



Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Manejo e Implementación de Archivos

Vacaciones junio 2023

**Catedrático:** Ing. Marlon Orellana

**Tutor académico:** Aarón Juárez

# Proyecto 1

(Unidad 1 y 2)

## Introducción:

El curso de Manejo e Implementación de Archivos busca que los estudiantes aprendan los conceptos sobre la administración de archivos, tanto en hardware como software, sistemas de archivos, particiones, entre otros conceptos, así mismo trata que los estudiantes apliquen estos conceptos en el desarrollo de un proyecto para que así de esta manera puedan aprender cada uno de los temas impartidos durante la clase magistral y el laboratorio para que luego se le pueda dar paso a los conocimientos que se impartirán en cursos posteriores como lo son las bases de datos.

## Objetivos:

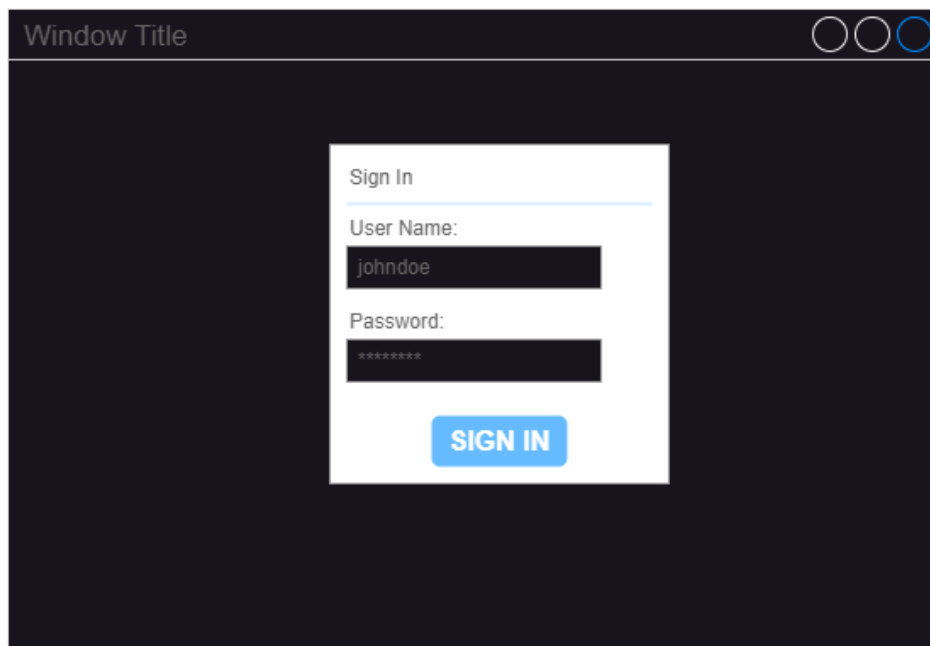
- Aprender a administrar archivos en Python
- Creación de carpetas y archivos en la nube
- Creación de carpetas y archivos de manera local
- Encriptación y desencriptación de archivos de texto
- Manejo de bitácoras.
- Crear una aplicación de comandos

## Aplicación de comandos e interfaz grafica

La aplicación debe contener interfaz gráfica, además de ser capaz de correr series de comandos, No distinguirá entre mayúsculas y minúsculas. Hay parámetros obligatorios y opcionales. Solo se puede colocar un comando por línea. Se debe crear una carpeta con el nombre "Archivos" tanto local como en la nube, en dicha carpeta se realizarán las operaciones que se detallan más adelante.

Si se utiliza un parámetro que no está especificado en este documento, debe mostrar un mensaje de error y guardar el mensaje en bitácora. Se utilizarán espacios en blanco para separar cada parámetro. Si se necesita que algún valor lleve espacios en blanco se encerrará entre comillas ". **Los parámetros pueden venir en cualquier orden**

Login:



La aplicación debe contar con un inicio de sesión, los usuarios y las contraseñas se verificarán a través de un archivo .txt que se proporcionara en la calificación, en el cual las contraseñas vendrán encriptadas en AES utilizando como llave el siguiente texto: miaproyecto12345. La estructura del archivo es la siguiente

```
Usuario1
E42810E54CFFFE9772D2940331350663
Usuario2
06329D9B25E809C660F132B8B85D08D2
```

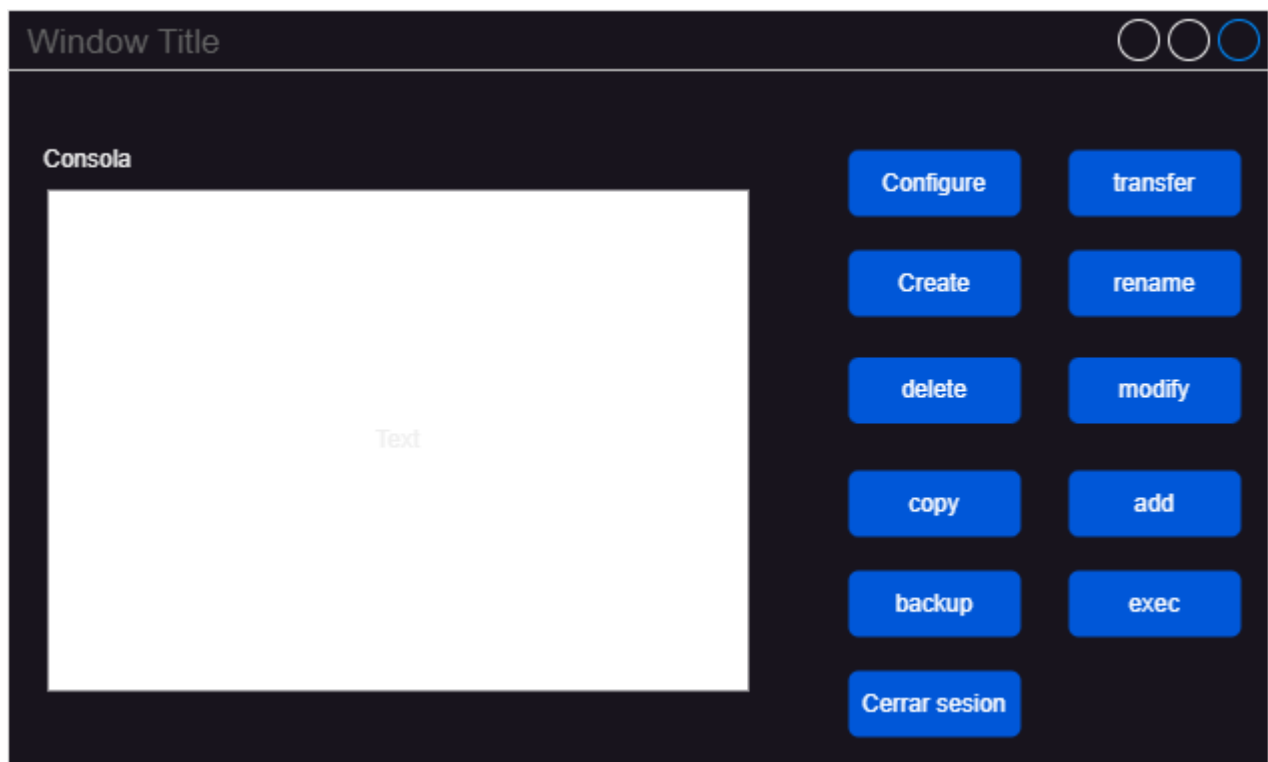
## Ventana Principal:

La aplicación debe contar con una consola para ingresar comandos de manera manual, además de las siguientes opciones

- Todos los botones de la interfaz grafica deben levantar una ventana emergente que soliciten los parámetros que se definen más adelante según la operación. Por ejemplo, el comando configure debe solicitar los parametros type, encrypt\_log, encrypt\_read, llave. Los campos con opciones deben ser representados por un select.
- El botón realizar backup nos permite guardar de manera local (si la configuración inicial es almacenamiento en la nube) o en la nube (si la configuración inicial es almacenamiento local) todo el contenido de la carpeta "Archivos", realizando una unión de los ficheros ya existentes con los del backup en el destino.
- El botón cerrar sesión nos redirigirá a la pantalla de inicio de sesión para ingresar nuevas credenciales.

Todas estas operaciones deben quedar registradas en la bitácora.

Ejemplo Ventana Principal:



Ejemplo opción configure

Window Title

type local

encrypt\_log true

encrypt\_read false

llave

## Bitácora:

Se debe guardar las operaciones realizadas durante el uso de la aplicación, desde el inicio de sesión, hasta las diferentes operaciones con los archivos. Este archivo de logs debe guardarse creando las carpetas del año, mes y día que se utilice, siempre y cuando no exista aun, por ejemplo, si la fecha es el 04/06/2023, la ruta del archivo debe quedar de la siguiente manera /Archivos/logs/2023/06/04/log\_archivos.txt. El archivo debe contener el comando de entrada y sus parámetros, así como el producto de la salida de este, se debe registrar la fecha y la hora en la que se realiza la operación. Queda discreción del estudiante agregar más información.

Ejemplo de salida a bitácora:

```
1 04/06/2023 17:44:20 - Input - Inicio Sesion - Usuario: Usuario 1
2 04/06/2023 17:44:25 - Output - Inicio Sesion - Inicio de sesion Exitoso
3 04/06/2023 17:45:10 - Input - Create - name: pruebal.txt|path: /carpetal/
4 04/06/2023 17:45:15 - Output - Create - Archivo creado exitosamente
5 04/06/2023 17:46:10 - Input - Transfer - from: /carpetal/pruebal.txt | to: /carpeta 2/
6 04/06/2023 17:46:15 - Output - Transfer - Error: la carpeta 2 no existe
```

## Administración de Archivos:

Estos comandos permitirán crear archivos de texto plano, tanto de manera local como en la nube. Estos comandos estarán disponibles desde que se inicia el programa. Estos comandos son:

### 1. Configure

Comando que indica la configuración del sistema de archivos. Debe de incluirse antes de correr cualquier otro comando. Los parámetros por recibir

son los siguientes:

PARÁMETRO	CATEGORÍA	DESCRIPCIÓN
-type	Obligatorio	Indica el tipo de almacenamiento, los valores pueden ser <b>Local</b> o <b>cloud</b> . <ul style="list-style-type: none"><li>• <b>Local:</b> todas las operaciones se realizarán de manera local</li><li>• <b>Cloud:</b> todas las operaciones se realizarán en la nube.</li></ul>
-encrypt_log	Obligatorio	Indica si el contenido del archivo de la bitácora debe de estar encriptado, pueden ser: true o false. El método de encriptado debe ser AES
-encrypt_read	Obligatorio	Este parámetro indica si el archivo a leer es encriptado, pueden ser: true o false. El método de encriptado es AES
-llave	Opcional	Indica la llave para encriptar y desencriptar la bitácora y el archivo de lectura

Ejemplo:

#configuracion de tipo local sin encriptación de bitácora ni archivo a leer encriptado

Configure -type->local -encrypt\_log->>false -encrypt\_read->>false

## 2. Create

Este comando creará un archivo de texto plano, deberá contener la información indicada por el parámetro. Recibirá la ruta en donde se debe crear. Además del nombre con el que se debe guardar

PARÁMETRO	CATEGORÍA	DESCRIPCIÓN
-name	Obligatorio	Nombre con el cual se guardará el archivo de texto plano

<b>-body</b>	Obligatorio	Contenido del archivo del texto plano
<b>-path</b>	Obligatorio	Este parámetro será la ruta en el que se creará el archivo. Si las carpetas de la ruta no existen deberán crearse.

#Crea un archivo llamado prueba1.txt con el contenido Este es el contenido del archivo 1

```
create -name->prueba1.txt -path->/carpeta1/ -body->"Este es el contenido del archivo 1"
```

#Se ponen comillas por el archivo "prueba 2" y para la carpeta "carpeta 2" ya que tiene espacios en blanco, se crea si no existe

```
create -name->"prueba 2.txt" -path->"/carpeta 2/" -body->"Este es el contenido del archivo 2"
```

### 3. delete

Este parámetro elimina un archivo mostrando un mensaje de confirmación para eliminar. Tendrá los siguientes parámetros

PARÁMETRO	CATEGORÍA	DESCRIPCIÓN
<b>-path</b>	Obligatorio	Este parámetro será la ruta en el que se encuentra el archivo. Si la ruta no existe, debe mostrar un mensaje de error en la bitácora.
<b>-name</b>	opcional	Este parámetro será el nombre del archivo a eliminar, si el archivo no existe, debe guardar un mensaje de error en la bitácora. De no venir el parámetro, se debe eliminar toda la carpeta.

#### Ejemplo:

#Elimina prueba1.txt

```
delete -path->/carpeta1/ -name->prueba1.txt
```

#Elimina carpeta 2

```
delete -path->"/carpeta 2/"
```

### 4. copy

Este comando copia un archivo, o bien una carpeta, a otra ubicación indicada a través de los parámetros.

Parámetro	Categoría	Descripción
<b>-from</b>	obligatorio	Ubicación del archivo el cual se debe copiar, de no tener el nombre del archivo se debe copiar toda la carpeta. Si no existe el archivo o carpeta deberá guardar el mensaje de error en la bitácora.
<b>-to</b>	obligatorio	Ubicación de la carpeta a la cual se debe copiar. Si no existe el archivo o carpeta deberá guardar el mensaje de error en la bitácora.

### Ejemplo:

#copia el archivo prueba1.txt a la carpeta 2

Copy -from->/carpeta1/prueba1.txt -to->"/carpeta 2/"

#Copia el contenido de la carpeta 2 a la carpeta1

Copy -from->"/carpeta 2/" -to->/carpeta1/

## 5. Transfer

Este comando transfiere un archivo, o bien una carpeta, a otra ubicación indicada a través de los parámetros.

Parámetro	Categoría	Descripción
<b>-from</b>	obligatorio	Ubicación del archivo el cual se debe transferir, de no tener el nombre del archivo se debe transferir toda la carpeta. Si no existe el archivo o carpeta deberá guardar el mensaje de error en la bitácora.
<b>-to</b>	obligatorio	Ubicación de la carpeta a la cual se debe transferir. Si no existe la carpeta deberá crearse, si ya existe un archivo con el mismo nombre, deberá cambiar el nombre del archivo y guardar ambos.
<b>-mode</b>	obligatorio	Indica el tipo de almacenamiento, los valores pueden ser <b>Local</b> o <b>cloud</b> . <ul style="list-style-type: none"> <li>• <b>Local:</b> todas las operaciones se realizarán de manera local</li> <li>• <b>Cloud:</b> todas las operaciones se realizarán en la nube.</li> </ul>

### Ejemplos:

#transfiere el archivo prueba1.txt a la carpeta 2 de manera local

transfer -from->/carpeta1/prueba1.txt -to->"/carpeta 2/" -mode->"local"

#transfiere el contenido de la carpeta 2 a la carpeta1 en la nube

transfer -from->"/carpeta 2/" -to->"/carpeta1/ -mode->"cloud"

## 6. Rename

Este comando permitirá cambiar el nombre de un archivo o carpeta. Tendrá los siguientes parámetros:

PARÁMETRO	CATEGORÍA	DESCRIPCIÓN
-path	Obligatorio	Este parámetro será la ruta del archivo o carpeta al que se le cambiará el nombre. Si lleva espacios en blanco deberá encerrarse entre comillas. Si no existe el archivo o carpeta deberá guardar el mensaje de error en la bitácora.
-name	obligatorio	Especificará el nuevo nombre del archivo, debe verificar que no exista un archivo con el mismo nombre, de ser así deberá guardar el mensaje de error en la bitácora.

### Ejemplos:

#Cambia el nombre del archivo prueba1.txt a b1.txt

rename -path->"/carpeta1/prueba1.txt -name->b1.txt

#Deberá mostrar error ya que el archivo b1.txt ya existe

rename -path->"/carpeta1/prueba1.txt -name->b1.txt

## 7. Modify

Este comando sustituirá el contenido de un archivo. Tendrá los siguientes parámetros:

PARÁMETRO	CATEGORÍA	DESCRIPCIÓN
-path	Obligatorio	Este parámetro será la ruta del archivo a la que se sustituirá el contenido. Si lleva espacios en blanco deberá encerrarse entre comillas. Si no existe el archivo o carpeta deberá guardar el mensaje de error en la bitácora.
-body	obligatorio	Será el nuevo contenido del archivo.



### Ejemplos:

#Cambia el contenido del archivo prueba1.txt

modify -path->/carpeta1/prueba1.txt -body->" este  
es el nuevo contenido del archivo"

## 8. Add

Este comando añadirá contenido a un archivo. Tendrá los siguientes parámetros:

PARÁMETRO	CATEGORÍA	DESCRIPCIÓN
-path	Obligatorio	Este parámetro será la ruta del archivo a la que se sustituirá el contenido. Si lleva espacios en blanco deberá encerrarse entre comillas. Si no existe el archivo o carpeta deberá guardar el mensaje de error en la bitácora.
-body	obligatorio	Será el nuevo contenido del archivo.

### Ejemplos:

#Se añade contenido al archivo prueba1.txt

modify -path->/carpeta1/prueba1.txt -body->" este  
es el nuevo contenido del archivo"

## 9. backup

Nos permite guardar de manera local (si la configuración inicial es almacenamiento en la nube) o en la nube (si la configuración inicial es almacenamiento local) todo el contenido de la carpeta "Archivos", realizando una unión de los ficheros ya existentes con los del backup en el destino.

### Ejemplos:

#Se realiza un backup de la información

backup

## EXEC

El programa podrá ejecutar scripts con el comando exec. Debe mostrar el contenido de la línea que está leyendo y su resultado. Al finalizar la lectura del archivo debe indicar la cantidad de archivos procesados de manera local y el tiempo de procesamiento, la cantidad de archivos procesados de manera en la nube y el tiempo de procesamiento. Tanto en la consola como en el archivo de bitácora

PARÁMETRO	CATEGORÍA	DESCRIPCIÓN
-path	Obligatorio	Especifica el nombre del script que se va a ejecutar.

**Ejemplo:**

#ejecuta el script

exec -path->/home/Desktop/calificacion.mia

## Entrega

El proyecto se entregará el jueves 15 de junio del 2023 hasta las 23:59. Se utilizará un repositorio de github para que suban su proyecto y se habilitará una opción en UEDI para que puedan subir el link de su repositorio, los auxiliares de cada curso deberán tener acceso a los repositorios respectivos en cualquier momento de la duración del laboratorio, si no se cuenta con acceso se anulara el proyecto, se recomienda que sea un repositorio privado para evitar copias. La impuntualidad anulara el trabajo entregado. Se calificará el último commit que suban a la hora estipulada y se deberá de encontrar dentro del repositorio un ejecutable desde el cual se calificará. De no contar con commits de uno de los estudiantes no se tomará en cuenta dentro del puntaje obtenido. **Está prohibido generar el ejecutable durante la calificación.**

Usuario de github:

1. AaronJuarez300

El proyecto debe realizarse de forma individual, **Se utilizará software para la detección de copias. las copias tendrán una nota de 0 y serán reportadas a la escuela.**

El lenguaje por utilizar es Python. No se permite el uso de otro lenguaje.

El proveedor de la nube queda a su discreción pueden ser como ejemplo: Google Drive, Outlook o Amazon S3.

NO se permite la modificación de código durante la calificación. El estudiante únicamente podrá utilizar el ejecutable entregado.

La hoja de calificación se proporcionará 24hrs antes de la calificación incluyendo un apartado de control de versionamiento en GitHub de los commits de su proyecto.

## Anexos:

### Ejemplo de archivo de entrada:

```
configure -type->local -encrypt_log->>false -encrypt_read->>false
create -name->prueba1.txt -path->/carpetal/ -body->"Este es el contenido del archivo 1"
create -name->prueba2.txt -path->/carpetal/ -body->"Este es el contenido del archivo 2"
backup
```

bitácora del procesamiento del archivo anterior, el comando exec se debe correr para leer el archivo.

```
04/06/2023 17:44:20 - Input - Inicio Sesion - Usuario: Usuario 1
04/06/2023 17:44:25 - Output - Inicio Sesion - Inicio de sesion Exitoso
04/06/2023 17:44:20 - Input - Exec - path: /entradas/calificacion.mia
04/06/2023 17:44:20 - Input - configure - type:local | encrypt_log:false | encrypt_read:false
04/06/2023 17:44:25 - Output - configure - configuracion exitosa
04/06/2023 17:45:10 - Input - Create - name: prueba1.txt|path: /carpetal/
04/06/2023 17:45:15 - Output - Create - Archivo creado exitosamente
04/06/2023 17:45:10 - Input - Create - name: prueba2.txt|path: /carpetal/
04/06/2023 17:45:15 - Output - Create - Archivo creado exitosamente
04/06/2023 17:46:10 - Input - Backup -
04/06/2023 17:46:15 - Output - Backup - Backup exitoso
04/06/2023 17:44:25 - Output - Exec - Archivos procesados localmente: 2 | tiempo de procesamiento: 300ms | archivos procesados en la nube:2 | tiempo de procesamiento: 400ms
```