

# ARCHIVOS

## Fundamentos de Informática Grado en Ingeniería Química

Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicación

# ARCHIVOS

Hasta ahora, los programas leen los datos de teclado y escriben los resultados en pantalla.

Esto es suficiente cuando se leen/escriben pocos datos, pero cuando son muchos hay que usar archivos de lectura/escritura o entrada/salida (Input/Output).

Se llama entrada o lectura de datos al proceso de pasar esos datos desde un dispositivo de entrada a un programa. El dispositivo de entrada estándar o por defecto es el teclado, pero también se pueden usar archivos de lectura.

Se llama salida o escritura de datos al proceso de escribir esos datos, resultados de un programa, en un dispositivo de salida. El dispositivo de salida estándar es la pantalla pero también se pueden usar archivos de escritura.

# ABRIR UN ARCHIVO EN UN PROGRAMA

Para abrir un archivo de texto (.txt) en un programa, se usa la función `open` con dos argumentos básicos:

```
f = open(nombre, acceso)
```

*nombre* es una cadena que contiene el camino (path) y nombre del archivo de texto que se va a abrir. Si no hay camino, el archivo de texto se encuentra en la misma carpeta que el programa que lo ejecuta, el llamado directorio de trabajo. Si hay camino, se puede dar referencia absoluta o relativa al directorio de trabajo. Los separadores dependen del sistema operativo: en Windows \\ en Linux o Mac /

*acceso* puede ser uno de los caracteres siguientes:

'**r**' para leer (read), valor por defecto.

'**w**' para escribir (write), en cada ejecución se crea uno nuevo borrándose el anterior.

'**a**' para añadir al final de un archivo de escritura (append), en cada ejecución se actualiza añadiendo al final sin borrar. La primera vez que se ejecuta, se crea y se escribe. Las siguientes se abre y se añade.

`open` devuelve un 'canal de comunicación' `archivo_texto - programa` que se guarda en una variable *f*.

# CERRAR UN ARCHIVO EN UN PROGRAMA

Para cerrar un archivo de texto (.txt) en un programa previamente abierto para leer /escribir en él, se usa el método close.

```
f = open(nombre, acceso)
```

```
f.close()
```

El método close requiere una variable de archivo (previamente abierto con open) a la izquierda del método separado por punto y un paréntesis vacío.

Un archivo de lectura se abre, se lee y se cierra.

Un archivo de escritura se abre, se escribe y se cierra.

# LEER UN ARCHIVO DE TEXTO: for-in

Para leer un archivo de texto (.txt) previamente abierto con `open` en un programa, hay dos formas.

Primera forma:

```
fe = open(nombre, 'r')
```

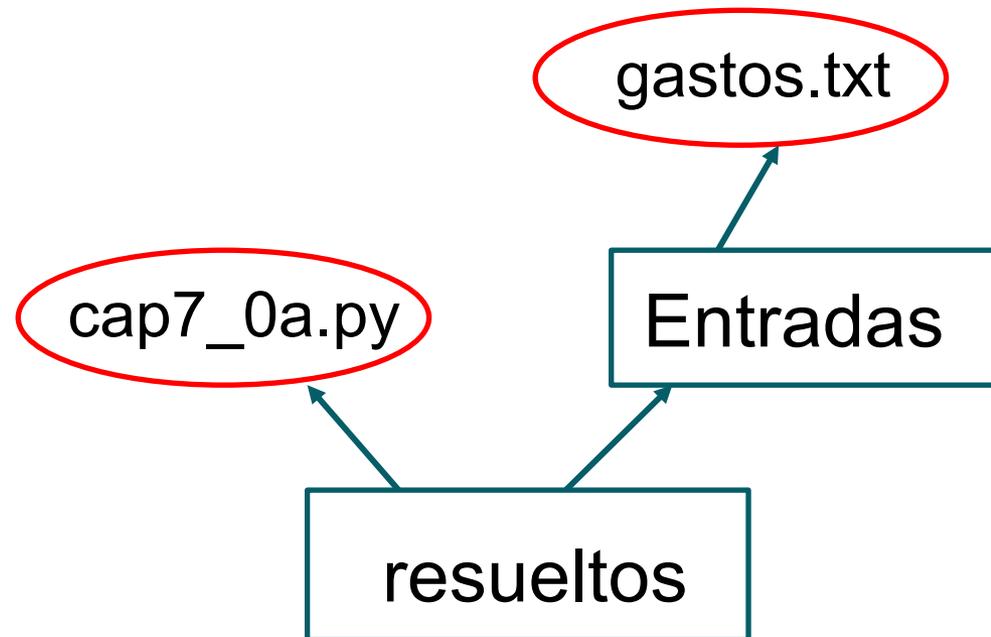
```
for linea in fe:
```

*fe* es iterable, por tanto, permite un `for`.

En cada iteración, *linea* es una cadena con todo el contenido de una línea. El último carácter de cada línea es `\n`.

# EJERCICIO 0\_a I

Construir un programa `cap7_0a.py` que lea el archivo de texto con nombre `gastos.txt` y escriba la suma de gastos en pantalla. La ubicación del programa `cap7_0a.py` y archivo de texto `gastos.txt` deben ser las siguientes:

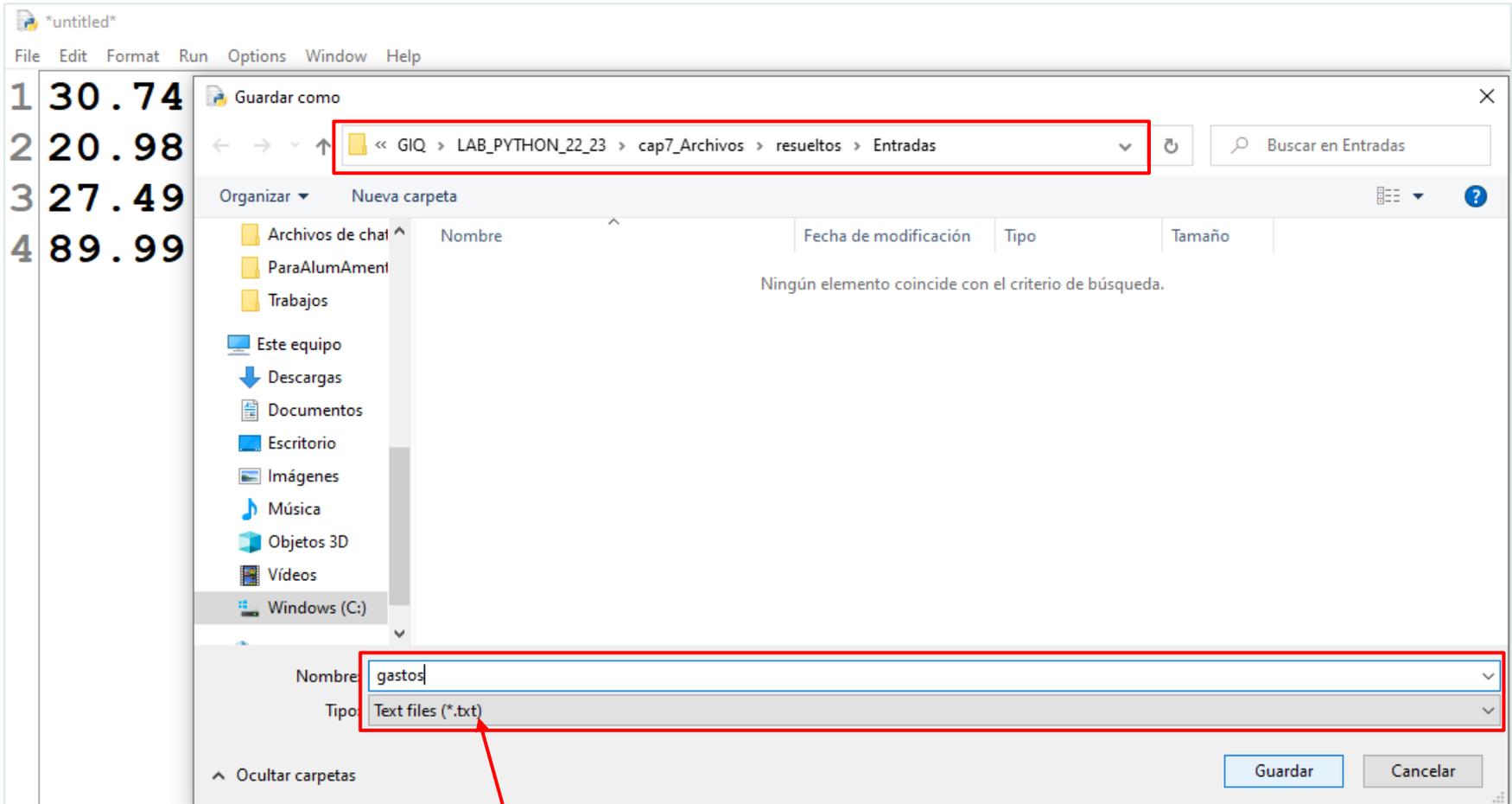


PASOS:

1. Crear la carpeta Entradas.
2. Editar con IDLE el archivo de texto `gastos.txt` y guardarlo en la carpeta Entradas.
3. Editar con IDLE el programa `cap7_0a.py` y guardarlo en la carpeta resueltos.
4. Ejecutar el programa.

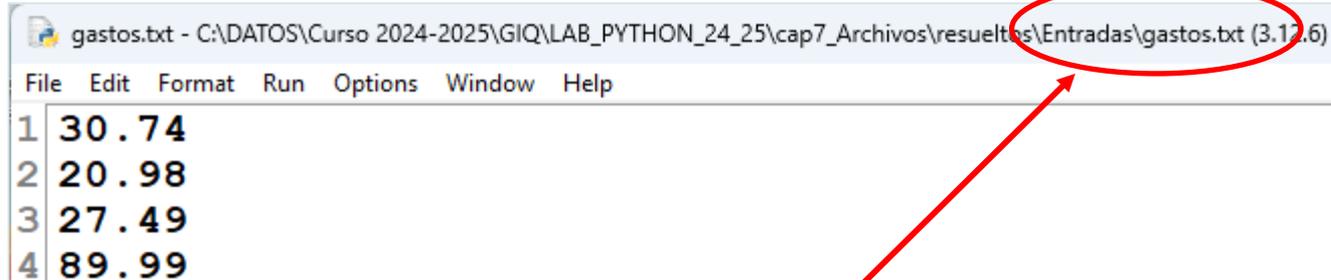
# EJERCICIO 0\_a II

Usando IDLE para editar el archivo de texto gastos.txt:



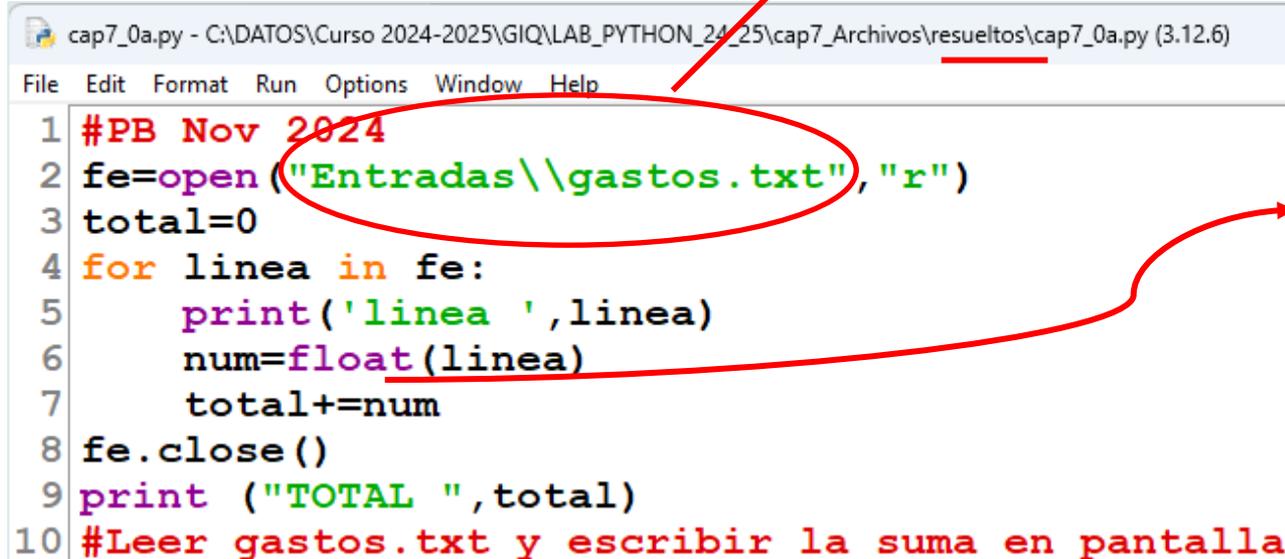
# EJERCICIO 0\_a III

Usando IDLE para editar el programa cap7\_0a.py:



A screenshot of a text editor window titled 'gastos.txt - C:\DATOS\Curso 2024-2025\GIQ\LAB\_PYTHON\_24\_25\cap7\_Archivos\resueltos\Entradas\gastos.txt (3.12.6)'. The window contains a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Window', and 'Help'. The text content is as follows:

```
1 30.74
2 20.98
3 27.49
4 89.99
```



A screenshot of a Python script editor window titled 'cap7\_0a.py - C:\DATOS\Curso 2024-2025\GIQ\LAB\_PYTHON\_24\_25\cap7\_Archivos\resueltos\cap7\_0a.py (3.12.6)'. The window contains a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Window', and 'Help'. The script content is as follows:

```
1 #PB Nov 2024
2 fe=open("Entradas\gastos.txt", "r")
3 total=0
4 for linea in fe:
5     print('linea ',linea)
6     num=float(linea)
7     total+=num
8 fe.close()
9 print ("TOTAL ",total)
10 #Leer gastos.txt y escribir la suma en pantalla
```

Red annotations include a circle around the file path in line 2 and an arrow pointing from the text on the right to the script.

Al convertir línea a real, desaparece \n automáticamente!

# LEER UN ARCHIVO DE TEXTO: método readlines

Segunda forma:

```
fe = open(nombre, 'r')
```

```
lc = fe.readlines()
```

El método `readlines` requiere una variable de archivo a la izquierda del método separado por punto y un paréntesis vacío a la derecha. Lee todo el contenido del archivo *fe* abierto previamente con `open` devolviendo una lista de cadenas, cada una de las cuales es una línea del archivo.

El último carácter de cada línea leída es `\n`.

El número de líneas del archivo se calcula con `len(lc)`.

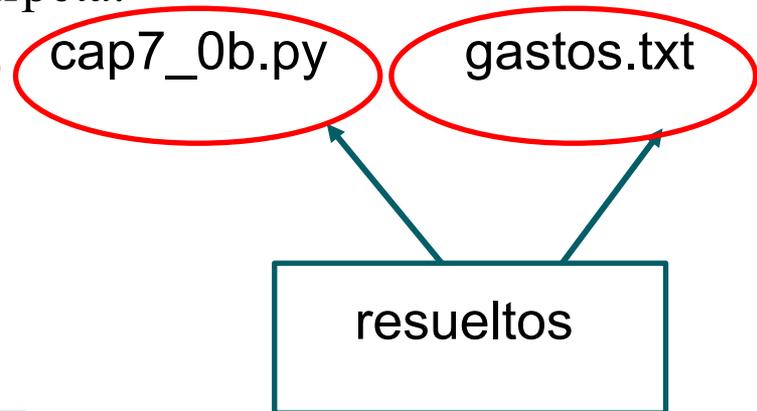
`cadena.split('separador')` es un método muy útil para separar los campos de una línea, si los hay, creando una lista con ellos. El último elemento de esa lista contendrá el carácter `\n`. Al convertir ese elemento a `int/float` (si procede) se elimina `\n` automáticamente.

# EJERCICIO 0\_b

Igual que la versión a pero usando el método readlines para leer el archivo gastos.txt y, en este caso, la ubicación del programa cap7\_0b.py y el archivo de texto gastos.txt debe ser la misma carpeta:

```
gastos.txt - C:\DATOS\Curso 2021-2022\GIQ\Lab_GIQ\cap7_Archivos\resueltos\gastos.txt |
File Edit Format Run Options Window Help
1 30.74
2 20.98
3 27.49
4 89.99
```

```
cap7_0b.py - C:/DATOS/Curso 2021-2022/GIQ/Lab_GIQ/cap7_Archivos/resueltos/cap7_0b.py |
File Edit Format Run Options Window Help
1 fe=open("gastos.txt","r")
2 total=0
3 lc=fe.readlines()
4 # print('lc ',lc)
5 for cadena in lc:
6     num=float(cadena)
7     total+=num
8 fe.close()
9 print("TOTAL ",total)
```



## PASOS:

1. Copiar gastos.txt a la carpeta resueltos.
2. Editar con IDLE el programa cap7\_0b.py y guardarlo en la carpeta resueltos.
3. Ejecutar el programa.

# ESCRIBIR EN UN ARCHIVO DE TEXTO

Para escribir en un archivo de texto (.txt) previamente abierto con `open` en un programa:

```
fs = open(nombre, 'w')
```

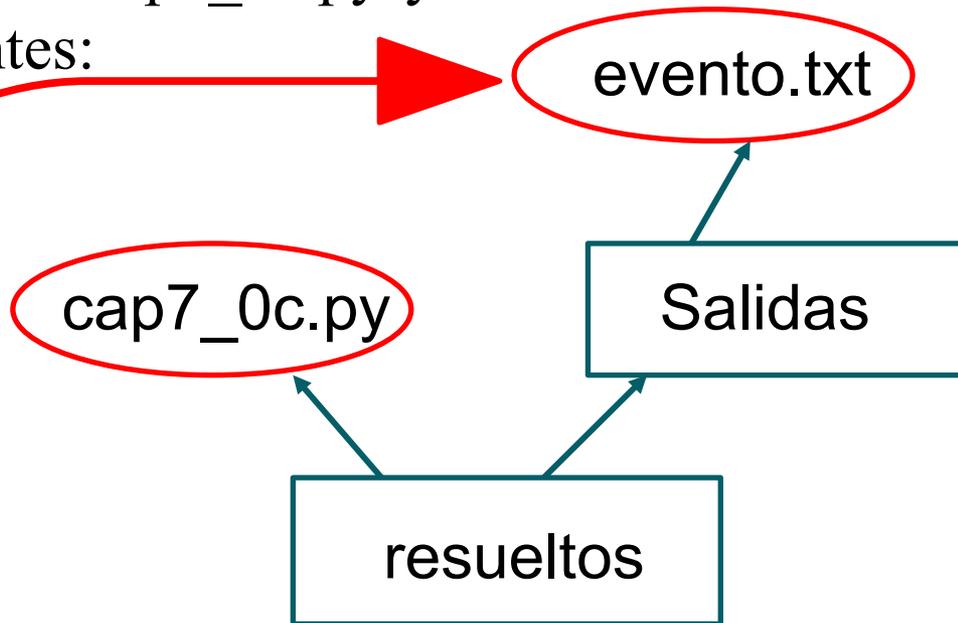
```
fs.write(cadena)
```

El método **write** requiere una variable de archivo a la izquierda del método separado por punto y un paréntesis a la derecha con la cadena que se quiere escribir.

Todo lo que se quiera escribir que no sea cadena hay que convertirlo a cadena con `str()` y concatenarlo a lo demás con el operador de concatenación `+`. Generalmente, el último carácter de la cadena es `\n` para escribir lo siguiente en una línea nueva.

# Ejercicio 0\_c I

Leer de teclado nombres hasta teclear FIN y escribirlos en un archivo de texto con nombre evento.txt. La ubicación del programa cap7\_0c.py y archivo de texto evento.txt deben ser las siguientes:



PASOS:

1. Crear la carpeta Salidas.
2. Editar con IDLE el programa cap7\_0c.py y guardarlo en la carpeta resueltos.
3. Ejecutar el programa.
4. Comprobar que se ha creado evento.txt en Salidas y contiene lo tecleado.

# Ejercicio 0\_c II

```
evento.txt - C:\DATOS\Curso 2024-2025\GIQ\LAB_PYTHON_24_25\cap7_Archivos\resueltos\Salidas\evento.txt (3.12.6)
File Edit Format Run Options Window Help
1 ASISTENTES
2 Ana Garcia
3 Pedro Gomez
4 Juan Lopez
5 Ines Perez
```

```
cap7_0c.py - C:\DATOS\Curso 2024-2025\GIQ\LAB_PYTHON_24_25\cap7_Archivos\resueltos\cap7_0c.py (3.12.6)
File Edit Format Run Options Window Help
1 #PB Nov-2024
2 print("Vayan escribiendo sus nombres.")
3 print('Escriban "FIN" cuando no quede nadie más.')
4 fs=open("Salidas\\evento.txt", "a")
5 fs.write("ASISTENTES\n") #Comentar en segunda ejecución y siguientes
6 cadena=''
7 while cadena!="FIN":
8     if cadena!='': fs.write(cadena+"\n")
9     cadena=input("Nombre (FIN para acabar): ")
10
11 fs.close()
```

En cada ejecución, los nombres que se tecleen se añaden a evento.txt

Comentar la línea 5 para la segunda ejecución y siguientes.

# Ejercicio 1

Programa que escriba en un archivo con nombre cap7\_1out.txt los cien primeros números naturales, un número por línea.

```
cap7_1.py - C:\DATOS\Curso 2022-2023\GIQ\LAB_PYTHON_22_23\cap7_Archivos\resueltos\cap7_1.py (3.10.7)
File Edit Format Run Options Window Help
1 #PB Nov 2022
2 fich=open("cap7_1out.txt","w")
3 for i in range(1,101):
4     fich.write(str(i)+'\n')
5 fich.close()
```

## Ejercicio 2

Renombra cap7\_1out.txt como cap7\_2in.txt. Elabora un programa que lea ese archivo de texto y calcule y escriba en pantalla y en el archivo cap7\_2out.txt los números leídos junto con sus raíces cuadradas y cúbicas expresadas con una cifra decimal, tres datos por línea.

cap7\_2.py - C:\DATOS\Curso 2022-2023\GIQ\LAB\_PYTHON\_22\_23\cap7\_Archivos\resueltos\cap7\_2.py (3.10.7)

File Edit Format Run Options Window Help

```
1 #PB Nov 2022
2 fiche=open("cap7_2in.txt","r")
3 fichs=open("cap7_2out.txt","w")
4
5 for linea in fiche:
6 #al convertir a entero se elimina \n
7     n=int(linea)
8     print(n,round(n**(1/2),1),round(n**(1/3),1))
9     fichs.write(str(n)+'\t'+str(round(n**(1/2),1))+
10                '\t'+str(round(n**(1/3),1))+'\n')
11 fiche.close()
12 fichs.close()
```

## Ejercicio 3

Programa que sume los datos de un archivo de compras con nombre compras.txt y escribir el resultado en pantalla. Leer de teclado el nombre del archivo de lectura.

```
cap7_3.py - C:\DATOS\Curso 2022-2023\GIQ\LAB_PYTHON_22_23\cap7_Archivos\resueltos\cap7_3.py (3.10.7)
File Edit Format Run Options Window Help
1 #PB Nov 2022
2 nombre=input("¿Nombre archivo de lectura? ")
3 fich=open(nombre, 'r')
4 total=0
5 for linea in fich:
6     num=float(linea)
7     total+=num
8 fich.close()
9 print("Total ", total)
```

## Ejercicio 4

Programa que lea un archivo fuente.txt y escriba en el archivo parte.txt únicamente las líneas impares.

```
cap7_4.py - C:\DATOS\Curso 2022-2023\GIQ\LAB_PYTHON_22_23\cap7_Archivos\resueltos\cap7_4.py (3.10.7)
File Edit Format Run Options Window Help
1 #PB Nov 2022
2 fiche=open("fuente.txt",'r')
3 fichs=open("parte.txt",'w')
4 n=0 # contador de líneas
5 for linea in fiche:
6     n+=1
7     # print(linea,n)
8     if n%2==1:
9         fichs.write(linea)
10 fiche.close()
11 fichs.close()
```

# Ejercicio 5

Elaborar un programa que lea de un archivo con nombre cap7\_5in.txt los nombres y las notas de los alumnos de una clase y escriba en un archivo con nombre cap7\_5out.txt los nombres de los alumnos y clase según sus notas, de acuerdo con la siguiente clasificación:

Clase	Nota
1	[8.5, 10]
2	[5, 8.5)
3	[0, 5)

```
cap7_5.py - C:\DATOS\Curso 2022-2023\GIQ\LAB_PYTHON_22_23\cap7_Archivos\resueltos\cap7_5.py (3.10.7)
File Edit Format Run Options Window Help
1 #PB Nov 2022
2 fiche=open("cap7_5in.txt","r")
3 fichs=open("cap7_5out.txt","w")
4 fichs.write('ALUMNO,CLASE \n')
5 for linea in fiche:
6     lista=linea.split(",")
7     nota=float(lista[1])
8     if nota<5:
9         clase=3
10    elif nota<8.5:
11        clase=2
12    else:
13        clase=1
14    fichs.write(lista[0]+' '+str(clase)+'\n')
15 fiche.close()
16 fichs.close()
```

## Ejercicio 6a I

Elaborar un programa que lea los nombres de los alumnos y sus notas de un archivo con nombre cap7\_6in.txt.

Obtener el nombre y la nota del mejor y peor alumno de la clase.

Calcular la media de la clase y la cantidad de aprobados.

Escribir los resultados en un archivo con nombre cap7\_6out.txt usando una cifra decimal para las notas.

Sin funciones y leyendo con for-in.

# Ejercicio 6a II

cap7\_6a.py - C:\DATOS\Curso 2024-2025\GIQ\LAB\_PYTHON\_24\_25\cap7\_Archivos\resueltos\cap7\_6a.py (3.12.6)

File Edit Format Run Options Window Help

```
1 #PB Nov 2024
2 fe=open("Entradas\\cap7_6in.txt","r")
3 n=0
4 nsufis,suma=0,0
5 for linea in fe:
6     n+=1
7     alumno=linea.split(",")
8     nota=float(alumno[1])
9     suma+=nota
10    if n==1:
11        peor=alumno #inicializacion, nombre y nota
12        mejor=alumno #idem
13    else:
14        if nota<float(peor[1]):
15            peor=alumno
16        if nota>float(mejor[1]):
17            mejor=alumno
18        if nota>=5 and nota<7:
19            nsufis+=1
20 media=suma/n
21 fe.close()
22 # Escritura de resultados en archivo
23 fs=open("Salidas\\cap7_6out.txt","w")
24 fs.write("Hay "+str(nsufis)+' alumnos con aprobado.\n')
25 fs.write('Nota media de la clase: '+str(round(media,2))+'\n')
26 fs.write('Peor: '+peor[0]+' nota: '+peor[1])
27 fs.write('Mejor: '+mejor[0]+' nota: '+mejor[1])
28 fs.close()
```

cadena

## Ejercicio 6b I

Usar una función para la lectura del archivo `fleearchivo()`: la función no recibe nada y devuelve una lista de cadenas.

Usar una función para calcular la media de la clase y la cantidad de aprobados, `faprob_media()`.

Leer el archivo usando el método `readlines`.

# Ejercicio 6b II

```
cap7_6b.py - C:\DATOS\Curso 2024-2025\GIQ\LAB_PYTHON_24_25\cap7_Archivos\resueltos\cap7_6b.py (3.12.6)
File Edit Format Run Options Window Help
1 #PB Nov 2024
2 def fleer_archivo():
3     fe=open("Entradas\\cap7_6in.txt","r")
4     todo=fe.readlines()
5     fe.close()
6     return todo #lista cadenas
7
8 def faprob_media(lineas): #lista cadenas
9     nsufis,suma=0,0
10    for i in range(len(lineas)):
11        campos=lineas[i].split(",")
12        nota=float(campos[1])
13        suma+=nota
14        if nota>=5 and nota<7:
15            nsufis+=1
16    return nsufis,suma/len(lineas)
17
18 def cap7_6b():
19 # leer archivo
20    completo=fleer_archivo() #lista de cadenas
21 # peor y mejor nota
22    peor=completo[0].split(",") #inicializacion, nombre y nota
23    mejor=completo[0].split(",") #idem
24    for i in range(1,len(completo)):
25        alumno=completo[i].split(",")
26        if float(alumno[1])<float(peor[1]):
27            peor=alumno
28        if float(alumno[1])>float(mejor[1]):
29            mejor=alumno
30    tupla=faprob_media(completo)
31 # Escritura de resultados en archivo
32    fs=open("Salidas\\cap7_6out.txt","w")
33    fs.write("Hay "+str(tupla[0])+' alumnos con aprobado.\n')
34    fs.write('Nota media de la clase: '+str(round(tupla[1],2))+'\n')
35    fs.write('Peor: '+peor[0]+' nota: '+peor[1])
36    fs.write('Mejor: '+mejor[0]+' nota: '+mejor[1])
37    fs.close()
```

# Ejercicio 7 I

Dado un fichero con nombre EQ.txt que contiene información acerca de elementos químicos (símbolo, nombre, número atómico, número másico), realizar un programa que:

Por medio de una función lea de un fichero los diferentes elementos químicos y los almacene en un diccionario (el índice de los elementos del diccionario será el símbolo de cada uno de ellos y el resto su contenido).

Mediante un menú permita:

- Listar los elementos químicos.
- Buscar y mostrar información de un elemento químico.
- Añadir un elemento químico al diccionario.
- Borrar un elemento químico del diccionario.
- Salir y guardar la información del diccionario en el archivo de los elementos químicos.

Utilizar una función para cada opción del menú y para el propio menú.

# Ejercicio 7 II

```
cap7_7.py - C:\DATOS\Curso 2022-2023\GIQ\LAB_PYTHON_22_23\cap7_Archivos\resueltos\cap7_7\cap7_7.py (3.10.7)
File Edit Format Run Options Window Help
1 #PB Nov 2022
2 def leer_fichero():
3     dic = {}
4     fichero = open('EQ.txt','r')
5     for linea in fichero:
6         elem=linea.split(",")
7         dic[elem[0]] = [elem[1],elem[2],elem[3][0:-1]]
8     fichero.close()
9     return dic
10
11 #Función que muestra el menú de la aplicación
12 def menu():
13     print('\n')
14     print('1- Listado elementos químicos')
15     print('2- Buscar información')
16     print('3- Añadir elemento')
17     print('4- Borrar elemento')
18     print('5- Salir y guardar cambios')
19     opcion = input('Seleccione una opción: ')
20     return opcion
21
22 #Función que muestra el listado, en forma de tabla, de los elementos químicos
23 def Listado_EQ(elDic):
24     print('Símbolo', 'Descripción', 'Nº atómico', 'Nº másico')
25     print('_ '*40)
26     print()
27     print(elDic)
28     for clave in elDic:
29         print ('{0} {1} {2} {3} '.format(clave,elDic[clave][0],elDic[clave][1],elDic[clave][2]))
30
```

# Ejercicio 7 III

```
31 #Función que pide el símbolo de un elemento químico y muestra la información del mismo
32 #que se encuentra en el diccionario pasado como parámetro
33 def mostrar_EQ(eLDic):
34     elelem= input('Símbolo del elemento quimico: ')
35     if elelem in eLDic:
36         print ('Descripción: ' + eLDic[elelem][0] + ' (' + elelem + ')')
37         print ('Número atómico: ' + eLDic[elelem][1])
38         print ('Número másico: ' + eLDic[elelem][2] + '\n')
39     else:
40         print ('No se ha encontrado el elemento de símbolo ' + elelem + '\n')
41
42 #Función que añade un elemento químico al diccionario pasado como parámetro
43 def add_EQ (eLDic):
44     print('Introduzca la información del nuevo elemento')
45     simbolo=input('Símbolo: ')
46     desc=input('Descripción: ')
47     Z=input('Número atómico: ')
48     A=input('Número másico: ')
49     eLDic[simbolo]=[desc,Z,A]
50
51 def del_EQ (eLDic):
52     elelem= input('Símbolo del elemento químico a borrar:')
53     if elelem in eLDic:
54         del eLDic[elelem]
55         print('Elemento eliminado correctamente')
56     else:
57         print('Elemento no encontrado')
58
59 def salir_programa (eLDic):
60     fichero = open ('EQ.txt','w')
61     for clave in eLDic:
62         linea= clave + ',' + eLDic[clave][0] + ',' + \
63             + eLDic[clave][1] + ',' + eLDic[clave][2] + '\n'
64         fichero.write(linea)
65     fichero.close()
```

# Ejercicio 7 IV

```
68 #PRINCIPAL
69 def main():      #procedimiento
70     eleccion=0
71     elem_quimicos=leer_fichero()
72     while eleccion!=5:
73         eleccion=menu()
74         if eleccion == '1':
75             Listado_EQ (elem_quimicos)
76         elif eleccion == '2':
77             mostrar_EQ(elem_quimicos)
78         elif eleccion == '3':
79             add_EQ(elem_quimicos)
80         elif eleccion == '4':
81             del_EQ(elem_quimicos)
82         elif eleccion == '5':
83             salir_programa (elem_quimicos)
84             break
85         else:
86             print('Selección incorrecta. Elija de nuevo')
87     print('Gracias por utilizar el programa')
88 main()
```