ARCHIVOS DE TEXTO

Pere Manel Verdugo Zamora

pereverdugo@gamil.com

Contenido

1.- Leyendo archivos de texto 1
2.- Codificación de caracteres 7
3.- Contando líneas 14
4.- Contando palabras 20
5.- Métodos seek y tell 22
6.- Gestor de contexto 25
7.- Modos de abrir un archivo de texto 28
8.- Cuaderno de notas con archivo de texto (Presentación) 32
9.- Cuaderno de notas (Código primera parte) 33
10.- Cuaderno de notas (Código segunda parte) 36
11.- Generador de test (Presentación) 41
12.- Generador de test (Código inicio) 49
13.- Generador de test (Código final) 55

Puedes seguir los tutoriales en YouTube en el canal de Manuel Gonzales, llamado 'Programar en Python es como un juego'.

1.- Leyendo archivos de texto

En este curso vamos a trabajar con archivos de texto. Hasta ahora solo utilizábamos el editor de texto para la creación de los programas, pero una vez cerrábamos el programa cualquier dato que se hubiere generado con este programa se perdía. No podíamos mantener nuestros datos en sesiones posteriores, bueno pues ahora vamos hacer que nuestros datos persistan, es decir al hecho que los datos de un programa cuando termina su ejecución se mantenga para posteriores ejecuciones se llama persistencia.

Lo vamos hacer pasando nuestros datos en un archivo de texto y de igual manera también leyendo datos de un archivo de texto para utilizarlos en la ejecución de un programa. Para ello he abierto un fichero de texto llamado saludo.txt y el archivo de Python llamado archivo.py.

El archivo de texto tiene 4 líneas.

```
saludo.txt
1   Hola.
2   Qué tal.
3   Encantado.
4   Adiós
```

Para poder utilizar un archivo de texto lo primero que hay que hacer es abrirlo y la forma de hacerlo es creando un objeto de tipo archivo, que nos permita acceder al contenido de ese archivo.

```
1   archivo = open("saludo.txt")
```

Vamos a hacer un print de este archivo.

```
2   print(archivo)
```

Vamos a ejecutar:

```
<_io.TextIOWrapper name='saludo.txt' mode='r' encoding='cp1252'>
```

Es un objeto de la subclase _io.TextIOWrapper, que permite tratar con ficheros.

Nos muestra el nombre del fichero 'saludo.txt'.

Tenemos el mode='r' de Reading (lectura).

Sistema de codificación → encoding='cp1252' es el de Windows por defecto, esto se comentará en capítulos posteriores.

Ahora nos vamos a centrar el modo de lectura, en capítulos posteriores veremos los otros modos que hay de abrir un archivo.

Si no le indicamos nada por defecto un archivo se abre en modo lectura.

```
1   archivo = open("saludo.txt", "r")
```

También se lo podemos especificar.

Solo podremos leer pero no modificar.

```
1   archivo = open("saludo.txt", "r")
2   texto = archivo.read()
3   print(type(texto))
```

Queremos saber de que tipo es esta variable, vamos a ejecutar:

```
<class 'str'>
```

Es de tipo string.

El método read() devuelve una serie de caracteres.

```
1   archivo = open("saludo.txt", "r")
2   texto = archivo.read()
3   print(texto)
```

Vamos a ejecutar.

```
Hola.
Qué tal.
Encantado.
Adiós
```

```
1   archivo = open("saludo.txt", "r", encoding="utf-8")
2   texto = archivo.read()
3   print(texto)
4   archivo.close()
```

Al finalizar es importante cerrar el archivo para que no se corrompan los datos.

A parte del método read(), otro va a ser readlines().

```
1   archivo = open("saludo.txt", "r", encoding="utf-8")
2   texto = archivo.readlines()
3   print(type(texto))
```

Vamos a ejecutar.

```
<class 'list'>
```

Ahora en un objeto de tipo lista.

```
1   archivo = open("saludo.txt", "r", encoding="utf-8")
2   texto = archivo.readlines()
3   print(texto)
4   archivo.close()
```

```
['Hola.\n', 'Qué tal.\n', 'Encantado.\n', 'Adiós']
```

Cada línea es un elemento de la lista.

El carácter de escape '\n' es el salto de línea.

Lo que podemos hacer.

```
1   archivo = open("saludo.txt", "r", encoding="utf-8")
2   texto = archivo.readlines()
3   for linea in texto:
4       print(linea)
5   archivo.close()
```

Este será el resultado:

```
Hola.

Qué tal.

Encantado.

Adiós
```

Observamos que nos a generado un salto de línea en cada línea, lo vamos a solucionar.

```
1   archivo = open("saludo.txt", "r", encoding="utf-8")
2   texto = archivo.readlines()
3   for linea in texto:
4       linea_1 = linea.strip("\n")
5       print(linea_1)
6   archivo.close()
```

strip() por defecto si no pones nada también elimina los saltos de línea.

Con el método .strip() le decimos el carácter que queremos eliminar, este será el resultado.

```
Hola.
Qué tal.
Encantado.
Adiós
```

Si lo queremos guardar en una lista.

```
1   archivo = open("saludo.txt", "r", encoding="utf-8")
2   texto = archivo.readlines()
3   lista_formateada = []
4   for linea in texto:
5       linea = linea.strip("\n")
6       lista_formateada.append(linea)
7   archivo.close()
8   print(texto)
```

```
9   print(lista_formateada)
```

Este será el resultado:

```
['Hola.\n', 'Qué tal.\n', 'Encantado.\n', 'Adiós']
['Hola.', 'Qué tal.', 'Encantado.', 'Adiós']
```

Aún tenemos otro método que nos permite leer el contenido de un archivo, readline().

Este método lo lee el archivo completo sino solo una línea.

```
1   archivo = open("saludo.txt", "r", encoding="utf-8")
2   linea = archivo.readline()
3
4   archivo.close()
5   print(linea)
```

Este será el resultado:

```
Hola.
```

Con el salto de línea.

Cuando se abre un archivo se genera un puntero que apunta al primer carácter del texto en el archivo, una vez que se lee una línea el puntero se sitúa al final de la línea leída, si repetimos es acción este será el resultado:

```
1   archivo = open("saludo.txt", "r", encoding="utf-8")
2   linea = archivo.readline()
3   print(linea)
4   linea = archivo.readline()
5   print(linea)
6   archivo.close()
```

Este será el resultado:

```
Hola.

Qué tal.
```

Con sus respectivos saltos de línea.

Este método va a ser muy útil para cuando tengamos archivos muy grandes y en vez de cargar todo el archivo en memoria lo que hacemos es ir leyendo línea por línea cargando solamente una línea en memoria, por lo tanto no cargando con exceso la memoria.

Lo que podemos hacer con este método utilizar un bucle para ir leyendo todas las líneas una por una.

```
1   archivo = open("saludo.txt", "r", encoding="utf-8")
2   while True:
```

```
3        linea = archivo.readline().strip()
4        if linea == "":
5            break
6        else:
7            print(linea)
8    archivo.close()
```

Este será el resultado:

```
Hola.
Qué tal.
Encantado.
Adiós
```

2.- Codificación de caracteres

Empezamos un nuevo capítulo en el que vamos a seguir con los archivos de texto.

Del capítulo anterior vimos como abrir un archivo y como podíamos leer su contenido, en este capítulo vamos a ver un aspecto de los archivos que no vimos en el capitulo anterior que va a ser muy importante a la hora de abrir y leer archivos.

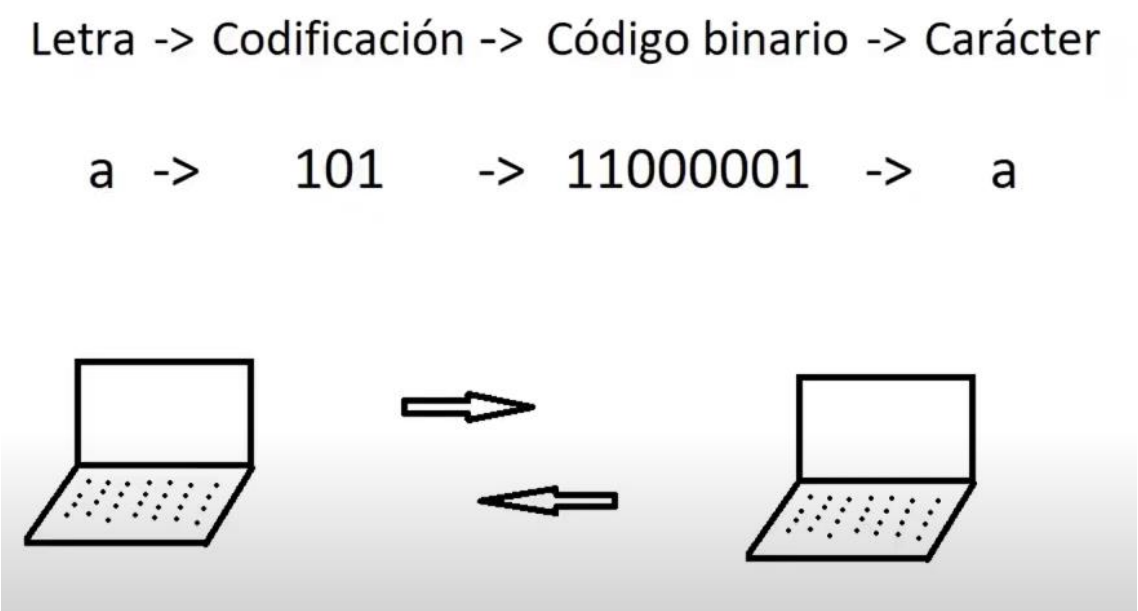
En el capítulo anterior vimos que para abrir un archivo creábamos un objeto de tipo archivo si lo imprimíamos por consola nos decía el tipo de archivo que era.

```
<_io.TextIOWrapper name='saludo.txt' mode='r' encoding='cp1252'>
```

Si vamos a leer archivos externos en nuestro ordenador vamos a tener en cuenta.

La codificación de caracteres es el método que permite convertir un carácter de un lenguaje natural (como el de kun alfabeto o silabario) en kun símbolo de otro sistema de representación, como un número o una secuencia de pulsos eléctricos en un sistema electrónico, aplicando nomas o reglas de codificación.

Defina la forma en la que se codifica un carácter dado en un símbolo en otro sistema de representación. Ejemplos de esto son el código Morse, la norma ASCII o la UTF-8, entre otros.



Imaginemos que tenemos dos ordenadores, cuando nosotros le damos a una tecla en un ordenador el ordenador sabe que letra es por un código que tiene asignado esa letra, que es el sistema de codificación, la convierte en un código binario, así puede escribir esta letra en la pantalla del ordenador o en un archivo, si este archivo se abre en otro ordenador, ese ordenador lee esa letra si tienen el mismo sistema de codificación sabe que letra es, la convierte en código binario y la reconoce con la misma letra que estaba en el otro ordenador. Si los dos ordenadores no tienen el mismo sistema de codificación no sabrán reconocer los caracteres de esos archivos.

Para poder representar los caracteres en un ordenador se necesita un sistema de codificación y ello supone reservar un espacio de memoria para cada uno de ellos.

Hasta hace poco este espacio era de:

1 byte = 8 bits = 0000000

Las combinaciones posible de 8 bits con 1 y 0 son 256.

Con lo cual si se usa 1 byte para guardar la información necesaria para procesar un carácter podemos tener 256 caracteres guardados en un sistema de 8 bits.

El formato ASCII al principio usaba 7 bits, y podía contener 128 caracteres. Este sistema de codificación fue creado por el Comité estadounidense de estándares y contenía sólo los caracteres ingleses, nada de letras con tildes u otras letras. Posteriormente pasó a usar los 8 bits de un byte y podía contener 256 caracteres.

El formato ANSI igualmente usa 1 byte y contiene 256 caracteres. Es usado por Microsoft en Windows. Este formato no es un estándar y en realidad hay muchos dependiendo del juego de caracteres que use Windows, en Europa Occidental sería cp1252, pero no contiene caracteres de otros idiomas.

El formato UNICODE es un estándar establecido por la Unicode Techinical Committee del que forman parte grandes empresas, universidades, etc, que se encarga de su mantenimiento.

UNICODE es un juego de caracteres establecido internacionalmente que contiene los caracteres posibles y necesarios para ser representados en un ordenador. Es un conjunto de caracteres, no un algoritmo de codificación.

Este formato no es un sistema de codificación en sí, sino que se implementa en los ordenadores a través del sistema UTF-8, u en otros casos el UTF-16, o UTF-32.

WIKIPEDIA: UNICODE, UTF-8, ANSI, ASCII, CODIFICACIÓN DE CARACTERES, etc…

www.atc.uniovi.es/inf_superior/4atc/DISTRIBUIDAS/03-L6-Caracteres.pdf.

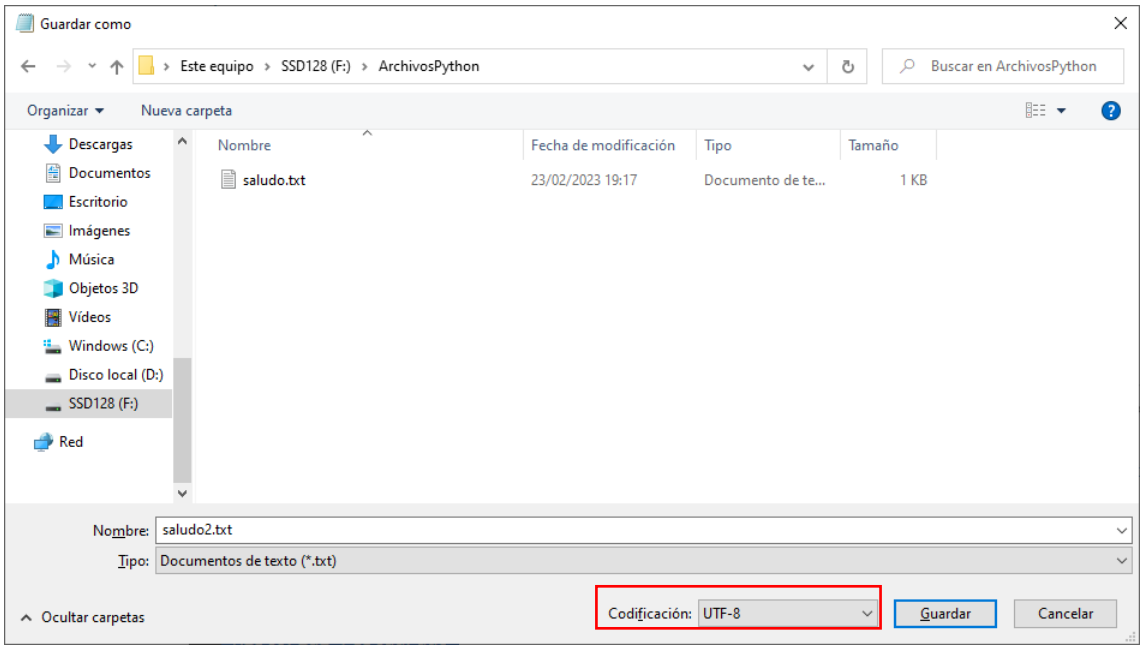
https://www.joelonsoftware.com/2003/10/08/the-absolute-minimun-every-software-developer-absolutely-positively-must-know-about-unicode-and-character-sets-no-excuses/

Vamos a ver un ejemplo:

Vamos a crear un documento con el bloc de notas que guardaremos donde tenemos el proyecto.

```
1   Hola.
2   ¿Qué tal?
3   Encantado.
4   Me gusta la eñe.
5   Adiós
```

Lo guardaremos con el nombre saludo2.txt y con una codificación de UTF-8




```
1   archivo = open("saludo.txt", "r")
2   texto = archivo.read()
3   archivo.close()
4   print(texto)
```

Queremos ver el archivo saludo.txt que está grabado en formato ANSI.

```
Hola.
¿Qué tal?
Encantado.
Me gusta la eñe.
Adiós
```

Se muestra correctamente.

Ahora vamos a hacer lo mismo con el archivo saludo2.txt.

```
1   archivo = open("saludo2.txt", "r")
2   texto = archivo.read()
3   archivo.close()
4   print(texto)
```

Está codificado con el formato UTF-8.

Este será el resultado:

```
Hola.
Â¿QuÃ© tal?
Encantado.
Me gusta la eÃ±e.
AdiÃ³s
```

El texto no se muestra correctamente debido a que la codificación ANSI y la codificación UTF-8 difieren en algunos caracteres.

Es importante saber cómo esta codificado un archivo si proviene de una fuente externa.

```
1   archivo = open("saludo2.txt", "r", encoding = "utf-8")
2   texto = archivo.read()
3   archivo.close()
4   print(texto)
```

Agregamos la codificación encoding = “utf-8”, vamos a ejecutar:

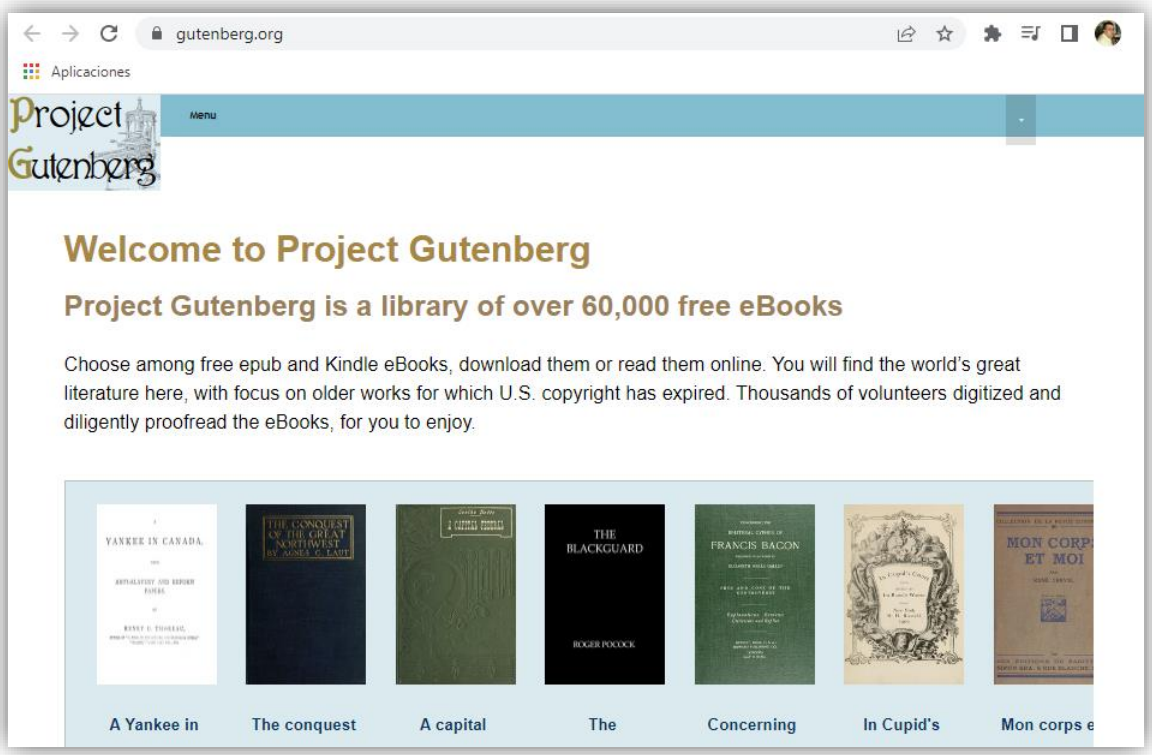
```
Hola.
¿Qué tal?
Encantado.
Me gusta la eñe.
Adiós
```

Ahora sí, nos muestra todos los caracteres correctamente, porque los hemos descodificado con el código de caracteres adecuado.

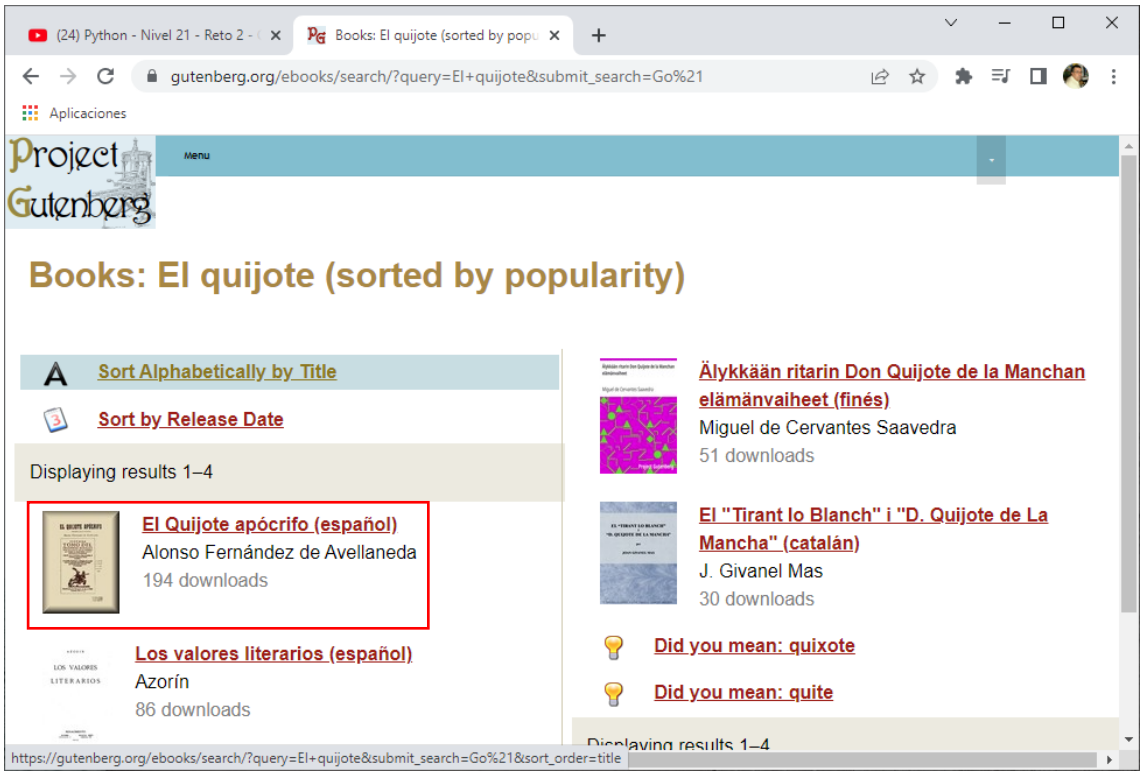
Para el próximo capítulo te propongo el siguiente reto.

Vamos al proyecto gutenberg.

https://gutenberg.org



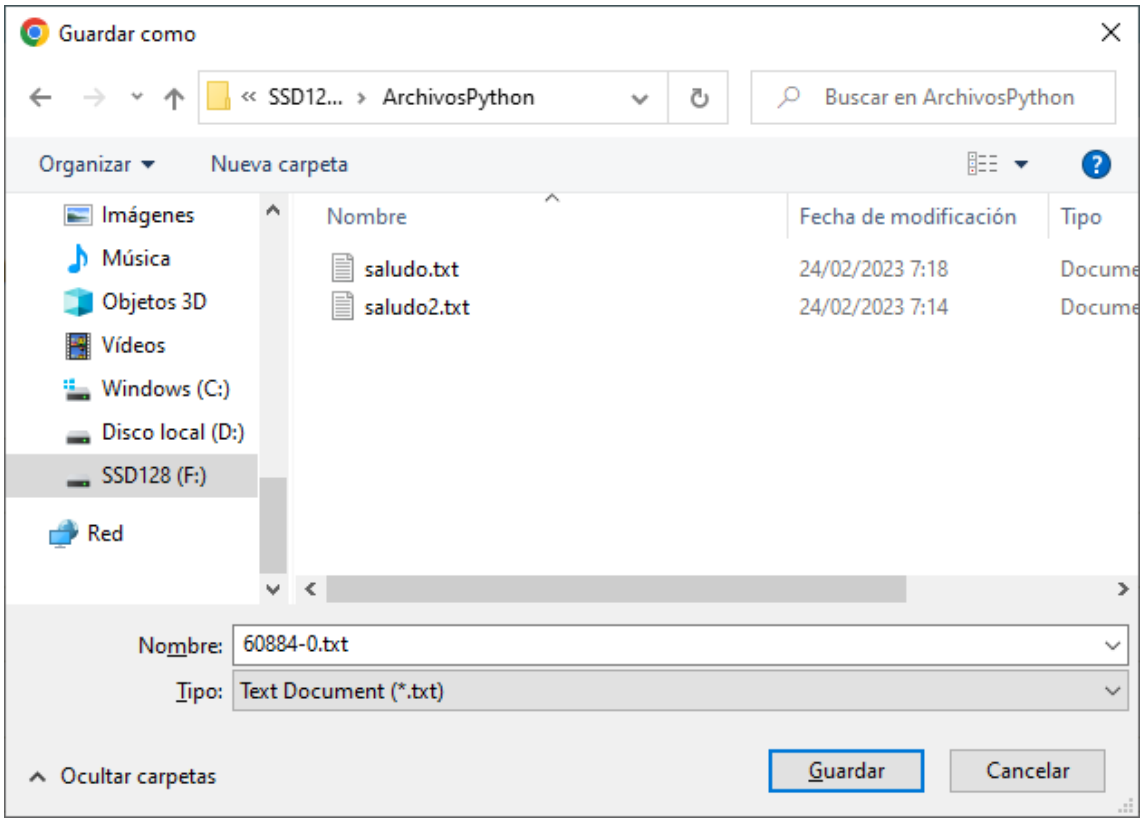
Buscamos el Quijote.



Lo seleccionamos.

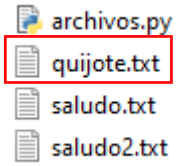
Lo tenemos con la codificación UTF-8.

Con el botón derecha le damos a guardar enlace como.

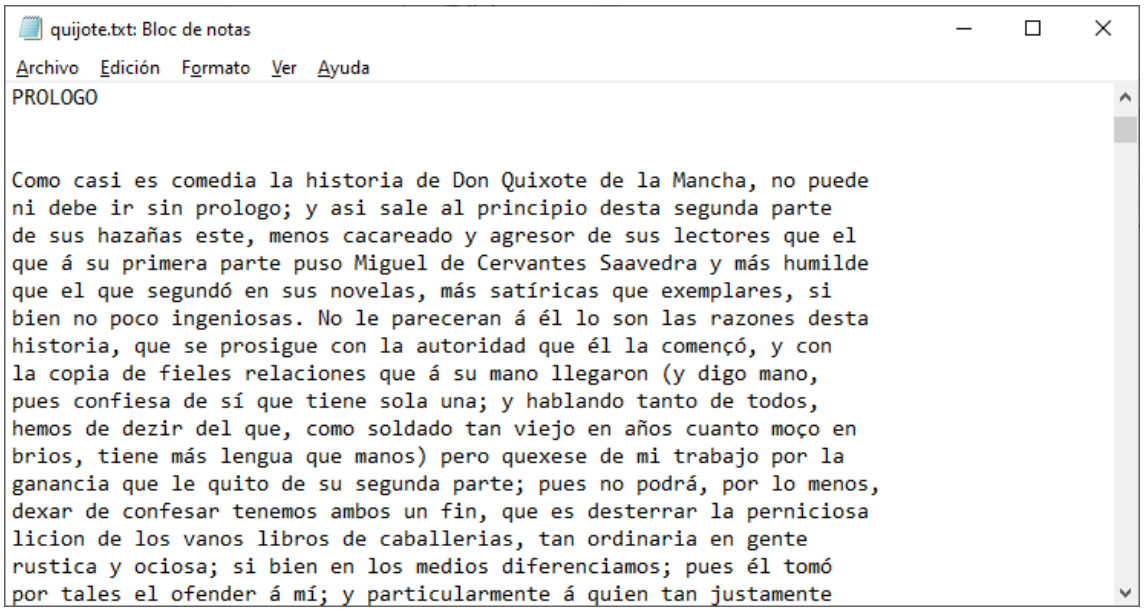


Nos vamos a la carpeta donde tenemos nuestro proyecto.

Si quieres lo renombras como quijote.



Si lo abrimos tenemos a Don Quijote de la Mancha.



Como reto te propongo las líneas que tiene este archivo.

Ejemplo 1

```
1   archivo = open("quijote.txt", "r", encoding = "utf-8")
2   texto = archivo.read()
3   archivo.close()
4   cont = 0
5   for letra in texto:
6       if letra == "\n":
7           cont += 1
8
9   print(cont)
```

Ejemplo 2

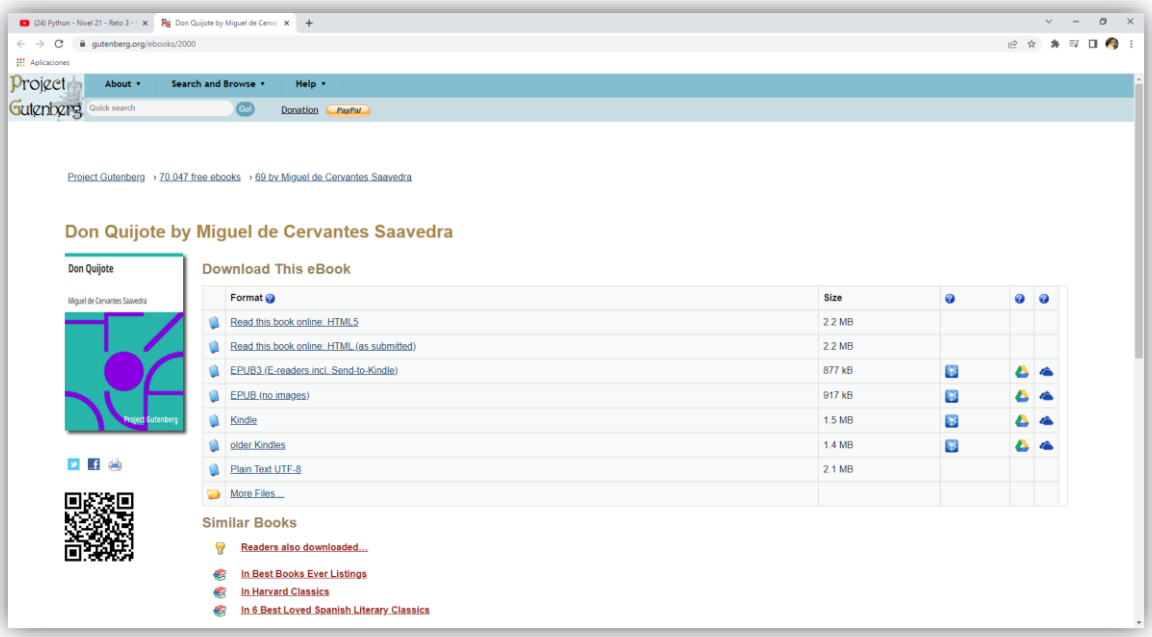
```
1   archivo = open("quijote.txt", "r", encoding = "utf-8")
2   texto = archivo.readline()
3   contador = 0
4   while not texto == "":
5       contador = contador + 1
6       texto = archivo.readline()
7   archivo.close()
8   print(contador)
```

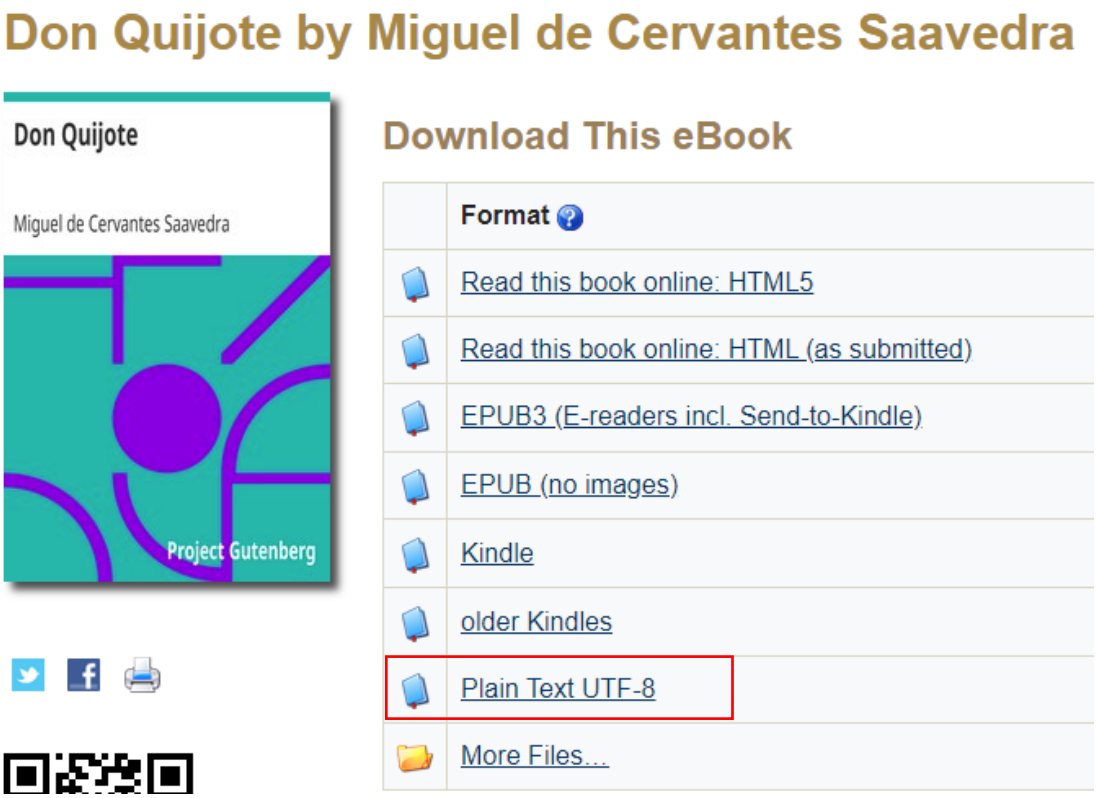
Ejemplo 3

```
archivo = open("quijote.txt", "r", encoding = "utf-8")
texto = archivo.readlines()
print(len(texto))
```

3.- Contando líneas

En el capítulo anterior fuimos a la página web https://www.gutenberg.org/ebooks/2000





Lo descargamos en formato de texto y codificado UTF-8.

La guardamos donde estamos realizando los proyectos con el nombre de quijote.txt.

El objetivo es realizar un programa que nos diga el número de líneas que contiene el Quijote.

Utilizando el método readlines()

```
1   archivo = open("quijote.txt", "r", encoding = "utf-8")
2   lineas = archivo.readlines()
3   print(len(lineas))
4   archivo.close()
```

Utilizando el método read()

```
1   archivo = open("quijote.txt", "r", encoding = "utf-8")
2   texto = archivo.read()
3   archivo.close()
4   cont = 0
5   for letra in texto:
6       if letra == "\n":
7           cont += 1
8
9   print(cont)
```

Otro ejemplo:

```
1   archivo = open("quijote.txt", "r", encoding = "utf-8")
2
3   n = 0
4
5   while True:
6       if archivo.readline() != "":
7           n +=1
8       else:
9           break
10  print(n)
11  archivo.close()
```

Otro ejemplo sin utilizar ningún método.

```
1   archivo = open("quijote.txt", "r", encoding = "utf-8")
2   n = 0
3   for linea in archivo:
4       n += 1
5
6   print(n)
7   archivo.close()
```

Por defecto va leyendo línea por línea.

Queremos ver las 10 primeras líneas del Quijote.

```
1   archivo = open("quijote.txt", "r", encoding = "utf-8")
2   n = 0
3   for linea in archivo:
4       print(linea)
5       n += 1
6       if n == 10:
7           break
8   print(n)
9   archivo.close()
```

Este será el resultado:

```
Project Gutenberg's El Quijote apócrifo, by Alonso Fernández de Avellaneda


This eBook is for the use of anyone anywhere in the United States and

most other parts of the world at no cost and with almost no restrictions

whatsoever.  You may copy it, give it away or re-use it under the terms

of the Project Gutenberg License included with this eBook or online at

www.gutenberg.org.  If you are not located in the United States, you'll

have to check the laws of the country where you are located before using

this ebook.


10
```

Te adjunto un ejemplo de cómo tienes que utilizar los diccionarios:

```
1   personal = {}
2   nombre = ""
3
4   while nombre != "fin":
5       nombre = input("Nombre que entra: ")
6       if nombre != "" and nombre != "fin":
7           personal.setdefault(nombre, 0)
8           personal[nombre] += 1
```

```
9   print("Control entradas")
10  print("----------------")
11  for clave in personal:
12      valor = personal[clave]
13      print(clave + "--> " + str(valor))
```

Este ejemplo consiste en un control de entradas que cuando pasa una trabajador introducimos el nombre, cuando se cierra contestamos con la palabra fin.

Nos tiene que decir cuantas veces a pasada cada uno de ellos.

Vamos a ejecutar:

```
Nombre que entra: Carlos
Nombre que entra: Luis
Nombre que entra: Pedro
Nombre que entra: Antonio
Nombre que entra: Luis
Nombre que entra: Pedro
Nombre que entra: Ana
Nombre que entra: Carlos
Nombre que entra: Luis
Nombre que entra: fin
Control entradas
----------------
Carlos--> 2
Luis--> 3
Pedro--> 2
Antonio--> 1
Ana--> 1
```

Imaginaros que al introducir los nombres en que los introduce algunas veces tiene la costumbre de agregar al nombre un punto, guion o como es este ejemplo lo que hacemos es eliminarlo.

```
1   personal = {}
2   nombre = ""
3
4   while nombre != "fin":
5       nombre = input("Nombre que entra: ")
6       if nombre != "" and nombre != "fin":
7           nombre = nombre.strip(".").strip("-").strip(",")
8           personal.setdefault(nombre, 0)
9           personal[nombre] += 1
10  print("Control entradas")
11  print("----------------")
12  for clave in personal:
```

```
13        valor = personal[clave]
14        print(clave + "--> " + str(valor))
```

Vamos a ejecutar:

```
Nombre que entra: Carlos,
Nombre que entra: Luis,
Nombre que entra: Luis-
Nombre que entra: Pedro.
Nombre que entra: Carlos-
Nombre que entra: fin
Control entradas
----------------
Pedro--> 2
Luis--> 3
Carlos--> 2
```

Ahora queremos que me lo ordene por personal que ha entrado más veces:

```
1   personal = {}
2   nombre = ""
3
4   while nombre != "fin":
5       nombre = input("Nombre que entra: ")
6       if nombre != "" and nombre != "fin":
7           nombre = nombre.strip(".").strip("-").strip(",")
8           personal.setdefault(nombre, 0)
9           personal[nombre] += 1
10  print("Control entradas")
11  print("----------------")
12  for i in range(len(personal)):
13      entradas = 0
14      for op in personal:
15          if personal[op] > entradas:
16              entradas = personal[op]
17              personal_mayor = op
18              valor = personal[op]
19      print(personal_mayor + "--> " + str(entradas))
20      del personal[personal_mayor]
```

Este será el resultado:

```
Nombre que entra: Carlos.
Nombre que entra: Luis,
Nombre que entra: Pedro
Nombre que entra: Antonio,
```

```
Nombre que entra: Ana
Nombre que entra: Pedro
Nombre que entra: Carlos.
Nombre que entra: Antonio
Nombre que entra: Ana,
Nombre que entra: fin
Control entradas
----------------
Carlos--> 2
Pedro--> 2
Antonio--> 2
Ana--> 2
Luis--> 1
```

Para el próximo capitulo te propongo el siguiente reto:

Mostrar qué cinco palabras con más de 12 letras aparecen más veces en el Quijote, y cuantas veces aparecen cada una.

4.- Contando palabras

```
1   archivo = open("quijote.txt", "r", encoding = "utf-8")
2   texto = archivo.read()
3   archivo.close()
4   lista = texto.split()
5   palabras = {}
6   for elemento in lista:
7       palabra = elemento.strip(".").strip(",").strip(";").strip(":")
8       if len(palabra) > 12:
9           palabras.setdefault(palabra, 0)
10          palabras[palabra] += 1
11
12  for i in range(5):
13      numero_mayor = 0
14      for palabra in palabras:
15          if palabras[palabra] > numero_mayor:
16              numero_mayor = palabras[palabra]
17              palabra_mayor = palabra
18      print(palabra_mayor, ":", numero_mayor )
19      del palabras[palabra_mayor]
```

Vamos a comentar el programa:

Línea 1 abrimos el archivo en modo lectura con la codificación “utf-8”

Línea 2 la variable texto contiene todas las palabras del Quijote.

Línea 3 cerramos el archivo, ya que disponemos la variable texto con toda la información.

Línea 4 creamos una lista con todas las palabras, si imprimimos esta lista este sería el resultado:

```
'contrario.', 'Pero', 'como', 'tarde', 'la', 'locura', 'se', 'cura,', 'dizen', 'que', 'en', 'saliendo', 'de
', 'la', 'corte,', 'volvió', 'á', 'su', 'tema,', 'y', 'que', 'comprando', 'otro', 'mejor', 'caballo,', 'se'
, 'fue', 'la', 'vuelta', 'de', 'Castilla', 'la', 'Vieja,', 'en', 'la', 'cual', 'le', 'sucedieron', 'estupen
das', 'y', 'jamas', 'oidas', 'aventuras,', 'llevando', 'por', 'escudero', 'á', 'una', 'moça', 'de', 'soldad
a', 'que', 'halló', 'junto', 'á', 'Torre', 'de', 'Lodones,', 'vestida', 'de', 'hombre,', 'la', 'cual', 'iba
', 'huyendo', 'de', 'su', 'amo', 'porque', 'en', 'su', 'casa', 'se', 'hizo', 'ó', 'la', 'hizieron', 'preñad
a', 'sin', 'pensarlo', 'ella,', 'si', 'bien', 'no', 'sin', 'dar', 'cumplida', 'causa', 'para', 'ello;', 'y'
, 'con', 'el', 'temor', 'se', 'iba', 'por', 'el', 'mundo.', 'Llevola', 'el', 'buen', 'caballero', 'sin', 's
aber', 'que', 'fuese', 'muger,', 'hasta', 'que', 'vino', 'á', 'parir', 'en', 'medio', 'de', 'un', 'camino,'
, 'en', 'presencia', 'suya,', 'dexandole', 'sumamente', 'maravillado', 'el', 'parto,', 'y', 'haziendo', 'gr
```

Con todas las palabras del libro.

Línea 5 Definimos un diccionario vacío llamado palabras.

Línea 6 hacemos un bucle que recorre cada elemento de la lista que es la información que contiene en la línea 4.

Línea 7 A la variable palabra le asignamos el valor elemento pero eliminando el punto, la coma, el punto y coma y los dos puntos gracias al método strip().

Línea 8 Si la palabra tiene más de 12 letras.

Línea 9 se agrega al diccionario, pero si esta palabra ya existe modifica el elemento existente y el la línea 10 por mediación de un contador le incrementamos 1.

Hemos puesto en la línea 11 un print(palabras) para ver el valor del diccionario palabras.

```
abrimiento': 1, 'desvergonçado': 2, 'enriquezeremos': 1, 'guerreamientos': 1, 'Mari-Gutierrez': 29, 'desagr
adecida': 2, 'correspondencia': 3, 'verdaderamente': 8, 'perfectamente': 1, 'infinitamente': 4, 'Despidiero
nse': 3, 'correspondiese': 1, 'descomponerme': 1, '¡Desventurada': 1, 'ensayamientos': 1, 'proseguiremos':
3, 'dificultosamente': 1, 'inconveniente': 3, 'deshonestidad': 1, 'desvergonçada': 1, 'Alborotaronse': 1, '
contradezirle': 2, 'desquixaramientos': 1, 'pantorrillas?': 1, 'despanzorremos': 1, 'desaguisadas?': 1, 'pr
incipalmente': 4, 'Constantinopla': 7, 'encantamientos': 6, 'bravisimamente': 1, 'impertinentes': 1, 'Belli
do.--¡Oh': 1, 'melonar!--Este': 1, 'tribulaciones': 2, 'llegaronseles': 1, 'llegandoseles': 1, 'circunstant
es': 24, 'zorrinloquios?': 1, 'recomendacion': 2, 'escandaliçando': 1, 'perjudiciales[15]': 1, 'pusilanimid
```

En la línea 12 hacemos un bucle con 5 iteraciones porque queremos que nos muestre las 5 palabras con más de 12 letras que más sale en el texto.

En la línea 13 definimos una variable llamada numero_mayor a un valor de 0.

En la línea 14 hacemos tantas iteraciones como el número de elementos que tiene el diccionario palabras.

En la línea 15 si el valor de un elemento del diccionario palabras en mayor a la variable numero_mayor.

En la línea 16 si se cumple la línea 15 la variable numero_mayor asume el valor del elemento actual.

En la línea 17 si se cumple al línea 15 palabra_mayor asume el valor del elemento del diccionario (la palabra).

En línea 18 Imprime la palabra que más se repite y las veces que se repite.

En la línea 19 borramos dicha palabra del diccionario, para que en la siguiente iteración muestre la segunda palabra que se repete más veces, pues la que se repetía más veces a sido eliminada.

Este será el resultado:

```
Mari-Gutierrez : 29
circunstantes : 24
entretenimiento : 17
simplicidades : 14
consideracion : 14
```

5.- Métodos seek y tell

En este capítulo vamos a ver nuevos métodos para trabajar con archivos de texto, el método seek y el método tell.

Hemos creado un archivo de texto llamado alfabeto.txt con el siguiente contenido:

alfabeto.txt

```
1   abcdefghijklmnñopqrstuvwxyz
```

Todo está en una sola línea.

Ahora vamos a ver el código:

```
1   archivo = open("alfabeto.txt", "r", encoding= "utf-8")
2   texto = archivo.read()
3   print(texto)
4   archivo.close()
```

Si lo ejecutamos este será el resultado:

```
abcdefghijklmnñopqrstuvwxyz
```

```
1   archivo = open("alfabeto.txt", "r", encoding= "utf-8")
2   texto = archivo.read()
3   print(texto)
4   texto2 = archivo.read()
5   print(texto2)
6   archivo.close()
```

En este ejemplo utilizamos texto para leer el archivo y después con texto2 volvemos a leer el archivo, si ejecutamos que pasa:

```
abcdefghijklmnñopqrstuvwxyz
```

Que solo lo muestra una vez, las segunda vez lo que nos muestra es una línea vacía.

¿Qué ha sucedido? Al crearse un objeto de tipo archivo se genera un puntero en el archivo de texto que apunta al comienzo del archivo, cuando lo hemos leído por primera vez ese puntero ha leído todo el texto y el puntero se ha posicionado el final del archivo, por lo tanto al querer leer de nuevo el archivo lo que hace es leer desde donde está situado el puntero, pero como está en el final no lee nada.

```
1   archivo = open("alfabeto.txt", "r", encoding= "utf-8")
2   texto = archivo.read(10)
3   print(texto)
4   texto2 = archivo.read()
5   print(texto2)
6   archivo.close()
```

En la línea 2 le estamos diciendo que solo lea los primeros 10 caracteres, cuando llegamos a la línea 4 el puntero lee desde la posición 11 hasta el final, si ejecutamos este será el resultado.

```
abcdefghij
klmnñopqrstuvwxyz
```

Tenemos dos métodos que podremos manejar el puntero, uno de ellos es el método tell.

```
1   archivo = open("alfabeto.txt", "r", encoding= "utf-8")
2   texto = archivo.read(10)
3   print(archivo.tell())
4   print(texto)
5   texto2 = archivo.read()
6   print(texto2)
7   archivo.close()
```

Nos va a indicar en que posición se encuentra el puntero.

```
1   archivo = open("alfabeto.txt", "r", encoding= "utf-8")
2   texto = archivo.read(10)
3   print(archivo.tell())
4   print(texto)
5   texto2 = archivo.read()
6   print(texto2)
7   archivo.seek(0)
8   texto3 = archivo.read()
9   print(texto3)
10  archivo.close()
```

Con el método seek le decimos en que posición queremos que se sitúe el puntero, vamos a ejecutar para ver lo que ocurre.

```
10
abcdefghij
klmnñopqrstuvwxyz

abcdefghijklmnñopqrstuvwxyz
```

Podemos volver a leer de nuevo todo el archivo de texto.

```
7   archivo.seek(5)
```

Si lo cambiamos por el número 5 veremos los siguiente:

```
fghijklmnñopqrstuvwxyz
```

Los primeros 5 caracteres no los muestra.

Vamos a cambiar el contenido del archivo alfabeto.txt por:

ñññññ

lo guardaremos con codificación UTF-8

```
1   archivo = open("alfabeto.txt", "r", encoding= "utf-8")
2   archivo.seek(0)
3   texto = archivo.read()
4   print(texto)
5   archivo.close()
```

Cuando queremos leer a partir de una determinada posición utilizamos el método seek, pero cuando utilizamos algunos caracteres como es la ñ el encoding utf-8 utiliza un byte para agregar información sobre dicho carácter, esto significa quedaría de la siguiente forma:

	ñ		ñ		ñ		ñ		ñ
1	2	3	4	5	6	7	8	9	10

Los caracteres que están en blanco están siendo reservados para que utf-8 guarde información del carácter que le precede, esto quiere decir que si queremos leer las posiciones 1, 3, 5, 7 y 9 vamos a tener un error, en cambio se leemos desde la posición 2, 4, 6, 8, 10 veremos como nos muestra dicha información desde la posición especificada hasta el final.

```
2   archivo.seek(1)
```

Ejecutamos:

```
Traceback (most recent call last):
  File "f:\ArchivosPython\archivos1.py", line 3, in <module>
    texto = archivo.read()
            ^^^^^^^^^^^^^^
  File "<frozen codecs>", line 322, in decode
UnicodeDecodeError: 'utf-8' codec can't decode byte 0xb1 in position 0: invalid start byte
PS F:\ArchivosPython>
```

```
2   archivo.seek(2)
```

Ejecutamos:

```
ññññ
```

La pregunta es si quiero que me imprima 2 eñes desde que posición me tengo que situar.

Pues tiene que ser desde la posición 6.

6.- Gestor de contexto

Lo que hacíamos hasta ahora era crear un objeto de tipo archivo mediante la función predefinida open llamaba a uno de sus métodos, por ejemplo el método read() y cerrarlo una vez no vayamos a utilizarlo.

Vamos a ver una forma más eficiente de llevar a cabo esta tarea con un gestor de contexto.

Que es un gestor de contexto es un objeto que define el contexto que se establece al ejecutar una sentencia with.

El gestor de contexto maneja la entrada y la salida a la ejecución de un bloque de código, que es el que está contenido en la sentencia with.

```
1   # archivo = open("notas1.txt", "r")
2   # texto = archivo.read()
3   # print(texto)
4   # archivo.close()
5
6   with open("nota1.txt", "r") as archivo:
7       texto = archivo.read()
8
9       print(texto)
```

Hace lo mismo que el código comentado, pero con la ventaja de que no hay que cerrar el archivo ya que al salir del bloque with cierra el archivo e incluso se produjese algún error antes de salir del bloque igualmente se cerraría correctamente el archivo.

Contenido del archivo nota.txt.

```
1   Esto es una nota 1.
```

Vamos a ejecutar.

```
Esto es una nota 1.
```

Hemos creado una carpeta llamada notas y dentro hemos pasado el archivo nota1.txt.

```
1   with open("notas/nota1.txt", "r") as archivo:
2       texto = archivo.read()
3
4       print(texto)
```

La barra es la barra normal no la invertida.

```
1   with open("notas\\nota1.txt", "r") as archivo:
2       texto = archivo.read()
3
4       print(texto)
```

También se pueden utilizar dos barras invertidas.

```
1   with open("notas/nota3.txt", "r") as archivo:
2       texto = archivo.read()
3
4       print(texto)
```

Imagina que queremos abrir un archivo que no existe.

```
Traceback (most recent call last):
  File "f:\ArchivosPython\archivos2.py", line 1, in <module>
    with open("notas/nota3.txt", "r") as archivo:
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
FileNotFoundError: [Errno 2] No such file or directory: 'notas/nota3.txt'
PS F:\ArchivosPython>
```

Nos sale el correspondiente mensaje.

```
1   import os
2
3   if os.path.exists("notas/nota3.txt"):
4
5       with open("notas/nota3.txt", "r") as archivo:
6           texto = archivo.read()
7
8           print(texto)
9   else:
10      print("Este archivo no existe.")
```

Este será el resultado:

```
Este archivo no existe.
```

Vamos a cambiarlo por nota2.txt

archivos2.py > ...

```
1   import os
2
3   if os.path.exists("notas/nota2.txt"):
4
5       with open("notas/nota2.txt", "r") as archivo:
6           texto = archivo.read()
7
8           print(texto)
9   else:
10      print("Este archivo no existe.")
```

Este será el resultado:

```
Esto es una nota 2.
```

Con el módulo os podemos comprobar si existe, renombrarlo, eliminarlo, etc.

7.- Modos de abrir un archivo de texto

Vamos a ver los diferentes modos de abrir un archivo de texto.

Hasta solo hemos visto el modo lectura "r" con el que podemos abrir un archivo y leer su contenido, pero no podemos escribir nuevos datos.

Modo	FICHERO EXISTENTE	FICHRO NO EXISTENTE
r	Lee el fichero	Error
w	Sobrescribe el fichero	Crea fichero
a	Sólo añade texto al Final del fichero	Crea fichero
r+	Puede leer y escribir en cualquier posición	Error
w+	Sobrescribir el fichero y Puede leer lo escrito	Crear fichero
a+	Lee y añade texto al final	Crea fichero
x	Error	Crea fichero y escribe

```
1   with open("nota.txt", "w") as archivo:
2
3       archivo.write("Hola qué tal.")
4
5   with open("nota.txt", "r") as archivo:
6
7       texto = archivo.read()
8
9       print(texto)
```

```
Hola qué tal
```

Abrimos un archivo en modo de escritura, si este no existe lo crea, escribimos en el “Hola qué tal.”, el archivo se cierra automáticamente. A continuación lo abrimos en modo lectura, le pasamos el contenido a la variable texto y al final se muestra el contenido.

```
1   with open("nota.txt", "w") as archivo:
2
3       archivo.write("Buenos días")
4
5   with open("nota.txt", "r") as archivo:
6
7       texto = archivo.read()
8
9       print(texto)
```

Modificamos lo que vamos a escribir en el archivo y lo ejecutamos de nuevo.

```
Buenos días
```

Al ejecutar de nuevo lo que ha hecho a sido sobrescribirlo haciendo que la información anterior se pierda.

```
1   with open("nota.txt", "w") as archivo:
2
3       archivo.write("Buenos días" + "\n")
4       archivo.write("Hasta luego.")
5
6   with open("nota.txt", "r") as archivo:
7
8       texto = archivo.read()
9
10      print(texto)
```

Abrimos el archivo como escritura, escribimos dos veces sin cerrar el archivo y finalmente lo abrimos para leer el contenido del archivo.

```
Buenos días
Hasta luego.
```

```
1   with open("nota.txt", "a") as archivo:
2
3       archivo.write("Buenos días 2" + "\n")
4       archivo.write("Hasta luego 2.")
5
6   with open("nota.txt", "r") as archivo:
```

```
7
8       texto = archivo.read()
9
10      print(texto)
```

Abrimos el archivo en modo añadir, agregamos dos líneas y a continuación mostramos su contenido.

```
Buenos días
Hasta luego.Buenos días 2
Hasta luego 2.
```

Si volvemos a ejecutar.

```
Buenos días
Hasta luego.Buenos días 2
Hasta luego 2.Buenos días 2
Hasta luego 2.
```

```
1   with open("nota.txt", "r+") as archivo:
2       texto = archivo.read()
3       print(texto)
4       archivo.write("Buenos días 3." + "\n")
5       archivo.write("Hasta luego 3.")
6       texto = archivo.read()
7       print(texto)
```

Si ejecutamos

```
Buenos días 2
Hasta luego.Buenos días 2
Hasta luego 2.
```

Las dos líneas nuevas no las ha podido leer porque el puntero esta al final, si vemos el contenido del fichero:

```
Buenos días 2
Hasta luego.Buenos días 2
Hasta luego 2.

Buenos días 3.
Hasta luego 3.
```

También lo podemos realizar de la siguiente forma.

Tenemos el siguiente archivo con su contenido.

```
nota.txt
1    aaaaaaaaaa
```

Es una línea que contiene solo la letra a repetida varias veces.

```
1  with open("nota.txt", "r+") as archivo:
2      archivo.write("b")
3      archivo.seek(0)
4      texto = archivo.read()
5      print(texto)
```

Este será el resultado:

```
baaaaaaaaa
```

A cambiado el primer carácter que era una a por una b.

```
1  with open("nota.txt", "r+") as archivo:
2      archivo.seek(0,2)  # primera posición por el final
3      archivo.write("b")
4      archivo.seek(0)
5      texto = archivo.read()
6      print(texto)
```

Este será el resultado:

```
baaaaaaaaab
```

A cambiado el último carácter que era una a por una b.

```
1  with open("nota.txt", "r+") as archivo:
2      archivo.seek(5)
3      archivo.write("b")
4      archivo.seek(0)
5      texto = archivo.read()
6      print(texto)
```

Este será el resultado:

```
baaaabaaaab
```

A cambiado el carácter de la posición 5 que antes era una a por una b.

8.- Cuaderno de notas con archivo de texto (Presentación)

En este capítulo vamos a presentar un programa que vamos a llevar a cabo en los próximos capítulos y en el que vamos a utilizar archivos de texto.

Se trata de un cuaderno notas, es decir un programa que nos va a permitir ir a tomar notas y que las guarde para luego si salimos del programa y volvemos a entrar en el en otro momento podamos leerlas, borrarlas o crear nuevas notas.

Este proyecto consiste en un archivo de Python no hay ningún archivo de texto.

Vamos a ejecutarlo:

```
-----------------------------------------------------
                CUADERNO DE NOTAS
--------------------- NOTAS ---------------------
No hay notas
-------------------- OPCIONES --------------------
1. Crear nueva nota
2. Leer una nota
3. Cambiar nombre nota
4. Borrar una nota
5. Salir del programa
-----------------------------------------------------
---> _
```

Una vez ejecutado el programa y habiendo agregado nuestra primera nota se nos creará dos archivos de texto. Uno notas y otro primera.

Si abrimos el de Notas aparece el título de la nota.

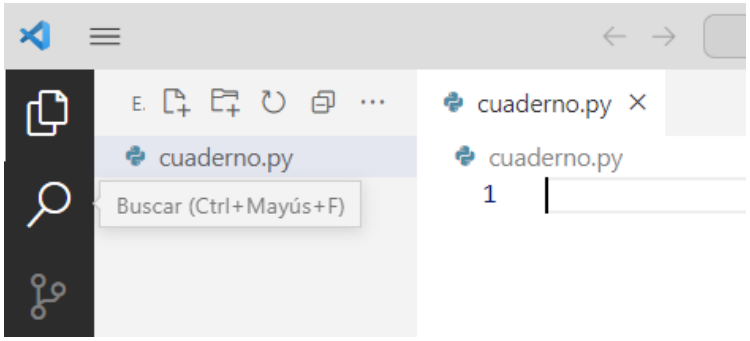
Si abrimos el archivo primera el contenido de la nota.

Lo que vamos a hacer es utilizar un archivo de texto para guardar todos los nombres de las notas, y otro archivo de texto para guardar el contenido de cada una de las notas.

En este ejercicio vamos a crear un archivo para guardar los nombres de las notas y un archivo por cada una de las notas que tomemos.

Para este proyecto vamos a crear una carpeta llamado Cuaderno.

Como nombre del proyecto cuaderno.py.



9.- Cuaderno de notas (Código primera parte)

```
"""
Cuaderno de notas, que permite tomar notas y
guardarlas.
Permite también leerlas, borrarlas y cambiarles de
nombre.
"""
# Con esta librería podemos interactuar con los
archivos
import os
```

Si el archivo notas.txt no existe, pues lo creas.

```
if not os.path.exists("notas.txt"):
    with open("notas.txt", "w", encoding = "utf-8") as
archivo_notas:
        pass  # Solo queremos crear el archivo

while True:
    os.system("cls")
```

Abrimos el archivo notas.txt en modo lectura.

```
    with open("notas.txt", "r", encoding = "utf-8") as
archivo_notas:
        lista_notas = archivo_notas.readlines()
```

Pasamos las líneas que contiene notas.txt a una lista.

```
    for i in range(len(lista_notas)):
        lista_notas[i] = lista_notas[i].strip()
```

A la lista_notas le eliminamos los saltos de línea \n y los espacios en blanco

```
    print("-----------------------------------------
--------")
    print("                    CUADERNO DE
NOTAS                    ")
    print("-----------------------NOTAS---------------
--------")
```

Si len(lista_notas) es igual a 0 significa que no tiene ningún elemento de lo contrario muestra el número de nota más la nota.

```
    if len(lista_notas) == 0:
        print("No hay notas ")
    else:
        n = 0
        for nota in lista_notas:
            n += 1
            print("Nota {}: {}".format(n, nota))
```

```
        # for n, nota in enumerate(lista_notas, 1):
        # print(n, nota)
        # Hace lo mismo que el bucle anterior.
```

```
    print("------------------OPCIONES-----------------
--")
    print("1. Crea nueva nota")
    print("2. Leer una nota")
    print("3. Cambiar nombre nota")
    print("4. Borrar una nota")
    print("5. Salir del programa")
    print("-----------------------------------------
--")
```

Muestra en consola el menú con 5 opciones que podremos selelccionar.

```
    opcion = ""
    while opcion not in ("1", "2", "3", "4", "5"):
        opcion = input("--->")
```

Este bucle controla que si contestamos por un valor distinto a 1, 2, 3, 4 o 5, no salgamos del bucle.

```
    if opcion == "1":
```

Si elegimos la opción 1.

```
        nombre = input("Nombre nota: ").strip()
        if os.path.exists(nombre + ".txt"):
            print("Ese nombre ya existe")
            print()
            input("Enter para continuar ...")
        else:
            with open("notas.txt", "a", encoding =
"utf-8") as archiovo_notas:
                archiovo_notas.write(nombre + "\n")

            with open(nombre + ".txt", "w", encoding =
"utf-8") as nuevo_archivo:
                while True:
                    print("Escribe contenido ('q' para
salir)")
                    contenido = input("> ")
                    if contenido == "q":
                        break
                    else:
                        nuevo_archivo.write(contenido +
"\n")
```

Nos preguntará el nombre de una nueva nota, sí ésta ya existe, nos avisará.

De lo contrario abriremos el archivo de notas.txt en modo de añadir, escribiremos el título de la nueva nota. Creamos un nuevo archivo con el nombre que le hemos dado para escribir el contenido de la nota.

Si elegimos la opción 2.

Nos preguntará por el número de la nota.

En la lista números_posibles se guardarán las notas que hay, si el número introducido no está entre los que hemos introducido, nos mostrará un mensaje.

Si la nota existe nos mostrará su contenido.

```
elif opcion == "2":
    numero = input("Número nota: ")
    numeros_posibles = []
    for i in range(1 , len(lista_notas)+1):
        numeros_posibles.append(str(i))
    if numero not in numeros_posibles:
        print("Número de nota incorrecto.")
        print()
        input("Enter para continuar...")
    else:
        numero = int(numero)
        nota = lista_notas[numero-1]
    with open(nota + ".txt", "r", encoding =
"utf-8") as archivo_nota_leer:
        contenido_nota =
archivo_nota_leer.read()
    print("-----------------------------------
--------------")
    print()
    print(contenido_nota)
    print("-----------------------------------
--------------")
    input("Enter para continuar...")
```

10.- Cuaderno de notas (Código segunda parte)

Comprueba si el número de nota no existe.

Eligiendo la opción 3

```
elif opcion == "3":
        numero = input("Número nota: ")
        numeros_posible = []
        for i in range(1, len(lista_notas)+1):
            numeros_posible.append(str(i))
        if numero not in numeros_posible:
            print("Número de nota incorrecto.")
            print()
            input("Enter para continuar...")
        else:
            numero = int(numero)
            nuevo_nombre = input("Nuevo nombre: ")
            if os.path.exists(nuevo_nombre + ".txt"):
                print("Ese nombre ya existe")
                print()
                input("Enter para continuar...")
            else:
                nombre_cambiar = lista_notas[numero-1]
                os.rename(nombre_cambiar + ".txt",
nuevo_nombre + ".txt")
                lista_notas[numero-1]= nuevo_nombre
                with open("notas.txt", "w", encoding =
"utf-8") as archivo_notas:
                    for nota in lista_notas:
                        archivo_notas.write(nota +
"\n")
```

Comprueba que el número de nota noexiste.

De lo contrario cambia el nombre al archivo y modifica el archivo notas.txt para cambiar el título de la nota.

Eligiendo la opción 4

```
elif opcion == "4":
        numero = input("Número nota: ")
        numeros_posible = []
        for i in range(1, len(lista_notas)+1):
            numeros_posible.append(str(i))
        if numero not in numeros_posible:
            print("Número de nota incorrecto.")
            print()
            input("Enter para continuar...")
        else:
            borrar = int(numero)
            nota_borrar = lista_notas[borrar-1]
```

Pregunta por el numero de nota que queremos borrar, si no existe nos avisará con un mensaje de lo contrario borra el archivo de esa nota y además modifica el archivo notas.txt.

```
            os.remove(nota_borrar + ".txt")

            del lista_notas[borrar-1]
            with open("notas.txt", "w", encoding =
"utf-8") as archivo_notas:
                for nota in lista_notas:
                    archivo_notas.write(nota + "\n")

    elif opcion == "5":
        break
```

Eligiendo la opción 5

Salimos del bucle y finaliza el programa.

Vamos a ejecutar:

La primera vez que ejecutamos el programa ser crea el archivo notas.txt.

```
-----------------------------------------------------
                 CUADERNO DE NOTAS
-----------------------NOTAS-----------------------
No hay notas
-------------------OPCIONES-------------------
1. Crea nueva nota
2. Leer una nota
3. Cambiar nombre nota
4. Borrar una nota
5. Salir del programa
----------------------------------------------
--->1
```

Vamos a crear una nueva nota.

```
-----------------------------------------------------
                 CUADERNO DE NOTAS
-----------------------NOTAS-----------------------
No hay notas
-------------------OPCIONES-------------------
1. Crea nueva nota
2. Leer una nota
3. Cambiar nombre nota
4. Borrar una nota
5. Salir del programa
----------------------------------------------
--->1
Nombre nota: Visita al médico
Escribe contenido ('q' para salir)
> El 15 de Febrero tengo
Escribe contenido ('q' para salir)
> hora con el média a las 16 horas
Escribe contenido ('q' para salir)
> q
```

Con la q salimos.

```
-----------------------------------------------------
                 CUADERNO DE NOTAS
-----------------------NOTAS-----------------------
Nota 1: Visita al médico ←
-------------------OPCIONES-------------------
1. Crea nueva nota
2. Leer una nota
3. Cambiar nombre nota
4. Borrar una nota
5. Salir del programa
-----------------------------------------------
--->
```

Vamos a agregar dos notas de más.

```
-----------------------------------------------------
                 CUADERNO DE NOTAS
-----------------------NOTAS-----------------------
Nota 1: Visita al médico
Nota 2: Pasar ITV
Nota 3: Pagar contribucion
-------------------OPCIONES-------------------
1. Crea nueva nota
2. Leer una nota
3. Cambiar nombre nota
4. Borrar una nota
5. Salir del programa
-----------------------------------------------
--->
```

Quiero leer la nota 5 (ojo no hay nota 5).

```
-----------------------------------------------
                CUADERNO DE NOTAS
-----------------------NOTAS-----------------------
Nota 1: Visita al médico
Nota 2: Pasar ITV
Nota 3: Pagar contribucion
-------------------OPCIONES-------------------
1. Crea nueva nota
2. Leer una nota
3. Cambiar nombre nota
4. Borrar una nota
5. Salir del programa
-----------------------------------------------
--->2
Número nota: 5
Número de nota incorrecto.

Enter para continuar...
```

Un mensaje diciendo nota incorrecta, ahora vamos leer la nota 2.

```
-----------------------------------------------------
                 CUADERNO DE NOTAS
-----------------------NOTAS-----------------------
Nota 1: Visita al médico
Nota 2: Pasar ITV
Nota 3: Pagar contribucion
-------------------OPCIONES-------------------
1. Crea nueva nota
2. Leer una nota
3. Cambiar nombre nota
4. Borrar una nota
5. Salir del programa
----------------------------------------------
--->2
Número nota: 2
-----------------------------------------------------

El día 18 de abril
tengo hora con el mecanico
Para pasar itv

-----------------------------------------------------
Enter para continuar...
```

Vamos a cambiar el titulo 2 por llevar coche mecánico.

```
-----------------------------------------------------
                 CUADERNO DE NOTAS
-----------------------NOTAS-----------------------
Nota 1: Visita al médico
Nota 2: Pasar ITV
Nota 3: Pagar contribucion
-------------------OPCIONES-------------------
1. Crea nueva nota
2. Leer una nota
3. Cambiar nombre nota
4. Borrar una nota
5. Salir del programa
----------------------------------------------
--->3
Número nota: 2
Nuevo nombre: Llevar coche mecanico
```

```
-----------------------------------------------------
                  CUADERNO DE NOTAS
-----------------------NOTAS-----------------------
Nota 1: Visita al médico
Nota 2: Llevar coche mecanico ←
Nota 3: Pagar contribucion
-------------------OPCIONES-------------------
1. Crea nueva nota
2. Leer una nota
3. Cambiar nombre nota
4. Borrar una nota
5. Salir del programa
----------------------------------------------
--->
```

Vamos a liminar la nota 1.

```
-----------------------------------------------------
                  CUADERNO DE NOTAS
-----------------------NOTAS-----------------------
Nota 1: Llevar coche mecanico
Nota 2: Pagar contribucion
-------------------OPCIONES-------------------
1. Crea nueva nota
2. Leer una nota
3. Cambiar nombre nota
4. Borrar una nota
5. Salir del programa
----------------------------------------------
--->
```

11.- Generador de test (Presentación)

Vamos a llevar a cabo un nuevo proyecto en que vamos a trabajar con archivos de texto, se va a tratar de un generador de test.

```
------------------------
   GENERADOR DE TEST
------------------------
1. Entrar al sistema
2. Darse de alta
3. Salir
-----------------------
-->
```

En este proyecto no va a ser importante los test que hagamos que va a ser un test de matemáticas muy sencillo, sino que lo importante va a ser entrar al sistema con un usuario y contraseña, sino se está dado de alta darse de alta, acceder al test, conseguir una puntuación y que la puntuación se guarde de tal forma que si salimos del programa y volvemos a entrar se mantengan los datos.

```
------------------------
   GENERADOR DE TEST
------------------------
1. Entrar al sistema
2. Darse de alta
3. Salir
-----------------------
--> 1
Introduce usuario: manuel
Introduce contraseña: 1234
Datos incorrectos ...
```

Intentamos acceder al sistema pero Manuel aun no se ha registrado.

```
-------------------------
   GENERADOR DE TEST
-------------------------
1. Entrar al sistema
2. Darse de alta
3. Salir
-----------------------
--> 2
Elige usuario: manuel
Elige contraseña: 1234
Repite contraseña: 1234
Usuario dado de alta.
Enter para continuar ...
```

Ya estamos dado de alta ahora volveremos a seleccionar la opción 1.

```
-------------------------
   GENERADOR DE TEST
-------------------------
1. Entrar al sistema
2. Darse de alta
3. Salir
-----------------------
--> 1
Introduce usuario: manuel
Introduce contraseña: 1234
```

Ya hemos entrado en el sistema.

```
---------------------------
    USUARIO: manuel
---------------------------
1. Hacer test
2. Puntuaciones
3. Salir
---------------------------
-->
```

Vamos a ver las puntuaciones seleccionando la opción 2.

```
---------------------------
   PUNTUACIONES: manuel
---------------------------
  Veces | Mejor | Media
---------------------------
    0   |   0   |  0
---------------------------

Enter para continuar ...
```

No hay ninguna puntuación.

Pues vamos a hacer un test.

```
---------------------------
    USUARIO: manuel
---------------------------
1. Hacer test
2. Puntuaciones
3. Salir
---------------------------
--> 1
```

Nos realiza 5 preguntas.

```
-----------------------
      Pregunta 2
-----------------------
4 + 9 = 13

    ¡ Correcto !

Enter para continuar ...
```

Alguna la aciertas y en otras te equivocas.

```
-----------------------
      Pregunta 4
-----------------------
4 + 6 = 5

    ¡ Incorrecto !

Enter para continuar ...
```

Al final te dará la puntuación.

```
-----------------------
      FIN DEL TEST
-----------------------

Has conseguido 2 puntos.

Enter para continuar ...
```

Le damos a Enter para continuar y vamos a ver las puntuaciones.

```
---------------------------
    USUARIO: manuel
---------------------------
1. Hacer test
2. Puntuaciones
3. Salir
---------------------------
--> 2
```

```
---------------------------
   PUNTUACIONES: manuel
---------------------------
  Veces | Mejor | Media
---------------------------
    1   |   2   |  2
---------------------------

Enter para continuar ...
```

Le damos a Enter para continuar.

Si quieres puede realizar un segundo test.

```
------------------------
      FIN DEL TEST
------------------------

Has conseguido 4 puntos.

Enter para continuar ...
```

El nuevo resultado.

Presionamos Enter para continuar.

Vamos a ver la puntuación.

```
---------------------------
    USUARIO: manuel
---------------------------
1. Hacer test
2. Puntuaciones
3. Salir
---------------------------
--> 2
```

```
---------------------------
   PUNTUACIONES: manuel
---------------------------
  Veces | Mejor | Media
---------------------------
    2   |   4   |  3.0
---------------------------

Enter para continuar ...
```

Enter para continuar, vamos a salir.

```
---------------------------
    USUARIO: manuel
---------------------------
1. Hacer test
2. Puntuaciones
3. Salir
---------------------------
--> 3
```

```
-------------------------
   GENERADOR DE TEST
-------------------------
1. Entrar al sistema
2. Darse de alta
3. Salir
-------------------------
--> 3
```

Salimos del sistema y del programa.

Ahora volvemos a entrar.

```
-------------------------
   GENERADOR DE TEST
-------------------------
1. Entrar al sistema
2. Darse de alta
3. Salir
-------------------------
--> 1
Introduce usuario: manuel
Introduce contraseña: 1234
```

Comprobamos que Manuel está registrado.

```
-------------------------
    USUARIO: manuel
-------------------------
1. Hacer test
2. Puntuaciones
3. Salir
-------------------------
-->
```

Puede acceder está dado de alta, vamos a ver las puntuaciones.

```
---------------------------
    USUARIO: manuel
---------------------------
1. Hacer test
2. Puntuaciones
3. Salir
---------------------------
--> 2_
```

```
---------------------------
   PUNTUACIONES: manuel
---------------------------
  Veces | Mejor | Media
---------------------------
    2   |   4   |  3.0
---------------------------

Enter para continuar ...
```

Vemos que la puntuación sigue siendo la que conseguimos en la última vez que jugamos.

Pues el programa consiste en poder acceder al sistema, sino se tiene usuario y contraseña poderse dar de alta.

Una vez se tiene usuario y contraseña se puede entrar.

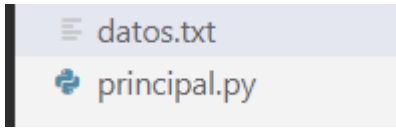
Una vez dentro del sistema nos permite hacer test y consultar la puntuación.

Vamos a trabajar con archivos de texto y lo realizaremos con funciones ya que hay diferentes opciones y el código se va a poder llevar a cabo más organizadamente con funciones.

12.- Generador de test (Código inicio)

Vamos a dar las opciones entrar al sistema y darse de alta.

Trabajaremos con el archivo principal.py que es el que contendrá el programa y para guardarse los dato el archivo datos.txt.



Vamos a empezar:

```
"""
Programa genera un test de matemáticas y guarda los puntos conseguidos.
El programa pide usuario y contraseña para acceder al sistema.
Si no está dado de alta permite darse de alta en el sistama.
"""
import random
import os

def listar_usuarios():
    ''' Abre el archivo donde están los usuarios y sus datos, y guarda
        los datos en una lista para su posterior uso'''
    with open("datos.txt", "r", encoding = "utf-8") as archivo_datos:
        lista_datos = archivo_datos.readlines()

    lista_usuarios = []
    for dato in lista_datos:
        lista_usuarios.append(dato.strip().split(","))

    return lista_usuarios

def menu_entrada():
    '''Menú de acceso al programa'''
    print("------------------------------")
    print("      GENERADOR DE TEST       ")
    print("------------------------------")
    print(" 1. Entrar al sistema         ")
    print(" 2. Darse de alta             ")
    print(" 3. Salir                     ")
    print("------------------------------")
    opcion = ""
    while opcion not in("1", "2", "3"):
        opcion = input(" --> ")
    return opcion

def menu_principal(nombre_usuario):
    print("------------------------------")
    print("      USUARIO: {}".format(nombre_usuario))
```

```
    print("-----------------------------")
    print(" 1. Hacer test")
    print(" 2. Puntuaciones")
    print(" 3. Salir")
    print("-----------------------------")
    opcion = ""
    while opcion not in("1", "2", "3"):
        opcion = input(" --> ")
    return opcion

def entrar_al_sistema(lista_usuarios):
    '''Pide usuario y contraseña y comprueba que los datos son
    correctos en la base de datos de usuarios'''
    usr = input(" Introduce usuario: ")
    con = input(" Introduce contraseña: ")

    for usuario in lista_usuarios:
        if usuario[0] == usr and usuario[1] == con:
            return usuario
        else:
            return None

def darse_de_alta(lista_usuarios):
    '''Pide usuario y la contraseña para dar de alta a un usuario'''
    while True:
        usr = input(" Elige usuario: ")
        if len(usr) < 4:
            print(" Ha de tener al menos 4 caracteres")
        else:
            for usuario in lista_usuarios:
                if usr == usuario[0]:
                    print(" Ese nombre está cogido")
                    break
            else:
                    while True:
                        con1 = input(" Elige contraseña: ")
                        if len(con1) < 4:
                            print(" Ha de tener al menos 4 caracteres")
                        else:
                            con2 = input(" Repite contraseña: ")
                            if con1 != con2:
                                print(" Contraseña no coincide")
                                break
                            else:
                                lista_usuarios.append([usr, con1, "0",
"0", "0"])
                                print(" Usuario dado de alta ")
                                return lista_usuarios
```

```
def grabar_datos(lista_usuarios):
    '''Guarda los nuevos datos de usuario en el archivo de texto'''
    with open("datos.txt", "w", encoding = "utf-8") as archivo_datos:
        for u in lista_usuarios:
            archivo_datos.write(u[0]+","+u[1]+","+u[2]+","+u[3]+","+u[4]+
"\n")
    return None

###########  FLUJO  DEL  PROGRAMA  ##########
if not os.path.exists("datos.txt"):
    with open("datos.txt", "w", encoding = "utf-8") as archivo_datos:
        pass

while True:
    jugando = False
    os.system("cls")
    lista_usuarios = listar_usuarios()
    opcion_entrada = menu_entrada()
    if opcion_entrada == "1":
        usuario = entrar_al_sistema(lista_usuarios)
        if usuario == None:
            input(" Datos incorrectos ...")
        else:
            jugando = True
    elif opcion_entrada == "2":
        lista_usuarios = darse_de_alta(lista_usuarios)
        grabar_datos(lista_usuarios)
        input(" Enter para continuar ...")

    elif opcion_entrada == "3":
        break
"""
Programa genera un test de matemáticas y guarda los puntos conseguidos.
El programa pide usuario y contraseña para acceder al sistema.
Si no está dado de alta permite darse de alta en el sistama.
"""
import random
import os

def listar_usuarios():
    ''' Abre el archivo donde están los usuarios y sus datos, y guarda
        los datos en una lista para su posterior uso'''
    with open("datos.txt", "r", encoding = "utf-8") as archivo_datos:
        lista_datos = archivo_datos.readlines()

    lista_usuarios = []
    for dato in lista_datos:
        lista_usuarios.append(dato.strip().split(","))
```

```
    return lista_usuarios

def menu_entrada():
    '''Menú de acceso al programa'''
    print("------------------------------")
    print("      GENERADOR DE TEST       ")
    print("------------------------------")
    print(" 1. Entrar al sistema         ")
    print(" 2. Darse de alta             ")
    print(" 3. Salir                     ")
    print("------------------------------")
    opcion = ""
    while opcion not in("1", "2", "3"):
        opcion = input(" --> ")
    return opcion

def menu_principal(nombre_usuario):
    print("------------------------------")
    print("      USUARIO: {}".format(nombre_usuario))
    print("------------------------------")
    print(" 1. Hacer test")
    print(" 2. Puntuaciones")
    print(" 3. Salir")
    print("------------------------------")
    opcion = ""
    while opcion not in("1", "2", "3"):
        opcion = input(" --> ")
    return opcion

def entrar_al_sistema(lista_usuarios):
    '''Pide usuario y contraseña y comprueba que los datos son
    correctos en la base de datos de usuarios'''
    usr = input(" Introduce usuario: ")
    con = input(" Introduce contraseña: ")

    for usuario in lista_usuarios:
        if usuario[0] == usr and usuario[1] == con:
            return usuario
        else:
            return None

def darse_de_alta(lista_usuarios):
    '''Pide usuario y la contraseña para dar de alta a un usuario'''
    while True:
        usr = input(" Elige usuario: ")
        if len(usr) < 4:
            print(" Ha de tener al menos 4 caracteres")
        else:
            for usuario in lista_usuarios:
```

```
                if usr == usuario[0]:
                    print(" Ese nombre está cogido")
                    break
            else:
                    while True:
                        con1 = input(" Elige contraseña: ")
                        if len(con1) < 4:
                            print(" Ha de tener al menos 4 caracteres")
                        else:
                            con2 = input(" Repite contraseña: ")
                            if con1 != con2:
                                print(" Contraseña no coincide")
                                break
                            else:
                                lista_usuarios.append([usr, con1, "0",
"0", "0"])
                                print(" Usuario dado de alta ")
                                return lista_usuarios

def grabar_datos(lista_usuarios):
    '''Guarda los nuevos datos de usuario en el archivo de texto'''
    with open("datos.txt", "w", encoding = "utf-8") as archivo_datos:
        for u in lista_usuarios:
            archivo_datos.write(u[0]+","+u[1]+","+u[2]+","+u[3]+","+u[4]+
"\n")
    return None


###########  FLUJO  DEL  PROGRAMA  ##########
if not os.path.exists("datos.txt"):
    with open("datos.txt", "w", encoding = "utf-8") as archivo_datos:
        pass

while True:
    jugando = False
    os.system("cls")
    lista_usuarios = listar_usuarios()
    opcion_entrada = menu_entrada()
    if opcion_entrada == "1":
        usuario = entrar_al_sistema(lista_usuarios)
        if usuario == None:
            input(" Datos incorrectos ...")
        else:
            jugando = True
```

```
elif opcion_entrada == "2":
    lista_usuarios = darse_de_alta(lista_usuarios)
    grabar_datos(lista_usuarios)
    input(" Enter para continuar ...")

elif opcion_entrada == "3":
    break
```

13.- Generador de test (Código final)

Lo seleccionado es el codigo añadido.

```
"""
Programa genera un test de matemáticas y guarda los puntos conseguidos.
El programa pide usuario y contraseña para acceder al sistema.
Si no está dado de alta permite darse de alta en el sistama.
"""
import random
import os

def listar_usuarios():
    ''' Abre el archivo donde están los usuarios y sus datos, y guarda
        los datos en una lista para su posterior uso'''
    with open("datos.txt", "r", encoding = "utf-8") as archivo_datos:
        lista_datos = archivo_datos.readlines()

    lista_usuarios = []
    for dato in lista_datos:
        lista_usuarios.append(dato.strip().split(","))

    return lista_usuarios

def menu_entrada():
    '''Menú de acceso al programa'''
    print("------------------------------")
    print("      GENERADOR DE TEST       ")
    print("------------------------------")
    print(" 1. Entrar al sistema         ")
    print(" 2. Darse de alta             ")
    print(" 3. Salir                     ")
    print("------------------------------")
    opcion = ""
    while opcion not in("1", "2", "3"):
        opcion = input(" --> ")
    return opcion

def menu_principal(nombre_usuario):
    print("------------------------------")
    print("      USUARIO: {}".format(nombre_usuario))
    print("------------------------------")
    print(" 1. Hacer test")
    print(" 2. Puntuaciones")
    print(" 3. Salir")
    print("------------------------------")
    opcion = ""
    while opcion not in("1", "2", "3"):
        opcion = input(" --> ")
```

```
    return opcion

def entrar_al_sistema(lista_usuarios):
    '''Pide usuario y contraseña y comprueba que los datos son
    correctos en la base de datos de usuarios'''
    usr = input(" Introduce usuario: ")
    con = input(" Introduce contraseña: ")

    for usuario in lista_usuarios:
        if usuario[0] == usr and usuario[1] == con:
            return usuario
        else:
            return None

def darse_de_alta(lista_usuarios):
    '''Pide usuario y la contraseña para dar de alta a un usuario'''
    while True:
        usr = input(" Elige usuario: ")
        if len(usr) < 4:
            print(" Ha de tener al menos 4 caracteres")
        else:
            for usuario in lista_usuarios:
                if usr == usuario[0]:
                    print(" Ese nombre está cogido")
                    break
            else:
                    while True:
                        con1 = input(" Elige contraseña: ")
                        if len(con1) < 4:
                            print(" Ha de tener al menos 4 caracteres")
                        else:
                            con2 = input(" Repite contraseña: ")
                            if con1 != con2:
                                print(" Contraseña no coincide")
                                break
                            else:
                                lista_usuarios.append([usr, con1, "0",
"0", "0"])
                                print(" Usuario dado de alta ")
                                return lista_usuarios

def grabar_datos(lista_usuarios):
    '''Guarda los nuevos datos de usuario en el archivo de texto'''
    with open("datos.txt", "w", encoding = "utf-8") as archivo_datos:
        for u in lista_usuarios:
            archivo_datos.write(u[0]+","+u[1]+","+u[2]+","+u[3]+","+u[4]+
"\n")
    return None
```

```
def test():
    '''Genera un test de matemáticas'''
    puntos = 0
    for i in range(5):
        os.system("cls")

        print("-----------------------")
        print("      Pregunta {}".format(i+1))
        print("-----------------------")

        op1 = random.randint(1, 9)
        op2 = random.randint(1, 9)
        pregunta = int(input(" {} + {} = ".format(op1, op2)))
        if pregunta == op1 + op2:
            print()
            print("     ¡  Correcto !")
            puntos += 1
        else:
            print()
            print("     ¡  Incorrecto !")

        print()
        input(" Enter para continuar ...")
    os.system("cls")

    print("-------------------------")
    print("       FIN DEL TEST      ")
    print("-------------------------")
    print()
    print(" Has conseguido {} puntos.".format(puntos))
    print()
    input(" Enter para continuar ...")
    return puntos
```

```
def actualizar_puntuacion(usuario, puntos):
    '''En base a los puntos obtenidos en una jugada, se actulzian los
datos
       de veces, mejor y media de usuario en la lista de sus datos.'''

    usuario[2] = int(usuario[2])
    usuario[3] = int(usuario[3])
    usuario[4] = float(usuario[4])

    usuario[2] += 1
    if puntos > usuario[3]:
        usuario[3] = puntos
    if usuario[2] == 1:
        usuario[4]= puntos
    else:
```

En la página siguiente seguimos en la misma función.

```
        usuario[4]= (usuario[4] + puntos) / 2

    usuario[2] = str(usuario[2])
    usuario[3] = str(usuario[3])
    usuario[4] = str(round(usuario[4],2))
    return usuario
```

```
def mostrar_puntuacion(usuario):
    '''Se muestra la puntuación de un usuario'''
    os.system("cls")
    print("------------------------------")
    print("      PUNTUACIONES: {}".format(usuario[0]))
    print("------------------------------")
    print("   Veces  |  Mejor  |  Media  ")
    print("------------------------------")
    print("     {}    |   {}     |    {}   ".format(usuario[2],
usuario[3], usuario[4]))
    print("------------------------------")
    return None
```

```
###########  FLUJO  DEL  PROGRAMA  ##########
# .exists se puede cambiar por .isfile
if not os.path.exists("datos.txt"):
    with open("datos.txt", "w", encoding = "utf-8") as archivo_datos:
        pass

while True:
    jugando = False
    os.system("cls")
    lista_usuarios = listar_usuarios()
    opcion_entrada = menu_entrada()
    if opcion_entrada == "1":
        usuario = entrar_al_sistema(lista_usuarios)
        if usuario == None:
            input(" Datos incorrectos ...")
        else:
            jugando = True
    elif opcion_entrada == "2":
        lista_usuarios = darse_de_alta(lista_usuarios)
        grabar_datos(lista_usuarios)
        input(" Enter para continuar ...")

    elif opcion_entrada == "3":
        break
    while jugando:
        os.system("cls")
        opcion = menu_principal(usuario[0])
        if opcion == "1":
            puntos = test()
```

El bucle continua en la siguiente página.

```
            usuario = actualizar_puntuacion(usuario, puntos)
            # No hace falta actualizar list_usuarios, porque usuario está
            # vinculado a ella:usuario =
lista_usuarios[indice_corespondiente]
            grabar_datos(lista_usuarios)
        elif opcion == "2":
            mostrar_puntuacion(usuario)
            print()
            input(" Enter para continuar ...")
        elif opcion == "3":
            jugando = False
```

Vamos a probar el programa.

Primero borra el archivo datos.txt y a continuación ejecutamos el programa.

```
-------------------------------
      GENERADOR DE TEST
-------------------------------
 1. Entrar al sistema
 2. Darse de alta
 3. Salir
-------------------------------
 --> 
```

Comprobarás que al ejecutar el programa el automáticamente crea el archivo datos.txt.

Vamos a elegir la opción 1 Entrar al sistema.

```
-------------------------------
      GENERADOR DE TEST
-------------------------------
 1. Entrar al sistema
 2. Darse de alta
 3. Salir
-------------------------------
 --> 1
 Introduce usuario: Pere
 Introduce contraseña: 1234
 Datos incorrectos ...
```

Como no estoy registrado, el sistema me dice datos incorrectos.

Lo que tenemos que hacer es registrarnos, opción 2 darse de alta.

```
------------------------------
      GENERADOR DE TEST
------------------------------
 1. Entrar al sistema
 2. Darse de alta
 3. Salir
------------------------------
 --> 2
 Elige usuario: Pere
 Elige contraseña: 1234
 Repite contraseña: 1234
 Usuario dado de alta
 Enter para continuar ...
```

Ya estamos dados de alta, ahora vamos a ver lo que ha pasado en el archivo datos.txt.

datos.txt

```
1    Pere,1234,0,0,0
```

Ya contiene los datos de un usuario.

Ahora ya podemos entrar al sistema, seleccionando de nuevo la opción 1.

```
------------------------------
      GENERADOR DE TEST
------------------------------
 1. Entrar al sistema
 2. Darse de alta
 3. Salir
------------------------------
 --> 1
 Introduce usuario: Pere
 Introduce contraseña: 1234
```

Seleccionamos la opción 1 de Entrar al sistema, y después de haber introducido correctamente el nombre de usuario y contraseña, ya hemos entrado en el sistema.

```
------------------------------
      USUARIO: Pere
------------------------------
 1. Hacer test
 2. Puntuaciones
 3. Salir
------------------------------
 -->
```

Es la primera vez que hemos entrado, si nos vamos al apartado 2 de Puntuaciones, tendrá que estar vacío.

```
------------------------------
      PUNTUACIONES: Pere
------------------------------
   Veces  |  Mejor  |  Media
------------------------------
     0    |    0    |    0
------------------------------

 Enter para continuar ...
```

Le damos Enter para continuar.

```
------------------------------
      USUARIO: Pere
------------------------------
 1. Hacer test
 2. Puntuaciones
 3. Salir
------------------------------
 --> 1
```

Seleccionamos la opción 1 para realizar un test.

Vamos a contestar 5 preguntas de matemáticas vamos a contestar 3 bien y dos mal.

```
-----------------------
      Pregunta 1
-----------------------
 1 + 5 = 6

     ¡  Correcto !

 Enter para continuar ...
```

```
-----------------------
      Pregunta 2
-----------------------
 5 + 6 = 15

     ¡  Incorrecto !

 Enter para continuar ...
```

```
-----------------------
      Pregunta 3
-----------------------
 6 + 7 = 13

     ¡  Correcto !

 Enter para continuar ...
```

```
-----------------------
      Pregunta 4
-----------------------
 5 + 6 = 15

     ¡  Incorrecto !

 Enter para continuar ...

 -----------------------
       Pregunta 5
 -----------------------
  8 + 4 = 12

      ¡  Correcto !

  Enter para continuar ...

 -------------------------
        FIN DEL TEST
 -------------------------

  Has conseguido 3 puntos.

  Enter para continuar ...
```

Ya hemos obtenido la puntuación de nuestro primer test.

Enter para continuar…

```
------------------------------
      USUARIO: Pere
------------------------------
 1. Hacer test
 2. Puntuaciones
 3. Salir
------------------------------
 --> 2
```

Vamos a seleccionar la opción 2 para ver las puntuaciones.

```
-----------------------------
      PUNTUACIONES: Pere
-----------------------------
   Veces   |  Mejor   |  Media
-----------------------------
     1     |   3      |    3
-----------------------------

 Enter para continuar ...
```

Ahora vamos a salir del programa, volvemos a entrar con nuestro nombre de usuario y contraseña y nos vamos de nuevo a la opción de puntuaciones.

```
-----------------------------
      PUNTUACIONES: Pere
-----------------------------
   Veces   |  Mejor   |  Media
-----------------------------
     1     |   3      |    3
-----------------------------

 Enter para continuar ...
```

Los datos se mantienen gracias al fichero datos.txt que nos permite recuperar la información que anteriormente se grabó en dicho fichero.