

**ALGORITMOS Y ESTRUCTURAS DE DATOS**

|  |  |
| --- | --- |
|  | |
| ASIGNATURA: | ALGORITMOS Y ESTRUCTURAS DE DATOS |
| PROFESOR:  FECHA: | Ing. Lorena Chulde  27 – 06 - 2025 |
| PERÍODO ACADÉMICO: | 2025\_A |
|  | |

**TALLER**

|  |
| --- |
|  |
| **TÍTULO**  **ARCHIVOS** |
|  |
|  |
|  |

**Nombre del estudiante:**

XXXXXXXXXXXXXXXXX

**PROPÓSITO DE LA TAREA**

Lograr persistencia de la información generada con listas y diccionarios mediante archivos.

**PARTE 1: ARCHIVOS**

#### **Nivel Básico**

**1. Crear un archivo de texto y escribir varias líneas**

Escribe 3 líneas en un archivo llamado saludo.txt.

|  |
| --- |
|  |
|  |

**2. Leer un archivo línea por línea**

Lee el archivo saludo.txt e imprime cada línea.

|  |
| --- |
|  |
|  |

**3. Crear un archivo solo si no existe (modo 'x')**

Intenta crear registro.txt, si ya existe, muestra un mensaje de error.

|  |
| --- |
| try:      with open("registro.txt","x") as archivo:          archivo.write("Creando solo un archivo con el modo x")  except FileExistsError:      print("Error. El archivo ya existe") |
|  |

**4. Contar cuántas líneas tiene un archivo**

Cuenta las líneas del archivo saludo.txt.

|  |
| --- |
| with open("Archivos/saludo.txt", "r") as archivo:      lineas = archivo.readlines()      print("Numero de lineas: ",len(lineas)) |
|  |

**5. Copiar el contenido de un archivo a otro**

|  |
| --- |
| with open("Archivos/saludo.txt", "r") as archivoOriginal, open("Archivos/copiaSaludo.txt", "w") as archivoCopia:      archivoCopia.write(archivoOriginal.read()) |
|  |

#### **Nivel Intermedio**

**6. Agregar texto a un archivo existente**

Agrega una línea nueva a saludo.txt sin borrar lo anterior.

|  |
| --- |
| with open("Archivos/saludo.txt", "a") as archivo:      archivo.write("\nSeguimos aprendiendo archivos.")      archivo.write("Es muy facil.\n") |
|  |

**7. Leer un archivo y contar cuántas veces aparece una palabra**

Cuenta cuántas veces aparece la palabra "Python".

|  |
| --- |
|  |
|  |

**8. Leer un archivo y mostrar solo las líneas que contienen una palabra clave**

Imprime solo las líneas que contengan "Hola".

|  |
| --- |
|  |
|  |

**9. Escribir una lista de nombres en un archivo**

Escribe una lista de nombres en nombres.txt, uno por línea.

|  |
| --- |
|  |
|  |

**10. Leer los nombres desde el archivo y mostrarlos en mayúsculas**

|  |
| --- |
| with open("Archivos/nombres.txt", "r") as archivo:      print(archivo.read().upper()) |
|  |

### **11. Crear y escribir en un archivo**

Crea un archivo llamado datos.txt y escribe 3 líneas.

|  |
| --- |
| with open("Archivos/datos.txt", "w") as archivo:      archivo.write("Hola")      archivo.write("como")      archivo.write("estas?") |
|  |

### **12. Leer e imprimir el contenido del archivo línea por línea**

Lee el archivo datos.txt e imprime cada línea sin saltos de línea dobles.

|  |
| --- |
|  |
|  |

### **13. Contar cuántas líneas tiene un archivo**

Cuenta cuántas líneas tiene el archivo datos.txt.

|  |
| --- |
|  |
|  |

### **14. Agregar una línea a un archivo existente**

Añade la línea "Cuarta línea" al final del archivo datos.txt.

|  |
| --- |
|  |
|  |

### **15. Buscar una palabra clave en el archivo**

Imprime solo las líneas que contienen la palabra "línea".

|  |
| --- |
|  |
|  |

### **16. Copiar el contenido de un archivo a otro**

### Copia todo el contenido de datos.txt a un nuevo archivo llamado copia\_datos.txt.

|  |
| --- |
|  |
|  |

### **17: Crear un archivo con nombre personalizado**

Pídele al usuario un nombre para el archivo y una línea de texto, y guárdala.

|  |
| --- |
|  |
|  |

### **18: Escribir múltiples líneas hasta que el usuario escriba "salir"**

Permite ingresar varias líneas y las guarda en mensajes.txt, hasta que se escriba "salir".

|  |
| --- |
|  |
|  |

### **19: Leer un archivo cuyo nombre lo da el usuario**

Solicita el nombre de un archivo y muestra su contenido línea por línea.

|  |
| --- |
|  |
|  |

### 20 **Guardar una lista de nombres ingresados por el usuario**

Pide al usuario varios nombres y guárdalos en nombres.txt.

|  |
| --- |
|  |
|  |

1. **Crear un Archivo**

Si se abre el archivo se le escribe contenido y se ejecuta nuevamente el script, se borra el contenido porque los sobreescribe.

|  |
| --- |
|  |
|  |

1. **Escribir en un archivo**

|  |
| --- |
|  |
|  |

1. **Escribir en un archivo más lineas**

|  |
| --- |
|  |
|  |

Si quiero sobreescribir , que no quede el texto anterior cambior la “a” por “w”

1. **Sobrescribir en un archivo mas lineas**

|  |
| --- |
|  |
|  |

1. **Seguir escribiendo líneas en un archivo**

|  |
| --- |
|  |
|  |

1. **Abrir una rchivo y leerlo**

|  |
| --- |
|  |
|  |

1. **Abrir una rchivo y leerlo desde otra ruta**

|  |
| --- |
|  |
|  |

1. **Abrir un archivo y leerlo línea por línea con while**

|  |
| --- |
|  |
|  |

1. **Abrir un archivo y leerlo línea por línea con FOR**

|  |
| --- |
|  |
|  |

Este CRUD permite:

* Crear personas
* Leer o listar todas
* Actualizar por índice
* Eliminar por índice
* Guardar en archivo personas.txt

**Algoritmo de ordenamiento Mergersort**

def merge\_sort(lista):

    if len(lista)>1:

        mitad = len(lista)//2

        mitad1 = lista[:mitad]

        mitad2 = lista[mitad:]

        merge\_sort(mitad1)

        merge\_sort(mitad2)

        i = 0

        j = 0

        k = 0

        while i < len(mitad1) and j < len(mitad2):

            if mitad1[i] < mitad2[j]:

                lista[k] = mitad1[i]

                i = i+1

            else:

                lista[k] = mitad2[j]

                j = j+1

            k = k+1

        while i < len(mitad1):

            lista[k] = mitad1[i]

            i = i+1

            k = k+1

        while j < len(mitad2):

            lista[k] = mitad2[j]

            j = j+1

            k = k+1

numeros = [12,4,15,2,7,1,8,4,9,3]

print(numeros)

merge\_sort(numeros)

print(numeros)

**ENTREGABLES:**

Una vez culminada tu tarea, capturar las pantallas de la ejecución del problema con tus datos y súbela en el apartado del aula virtual “S09-Matrices”

Subir los ejercicios al git o al drive y entrega la url de los archivos .cpp o, a su vez, entregue el archivo.

1. Recordar que el nombre del archivo deberá ser: **Deber-NApellido**(de todos los integrantes)

**RECURSOS NECESARIOS**

* Acceso a Internet.
* Imaginación.
* VSC