Ficheros - Lista de ejercicios - Soluciones en el curso

Lista de ejercicios que corresponden a los ficheros. Las soluciones las dejo junto con este documento en un fichero zip. ***En este momento ya podemos trabajar haciendo uso de funciones y excepciones como hemos aprendido en las dos secciones anteriores*** *con lo que hemos aprendido en la sección.*

1. Crear un programa que abra un fichero de texto y después imprima todo el contenido con un print y NO línea a línea

Contenido del fichero (exercise1.txt):

Hola, soy Anartz.
Estoy haciendo los ejercicios de ficheros.
Primer ejercicio.

Debe de imprimir ese contenido en tres líneas tal cual.

2. Escribe un programa en el que abrimos un fichero de varias líneas y cuando lo tengamos abierto, especificamos con una entrada desde teclado cuantas líneas queremos visualizar.

Contenido del fichero (exercise2-3.txt):

Linea 1.
Linea 2.
Linea 3.
Linea 4.
Linea 5.
Linea 6.

Resultado si especificamos que queremos mostrar las 3 primeras líneas:

Linea 1.

Linea 2.

Linea 3.

3. Le damos continuidad al ejercicio anterior y lo que hacemos es en vez de mostrar la información con los saltos de línea (el carácter de escape que tiene asignado el fichero y que deja un hueco por medio) es eliminar ese hueco por una de las funciones de string que hemos trabajado en la sección de Strings.

El contenido del fichero, es el mismo que hemos usado anteriormente y vamos a mostrar las primeras 3 líneas, siendo el resultado:

Linea 1.
Linea 2.
Linea 3.

4. Añadir texto línea a línea con una entrada por teclado hasta que se introduzca “EXIT”. Cuando ya dejamos de introducir, cerrar el flujo y posteriormente abrimos ese fichero y hacemos como en el ejercicio 1 mostrar todo el contenido después de abrirlo. **El fichero si no existe se creará y si existe, se irá añadiedo la información anterior sin sobrescribir lo anterior.**

5. Crear un programa que lea un fichero línea a línea y a su vez vaya almacenando esta información en una lista (Podemos usar el de cualquier ejercicio anterior) para mostrar todo el contenido de golpe con un print al finalizar el programa

6. Lo mismo que en el anterior pero limpiando el caracter de escape

(Almacenar la información limpia, sin ningún caracter de escape como un salto de línea)

7. Leer de un fichero todas las líneas y extraer la información sin caracteres de escape (por ejemplo salto de línea) y determinar cual de ellos es la palabra más larga. Procurad que solo haya una palabra más larga respecto a las otras.

En el resultado determinar la palabra más larga y cuantos caracteres de longitud tiene.

8. Utilizando como base el ejercicio anterior, vamos a implementar la función para que almacene una ó más palabras en una lista, en el caso de que haya más de una palabra con la longitud máxima que sea igual.

El resultado que obtendremos será el siguiente:

Si hay un resultado:

```
['rojo', 'colorado', 'bermellon', 'amaranto', 'esmerald', 'turquesa']
La palabra 'bermellon' tiene la longitud máxima con 9 caracteres
```

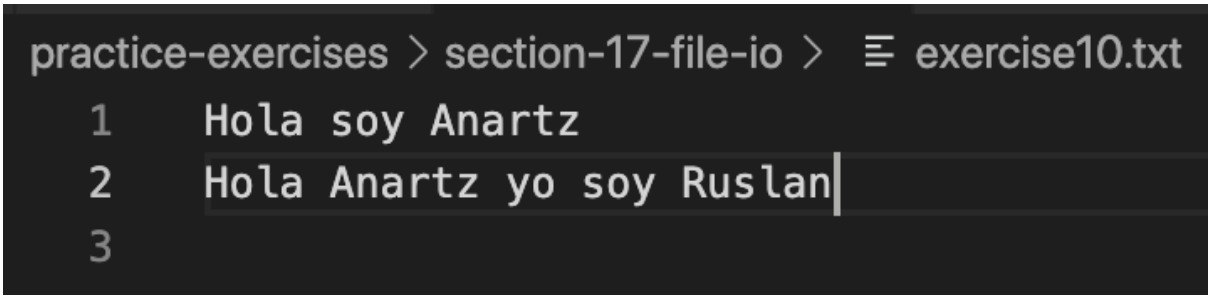
Si hay dos ó más resultados:

```
['rojo', 'colorado', 'bermellon', 'amaranto', 'esmeralda', 'turquesa']
Las palabras '['bermellon', 'esmeralda']' tienen la longitud máxima con 9 caracteres
```

9. Escribir un programa que cuente las líneas que contiene un fichero de texto.

10. Escribir un programa que obtenga las palabras para almacenarlas especificando cuantas veces aparece cada una de ellas.

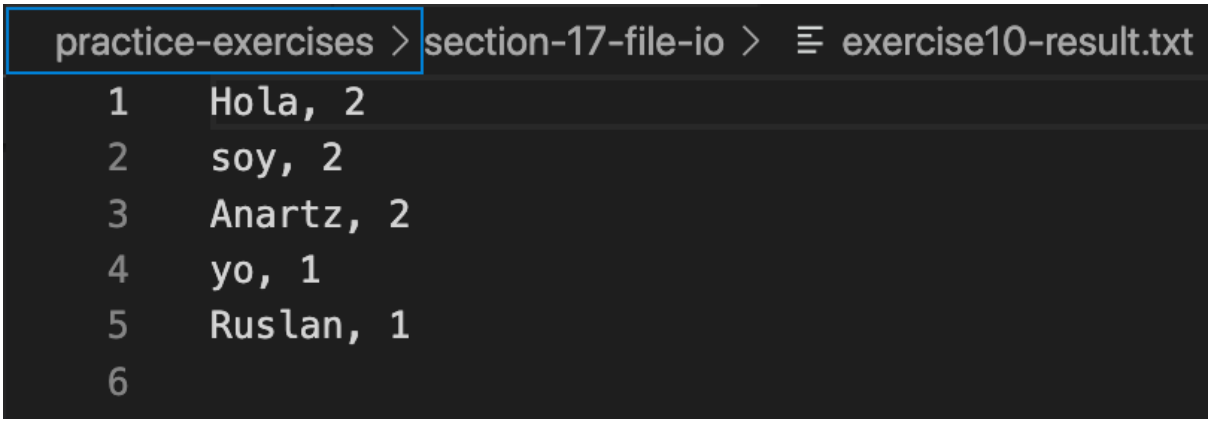
Ejemplo de contenido de fichero:



Resultado esperado:

```
[['Hola', 2], ['soy', 2], ['Anartz', 2], ['yo', 1], ['Ruslan', 1]]
```

Este resultado lo almacenamos en un fichero llamado “exercise10-result.txt” sobrescribiendo el contenido anterior, si lo hubiera.



11. Escribir un programa que **obtenga el contenido de un fichero de texto** y haga una **copia** almacenando esa información **en otro fichero nuevo**.

12. Escribe un programa abriendo un fichero de texto y comprueba que está abierto antes de cerrarlo y después de cerrarlo.

```
¿Cerrado? => False
¿Cerrado? => True
```

13. Escribir un programa que obtiene el contenido de un fichero de texto y devuelve el número de palabras que contiene.

Tener en cuenta: Algunas palabras están separadas por punto y coma, eliminarlos antes de hacer el cómputo.

Ejemplo de fichero con contenido:

Hola, soy Anartz.
Estoy haciendo los ejercicios de ficheros.
Primer ejercicio.

Resultado:

```
El fichero tiene 11 palabras
```

14. Escribe un programa que genere 26 ficheros de texto llamados desde A.txt, B.txt, ... hasta la Z.txt.

Usamos el siguiente string como referencia para poder crear los fiheros:

ABCDEFGHIJKLMNOPQRSTUVWXYZ

En la salida tenemos que tener 26 ficheros donde cada uno tiene como nombre el caracter seleccionado.

15. Desarrollar un programa que permita calcular el sueldo mensual de "n" empleados de una empresa.

Especificamos la cantidad de trabajadores desde teclado (evidentemente tiene que ser 1 ó más.

Una vez ingresado, iteramos añadiendo los siguientes datos por trabajador: nombre, categoría y número horas trabajadas.

Si la categoría es A, el pago por hora es de 50 dólares, si es B es de 80 dólares, si es C es de 90 dólares y si es D es de 120 dólares. Si no se introduce correctamente la categoría, el pago es de35 dólares.

Mostrar el pago que le corresponde a cada trabajador, el pago total que se debe hacer, el total de trabajadores que ganan menos de 5000 dólares, el total que ganan desde 5000 dólares a 9000 dólares y los que ganan más de 9000 dólares.

Toda esta información, para no perderla, la guardaremos en dos ficheros de texto línea a línea, por lo que todo los resutados que se han ido mostrando, han tenido que ser almacenados en una lista para después ir almacenándolo. Uno de los ficheros guardará la información, en cada línea la información del empleado y en el segundo, ya con el filtro aplicado de los sueldos.

Es importante que validemos las entradas, teniendo en cuenta las diferentes posibilidades.

Por ejemplo, si queremos introducir solo números, tendremos que manejar la excepción del tipo de valor, para controlar y no romper el programa.

Ficheros - Lista de ejercicios - Soluciones en el curso

CONSEJO: Ir desgranando el ejercicio en partes y depurando y a medida que vayáis haciendo las diferentes fases, seguir poco a poco

Sugerencia de lo que se verá, a la hora de pedir datos:

```
Introduzca número de empleados: 2
Introduce nombre / apellidos: Anartz
Introduce categoría A, B, C, D: A
Introduce horas trabajadas: 12
Introduce nombre / apellidos: Ruslan
Introduce categoría A, B, C, D: d
Introduce horas trabajadas: 23
```

Ficheros que se crean a partir del ejercicio:

- employees.txt

practice-exercises > section-17-file-io > employees.txt

```
1  name,category,salary_per_hour,hours_work,salary
2  Anartz,A,50,12,600
3  Ruslan,D,120,23,2760
4
```

- employees_ranking_salary.txt

practice-exercises > section-17-file-io > employees.txt

```
1  name,category,salary_per_hour,hours_work,salary
2  Anartz,A,50,12,600
3  Ruslan,D,120,23,2760
4
```

Ficheros - Lista de ejercicios - Soluciones en el curso

16.- Crear un programa en el que vamos a gestionar las notas de los alumnos y alumnas de la academia online “Anartz Mugika Ledo - Desarrollo / Formación”.

Para hacer la gestión y no perder los datos, lo vamos a guardar en un fichero de texto, tal y como hemos aprendido.

Necesitamos saber:

- Número de alumnos/as
- Nombre del curso.

Ir introduciendo los datos de los alumnos/as:

- Nombre y apellidos
- Email
- Nota

Ahora que ya tenemos las notas con los datos de las personas que estudian un curso en la academia, debemos de calcular:

Promedio de la nota de todas las personas que estudian.

<suma notas de todos estudiantes> / <nº estudiantes> = NOTA MEDIA

Teniendo la nota media, contamos las personas que están por encima y las que está por debajo de esa media.

Una vez que tenemos esto, es conveniente tener preparado para guardar esa información en un fichero de texto. **Añadimos para quese sobrescriba siempre que añadamos nueva información.**

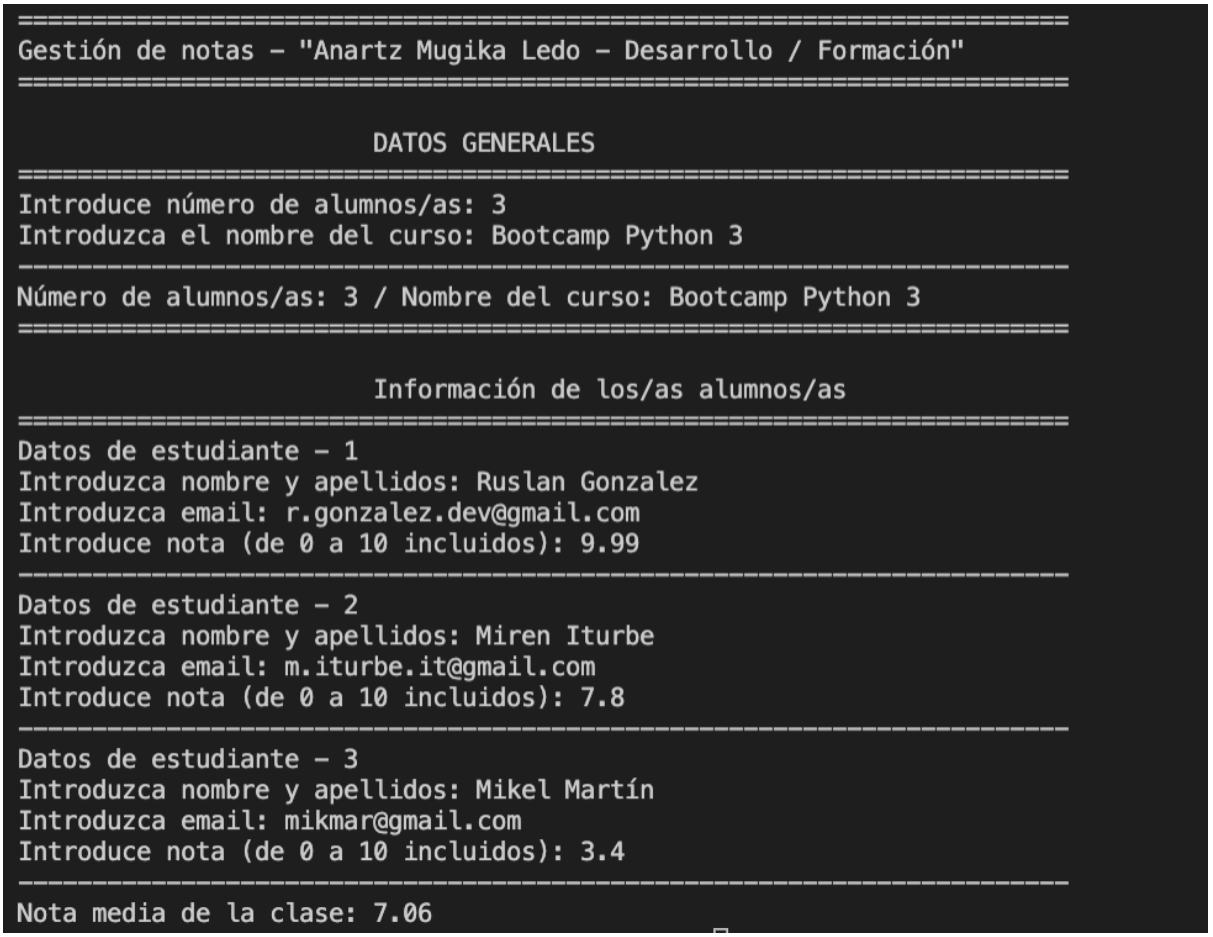
Lo que tendremos que tener al final del ejercicio almacenado, es lo siguiente:

- Listado de alumos/as con sus notas y el promediofinal **(con dos decimales solo)**: **students.txt**
- Estudiantes que están por encima de la media y por debajo (en dos bloques) en el fichero **report-students-avg-data.txt**

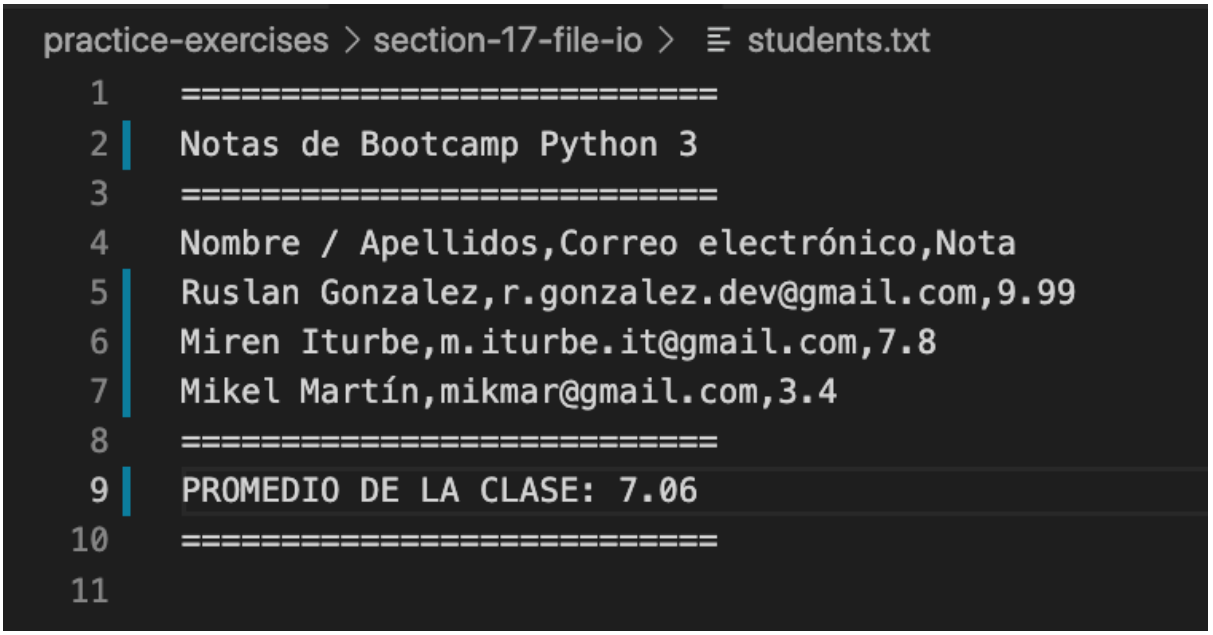
Añadir todas las validaciones, comprobaciones y puntos necesarios con condicionales, manejo de excepciones,...

Ejemplo:

Esto es lo que se mostraría cuando introducimos los datos (podéis darle el formato que queráis):



Resultado del fichero **students.txt**



Resultado del reporte con la clasificación de los/as alumnos/as teniendo en cuentala media de la clase.