Agenda v2

Christian Jhovany Flores Lozano

Departamento. DIVTIC

I. introducción.

En este reporte explicare lo que se solicitó de actividad número 5, la cual se trata de una agenda mejorada, tomada en base a la agenda simple de la actividad 4 con la inclusión de un archivo de respaldo, es decir, este tendrá la información anterior a la agenda, todo lo anterior en un menú iterativo.

II. Desarrollo de contenido.

Para comenzar en esta actividad fueron necesario más palabras reservadas, ya que con estas se logró la implementación deseada.

[1] '*La función "strip ()" retorna una copia de una cadena con ciertos caracteres eliminados de su principio y final. En esencia, la función strip compara los caracteres de inicio y fin con un conjunto de caracteres definidos por el usuario y los elimina hasta llegar a un carácter que no coincida.*'

[2] La función Split *'Para convertir una cadena en una lista, utilizando un separador.'*

III. Muestra y explicación.

Ahora que comprendemos estas dos funciones explicare los bloques del programa. Iniciare por mostrar el primer bloque (las imágenes se muestran más adelante).

1. *En el Fragmento de código 1* con él os.path.exists y la ruta comprobamos que este existe al existir se abrirá el archivo de original (agenda) en método de lectura y el de respaldo en escritura para sobrescribir la agenda en el respaldo, de no existir se crearan los archivos de agenda y de respaldo en modo escritura, esto es así para que se crean. Al final se cierran los dos archivos independientemente de caso.

Lo siguiente a esto es el menú, el cual consta de la opción de ingresar alumno, buscar y salir.

2. En el *fragmento de código 2* podemos observar el menú anteriormente explicado y anterior a este, métodos de apertura de archivos, en este caso en mi programa es más practico abrir la agenda en método de apertura append.

Ahora si iniciare con la explicación principal del programa.

3. En el *fragmento de código 3*. Lee el archivo de respaldo y al final solamente añade el ultimo registro con el write gracias a que está en modo append.

El insertar el es mas fácil, ahora el siguiente es el buscar al alumno y realizar distintas acciones con esto.

4. En el *fragmento de código 4*, lo primero que hará es solicitar nombre o código del alumno a buscar, en el ciclo for lo recorrerá hasta que lo encuentre en una línea, al ser encontrado se activará la bandera de encontrado y recuperará la información de la línea con ayuda de las funciones sobrecargadas explicadas en el desarrollo de contenido. Dependiendo de lo que encontró es la forma en que imprimirá la información.

Ahora que sabemos si existe o no entrara a una condición, una de estas condiciones es.

5. No existe, como se muestra en el *fragmento de código 5*. En este caso existen dos opciones, ingresarlo a la agenda o no hacer nada. Es lo mismo que al ingresar.

6. Si existe, como se muestra en el *fragmento de código 6*, observamos que existen 3 casos.
    1. Editar. El editar consta en remplazar el dato que deseamos, dependiendo de cada caso se creara una cadena con la nueva edición y los datos recuperados en el *fragmento de código 4*. Al final solamente se vuelve a leer y guardar con el respaldo y el original hasta llegar a la línea editada remplazándola por la nueva cadena.
    2. Eliminar. Este solamente sobrescribe los registros hasta llegar al deseado para eliminar y se lo saltan, como se muestra en el *fragmento de código 7*.
    3. No hacer nada. Este solamente regresa al menú principal.

Ahora que explique el código, hare unas pruebas de 0.

Al ser la primera ejecución se mostró lo siguiente como se puede ver en *impresión en pantalla 1*. Después ingresare varios alumnos para realizar lo siguiente, así como se muestra en *impresión de pantalla 2*.

Ahora que tenemos esto eliminare uno de ellos, así como se muestra en la *impresión de pantalla 3*.

Lo siguiente es editar un alumno como se muestra en *impresión de pantalla 4*.

Después buscar un usuario que no existe e ingresarlo a la agenda como se muestra en la *impresión de pantalla 5*.

Por último, más adelante se mostrará como quedaron los archivos.

IV. Conclusión.

En esta práctica comprendí un poco mas el manejo de archivos, así como sacarle un mejor provecho y la importancia de un archivo de respaldo, que utilice mucho en la ejecución del programa.

V. *Capturas*.

```
if os.path.exists(r"Agenda.txt"):
    archivo_original = open("Agenda.txt",'r')
    with open("Respaldo.txt", 'w') as archivo_respaldo:
        cadena = archivo_original.read()
        archivo_respaldo.write(cadena)
else:
    archivo_original = open("Agenda.txt",'w')
    archivo_respaldo = open("Respaldo.txt",'w')
    print("Se creo la Agenda")
archivo_original.close()
archivo_respaldo.close()

archivo_original = open("Agenda.txt", 'a')
archivo_respaldo = open("Respaldo.txt", 'r')
```

Fragmento de código 1

```
archivo_original = open("Agenda.txt", 'a')
archivo_respaldo = open("Respaldo.txt", 'r')

print("AGENDA V2")
print("1)Ingresar alumno")
print("2)Buscar alumno")
print("0)Salir")
opcion = int(input("Ingresar opcion < "))
```

Fragmento de código 2

```
if opcion == 1:
    nombre = input("Ingresa el nombre: ")
    codigo = input("Ingrese el codigo: ")
    carrera = input("Ingresa la carrera: ")
    cadena = archivo_respaldo.read()
    archivo_original.write("Nombre: " + nombre + " |" +
                           "Codigo: " + codigo + " |" +
                           "Carrera: " + carrera + "\n")
```

Fragmento de código 3

```
elif opcion == 2:
    alumno = input("Ingresa el nombre o codigo del alumno a buscar: ")
    encontrado = False
    for linea in archivo_respaldo:
        if alumno in linea:
            print("Alumno encontrado:")
            nombre, codigo, carrera = [elemento.strip().split(': ')[1]
                                       for elemento in linea.split('|')]
            if alumno == nombre:
                print("Codigo: " + codigo + " |" + "Carrera: " + carrera)
            elif alumno == codigo:
                print("Nombre: " + nombre + " |" + "Carrera: " + carrera)
            encontrado = True
```

Fragmento de código 4

```
if not encontrado:
    print(f"No se encontró al alumno con el nombre o codigo {alumno}")
    print("Desea añadirlo?")
    opcion = int(input('1)Si 2)No : '))

    if opcion == 1:
        nombre = input("Ingresa el nombre: ")
        codigo = input("Ingrese el codigo: ")
        carrera = input("Ingresa la carrera: ")
        cadena = archivo_respaldo.read()
        archivo_original.write("Nombre: " + nombre + " |" + "Codigo: " + codigo
                               + " |" + "Carrera: " + carrera + "\n")
```

Fragmento de código 5

```
else:
    print("Que desea hacer")
    print("1)Editar 2)Eliminar 3)Nada")
    opcion = int(input("Ingrese la opcion : "))

    if opcion == 1:
        print("Que desea editar")
        print("1)Nombre 2)Codigo 3)Carrera")
        opcion = int(input("Ingrese la opcion : "))

        if opcion == 1:
            nuevo_nombre = input("Ingresa el nuevo nombre: ")
            cadena = ("Nombre: " + nuevo_nombre + " |" + "Codigo: "
                      + codigo + " |" + "Carrera: " + carrera + "\n")

        elif opcion == 2:
            nuevo_codigo = input("Ingresa el nuevo codigo: ")
            cadena = ("Nombre: " + nombre + " |" + "Codigo: "
                      + nuevo_codigo + " |" + "Carrera: " + carrera + "\n")

        elif opcion == 3:
            nueva_carrera = input("Ingrese la nueva carrera: ")
            cadena = ("Nombre: " + nombre + " |" + "Codigo: " +
                      codigo + " |" + "Carrera: " + nueva_carrera + "\n")

        with open("Agenda.txt", 'r') as archivo_temp:
            lineas = archivo_temp.readlines()

        with open("Agenda.txt", 'w') as archivo_temp:
            for linea in lineas:
                if alumno not in linea:
                    archivo_temp.write(linea)
                else:
                    archivo_temp.write(cadena)
```

Fragmento de código 6

```
elif opcion == 2:
    with open("Agenda.txt", 'r') as archivo_temp:
        lineas = archivo_temp.readlines()

    with open("Agenda.txt", 'w') as archivo_temp:
        for linea in lineas:
            if alumno not in linea:
                archivo_temp.write(linea)
```

Fragmento de código 7

```
Se creo la Agenda
AGENDA V2
1)Ingresar alumno
2)Buscar alumno
0)Salir
Ingresar opcion <
```

Impresión de pantalla 1

```
AGENDA V2
1)Ingresar alumno
2)Buscar alumno
0)Salir
Ingresar opcion < 1
Ingresa el nombre: Christian Flores
Ingrese el codigo: 23421
Ingresa la carrera: INNI
```

Impresión de pantalla 2

```
AGENDA V2
1)Ingresar alumno
2)Buscar alumno
0)Salir
Ingresar opcion < 2
Ingresa el nombre o codigo del alumno a buscar: 35753
Alumno encontrado:
Nombre: Pedro Martinez |Carrera: INFO
Que desea hacer
1)Editar 2)Eliminar 3)Nada
Ingrese la opcion : 2
```

Impresión de pantalla 3

```
AGENDA V2
1)Ingresar alumno
2)Buscar alumno
0)Salir
Ingresar opcion < 2
Ingresa el nombre o codigo del alumno a buscar: Pablo Hernandez
No se encontró al alumno con el nombre o codigo Pablo Hernandez
Desea añadirlo?
1)Si 2)No : 1
Ingresa el nombre: Pablo Hernandez
Ingrese el codigo: 39342
Ingresa la carrera: INNI
```

Impresión de pantalla 4

```
AGENDA V2
1)Ingresar alumno
2)Buscar alumno
0)Salir
Ingresar opcion < 2
Ingresa el nombre o codigo del alumno a buscar: Miguel Candelas
Alumno encontrado:
Codigo: 23454 |Carrera: QFP
Que desea hacer
1)Editar 2)Eliminar 3)Nada
Ingrese la opcion : 1
Que desea editar
1)Nombre 2)Codigo 3)Carrera
Ingrese la opcion : 3
Ingrese la nueva carrera: QFB
```

Impresión de pantalla 5

main.py | Instrucciones.txt | *Agenda.txt* ×

Agenda.txt

```
1  Nombre: Christian Flores |Codigo: 23421 |Carrera: INNI
2  Nombre: Daniel Garcia |Codigo: 89652 |Carrera: INCO
3  Nombre: Yael Padilla |Codigo: 56743 |Carrera: QFB
4  Nombre: Miguel Candelas |Codigo: 23454 |Carrera: QFB
5  Nombre: Ronaldo Macias |Codigo: 46472 |Carrera: INCO
6  Nombre: Pablo Hernandez |Codigo: 39342 |Carrera: INNI
7
```

main.py | Instrucciones.txt | *Respaldo.txt* ×

Respaldo.txt

```
1  Nombre: Christian Flores |Codigo: 23421 |Carrera: INNI
2  Nombre: Daniel Garcia |Codigo: 89652 |Carrera: INCO
3  Nombre: Yael Padilla |Codigo: 56743 |Carrera: QFB
4  Nombre: Miguel Candelas |Codigo: 23454 |Carrera: QFP
5  Nombre: Ronaldo Macias |Codigo: 46472 |Carrera: INCO
6
```

(Para que se actualice el respaldo se debe iniciar el programa otra vez)

Referencias

[1] La función Strip de Python. (2017, November 20). Techlandia. https://techlandia.com/funcion-strip-python-info_254266/

[2] Parzibyte. (2018, December 10). Usos y ejemplos de split en Python para separar cadenas. Parzibyte's Blog. https://parzibyte.me/blog/2018/12/10/split-python-separar-cadenas/