

**TECNOLOGICO DE ESTUDIOS SUPERIORES  
DE JILOTEPEC**

**FECHA: 17 DE SEPTIEMBRE DE 2025**

**SISTEMAS OPERATIVOS 1**

**TERCER SEMESTRE**

**PROFESOR: JOSE LUCIO HERNANDEZ NOGUEZ**

**ALUMNO: EDGAR MIRANDA DE LA CRUZ**

**MATRICULA: 202423380**

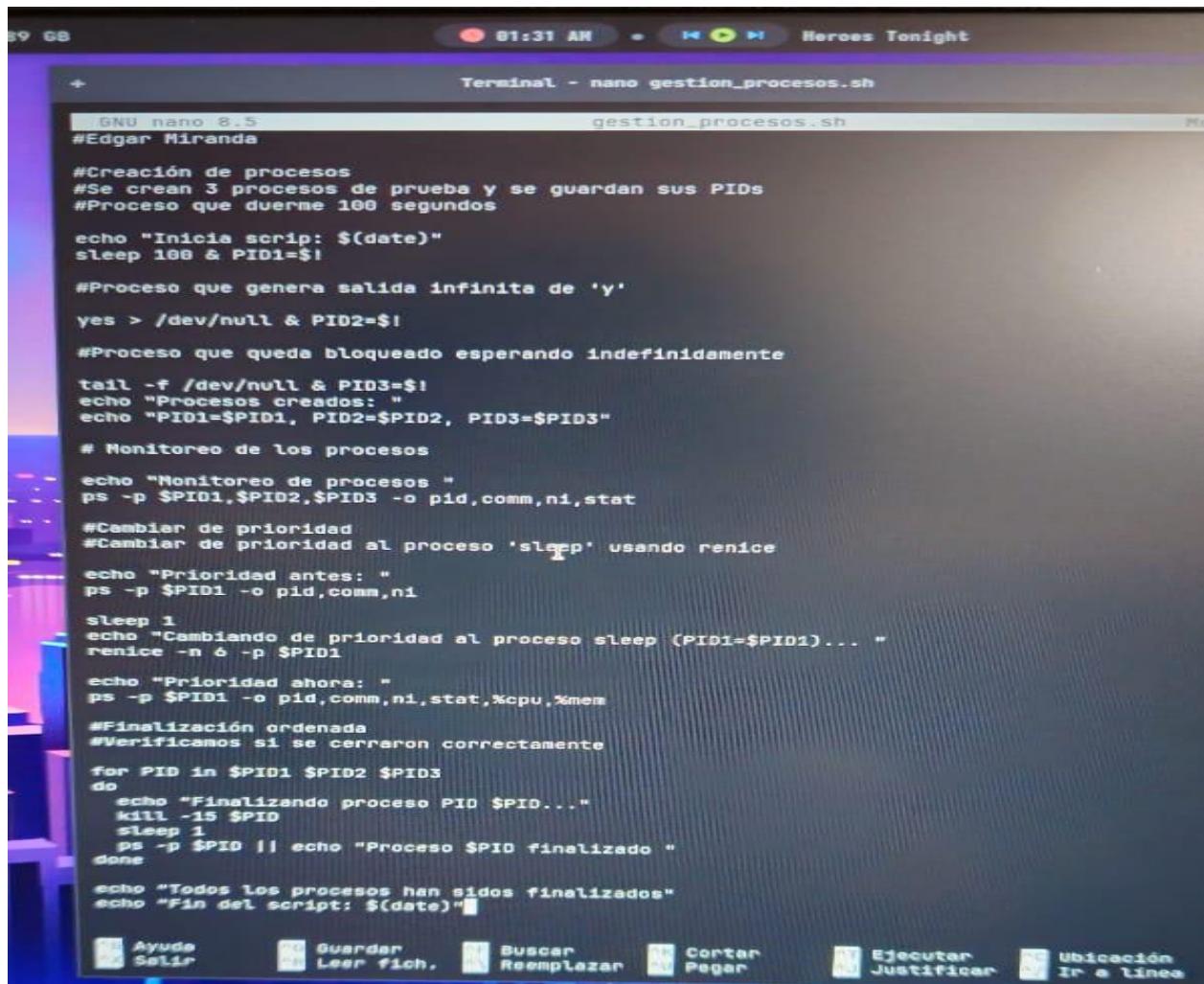
**GRUPO: 3302**

**2025 -2026**

# Introducción

En esta práctica el objetivo de todo esto era que utilizando el sistema operativo de Archcraft hiciéramos uso de la terminal para desarrollar un scrip en bash para monitorear los procesos en el diestro, teniendo como meta a alcanzar el poder modificarlos, cambiar su prioridad y finalizarlos de forma ordenada. Antes de comenzar con lo que hice me gustaría hablar un poco acerca del sistema y sobre lo que es un scrip. Primero Archcraft es una distribución de Linux basada en Arch Linux que utiliza aplicaciones y gestores de ventana ligeros lo que la hace demasiado rápida. Un scrip bash es un archivo de texto con comandos escritos en el lenguaje Shell bash que es la terminal de Linux que se va ejecutando de manera automática, son instrucciones que el sistema va a ejecutando sin necesidad de que tengas que escribirlas una por una, la palabra Bash significa “Bourne Again Shell” el cual es el interprete de Linux más común.

## Código del script comentado



```
89 GB 81:31 AM Heroes Tonight
+ Terminal - nano gestion_procesos.sh
GNU nano 8.5 gestion_procesos.sh
#Edgar Miranda

#Creación de procesos
#Se crean 3 procesos de prueba y se guardan sus PIDs
#Proceso que duerme 100 segundos

echo "Inicia scrip: $(date)"
sleep 100 & PID1=$!

#echo "Proceso que genera salida infinita de 'y'"
#yes > /dev/null & PID2=$!
#Proceso que queda bloqueado esperando indefinidamente
tail -f /dev/null & PID3=$!
echo "Procesos creados: "
echo "$PID1,$PID2,$PID3"

# Monitoreo de los procesos
echo "Monitoreo de procesos"
ps -p $PID1,$PID2,$PID3 -o pid,comm,ni,stat

#Cambiar de prioridad
#Cambiar de prioridad al proceso 'sleep' usando renice
echo "Prioridad antes: "
ps -p $PID1 -o pid,comm,ni

sleep 1
echo "Cambiando de prioridad al proceso sleep (PID1=$PID1)..."
renice -n 6 -p $PID1

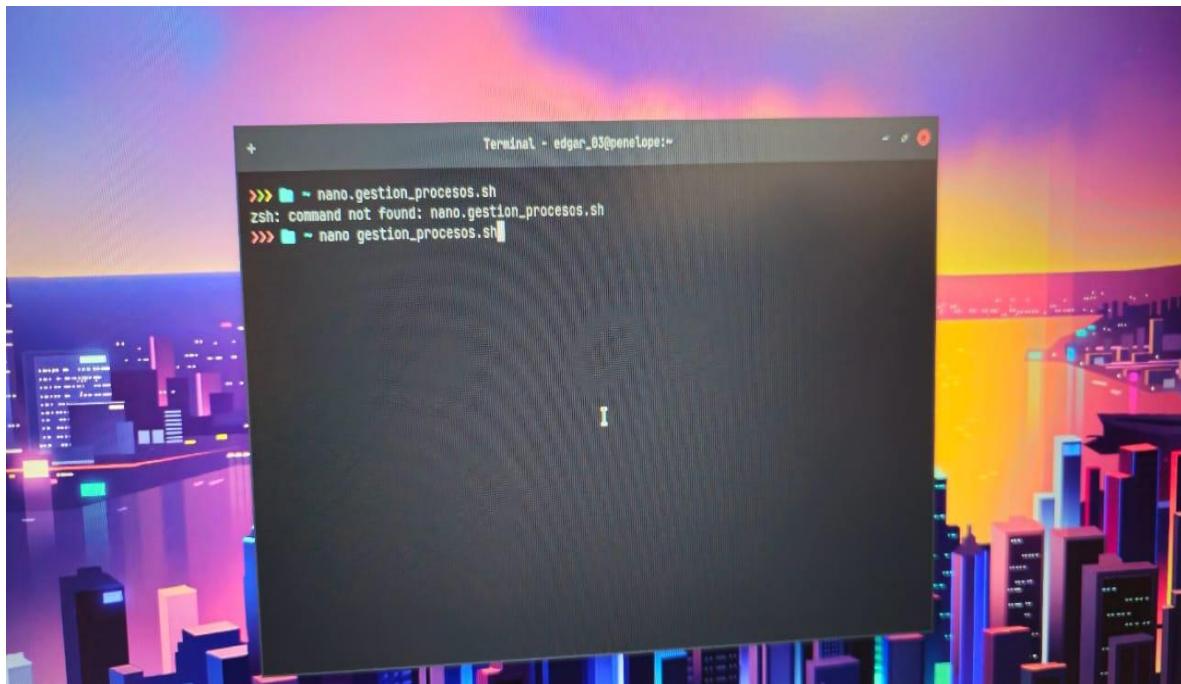
echo "Prioridad ahora: "
ps -p $PID1 -o pid,comm,ni,stat,%cpu,%mem

#Finalización ordenada
#Verificamos si se cerraron correctamente
for PID in $PID1 $PID2 $PID3
do
    echo "Finalizando proceso PID $PID..."
    kill -15 $PID
    sleep 1
    ps -p $PID || echo "Proceso $PID finalizado"
done

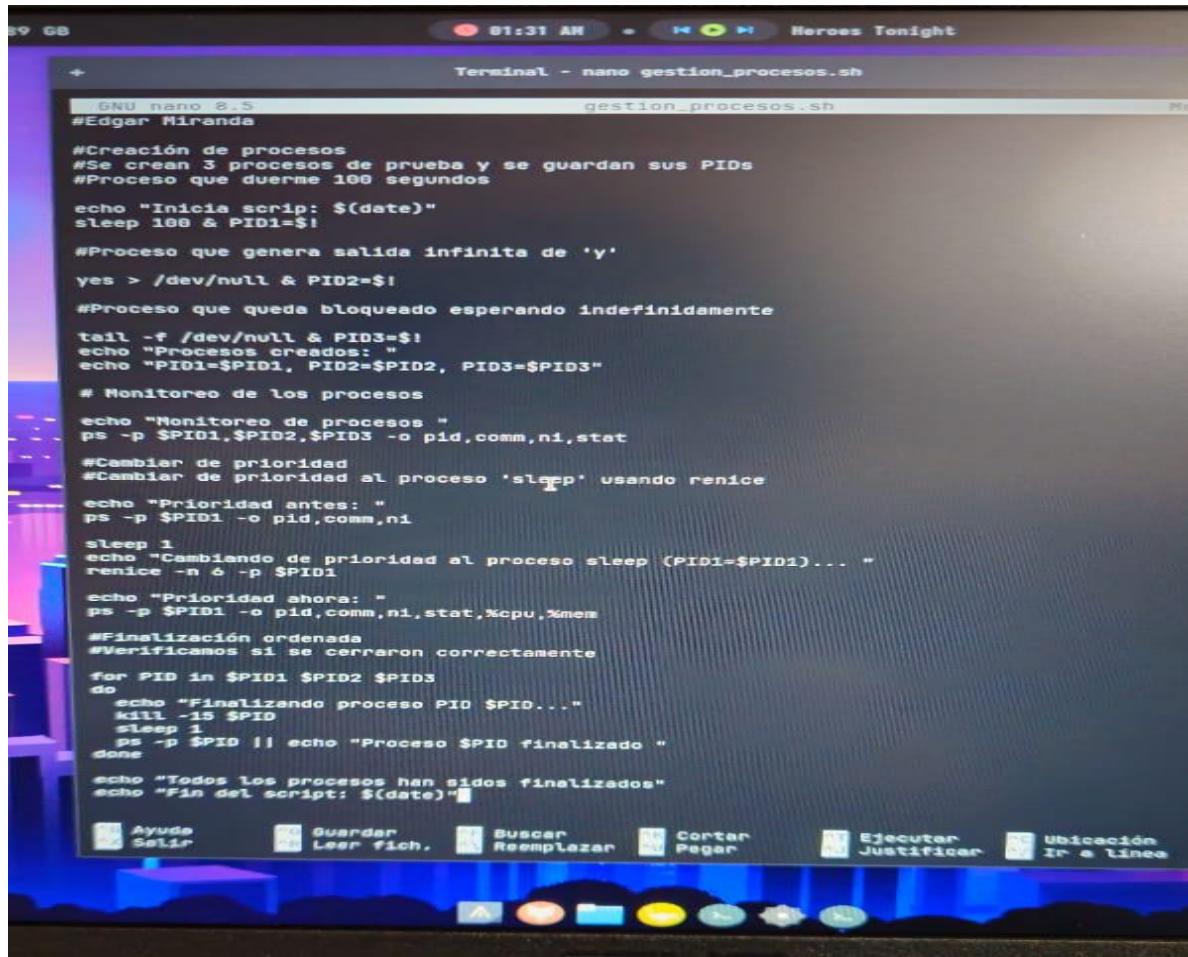
echo "Todos los procesos han sido finalizados"
echo "Fin del script: $(date)"

Ayuda Salir Guardar Leer fich. Buscar Reemplazar Cortar Pegar Ejecutar Justificar Ubicación Ir a línea
```

## Evidencias de la práctica realizada



- Comando que sirve para abrir un archivo, si el archivo aún no existe es creado en mi caso mi archivo tiene el nombre de gestión\_procesos



```

89 08 81:31 AM • Heroes Tonight
+ Terminal - nano gestion_procesos.sh
GNU nano 8.5
#Edgar Miranda

#Creación de procesos
#Se crean 3 procesos de prueba y se guardan sus PIDs
#Proceso que duerme 100 segundos

echo "Inicia script: $(date)"
sleep 100 & PID1=$!

#Proceso que genera salida infinita de 'y'
yes > /dev/null & PID2=$!

#Proceso que queda bloqueado esperando indefinidamente
tail -f /dev/null & PID3=$!
echo "Procesos creados:"
echo "PID1=$PID1, PID2=$PID2, PID3=$PID3"

# Monitoreo de los procesos
echo "Monitoreo de procesos"
ps -p $PID1,$PID2,$PID3 -o pid,comm,ni,stat

#Cambiar de prioridad
#Cambiar de prioridad al proceso 'sleep' usando renice
echo "Prioridad antes:"
ps -p $PID1 -o pid,comm,ni,stat,%cpu,%mem

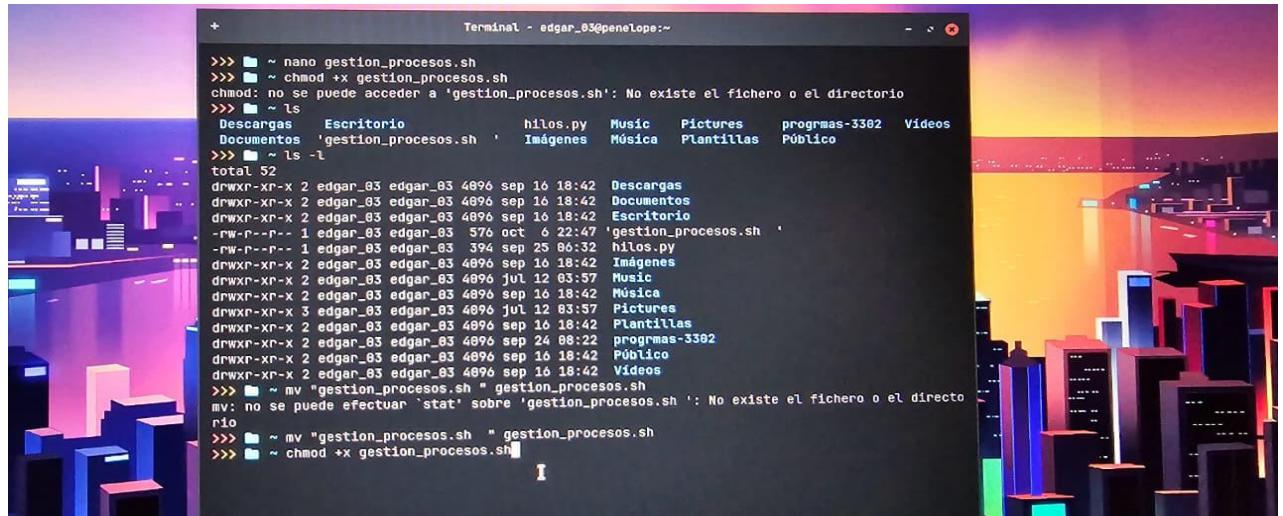
#Finalización ordenada
#Verificamos si se cerraron correctamente

for PID in $PID1 $PID2 $PID3
do
    echo "Finalizando proceso PID $PID..."
    kill -15 $PID
    sleep 1
    ps -p $PID || echo "Proceso $PID finalizado"
done

echo "Todos los procesos han sido finalizados"
echo "Fin del script: $(date)"
```

Ayuda Salir Guardar Leer fich. Buscar Reemplazar Cortar Pegar Ejecutar Justificar Ubicación Ir a línea

- Para guardar el proceso de mi código en el archivo lo que hacía es dar ctrl O y luego presionaba la tecla de enter
- Para salir del archivo lo que hacía era presionar ctrl X



```

>>> ~ nano gestion_procesos.sh
>>> ~ chmod +x gestion_procesos.sh
chmod: no se puede acceder a 'gestion_procesos.sh': No existe el fichero o el directorio
>>> ~ ls
Descargas Escritorio hilos.py Música Pictures programas-3302 Videos
Documentos 'gestion_procesos.sh' Imágenes Música Plantillas Público
>>> ~ ls -l
total 52
drwxr-xn-x 2 edgar_83 edgar_83 4096 sep 16 18:42 Descargas
drwxr-xn-x 2 edgar_83 edgar_83 4096 sep 16 18:42 Documentos
drwxr-xn-x 2 edgar_83 edgar_83 4096 sep 16 18:42 Escritorio
-rw-r--r-- 1 edgar_83 edgar_83 576 oct 6 22:47 'gestion_procesos.sh'
-rw-r--r-- 1 edgar_83 edgar_83 394 sep 25 06:32 hilos.py
drwxr-xn-x 2 edgar_83 edgar_83 4096 sep 16 18:42 Imágenes
drwxr-xn-x 2 edgar_83 edgar_83 4096 jul 12 03:57 Música
drwxr-xn-x 2 edgar_83 edgar_83 4096 sep 16 18:42 Pictures
drwxr-xn-x 3 edgar_83 edgar_83 4096 sep 16 18:42 Plantillas
drwxr-xn-x 2 edgar_83 edgar_83 4096 sep 24 08:22 programas-3392
drwxr-xn-x 2 edgar_83 edgar_83 4096 sep 16 18:42 Público
drwxr-xn-x 2 edgar_83 edgar_83 4096 sep 16 18:42 Videos
>>> ~ mv "gestion_procesos.sh" "gestion_procesos.sh"
mv: no se puede efectuar 'stat' sobre 'gestion_procesos.sh': No existe el fichero o el directorio
>>> ~ mv "gestion_procesos.sh" "gestion_procesos.sh"
>>> ~ chmod +x gestion_procesos.sh
```

- El comando chmod +x nombre\_archivo.sh hace que el archivo sea ejecutable

- Para ejecutar el archivo lo que se hace es ./nombre\_archivo.sh y al momento de presionar la tecla enter el archivo se empieza a ejecutar en la misma terminal.

```

>>> ~ - nano gestion_procesos.sh
>>> ~ - ./gestion_procesos.sh
./gestion_procesos.sh: línea 33: error sintáctico cerca del elemento
./gestion_procesos.sh: línea 33: `echo "Cambiando de prioridad al pr
".
>>> ~ bash -x gestion_procesos.sh
gestion_procesos.sh: línea 33: error sintáctico cerca del elemento i
gestion_procesos.sh: línea 33: `echo "Cambiando de prioridad al proce
>>> ~ nano gestion_procesos.sh
>>> ~ ./gestion_procesos.sh
./gestion_procesos.sh: línea 34: error sintáctico cerca del elemento
./gestion_procesos.sh: línea 34: `echo "Cambiando de prioridad al pro
".
>>> ~ nano gestion_procesos.sh
>>> ~ ./gestion_procesos.sh
Inicia scrip: mié 15 oct 2025 01:31:51 CST
Procesos creados:
PID1=109665, PID2=109666, PID3=109667
Monitoreo de procesos
  PID COMMAND      NI STAT
  109665 sleep      0 S+
  109666 yes        0 R+
  109667 tail       0 S+
Prioridad antes:
  PID COMMAND      NI
  109665 sleep      0
Cambiando de prioridad al proceso sleep (PID1=109665)...
109665 (process ID) prioridad anterior 0, nueva prioridad 6
Prioridad ahora:
  PID COMMAND      NI STAT %CPU %MEM
  109665 sleep      6 SN+  0.0  0.0
Finalizando proceso PID 109665...
  PID TTY      TIME CMD
Proceso 109665 finalizado
Finalizando proceso PID 109666...
  PID TTY      TIME CMD
Proceso 109666 finalizado
Finalizando proceso PID 109667...
  PID TTY      TIME CMD
Proceso 109667 finalizado
Todos los procesos han sido finalizados
Fin del script: mié 15 oct 2025 01:31:55 CST
>>> ~

```

- Al momento de ejecutarlo muestra el cambio de prioridad del proceso sleep con su anterior y nueva prioridad siendo 0 y 6 respectivamente, además también se muestra el consumo de CPU y memoria al momento de hacer el cambio de prioridad
- Todos los procesos son finalizados y para ver que se hizo bien se muestra un mensaje que indica que el proceso ha finalizado

## **Reflexión final**

En esta practica pude entender lo que significa el scrip y pude aprender como monitorear los procesos, tuve algunas dificultades al momento de querer ejecutar el archivo ya que al momento de colocar el comando para correrlo lo que paso fue que me marco un error ya que me indicaba que el archivo no existía cuando claramente si estaba, al momento de consultar la ubicación del archivo me di cuenta que el archivo tenía espacio en el nombre lo que me daba a entender que ese era el motivo por el cual el archivo no lo encontraba la solución que encontré para ello fue renombrar el archivo con el comando mv “nombre\_actual” nombre\_nuevo y con eso pude arreglarlo ya que después de ello al hacerlo ejecutable y al correrlo, el archivo se ejecutó sin ningún problema, las otras cosas que pude aprender durante todo esto es como crear un archivo, para que dentro de él vayan las instrucciones que tiene que hacer la terminal para de esa manera cumplir con el objetivo de esta practica que en pocas palabras es, crearlo, monitorearlo cambiar su prioridad y finalizarlos, también tuve otros problemas ya que me marcaba error en una líneas del código al momento de querer mejorarlo y ahora el código ya no se podía ejecutar, pero vi que el fallo estaba por la ausencia de las comillas en un agregado que hice para que se muestre la fecha al momento de iniciar y terminar el proceso.