

## Taller: Gestión de Archivos en Python

Nombre: Silvia Vera

### 1. ¿Qué significa escribir un archivo en programación?

Escribir un archivo en programación implica almacenar datos, como texto, números o estructuras complejas, en un archivo en el sistema de archivos del dispositivo. Este proceso permite guardar información de forma persistente para su uso posterior, como registrar resultados de un programa, guardar configuraciones o exportar datos. Se realiza abriendo un archivo en modo escritura, enviando el contenido y cerrándolo para garantizar que los datos queden guardados.

### 2. ¿Cuál es la diferencia entre los modos de apertura "w" y "a"?

El modo "w" (escritura) abre un archivo para escribir, eliminando cualquier contenido previo si el archivo ya existe, o crea un nuevo archivo si no existe. Por otro lado, el modo "a" (anexar) también permite escribir, pero en lugar de borrar el contenido, agrega los nuevos datos al final del archivo, preservando lo que ya estaba escrito. Ambos crean el archivo si no existe.

### 3. ¿Qué hace el siguiente código? Explica línea por línea.

```
with open("info.txt", "w") as archivo:
```

```
    archivo.write("info común")
```

- La línea `with open("info.txt", "w") as archivo:` abre un archivo llamado "info.txt" en modo escritura ("w"). Si el archivo ya existe, se sobrescribe; si no, se crea. El uso de `with` asegura que el archivo se cierre automáticamente al terminar.

- La línea `archivo.write("info común")` escribe la cadena "info común" en el archivo. No se agrega un salto de línea automáticamente.

El resultado es un archivo "info.txt" con el texto "info común".

### 4. ¿Qué función de Python se usa para leer todo el contenido de un archivo de texto de una sola vez?

La función `read()` se utiliza para leer todo el contenido de un archivo de texto de una sola vez, devolviendo el contenido como una cadena.

### 5. ¿Qué hace el método `strip()` al leer líneas de un archivo? ¿Por qué es útil?

El método ``strip()`` elimina los espacios en blanco, tabulaciones y caracteres de salto de línea (``\n``) al inicio y al final de una cadena. Es útil al leer líneas de un archivo porque elimina caracteres no deseados, como saltos de línea automáticos, dejando solo el contenido relevante. Esto facilita procesar o mostrar los datos de manera limpia.

6. Escribe un código en Python que lea un archivo llamado `datos.txt` línea por línea e imprima solo las líneas que contienen la palabra "clave".

```
with open("datos.txt", "r") as file:
```

```
    for line in file:
```

```
        if "clave" in line.lower():
```

```
            print(line.rstrip())
```

Este código abre `"datos.txt"` en modo lectura, recorre cada línea y, si la palabra "clave" (en minúsculas para evitar problemas de mayúsculas) está presente, imprime la línea sin el salto de línea final.

7. ¿Qué comando usarías en Linux para quitar todos los permisos de un archivo?

El comando en Linux para quitar todos los permisos de un archivo es: ``chmod 000 nombre_archivo``. Esto elimina los permisos de lectura, escritura y ejecución para el propietario, el grupo y otros usuarios.

8. ¿Cómo podrías simular una protección por contraseña antes de permitir leer un archivo en Python?

```
password = input("Introduce la contraseña: ")
```

```
correct_password = "mi_clave_123"
```

```
if password == correct_password:
```

```
    with open("archivo.txt", "r") as file:
```

```
        print(file.read())
```

```
else:
```

```
    print("Acceso denegado: contraseña incorrecta.")
```

Este código solicita una contraseña al usuario, la compara con una contraseña predefinida y, si coincide, lee y muestra el contenido del archivo. De lo contrario, muestra un mensaje de error.

9. Explica qué hace el siguiente código y qué archivo crea.

`with open("numeros.txt", "w") as archivo:`

`for i in range(1, 6):`

`archivo.write(str(i) + "\n")`

- La línea `with open("numeros.txt", "w") as archivo:` abre (o crea) un archivo llamado "numeros.txt" en modo escritura, sobrescribiendo cualquier contenido previo.

- El bucle `for i in range(1, 6):` itera sobre los números del 1 al 5.

- La línea `archivo.write(str(i) + "\n")` convierte cada número a cadena, agrega un salto de línea y lo escribe en el archivo.

El resultado es un archivo "numeros.txt" con los números 1, 2, 3, 4, 5, cada uno en una línea separada.

10. ¿Por qué es importante proteger un archivo con permisos o contraseñas? Da un ejemplo real.

Proteger un archivo con permisos o contraseñas es crucial para evitar accesos no autorizados, proteger datos sensibles y garantizar la privacidad o seguridad. Sin protección, los datos pueden ser leídos, modificados o robados por personas no autorizadas. Un ejemplo real es un archivo que contiene información médica de pacientes en un hospital. Si este archivo no está protegido con permisos estrictos o cifrado, un atacante podría acceder a datos confidenciales, como diagnósticos o tratamientos, lo que podría derivar en violaciones de privacidad, chantaje o incluso fraudes médicos.