**Redireccionamientos de Entrada, Salida y Errores**

**¿Qué es la redirección?**

Cuando usamos la terminal en Linux, cada comando genera una **salida** (lo que muestra por pantalla) y puede leer **entrada** (lo que recibe desde el teclado u otro archivo). La redirección nos permite **modificar de dónde toma la entrada un comando o a dónde envía su salida**

**Tipos de redirecciones y comandosCaptura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.**

**Entrada y salida estándar**

** 0: Entrada estándar (stdin)  
Es de donde el programa toma información. Por defecto, es el teclado.**

** 1: Salida estándar (stdout)  
Es a donde el programa envía sus resultados normales. Por defecto, es la pantalla.**

** 2: Salida de errores (stderr)  
Es a donde el programa envía sus mensajes de error. También, por defecto, aparece en pantalla.**

**ejemplo**

**ls carpeta > salida.txt 2> errores.txt**

✅ ¿Qué hace este comando?

ls carpeta:

Intenta listar el contenido de un directorio llamado carpeta.

> salida.txt:

Redirige la salida estándar (stdout) del comando ls al archivo salida.txt.

Es decir, si el directorio existe, su contenido se guarda en ese archivo en vez de mostrarse por pantalla.

2> errores.txt:

Redirige la salida de errores (stderr) al archivo errores.txt.

Si carpeta no existe o hay un problema, el mensaje de error va a parar a ese archivo, en vez de mostrarse en pantalla.

**Tuberías, filtros y bifurcaciones**

**¿Qué es una tubería (|)?**

Permite **pasar la salida de un comando como entrada de otro**. Esto nos da una forma muy poderosa de encadenar comandos para obtener información procesada.

**Comandos de filtrado comunes**

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

**Ejemplos encadenados**

**ps aux | grep firefox | wc -l**

**Explicación:**

* **ps aux: lista todos los procesos del sistema.**
* **grep firefox: filtra los procesos que contienen la palabra "firefox".**
* **wc -l: cuenta cuántas líneas hay (es decir, cuántos procesos relacionados con Firefox).**

**Sirve para saber cuántas instancias de Firefox están corriendo.**

**cat archivo.txt | sort | uniq > limpio.txt**

**Explicación:**

* **cat archivo.txt: muestra el contenido del archivo.**
* **sort: ordena las líneas alfabéticamente.**
* **uniq: elimina duplicados consecutivos.**
* **> limpio.txt: guarda el resultado limpio (sin duplicados) en el archivo "limpio.txt".**

**Este comando es útil para limpiar listas con muchos valores repetidos.**

**Actividad en clase: "Redireccionando la vida"**

1. Crear un archivo notas.txt y escribirle algo:

echo "Primera nota" > notas.txt

echo "Segunda nota" >> notas.txt

1. Mostrar su contenido:

cat notas.txt

1. Simular un error:

cat archivoinexistente 2> error.txt

1. Combinar salida y error:

ls -l > salida.txt 2>&1

1. Verificar que todo quedó guardado correctamente:

cat salida.txt

cat error.txt

**Actividad integradora: "Espiando procesos"**

1. Ver los procesos en ejecución:

ps aux

1. Contar cuántos procesos hay:

ps aux | wc -l

1. Buscar procesos que usan mucha memoria:

ps aux | sort -nrk 4 | head

1. Filtrar procesos de un usuario:

ps aux | grep nombreusuario