

Guía de Ejercicios Prácticos - Ficha 07

Sitio: [Universidad Virtual UTN FRC](https://uv.frc.utn.edu.ar)

Curso: Algoritmos y Estructuras de Datos (2020)

Libro: Guía de Ejercicios Prácticos - Ficha 07

Imprimido por: Luciana Lisette Montarce

Día: lunes, 23 de noviembre de 2020, 21:17



Descripción

Esta guía contiene enunciados de algunos ejercicios para aplicar los conceptos de programación en **Python** que se analizan en la **Ficha 07**. 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. Ciclistas

1.1. Solución

2. Secuencia de impares

2.1. Solución

2.2. Solución (otra variante)

3. Sueldos y aguinaldo

3.1. Solución (con secuencia)

3.2. Solución (con range)

4. Decimal a Hexadecimal

4.1. Solucion

5. Analisis de Texto

5.1. Solucion General

5.2. Solución Variante 1

6. Números Enteros

6.1. Solución

7. Sílabo 'sa'

7.1. Solución

8. Puntos en un plano

8.1. Solucion

9. Números: Mayor y Menor

9.1. Solución

10. Caracteres: Promedio de Letras.

10.1. Solución

11. Caracteres: Sílabo 'ci'

11.1. Solución

12. Caracteres: Longitudes Menores

12.1. Solución

13. Vocales y consonantes

13.1. Solución

14. Ejercicio Estadística de Guardería Náutica

14.1. Solución

15. Sílabo 'dre'

15.1. Solución

16. Sílabo 'mo'

16.1. Solución

1. Ciclistas

La final de una carrera de ciclistas tiene n competidores (n se ingresa por teclado).

Desarrollar un programa que permita cargar, por cada competidor, nombre y tiempo de carrera. Luego se pide:

- a) Determinar y mostrar el nombre del ganador de la carrera.
- b) Ingresar por teclado el tiempo record registrado para dicha carrera. Determinar si el tiempo del ganador es menor al tiempo record, mostrar un mensaje.
- c) Calcular y mostrar el tiempo promedio entre todos los ciclistas.



FILADD.COM

1.1. Solución

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

print('CARRERA DE CICLISTAS')
print('-' * 40)
cont = acum = 0
ganador = None

n = int(input('Ingrese la cantidad de ciclistas que participan de la carrera: '))

for i in range(n):
    print('Ciclista', i)
    nombre = input('Ingrese nombre: ')
    tiempo = int(input('Ingrese tiempo: '))
    if ganador is None or tiempo < ganador[1]:
        ganador = nombre, tiempo
    cont += 1
    acum += tiempo

print('-' * 40)
if n > 0:
    record = int(input('Ingrese record actual: '))
    print('El ganador es', ganador[0])
    if ganador[1] < record:
        print('El ganador supero el record!')
        if cont > 0:
            promedio = round(acum / cont, 2)
        else:
            promedio = 0
        print('Tiempo promedio general', promedio)
else:
    print('No se ingresaron datos')
```

2. Secuencia de impares

Cargar por teclado dos números, e imprimir los números impares que se encuentran comprendidos entre ellos, en forma ascendente y descendente.



2.1. Solución

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

print('SECUENCIA DE IMPARES')
print('=' * 40)

num1 = int(input('Ingrese un número: '))
num2 = int(input('Ingrese otro: '))

#Identificamos menor y mayor para saber como ordenar la secuencia
if num1 < num2:
    men, may = num1, num2
else:
    men, may = num2, num1

#Ajustamos valores si son pares
if men % 2 == 0:
    men += 1
if may % 2 == 0:
    may -= 1

#Resultados
print('Secuencia ascendente:')
ascend = range(men, may + 1, 2)
for num in ascend:
    print(num, end=' | ')

print('\nSecuencia descendente:')
descend = range(may, men - 1, -2)
for num in descend:
    print(num, end=' | ')
```

2.2. Solución (otra variante)

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

print('SECUENCIA DE IMPARES')
print('=' * 40)

num1 = int(input('Ingrese un número: '))
num2 = int(input('Ingrese otro: '))

#Definimos límites de la secuencia ascendente
if num1 < num2:
    inicio, fin = num1, num2 + 1
else:
    inicio, fin = num2, num1 + 1

#Corregimos inicio si es par
if inicio % 2 == 0:
    inicio += 1

#Mostramos secuencia
print('Secuencia ascendente')
for num in range(inicio, fin, 2):
    print(num, end=' ')

#Definimos límites de la secuencia descendente
if num1 > num2:
    inicio, fin = num1, num2 - 1
else:
    inicio, fin = num2, num1 - 1

#Corregimos inicio si es par
if inicio % 2 == 0:
    inicio += 1

#Mostramos secuencia
print('\nSecuencia descendente')
for num in range(inicio, fin, -2):
    print(num, end=' ')
```


3. Sueldos y aguinaldo

Ingresa por teclado los sueldos de un vendedor, correspondientes al primer semestre del año y luego:

- a) Calcular su aguinaldo, sabiendo que es la mitad del sueldo más alto del período.
- b) Determinar en qué mes recibió el sueldo más bajo del período.
- c) Informar el sueldo promedio del semestre.



3.1. Solución (con secuencia)

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

print('SUELDOS Y AGUINALDO')
print('*' * 80)

#Datos y proceso
total = 0
primero = True
semestre = ("Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio")
for mes in semestre:
    sueldo = float(input("Ingrese sueldo de " + str(mes) + ": "))
    if primero==True:
        may = sueldo
        men = sueldo, mes
        primero = False
    else:
        if sueldo > may:
            may = sueldo
        if sueldo < men[0]:
            men = sueldo, mes
    total += sueldo

#Resultados
aguinaldo = may / 2
print("\nEl aguinaldo es de $",aguinaldo)
print("El menor sueldo fue de $",men[0],"y lo obtuvo en el mes de",men[1])
promedio = round(total/len(semestre),2)
print ("El sueldo promedio es de $",promedio)
```

3.2. Solución (con range)

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

print('SUELDOS Y AGUINALDO')
print('*' * 80)

#Datos y proceso
total = 0
for mes in range(1,7):
    sueldo = float(input("Ingrese sueldo mes " + str(mes) + ": "))
    if mes==1:
        may = sueldo
        men = sueldo,1
    else:
        if sueldo > may:
            may = sueldo
        if sueldo < men[0]:
            men = sueldo, mes
    total += sueldo

#Resultados
aguinaldo = may / 2
print ("\nEl aguinaldo es de $",aguinaldo)
print("El menor sueldo fue de $",men[0],"y lo obtuvo en el mes",men[1])
promedio = round(total/6,2)
print ("El sueldo promedio es de $",promedio)
```

4. Decimal a Hexadecimal

Generar n numeros aleatorios entre el rango de 5000 y 450000, por cada uno de ellos mostrar y generar el numero hexadecimal



4.1. Solucion

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

import random
n = int(input('Ingrese la cantidad de numeros a procesar: '))

for i in range(n):
    numero = random.randrange(5000, 450000)
    hexadecimal = ''
    #proceso de conversion hexadecimal
    valor = numero % 16
    siguiente = numero // 16
    digito = ''
    while valor > 0:
        if valor < 10:
            digito = str(valor)
        else:
            digito = chr(55 + valor)
        hexadecimal = digito + hexadecimal
        valor = siguiente % 16
        siguiente //= 16

    print('El numero ', numero, 'en hexadecimal es', hexadecimal)
```

5. Analisis de Texto

El usuario ingresa una frase al comenzar el programa, la misma no puede tener longitud cero. La frase finaliza con un punto, y las palabras están separadas por espacios únicamente.

Se debe mostrar:

- a) Ver el porcentaje de vocales respecto del total de letras de la frase.
- b) La longitud promedio de las palabras
- c) La longitud de la palabra mas larga del texto
- c) Cantidad de palabras que comienzan con "ta"



5.1. Solucion General

```

__author__ = 'Catedra de Algoritmos y Estructuras de Datos'

# Inicialización de variables
vocales = 'aeiouAEIOUáéíóú'
texto = ''
cv = cp = cl = ctl = may = cpTa = 0
hayT = hayTa = False

# Validacion de que el texto no tenga longitud 0
while len(texto) <= 0:
    texto = input('Ingrese el texto a analizar: ')
    if len(texto) <= 0:
        texto = input('Ingrese el texto a analizar: ')

# comienza el procesamiento del texto ingresado por el usuario
for c in texto:
    if c.isalpha(): # si es un caracter alfanumerico
        cl += 1 # cuenta las letras de la palabra

        if c in vocales: # pregunta si la letra es una vocal
            cv += 1 # cuenta las vocales de la frase

        if cl == 1 and c.upper() == 'T': # pregunta si es el primer caracter y ademas es igual a T.
            # c.upper() convierte el caracter c a mayusculas
            hayT = True # activa la bandera
        else:
            if hayT and c.upper() == 'A': # pregunta si el primer caracter era T y el actual es A
                hayTa = True # activa la bandera indicando que encontro la silaba TA al comienzo de la palabra
            hayT = False # desactiva la bandera porque no encontro la silaba TA al comienzo de la palabra

    else: # si no es un caracter alfanumerico (espacio, punto, etc.)

        if cl > 0: # si la cantidad de letras es mayor a 0
            cp += 1 # cuenta una palabra mas
            ctl += cl # incrementa el acumulador de letras de la frase en la cantidad de letras de la palabra
            if cl > may: # hace la busqueda de la palabra de mayor longitud
                may = cl # si encuentra un mayor, almacena la longitud de la palabra

        if hayTa: # si encontro la silaba TA
            cpTa += 1 # incrementa el contador de silabas TA que encontro
            hayTa = False # resetea la bandera para que pueda seguir buscando la silaba en las palabras subsiguientes
        cl = 0 # resetea el contador de letras de la palabra para que siga contando la cantidad de letras de las palabras
        subsiguientes

if ctl!=0: # valida que el denominador del calculo del porcentaje sea distinto de 0 para evitar una indeterminacion
    porc = round(cv * 100 / ctl,2) # Calcula el porcentaje de vocales con respecto a la cantidad de letras de la frase
else:
    porc=0 # si el denominador es 0, entonces el calculo del porcentaje se determina como igual a 0.

if cp!=0: # valida que el denominador del calculo del promedio sea distinto de 0 para evitar una indeterminacion
    prom = ctl // cp # calcula la longitud promedio de las palabras de la frase
else:
    prom=0 # si el denominador es 0, entonces el calculo del promedio se determina como igual a 0.

# Muestra de resultados
print('El porcentaje de vocales respecto del total de letras es de ', porc, '%')
print('Hay', prom, 'letras por palabra')
print('La palabra mas larga del texto tiene', may, 'letras')
print('Hay', cpTa, 'palabras que comienzan con "ta"')

```

5.2. Solución Variante 1

```

__author__ = 'Catedra de Algoritmos y Estructuras de Datos'

# Variante 2

# Inicialización de variables
vocales = 'aeiouAEIOUáéíóú'
abecedario='abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZáéíóú'
texto = ''
cv = cp = cl = ctl = may = cpTa = 0
hayT = hayTa = False

# Validacion de que el texto no tenga longitud 0
while len(texto) <= 0:
    texto = input('Ingrese el texto a analizar: ')
    if len(texto) <= 0:
        texto = input('Ingrese el texto a analizar: ')

# comienza el procesamiento del texto ingresado por el usuario
for c in texto:
    if c in abecedario: # si es un caracter alfanumerico
        cl += 1 # cuenta las letras de la palabra

        if c in vocales: # pregunta si la letra es una vocal
            cv += 1 # cuenta las vocales de la frase

        if cl == 1 and (c == 'T' or c == 't'): # pregunta si es el primer caracter y ademas es igual a T.
            # c.upper() convierte el caracter c a mayusculas
            hayT = True # activa la bandera
        else:
            if hayT and (c == 'A' or c == 'a'): # pregunta si el primer caracter era T y el actual es A
                hayTa = True # activa la bandera indicando que encontro la silaba TA al comienzo de la palabra
            hayT = False # desactiva la bandera porque no encontro la silaba TA al comienzo de la palabra

    elif c==' ' or c=='.': # si es un espacio o punto
        if cl > 0: # si la cantidad de letras es mayor a 0
            cp += 1 # cuenta una palabra mas
            ctl += cl # incrementa el acumulador de letras de la frase en la cantidad de letras de la palabra
            if cl > may: # hace la busqueda de la palabra de mayor longitud
                may = cl # si encuentra un mayor, almacena la longitud de la palabra

            if hayTa: # si encontro la silaba TA
                cpTa += 1 # incrementa el contador de silabas TA que encontro
                hayTa = False # resetea la bandera para que pueda seguir buscando la silaba en las palabras subsiguientes
            cl = 0 # resetea el contador de letras de la palabra para que siga contando la cantidad de letras de las palabras subsiguientes

if ctl!=0: # valida que el denominador del calculo del porcentaje sea distinto de 0 para evitar una indeterminacion
    porc = round(cv * 100 / ctl,2) # Calcula el porcentaje de vocales con respecto a la cantidad de letras de la frase
else:
    porc=0 # si el denominador es 0, entonces el calculo del porcentaje se determina como igual a 0.

if cp!=0: # valida que el denominador del calculo del promedio sea distinto de 0 para evitar una indeterminacion
    prom = ctl // cp # calcula la longitud promedio de las palabras de la frase
else:
    prom=0 # si el denominador es 0, entonces el calculo del promedio se determina como igual a 0.

# Muestra de resultados
print('El porcentaje de vocales respecto del total de letras es de ', porc, '%')
print('Hay', prom, 'letras por palabra')
print('La palabra mas larga del texto tiene', may, 'letras')
print('Hay', cpTa, 'palabras que comienzan con \"ta\"')

```




6. Números Enteros

Escribir un programa que permita leer la cantidad de números enteros ingresados por el usuario y calcular lo siguiente:

- a) El segundo menor
- b) El promedio de los números positivos.
- c) El mayor de los números negativos.



6.1. Solución

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

n=int(input("Ingrese la cantidad de nros. a procesar: "))
contador_positivos=0
suma_positivos=0
mayor_negativos=0

for i in range(n):

    #Entrada
    print("Ingrese el número ", i+1)
    numero=int(input())

    #Subproblema 1: Búsqueda del segundo menor
    if i==0:
        menor=numero
    elif i==1:
        if numero<menor:
            segundoMenor=menor
            menor=numero
        else:
            segundoMenor=numero
    else:
        if numero<menor:
            segundoMenor=menor
            menor=numero
        elif numero<segundoMenor:
            segundoMenor=numero

    #Subproblema 2: El promedio de los numeros positivos
    if numero>=0:
        contador_positivos+=1
        suma_positivos+=numero

    #Subproblema 3: El mayor de los numeros negativos.
    else:
        if mayor_negativos==0:
            mayor_negativos=numero
        elif mayor_negativos<numero:
            mayor_negativos=numero

if contador_positivos!=0:
    promedio=suma_positivos/contador_positivos
else:
    promedio=0

#Salida
print("El segundo menor es:",segundoMenor)
print ("El promedio de numeros positivos es: ",promedio)
print ("El mayor de los numeros negativos es: ",mayor_negativos)
```

7. Sílabas 'sa'

Cargar por teclado una frase, pero a razón de un carácter por vez en una variable. La frase debe terminar con un punto (al aparecer el punto, la carga debe finalizar). El programa debe informar:

- a) Promedio de letras por palabra.
- b) Cantidad de palabras que terminan con la letra 's' (minúscula).
- c) Cantidad de palabras que contienen a la sílaba 'sa' (minúscula).



7.1. Solución

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

print('Análisis de Texto')
print('*' * 80)

#Inicialización
letras = 0
palabras = 0
palS = 0
tieneSA= False
palSA = 0
cl=0

#Carga de datos y proceso
ant = None
act = None
while act != '.':
    act = input('Ingrese caracter (con . termina): ')
    if (act == ' ' or act == '.') and ant != None:
        if cl>0:
            palabras += 1
            #Termina en s?
            if ant == 's':
                palS += 1
            #Contiene la expresión sa?
            if tieneSA == True:
                palSA += 1
            #Reiniciar las variables de palabra
            tieneSA = False
            cl=0
        else:
            letras += 1
            cl+=1
            #Identificar expresión sa
            if act=='a' and ant=='s':
                tieneSA = True
    ant = act

#Resultados
print('*' * 80)
if palabras == 0:
    prom = 0
else:
    prom = letras/palabras
print('El promedio de letras por palabra es',prom)
print('Las palabras terminadas en S son',palS)
print('Las palabarar con la expresión SA son',palSA)
print("cant. Palabras ",palabras)
```

8. Puntos en un plano

Desarrollar un programa que permita ingresar las coordenadas de n puntos en el plano, e informe:

- a) En qué cuadrante se encuentra cada uno.
- b) Determinar cuántos puntos se encuentran en el primer o tercer cuadrante.
- c) Determinar cuál de todos los puntos cargados se encuentra a mayor distancia del origen de coordenadas.



8.1. Solucion

```
import math

__author__ = 'Catedra de Algoritmos y Estructuras de Datos'

print('Procesamiento de puntos en el plano')
print('-' * 80)
n = int(input('Ingrese la cantidad de puntos a procesar: '))

cant_primer_cuadrante = 0
cant_tercer_cuadrante = 0
mayor_distancia = 0
mayor_punto = ()

for vuelta in range(n):
    eje_x = float(input('Ingrese el valor de la abscisa del punto: '))
    eje_y = float(input('Ingrese el valor de la ordenada del punto: '))

    if eje_x > 0:
        if eje_y > 0:
            cuadrante = "primer"
            cant_primer_cuadrante += 1
        else:
            cuadrante = "cuarto"
    else:
        if eje_y > 0:
            cuadrante = "segundo"
        else:
            cuadrante = "tercer"
            cant_tercer_cuadrante += 1

    print('El punto (', eje_x, ',', eje_y, ') ',
          'se encuentra en el ', cuadrante, ' cuadrante', sep='')

    distancia_origen = math.sqrt(pow(eje_x, 2) + pow(eje_y, 2))
    if distancia_origen > mayor_distancia:
        mayor_distancia = distancia_origen
        mayor_punto = eje_x, eje_y

    print('-' * 80)

print('-' * 80)
print('La cantidad de puntos en el primer cuadrante fue de: ',
      cant_primer_cuadrante)
print('La cantidad de puntos en el tercer cuadrante fue de: ',
      cant_tercer_cuadrante)
print('El punto con mayor distancia al origen fue (',
      mayor_punto[0], ',', mayor_punto[1], ")", sep='')
```

9. Números: Mayor y Menor

Cargar por teclado n números enteros positivos, uno a uno. Se deberá establecer qué número es el mayor de los números pares y el menor de los números impares.

Por ejemplo, en una secuencia de números: 8, 15, 9, 2, 27, 18, 0; el mayor de los pares sería el número 18 y el menor de los impares el número 9.



9.1. Solución

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

print('Busqueda de Mayores Pares y Menores Impares')
print('-' * 80)

print('Se procesarán enteros positivos, con el cero se termina la carga: ')
numero = int(input('Ingrese un numero entero positivo: '))

mayor_par = 0
menor_impar = 0
primer_impar = True

while numero > 0:
    paridad = numero % 2
    if paridad == 0:
        if numero > mayor_par:
            mayor_par = numero
    else:
        if primer_impar or numero < menor_impar:
            primer_impar = False
            menor_impar = numero

    numero = int(input('Ingrese un nuevo numero entero positivo: '))

if mayor_par != 0:
    print('El mayor numero de los pares es:', mayor_par)
else:
    print('No ingreso numeros pares')

if menor_impar != 0:
    print('El menor numero de los impares es:', menor_impar)
else:
    print('No ingreso numeros impares')
```

10. Caracteres: Promedio de Letras.

Cargar por teclado una frase (de a un caracter por vez). La carga solo debe terminar cuando se ingrese un punto (el cual no forma parte de la frase a procesar). Se debe informar la cantidad de palabras de la frase y cantidad promedio de letras por palabra. Por ejemplo, la frase "Este es un ejercicio muy sencillo." tiene 6 palabras y la cantidad promedio de letras por palabra es de 4,66 ($[4+2+2+9+3+8]/6$).



10.1. Solución

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

print('Tratamiento de Caracteres, lectura caracter por caracter')
print('-' * 90)

cant_letras = 0
cant_palabras = 0
total_letras = 0
termino = False

print('Se procesara un texto caracter a carater, termina con punto')
caracter = input('Ingrese una letra: ')
while caracter != '.':
    if caracter.isalpha():
        cant_letras += 1
    else:
        cant_palabras += 1
        total_letras += cant_letras
        cant_letras = 0

    if termino:
        break

    caracter = input('Ingrese una letra: ')
    if caracter == '.':
        caracter = ' '
        termino = True

print('Resultado del proceso')
print('-' * 90)
print('Cantidad de palabras del texto:', cant_palabras)
promedio = 0
if cant_palabras > 0:
    promedio = total_letras / cant_palabras
print('El promedio de letras por palabra fue de', round(promedio, 2))
```

11. Caracteres: Sílabas 'ci'

Cargar por teclado una frase, cada palabra separada únicamente por espacio y el texto terminar cuando se ingrese un punto (el cual no forma parte de la frase a procesar). Se debe obtener la cantidad de palabras que poseen al menos una vez la sílaba "ci". Por ejemplo, la frase "Este es un ejercicio moderado." tiene 1 palabra que posee la sílaba "ci" ("ejercicio").



11.1. Solución

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

print('Tratamiento de Caracteres, lectura caracter por caracter')
print('-' * 90)

hay_c = False
hay_ci = False
cantidad_palabras_ci = 0

texto = input('Ingrese el texto a procesar: ')
texto = texto.upper()

for car in texto:
    if car != ' ' and car != '.':
        if car == 'C':
            hay_c = True
        else:
            if car == 'I' and hay_c:
                hay_ci = True
            hay_c = False
    else:
        if hay_ci:
            cantidad_palabras_ci += 1
        hay_ci = False

print('La cantidad de palabras que tienen ci son', cantidad_palabras_ci)
```

12. Caracteres: Longitudes Menores

Cargar por teclado una frase separada por espacios y termina con un punto (el cual no forma parte de la frase a procesar). Se debe establecer cuántas palabras de la frase están antecedidas por otra palabra de menor o igual longitud (la primer palabra de la frase nunca podrá ser contabilizada ya que no posee palabra antecesora).

Por ejemplo, la frase "Este es un ejercicio un poco complicado." tiene 4 palabras cuya antecesora es igual o más corta en longitud ("un", "ejercicio", "poco", "complicado").



12.1. Solución

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

print('Tratamiento de Caracteres, longitud de palabras')
print('-' * 90)

cl = 0
cl_previa = 0
cantidad_palabras = 0

texto = input('Ingrese el texto a procesar: ')
texto = texto.upper()

for car in texto:
    if car != ' ' and car != '.':
        cl += 1
    else:
        if cl_previa != 0 and cl_previa <= cl:
            cantidad_palabras += 1
        cl_previa = cl
        cl = 0

print('Resultados')
print('-' * 90)
print('Hay', cantidad_palabras, ' que son antecedidas por una de menor o igual longitud en el texto')
```

13. Vocales y consonantes

Cargar por teclado una frase, pero a razón de un caracter por vez en una variable. La frase debe terminar con un punto (al aparecer el punto, la carga debe finalizar). El programa debe informar

- a) Cuántas vocales existen en la frase
- b) Cuántas letras en total existen en la frase (no contar símbolos como comas, puntos y comas, números, etc).
- c) El porcentaje de vocales respecto al total de letras.



13.1. Solución

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

print('Tratamiento de Caracteres, lectura caracter por caracter')
print('-' * 90)

vocales = ('a','e','i','o','u','A','E','I','O','U')
cant_letras = 0
cant_vocales = 0

print('Se procesara un texto caracter a carater, termina con punto')
caracter = input('Ingrese una letra: ')
while caracter != '.':
    if caracter in vocales:
        cant_vocales += 1

    if caracter.isalpha():
        cant_letras += 1

    caracter = input('Ingrese una letra: ')

print('Resultado del proceso')
print('-' * 90)
print('En el texto se cargaron', cant_vocales, 'vocales')
print('En el texto tuvo', cant_letras, 'de letras')
if cant_letras > 0:
    porc = round(cant_vocales * 100 / cant_letras, 2)
    print('Y representan un ', porc, '% sobre el total de letas del texto', sep='')
```

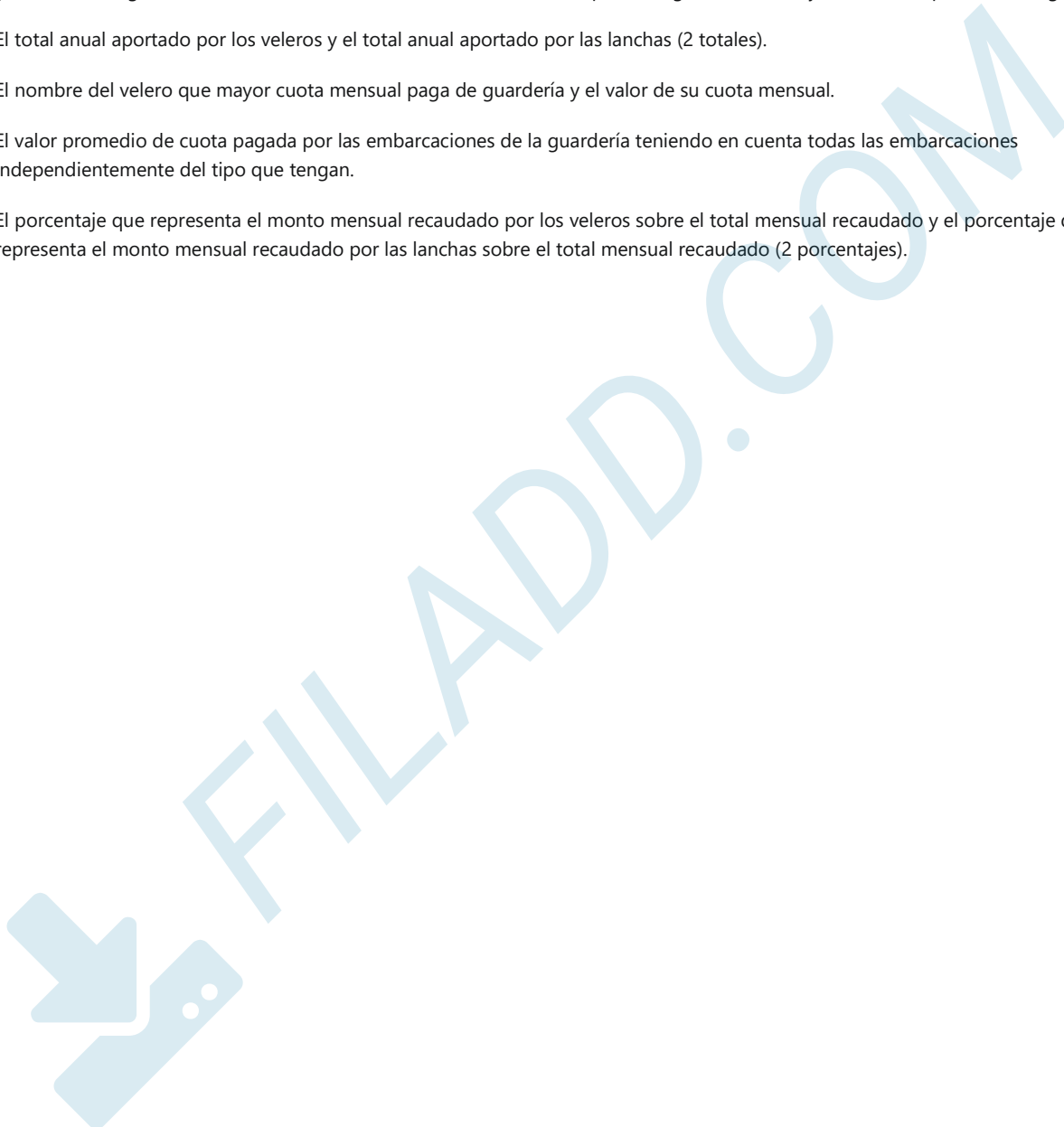
14. Ejercicio Estadística de Guardería Náutica

Un club náutico de la costa del lago San Roque necesita calcular estadísticas acerca de los barcos que tiene en la guardería.

Se pretende un programa que cargue uno por uno los datos de cada barco. De ellos se sabe el nombre, el tipo (1 si es velero, 2 si es lancha) y el monto que pagan por mes de guardería.

El programa debe cargar datos de los barcos de acuerdo a una cantidad n que se carga al comienzo y una vez completada la carga informar:

1. El total anual aportado por los veleros y el total anual aportado por las lanchas (2 totales).
2. El nombre del velero que mayor cuota mensual paga de guardería y el valor de su cuota mensual.
3. El valor promedio de cuota pagada por las embarcaciones de la guardería teniendo en cuenta todas las embarcaciones independientemente del tipo que tengan.
4. El porcentaje que representa el monto mensual recaudado por los veleros sobre el total mensual recaudado y el porcentaje que representa el monto mensual recaudado por las lanchas sobre el total mensual recaudado (2 porcentajes).



14.1. Solución



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

# Encabezado de la consola
print()
print('_' * 40)
print('Programa de estadísticas del Club Náutico')
print('_' * 40)
print()

# Leer la cantidad de datos a cargar (n que representa la cantidad de barcos)
n = int(input('Ingrese la cantidad de Embarcaciones a cargar: '))

# Segmento de inicialización de contadores acumuladores y banderas
total_anual_veleros = total_anual_lanchas = 0
primero = True
suma_total_mensual = suma_mensual_veleros = suma_mensual_lanchas = 0
cantidad = 0

# Comienza la carga y proceso de los datos
print()
print('Cargue los datos de las Embarcaciones...')
print()
for i in range(n):
    # Lectura de datos de cada barco
    nombre = input('Ingrese el nombre de la Embarcación: ')
    tipo = int(input('Ingrese el tipo de embarcación (1 - para Veleros o 2 - para Lanchas): '))
    importe_mensual = float(input('Ingrese el importe mensual pagado: '))

    if tipo == 1:
        # Incremento el total anual de los veleros
        # Esto también se podría resolver al final a partir del total mensual
        total_anual_veleros += importe_mensual * 12

        # Busco el velero con mayor cuota mensual
        # Utilizamos la alternativa que sirve para todos los casos por más que como dijimos en este caso había otras opciones
        if primero or importe_mensual > mayor_cuota:
            mayor_cuota = importe_mensual
            nombre_mayor = nombre
            primero = False

        # Incremento la suma mensual para los veleros para el cálculo de los porcentajes
        suma_mensual_veleros += importe_mensual
    else:
        # Incremento el total anual de las lanchas
        # También se podría resolver al final a partir del total mensual
        total_anual_lanchas += importe_mensual * 12

        # Incremento la suma mensual para las lanchas para el cálculo de los porcentajes
        suma_mensual_lanchas += importe_mensual

    # Incremento la suma total
    # (como también dijimos en clase este valor también se podía calcular porque en este caso en ambas ramas del if estamos sumando
    # cada importe
    suma_total_mensual += importe_mensual

    # Cuento la cantidad de barcos
    cantidad += 1

# Si se cargaron barcos, Calculo el promedio pedido en el punto 3
if cantidad > 0:
    promedio = suma_total_mensual / cantidad

# Calculo los porcentajes solicitados en el punto 4
porcentaje_veleros = suma_mensual_veleros * 100 / suma_total_mensual
porcentaje_lanchas = suma_mensual_lanchas * 100 / suma_total_mensual

# Muestro los resultados solicitados
```

```
print()
print('_' * 40)
print('Resultados estadísticos')
print('_' * 40)
print()

print('El total anual aportado por los Veleros es:', round(total_anual_veleros, 2))
print('El total anual aportado por las Lanchas es:', round(total_anual_lanchas, 2))

# Para el caso del mayor valido que se haya cargado algun velero y sino muestro un mensaje
print()
if not primero:
    print('El Velero que mayor cuota mensual paga es:', nombre_mayor, 'y paga:', mayor_cuota)
else:
    print('No se cargaron Veleros, por lo que no hay Velero con mayor cuota mensual...')

print()
print('La cuota mensual promedio abonada por todas las embarcaciones fue:', round(promedio, 2))

print()
print('De un total mensual recuadudado de:', suma_total_mensual)
print('Los Veleros aportaron un ' + str(round(porcentaje_veleros, 2)) + '%')
print('Las Lanchas aportaron un ' + str(round(porcentaje_lanchas, 2)) + '%')

# Muestro un mensaje de fin y solicito que presione enter para finalizar
print()
print('_' * 40)
print('Fin.')
print('_' * 40)
```

15. Sílabo 'dre'

Desarrollar un programa en Python que permita cargar por teclado un texto completo. Siempre se supone que el usuario cargará un 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:

- a) Determinar cuántas palabras tenían exactamente 3 letras.
- b) Determinar el porcentaje que las palabras del punto 1 representan en el total del palabras del texto.
- c) Determinar cuántas palabras terminaban con la letra "s".
- d) Determinar cuántas palabras contuvieron al menos una vez la expresión "dre".



15.1. Solucion

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

print('Análisis de texto con silabra dre')
print('-' * 80)

texto = input('Ingrese el texto a analizar, debe finalizar con punto: ')
anterior = ''
cant_letras = pal_3let = cant_pal = pal_terminan_s = pal_dre = 0
tiene_d = tiene_dr = tiene_dre = False
for letra in texto:
    if letra == ' ' or letra == '.':
        if cant_letras > 0:
            cant_pal += 1

            if cant_letras == 3:
                pal_3let += 1

            if anterior == 's':
                pal_terminan_s += 1

            if tiene_dre:
                pal_dre += 1

            tiene_dre = False
            cant_letras = 0
    else:
        cant_letras += 1
        if letra == 'd':
            tiene_d = True
            tiene_dr = False
        else:
            if letra == 'r' and tiene_d:
                tiene_dr = True
            else:
                if letra == 'e' and tiene_dr:
                    tiene_dre = True
                    tiene_dr = False
                tiene_d = False
        anterior = letra

porc = 0
if cant_pal != 0:
    porc = round(pal_3let * 100 / cant_pal, 2)

print('Presentacion de Resultados')
print('-' * 80)
print('Cantidad de palabras con exactamente tres letras:', pal_3let)
print('La cantidad de palabras que terminan con \'s\' son:', pal_terminan_s)
print('La cantidad de palabras que contienen \'dre\' son:', pal_dre)
print('El porcentaje de palabra de tres letras respecto del total de palabras del texto es: ', porc, '%', sep='')
```

16. Sílabo 'mo'

Desarrollar un programa en Python que permita cargar por teclado un texto completo. Siempre se supone que el usuario cargará un 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:

- a) Determinar cuántas palabras tenían más de 4 letras.
- b) Determinar cuántas palabras tenían al menos una vez la letra "x" o la letra "y".
- c) Determinar el promedio de letras por palabra en todo el texto.
- d) Determinar cuántas palabras contuvieron sólo una vez la expresión "mo".

Ejemplo: 'el mono momoxy toca el xilofon.'

Palabras con más de 4 letras: 2

Palabras tenían al menos una vez la letra "x" o la letra "y": 2

El promedio de letras por palabra en todo el texto es: 4.17

Determinar cuántas palabras contuvieron sólo una vez la expresión "mo": 1

16.1. Solución

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

print('Analisis de Texto silaba \"mo\"')
print('=' * 80)

texto = input('Ingrese un texto a analizar, debe finalizar con punto: ')
cont_letras = cont_pal_4letras = cont_palabras_xy = cont_mo = cont_palabras_mo = cont_palabras = total_letras = 0
tiene_xy = False
anterior = ''
for letra in texto:
    if letra == ' ' or letra == '.':
        if cont_letras > 0:
            cont_palabras += 1

            if cont_letras > 4:
                cont_pal_4letras += 1

            if tiene_xy:
                cont_palabras_xy += 1

            if cont_mo == 1:
                cont_palabras_mo += 1

        cont_mo = 0
        cont_letras = 0
        tiene_xy = False
    else:
        cont_letras += 1
        total_letras += 1
        if letra == 'x' or letra == 'y':
            tiene_xy = True

        if anterior == 'm' and letra == 'o':
            cont_mo += 1

        anterior = letra

promedio = 0
if cont_palabras > 0:
    promedio = round(total_letras / cont_palabras, 2)

print('La cantidad de palabras con mas de 4 letras es:', cont_pal_4letras)
print('El Promedio de letras por palabras es:', promedio)
print('La cantidad de palabras con silaba \"mo\" una vez es:', cont_palabras_mo)
print('La cantidad de palabras con letras \"x\" o \"y\" es:', cont_palabras_xy)
```