            print('La cantidad de palabras que incluyeron "l + <vocal>" fue:', cont_pal_l_vocal)
            pausa()
    else:
        print('Debe cargar el texto primero ingresando a la opción 1')

```

```
    pausa()  
test()
```

Descargar proyecto de la solución [aquí](#).



3.2. Solución sin menu



```
__author__ = 'Felipe'

"""
Programa que soluciona el enunciado tipo parcial 1 publicado como Ejercicio 3 en la Guía 11
"""

def es_vocal(letra):
    vocales = ('a', 'e', 'i', 'o', 'u')
    return letra in vocales

def leer_texto():
    texto = input('Ingrese el texto (termina con "." y las palabras se separan por espacio)\n')
    while(texto[-1] != '.'):
        print('Error: el texto debe terminar con ".")
        texto = input('Ingrese el texto (termina con "." y las palabras se separan por espacio)\n')
    texto = texto.lower()
    return texto

def pausa():
    input('\n\nPresione enter para continuar...\n')

def test():

    #Inicialización de contadores y acumuladores Generales
    texto_cargado = False
    acumulador_letras = 0
    contador_palabras = 0
    cont_pal_term_vocal = 0
    cont_pal_l_vocal = 0
    orden_mas_larga = 0
    letras_mas_larga = 0
    contador_letras = 0
    ultima_letra = ' '
    vino_l = False
    vino_l_vocal = False

    texto = leer_texto()

    for car in texto:
        if (car != ' ' and car != '.'): # Por verdadero estoy procesando las letras dentro de una palabra
            # Cuento las letras de la palabra
            contador_letras += 1

            # Detecto si vino l + vocal
            if (car == 'l'):
                vino_l = True
            else:
                if (vino_l and es_vocal(car)):
                    vino_l_vocal = True
                vino_l = False

            # Me guardo cada letra para tener la última cuando la palabra termine
            ultima_letra = car

        else: # Por el falso esto al final de una palabra (o entre 2 palabras, parado en el espacio)
            # Cuento las palabras
            contador_palabras += 1

            # Acumulo las letras para tener el total de letras para el promedio
            acumulador_letras += contador_letras

            # Cuentos las palabras con la última letra vocal
            if (es_vocal(ultima_letra)):
                cont_pal_term_vocal += 1

            # Chequeo si la palabra anterior fue más larga y guardo los datos si así fuera
```

```
        if (contador_palabras == 1 or contador_letras > letras_mas_larga):
            letras_mas_larga = contador_letras
            orden_mas_larga = contador_palabras

        #Cuento las palabras con l + vocal
        if (vino_l_vocal):
            cont_pal_l_vocal += 1

        # Vuelvo al inicio los contadores y banderas para la próxima palabra
        contador_letras = 0
        vino_l = False
        vino_l_vocal = False

    print('\n\nTexto procesado correctamente...\n\n')
    texto_cargado = True
    pausa()

    # Mostrar promedio
    if (contador_palabras > 0):
        promedio = acumulador_letras / contador_palabras
    else:
        promedio = 0
    print('El promedio de letras por palabra fue:', promedio)
    pausa()
    # Mostrar terminadas en vocal
    print('La cantidad de palabras terminadas en vocal fue:', cont_pal_term_vocal)
    pausa()
    # Mostrar terminadas en vocal
    print('La ', orden_mas_larga, '° fue la palabra más larga, y tuvo ', letras_mas_larga, ' letras', sep='')
    pausa()
    # Mostrar terminadas en vocal
    print('La cantidad de palabras que incluyeron "l + <vocal>" fue:', cont_pal_l_vocal)
    pausa()

test()
```

4. Parcial 1 [1k01/05/14] 2017

Se pide desarrollar un programa en Python que permita cargar por teclado un texto completo en una variable de tipo cadena de caracteres. El texto finaliza con '.' y se supone que el usuario cargará el punto para indicar el final del texto, y que cada palabra de ese texto está separada de las demás por un espacio en blanco. El programa debe:

- 1- Determinar la cantidad de palabras que terminaron en vocal. Por ejemplo, en el texto: "En el mar Dios me escucha.", tiene 2 palabras terminadas en vocal.
- 2- El porcentaje de consonantes y el porcentaje de vocales en todo el texto (tener en cuenta que puede haber otros caracteres): "La universidad es una etapa mas de la vida entre los 18 y los 25 años." Contiene 27 consonantes, 23 vocales y 54 caracteres en total.
- 3- Determinar qué palabra tuvo la mayor cantidad de consonantes del texto y mostrar su número de orden entendiendo que la primera palabra tiene orden 1. Por ejemplo, en el texto: "Los mandriles de Brasil son material de estudio.", la palabra "mandriles" con 6 consonantes es la que más consonantes tiene y su número de orden es 2.
- 4- Determinar la cantidad de palabras que comenzaron con primera letra de todo el texto y además incluyeron "st". Por ejemplo, en el texto: "En este parcial estamos evaluando lógica.", encontramos 2 palabras que cumplen la condición "este" y "estamos".



4.1. general.py

```
__author__ = 'Catedra de Algoritmos y Estructuras de Datos'

def es_vocal(car):
    vocales = 'aeiouAEIOU'
    if car in vocales:
        return True
    else:
        return False

def es_consonante(car):
    consonantes = 'bcdfghjklmnñpqrstvwxyzBCDFGHJKLMNÑPQRSTVWXYZ'
    if car in consonantes:
        return True
    else:
        return False

def calcular_porcentaje(cant,total):
    if total == 0:
        return 0
    else:
        return cant * 100 / total
```

4.2. principal.py

```
from general import *

__author__ = 'Catedra de Algoritmos y Estructuras de Datos'

def principal():
    #Iniciación
    anterior = ''
    pal_fin_vocal = letras = consonantes = vocales = palabras = cons_pal = letras_pal = pal_comienza_st = 0
    mayor = None
    comienza = tiene_st = False

    #Carga de datos y proceso
    texto = input('Ingrese el texto a analizar, separando las palabras con un espacio y terminando con punto: ')
    texto = texto.lower()
    for letra in texto:
        if letra != ' ' and letra != '.':
            #Dentro de la palabra
            letras_pal += 1
            if es_vocal(letra):
                vocales += 1
            elif es_consonante(letra):
                cons_pal += 1
            if palabras >= 1 and letras_pal == 1 and letra == texto[0]:
                comienza = True
            if letra == 't' and anterior == 's':
                tiene_st = True
        else:
            #Final de la palabra
            palabras += 1
            letras += letras_pal
            consonantes += cons_pal
            if es_vocal(anterior):
                pal_fin_vocal += 1
            if palabras == 1 or cons_pal > mayor[0]:
                mayor = cons_pal, palabras
            if comienza and tiene_st:
                pal_comienza_st += 1
            #Reiniciar
            cons_pal = 0
            letras_pal = 0
            comienza = False
            tiene_st = False
            anterior = letra

    #Resultados
    print('Palabras terminadas en vocal:', pal_fin_vocal)
    porc_voc = calcular_porcentaje(vocales, letras)
    porc_cons = calcular_porcentaje(consonantes, letras)
    print('Hay', round(porc_voc, 2), '% de vocales y', round(porc_cons, 2), '% de consonantes')
    print('La palabra con más consonantes tiene', mayor[0], 'consonantes, y aparece en el orden', mayor[1])
    print('Palabras que comienzan con la primera letra del texto y contienen st:', pal_comienza_st)

if __name__ == '__main__':
    principal()
```

5. Parcial 1 [1k06] 2017

Se pide desarrollar un programa en Python que permita cargar por teclado un texto completo en una variable de tipo cadena de caracteres. El texto finaliza con '.' y se supone que el usuario cargará el punto para indicar el final del texto, y que cada palabra de ese texto está separada de las demás por un espacio en blanco. El programa debe:

1. Determinar la cantidad de palabras que incluyeron al menos 2 dígitos, por ejemplo, el texto: "Argentina ganó 2 mundiales en 1978 y en 1986." contiene 2 palabras con al menos 2 dígitos.
2. Determinar la cantidad de palabras que comienzan con "la", por ejemplo, el texto: "Las laderas de las montañas están labradas." contiene 4 palabras que comienzan con "la".
3. Determinar el promedio de letras de las palabras que cumplieron con el punto 2, por ejemplo, en el texto anterior "laderas" y "labradas" cumplieron la condición y su promedio de letras por palabra fue 7.5.
4. Determinar la cantidad de palabras que comenzaron con "ll" y además incluyeron alguna "v". Por ejemplo, en el texto: "Las lluvias se llevaron los llantos.", contiene 2 palabras que cumplen la condición.



5.1. Solución




```
__author__ = 'Catedra de Algoritmos y Estructuras de Datos'

def es_digito(letra):
    return letra in '0123456789'

def calcular_promerio(cantidad, total):
    prom = 0
    if total != 0:
        prom = round(cantidad / total, 2)
    return prom

def principal():
    texto = input('Ingrese el texto a procesar. Debe finalizar con punto: ')
    texto = texto.lower()
    cant_digitos = pal_digitos = pal_tiene_la = pal_ll_con_v = 0
    cont_letras = cont_letras_la = 0
    comienza_l = tiene_la = tiene_ll = tiene_v = False
    for letra in texto:
        if letra != ' ' and letra != '.':
            cont_letras += 1

            if es_digito(letra):
                cant_digitos += 1

            if cont_letras == 1 and letra == 'l':
                comienza_l = True
            else:
                if cont_letras == 2:
                    if comienza_l and letra == 'a':
                        tiene_la = True

                    if comienza_l and letra == 'l':
                        tiene_ll = True

                comienza_l = False

            if letra == 'v':
                tiene_v = True
        else:
            if cant_digitos >= 2:
                pal_digitos += 1

            if tiene_la:
                pal_tiene_la += 1
                cont_letras_la += cont_letras

            if tiene_ll and tiene_v:
                pal_ll_con_v += 1

            cant_digitos = 0
            tiene_la = False
            tiene_ll = False
            tiene_v = False
            cont_letras = 0

    prom = calcular_promerio(cont_letras_la, pal_tiene_la)

    print('La cantidad de palabra con al menos 2 digitos son:', pal_digitos)
    print('La cantidad de palabras que comienzan con \"la\" son:', pal_tiene_la)
    print('El promedio de letras que comienzan con \"la\" es:', prom)
    print('La cantidad de palabras con \"ll\" con alguna \"v\" son', pal_ll_con_v)
```

```
if __name__ == '__main__':  
    principal()
```



6. Parcial 2 [1k03] 2018

Se pide desarrollar un programa en Python que permita cargar por teclado un texto completo en una variable de tipo cadena de caracteres.

El texto finaliza con '.' y se supone que el usuario cargará el punto para indicar el final del texto, y que cada palabra de ese texto está separada de las demás por un espacio en blanco. El programa debe:

1. Determinar cuántas palabras tenían más de cuatro letras y contenían al menos una "n". Por ejemplo, en el texto: "La universidad es una etapa más en el camino.", hay 2 palabras que cumplen la condición ("universidad" y "camino").
2. Determinar el promedio de letras por palabra entre las que comenzaban con "t". Por ejemplo, en el texto: "Ahora tenemos otra tarea.", hay 2 palabras que comienzan con "t" ("tenemos" y "tarea") y suman un total de 12 letras, por lo que el promedio pedido es $p = 12 / 2 = 6$ letras por palabra.
3. Determinar cuántas palabras contenían una "a" y también una "s", pero no contenían una "e". Por ejemplo, en el texto: "Ahora estamos en octavos de final del mundial.", hay una palabra que cumple la condición ("octavos"). La palabra "estamos" tiene una "a" y una "s", pero no cuenta porque tiene también una "e".
4. Determinar cuántas palabras contenían al menos una vez la expresión "re" pero terminaban con la letra "o". Por ejemplo, en el texto: "El registro de goles ha revelado que el réferi se equivoca.". hay dos palabras que cumplen la condición ("registro" y "revelado"). La palabra "réferi" tiene la expresión "re", pero no cumple porque no termina en "o".



6.1. Solución



```
__author__ = 'Algoritmos y Estructuras de Datos'

def calcular_promedio(suma, cantidad):
    prom = 0
    if cantidad != 0:
        prom = round(suma / cantidad, 2)
    return prom

def principal():
    letras_pal = cp4ln = palabras_t = letras_palt = palabras_as = palabras_re = 0
    tiene_n = empieza_t = tiene_a = tiene_s = tiene_e = tiene_r = tiene_re = False
    ultima_letra = ''
    texto = input('Ingrese el texto, debe finalizar con punto: ')
    for letra in texto:
        if letra == ' ' or letra == '.':

            if letras_pal > 4 and tiene_n:
                cp4ln += 1

            if empieza_t:
                letras_palt += letras_pal
                palabras_t += 1

            if not tiene_e and tiene_a and tiene_s:
                palabras_as += 1

            if tiene_re and (ultima_letra == 'o' or ultima_letra == 'O'):
                palabras_re += 1

            letras_pal = 0
            tiene_n = empieza_t = tiene_a = tiene_s = tiene_e = tiene_r = False
        else:
            letras_pal += 1
            if letra == 'n' or letra == 'N':
                tiene_n = True

            if letras_pal == 1 and (letra == 't' or letra == 'T'):
                empieza_t = True

            if letra == 'a' or letra == 'A':
                tiene_a = True

            if letra == 's' or letra == 'S':
                tiene_s = True

            if letra == 'e' or letra == 'E':
                tiene_e = True

            if letra == 'r' or letra == 'R':
                tiene_r = True
            else:
                if tiene_r and (letra == 'e' or letra == 'E'):
                    tiene_re = True
                else:
                    tiene_r = False
            ultima_letra = letra

    print('La cantidad de palabras con mas de cuatro letras y '
          'contienen "\\n\\" son:', cp4ln)
    promedio = calcular_promedio(letras_palt, palabras_t)
    print('El promedio de letras por palabra entre las que '
          'comenzaban con "\\t\\" es', promedio)
    print('La cantidad palabras contenían una "\\a\\" y también una "\\s\\", pero no '
          'contenían una "\\e\\" son:', palabras_as)
    print('La cantidad palabras contenían al menos una vez la expresión "re" pero'
```

```
        ' terminaban con la letra "o" son:', palabras_re)  
  
if __name__ == '__main__':  
    principal()
```



7. Parcial 2 [1k10] 2018

Se pide desarrollar un programa en Python que permita cargar por teclado un texto completo en una variable de tipo cadena de caracteres. El texto finaliza con '.' y se supone que el usuario cargará el punto para indicar el final del texto, y que cada palabra de ese texto está separada de las demás por un espacio en blanco.

El programa debe:

1. Determinar cuántas palabras tenían por lo menos tres vocales y más de cuatro letras. Por ejemplo, en el texto: "La universidad es una etapa más de la vida.", hay dos palabras que cumplen la condición ("universidad" y "etapa").
2. Determinar la longitud de la palabra más corta de entre las que contenían una consonante en la segunda posición. Por ejemplo, en el texto: "Vamos Argentina el sábado contra Francia", las palabras "Argentina", "el" y "Francia" tienen una consonante en la segunda letra, y la longitud de la más corta de esas tres palabras es 2 (corresponde a la palabra "el").
3. Determinar cuántas palabras empiezan con "v" o con "p" y terminan con "n" o con "a". Por ejemplo, en el texto: "Ahora que vengan y nos ganen si pueden.", hay dos palabras que cumplen la condición ("vengan" y "pueden").
4. Determinar el porcentaje de palabras que contenían la expresión "ga" con respecto al total del palabras del texto. Por ejemplo, en el texto: "Ahora que vengan y nos ganen si pueden.", hay 2 palabras con la expresión "ga" ("vengan" y "ganen") y hay un total de 8 palabras en todo el texto, por lo que el porcentaje pedido es $pr = 2 * 100 / 8 = 25\%$

7.1. Solución




```
__author__ = 'Algoritmos y Estructuras de Datos'

def es_vocal(letra):
    vocales = 'aeiouAEIOU'
    if letra in vocales:
        return True
    else:
        return False

def es_letra(letra):
    if (letra >= 'a' and letra <= 'z') or (letra >= 'A' and letra <= 'Z'):
        return True
    else:
        return False

def es_consonante(letra):
    if es_letra(letra) and es_vocal(letra) == False:
        return True
    else:
        return False

def calcular_porcentaje(cantidad, total):
    if total != 0:
        porcentaje = cantidad * 100 / total
    else:
        porcentaje = 0
    return porcentaje

def principal():
    print('PROCESAMIENTO DE TEXTO')
    print('='*80)
    letras_pal = vocales_pal = 0
    segunda_consonante = empieza_vp = expresion_ga = False
    palabras_3voc4let = palabras_cons2 = menor_cons2 = palabras_vpna = palabras_ga = palabras = 0
    texto = input('Ingrese el texto, debe finalizar con punto: ')
    for letra in texto:
        if letra == ' ' or letra == '.':
            #Final de la palabra
            palabras += 1
            if vocales_pal >= 3 and letras_pal > 4:
                palabras_3voc4let += 1
            if segunda_consonante:
                palabras_cons2 += 1
                if palabras_cons2 == 1:
                    menor_cons2 = letras_pal
                elif letras_pal < menor_cons2:
                    menor_cons2 = letras_pal
            if empieza_vp and (anterior == 'n' or anterior == 'a'):
                palabras_vpna += 1
            if expresion_ga:
                palabras_ga += 1
            #Reiniciar variables de palabra
            segunda_consonante = empieza_vp = expresion_ga = False
            letras_pal = vocales_pal = 0
        else:
            #Dentro de la palabra
            letras_pal += 1
            if es_vocal(letra):
                vocales_pal += 1
            if letras_pal == 2 and es_consonante(letra):
                segunda_consonante = True
            if letras_pal == 1 and (letra == 'v' or letra == 'p'):
                empieza_vp = True
```

```
        if letra == 'a' and anterior == 'g':
            expresion_ga = True
        anterior = letra
#Resultados
print('=' * 80)
print('Palabras que tenían por lo menos tres vocales y más de cuatro letras:', palabras_3voc4let)
if palabras_cons2 > 0:
    print('Longitud de la palabra más corta de entre las que contenían una consonante en la segunda posición:', menor_cons2)
else:
    print('No había palabras con una consonante en la segunda posición')
print('Palabras que empiezan con "v" o con "p" y terminan con "n" o con "a":', palabras_vpna)
porcentaje = calcular_porcentaje(palabras_ga, palabras)
print('Porcentaje de palabras que contenían la expresión "ga":', round(porcentaje, 2), '%')

if __name__ == '__main__':
    principal()
```