

Documentación del Script auto.sh para Gestión de Archivos – Actividad 3.2

1. Creación del archivo y edición con Nano

Para crear o editar el script se usa:

```
dguaadir@localhost:~$ nano auto.sh
```

Después de ingresar procederemos a realizar el script, el cual es el siguiente:

```
1 # validar que existen al menos 2 argumentos
2
3 if [ $# -ne 2 ]; then
4     echo "Uso: $0 <archivo> <crear|borrar|ver>"
5     exit 1
6 fi
7
8 ARCHIVO="$1"
9 ACCION="$2"
10
11 echo "Argumento de accion recibido: <$ACCION>"
12
13 case "$ACCION" in
14     crear)
15         if [ -f "$ARCHIVO" ]; then
16             echo "El archivo ya existe."
17         else
18             touch "$ARCHIVO"
19             echo "Archivo creado: $ARCHIVO"
20         fi
21         ;;
22     borrar)
23         if [ -f "$ARCHIVO" ]; then
24             rm "$ARCHIVO"
25             echo "Archivo borrado."
26         else
27             echo "No existe el archivo."
28         fi
29         ;;
30     ver)
31         if [ -f "$ARCHIVO" ]; then
32             echo "Contenido de archivo."
33             cat "$ARCHIVO"
34         else
35             echo "No se puede ver, no existe el archivo."
36         fi
37         ;;
38     *)
39         echo "Acción no valida. Use: crear | borrar | ver."
40         exit 1
41     ;;
42 esac
```

Luego se guarda con CTRL+O, ENTER y se sale con CTRL+X.

Finalmente se otorgan permisos de ejecución con:

✓ `chmod +x auto.sh`

2. Contenido del script

El script recibe dos argumentos: el nombre del archivo y la acción (crear, borrar o ver).

```
1 # validar que existen al menos 2 argumentos
2
3 if [ $# -ne 2 ]; then
4     echo "Uso: $0 <archivo> <crear|borrar|ver>"
5     exit 1
6 fi
7
8 ARCHIVO="$1"
9 ACCION="$2"
10
11 echo "Argumento de accion recibido: <$ACCION>"
12
```

3. Explicación del funcionamiento del script

A. Acción CREAR

```
15 crear)
16     if [ -f "$ARCHIVO" ]; then
17         echo "El archivo ya existe."
18     else
19         touch "$ARCHIVO"
20         echo "Archivo creado: $ARCHIVO"
21     fi
22 ;;
23
```

- ✓ Verifica si el archivo existe.
- ✓ Si no existe, lo crea con touch.
- ✓ Si ya existe, muestra un mensaje indicando que no es necesario crearlo.

C. Acción BORRAR

```
24 borrar)
25     if [ -f "$ARCHIVO" ]; then
26         rm "$ARCHIVO"
27         echo "Archivo borrado."
28     else
29         echo "No existe el archivo."
30     fi
31 ;;
32
```

- ✓ Verifica si el archivo existe.
- ✓ Si existe, lo borra con rm.
- ✓ Si no existe, avisa que no puede borrarse.

B. Acción VER

```
33 ver)
34 if [ -f "$ARCHIVO" ]; then
35     echo "Contenido de archivo."
36     cat "$ARCHIVO"
37 else
38     echo "No se puede ver, no existe el archivo."
39 fi
40 ;;
41
```

- ✓ Verifica si el archivo existe.
- ✓ Si existe, muestra su contenido con cat.
- ✓ Si no existe, indica que no se puede mostrar.

4. Cómo verificar que cada acción funciona

A. Probar CREAR:

- ✓ ./auto.sh prueba.txt crear

```
dguadir@localhost:~$ ./auto.sh prueba.txt crear
Argumento de accion recibido: <crear>
Archivo creado: prueba.txt
```

- ✓ Usar ls para confirmar: ls -l prueba.txt

```
dguadir@localhost:~$ ls -l prueba.txt
-rw-r--r--. 1 dguadir dguadir 0 dic  9 14:03 prueba.txt
```

B. Probar VER:

- ✓ ./auto.sh prueba.txt ver

```
dguadir@localhost:~$ ./auto.sh prueba.txt ver
Argumento de accion recibido: <ver>
Contenido de archivo.
```

C. Probar BORRAR:

- ✓ ./auto.sh prueba.txt borrar

```
dguadir@localhost:~$ ./auto.sh prueba.txt borrar
Argumento de accion recibido: <borrar>
Archivo borrado.
```

5. Verificación de manejo de errores

A. Acción no válida:

- `./auto.sh archivo.txt editar`

Debe mostrar: Acción no válida. Use: crear | borrar | ver.

```
dguadir@localhost:~$ ./auto.sh prueba.txt editar
Argumento de accion recibido: <editar>
Acción no valida. Use: crear | borrar | ver.
```

B. Falta de argumentos:

- `./auto.sh archivo.txt`

Debe mostrar: Uso: `./auto.sh <archivo> <crear|borrar|ver>`

```
dguadir@localhost:~$ ./auto.sh prueba.txt
Uso: ./auto.sh <archivo> <crear|borrar|ver>
```

6. Conclusión

El script `auto.sh` permite crear, borrar y visualizar archivos desde la consola de forma rápida y automatizada, aplicando validaciones y manejo básico de errores. A diferencia de un entorno gráfico, donde estas acciones se realizan mediante clics y ventanas, la consola ofrece mayor eficiencia, control y precisión, siendo ideal para servidores y tareas administrativas. Mientras la interfaz gráfica es más intuitiva, el uso del script en consola fortalece las habilidades de línea de comandos y permite ejecutar operaciones de forma más profesional y automatizada.