

**«Alumno»**

César Luis Longart Torres

**Trabajo Obligatorio. Python**

**Agosto 2023**

**GRUPO SAN VALERO**

**SEAS, Estudios Superiores Abiertos**

**ZARAGOZA**

|  |
| --- |
| Propuesta de trabajo |

El siguiente ejercicio consiste en escribir un programa para gestionar una agenda telefónica con los nombres y los teléfonos de los conocidos de una persona. El programa incorporará las funciones para:

* Consultar el teléfono de una persona.
* Añadir el teléfono de una persona nueva.
* Eliminar el teléfono de una persona.

La agenda debe estar guardada en el fichero de texto agenda.txt donde el nombre de las personas y su teléfono deben aparecer separados por comas y cada persona debe aparecer en una línea distinta.

Se deberán implementar las siguientes funciones:

* obten\_telefono(fichero, persona)

# esta función devuelve el teléfono de una persona.

* inserta\_telefono(fichero, persona, teléfono)

# esta función inserta el teléfono de una persona. Dará un mensaje de error si el fichero no existe

* elimina\_telefono(fichero, persona)

# esta función elimina el teléfono de una persona.

* menu()

# presenta un menú con las opciones disponibles para gestionar la agenda.

Como se acaba de ver, el programa estará conducido por un menú, que será el siguiente:

\_\_\_\_\_\_\_ MENÚ \_\_\_\_\_\_

1. Obtener un teléfono.
2. Insertar un teléfono.
3. Eliminar un teléfono.
4. Salir

NOTA: Todas las funciones devolverán un mensaje de error si no se han podido llevar a cabo correctamente.

|  |
| --- |
| Objetivos del trabajo |

Con el presente trabajo se pretende:

* Que el alumno desarrolle un programa gestionado por menús en el que se trabajen muchos de los conceptos estudiados en el curso.

|  |
| --- |
| Bibliografía |

* Manual de la asignatura.
* Ejercicios realizados a lo largo del curso.

|  |
| --- |
| Criterios de evaluación |

La evaluación, es una componente fundamental de la formación. Este trabajo obligatorio formará parte de tu calificación final. En esta tabla, se resumen los aspectos a valorar y el porcentaje que representa cada uno de los mismos.

|  |  |
| --- | --- |
| CONTENIDOS GENERALES | **10%** |
| Explicación general del programa, claridad de la programación, limpieza del código y presentación.  Claridad en los conceptos. Desarrollo modular de forma que sea aprovechable para otras aplicaciones. | 10% |
| TEMAS DE ESPECIALIDAD | **90%** |
| Explicación correcta del desarrollo de las funciones que intervienen, así como de la programación del menú. | 20% |
| Desarrollos de las funciones que intervienen en el programa. | 50% |
| Prueba de funcionamiento en todos los casos posibles que se puedan presentar, incluyendo aquellos que implican fallos en la introducción de los datos. | 20% |

|  |
| --- |
| Fecha límite de recepción de trabajos |

La fecha de entrega deberá ser anterior a la fecha fin de matrícula.

|  |
| --- |
| Ficha de Corrección del Trabajo |

(Espacio reservado para anotaciones del profesor y doble corrector).

|  |  |
| --- | --- |
| Profesor: Enrique Hernández Hernández | |
| Alumno (Código / Nombre): César Luis Longart Torres | |
| Fecha de Entrega: 10/08/2023 | Fecha de Calificación: |

|  |
| --- |
| Observaciones sobre el trabajo |

Este espacio está reservado para que el profesor titular describa anotaciones que considera importantes sobre la realización del trabajo

También está destinado para que el profesor que efectúa la doble corrección pueda realizar sus anotaciones, asimismo se podrán describir las conclusiones a las que se ha llegado tras realizar la doble corrección.

|  |
| --- |
| Formato de presentación |

1. Se presentará el código en el formato .py de los programas de Python. Las explicaciones se presentarán en formato informático.
2. Se valorará el realizar otras versiones del programa para trabajar con otros tipos de archivos, además del archivo .txt del enunciado.

|  |
| --- |
| Desarrollo del trabajo |

Se inicia con un archivo **Main.py** en el cual se importan dos módulos. El primer módulo Agenda\_Clase.py contiene una clase con nombre Agenda, esta clase contiene tres métodos, posteriormente se explicarán. El segundo módulo Funciones\_utilitarias.py contiene tres funciones, posteriormente se explicarán.

El código del archivo Main.py continúa declarando una variable con el nombre del fichero con el que vamos a trabajar “agenda.txt” y seguimos con otra variable llamada agenda, donde guardamos el objeto de la instancia de la clase Agenda\_Clase.Agenda().

Seguimos con una función llamada insertar\_telefeno(), función que se usa para buscar un número telefónico dentro de la variable fichero. Valida los caracteres ingresados por el usuario a través de la función utilitaria is\_valid\_name. Llama al método obten\_telefono de la clase Agenda validando si el nombre ingresado por el usuario existe. Una vez validado imprime en pantalla el número de teléfono, sino existe muestra un mensaje al usuario. Dentro de esta función se pide al usuario ingresar el nombre que desea busar y se guarda en la variable persona y se utiliza la función upper para convertir las letras en mayúsculas y la función strip para eliminar los espacios en blanco innecesarios. Se ejecuta un condicional con la función utilitaria is\_valid\_name con la variable persona, si la respuesta es True se ejecuta el objeto agenda que llama al método obtén\_telefono que se explicará posteriormente y finalmente se sale del bucle con un break.

A continuación la función inserta\_telefono(), función que se usa para guardar en el archivo agenda.txt el nombre y número telefónico de un nuevo contacto ingresado por el usuario. Se le solicita al usuario ingresar nombre y número, se validan con las funciones utilitarias is\_valid\_name y is\_valid\_phone respectivamente. Una vez validados se llama al método inserta\_telefono de la clase Agenda la cual procede a guardar las variables persona y número en el fichero. Dentro de esta función se encuentra un bucle while que nos permite solicitar nuevamente al usuario los datos sino han sido validados por las funciones utilitarias, al ser validados se ejecuta el objeto agenda llamando al método insertar\_telefono, al final se imprime en pantalla un mensaje confirmando que se han guardado con éxito los datos y finalmente se sale del bucle con un break.

Ahora tenemos la función elimina\_telefono(), Función que se usa para eliminar un registro introducido por el usuario. Se solicita al usuario ingresar el nombre y se valida con la función utilitaria is\_valid\_name. Una vez validado se llama al método de la clase Ageda la cual procede a eliminar el registro indicado del fichero. Dentro de esta función se encuentra un bucle while que nos permite solicitar nuevamente al usuario el nombre sino se ha podido validar por la función utilitaria is\_valid\_name. Una vez validado se ejecuta el objeto agenda que llama al método elimina\_telefono y finalmente se sale del bucle con un break.

La función salir(), simplemente se ejecuta para salir del programa, imprime en pantalla un aviso al usuario y ejecuta la función exit().

La función default(), se ejecuta en caso de que el usuario en el menú ingrese una opción no valida e imprime en pantalla un aviso al usuario.

cases es un diccionario que guarda las opciones disponibles en el menú, teniendo como clave los caracteres del 1 al 4 y como valor cada una de las respectivas opciones que llamarían a las funciones de este archivo anteriormente explicadas.

La función menu(), se encarga de imprimir en pantalla el menú y pedir al usuario una opción. Guarda en la variable op la opción ingresada por el usuario y se ejecuta la función get al diccionario cases, la cual llama a las opciones dentro del diccionario o sino a la función default().

Y para finalizar se ejecuta la función menu().

Continuamos con el módulo **Agenda\_Clase.py** donde se define una clase llamada Agenda que contiene tres métodos.

El primer método se llama obtén\_telefono, se reciben dos parámetos el primero es el fichero en el cual se leen los datos. El segundo parámetro es persona que es el nombre ingresado por el usuario. Una vez recibidos ambos parámetros se abre el fichero en modo lectura y se itera en cada línea del mismo. Se ejecuta un condicional que valida si el parámetro persona se encuentra en la línea que está iterando, si se valida se imprime la línea sino se encuentra se imprime un mensaje que dice que el registro no existe.

El segundo método es insertar\_telefono(), se reciben tres parámetros, el primero es el fichero, el cual se abre en modo anexado para que los nuevos datos no sobrescriban los datos existentes. El segundo parámetro es persona que es el nombre ingresado por el usuario. El tercero es el teléfono que corresponde al número de teléfono ingresado por el usuario. Con los tres parámetros recibidos se ejecuta la función write al archivo (fichero) y se le indica agregar en una misma línea separados por una coma y un espacio el parámetro persona y telefono.

El método elimina\_telefono recibe dos parámetros, el primero es fichero, el cual se abre en modo lectura. El segundo es persona que es el nombre ingresado por el usuario que desea eliminar. Se ejecuta la funcion readlines() para leer todo el fichero y se guarda en una variable llamada líneas. Se define la variable encontrado con el valor False. Se itera con la función enumerate así obtenemos la posición (número de línea) y el valor de la línea (nombre y teléfono). Se ejecuta el condicional verificando si el parámetro persona se encuentra en la línea iterada, si es así encontrado es True y se sale del ciclo for. Sino es encontrado se imprime en pantalla que no existe el dato en el archivo. A continuación, si encontrado es True se elimina la posición en la que se detuvo el ciclo for. Se abre nuevamente el fichero en modo escritura, se sobrescribe con el contenido de la variable lineas y se imprime en pantalla que se ha eliminado con éxito.

Para finalizar tenemos el módulo **Funciones\_Utilitarias.py** que se compone de tres funciones. La primera es is\_ony\_digits(), que recibe un parámetro. Ejecuta un bucle for en cada uno de sus caracteres y valida que cada uno este en el rango de la tabla ascii correspondiente a los dígitos numéricos, devuelve un true o False.

La segunda función es is\_valid\_phobe(), recibe un parámetro que verifica que contenga la longitud de 9 caracteres, a continuación valida que el carácter en la posición cero corresponda a un 9 o un 6, si es verdadero llama a la función is\_only\_digits() para validar que todos los caracteres sean numéricos. En caso contrario imprime un mensaje correspondiente al error y retorna un False.

La tercera y última función es is\_valid\_name(), recibe un parámetro, que valida la longitud de la cadena que debe ser mayor a un carácter. De ser verdadero inicia un contador en cero, ejecuta un bucle for iterando en el parámetro y convirtiendo cada uno de los caracteres en minúsculas y lo guardada en la variable aux. A continuación, ejecuta una condición para validar que cada carácter guardado en aux corresponda a los caracteres de la tabla ascii que involucran al abecedario en minúsculas y a los caracteres ñ y el espacio. Se incrementa el contador y se valida que tenga la misma longitud que el parámetro. En caso de errores se imprime un mensaje correspondiente y se retorna False.