


Preparación del entorno

Ejecuta esto en tu terminal para tener un archivo de trabajo:


```
cat << EOF > inventario.txt
101:Disco Duro:Hardware:50
102:Monitor:Hardware:150
103:Teclado:Hardware:20
104:Raton:Hardware:15
105:Office:Software:300
106:Antivirus:Software:50
101:Disco Duro:Hardware:50
EOF
```

Taller Práctico: Búsqueda, Ordenación y Edición (IFCT114PO)


Ejercicio 1: Ordenación básica (sort)

- **Objetivo:** Ordenar alfabéticamente el inventario para facilitar su lectura.
- **Paso a paso:** Ejecuta `sort inventario.txt`
- **Explicación:** `sort` ordena por defecto de forma ascendente basándose en el primer carácter de cada línea. 


Ejercicio 2: Eliminación de duplicados (sort -u)

- **Objetivo:** Limpiar el archivo de entradas repetidas.
- **Paso a paso:** Ejecuta `sort -u inventario.txt`
- **Explicación:** La opción `-u` (unique) ordena el archivo y elimina las líneas que son idénticas, dejando solo una copia de cada una. 

Ejercicio 3: Búsqueda de patrones simples (grep)

- **Objetivo:** Encontrar todos los productos que pertenezcan a la categoría "Software".
- **Paso a paso:** Ejecuta `grep "Software" inventario.txt`
- **Explicación:** `grep` busca una cadena de texto específica y muestra solo las líneas que la contienen. Es la herramienta de búsqueda por excelencia en Linux. 

Ejercicio 4: Búsqueda insensible a mayúsculas y numeración (grep -in)

- **Objetivo:** Buscar "disco" sin importar si está en mayúsculas y saber en qué línea está.
- **Paso a paso:** Ejecuta `grep -in "disco" inventario.txt`
- **Explicación:** `-i` ignora la diferencia entre mayúsculas/minúsculas y `-n` nos indica el número de línea original del archivo. 

Ejercicio 5: Inversión de búsqueda (grep -v)

- **Objetivo:** Mostrar todo el inventario excepto lo que sea "Hardware".
- **Paso a paso:** Ejecuta `grep -v "Hardware" inventario.txt`
- **Explicación:** La opción -v (invert-match) descarta las líneas que coinciden con el patrón y muestra todas las demás. 🚫

Ejercicio 6: Localización de archivos por nombre (find)

- **Objetivo:** Encontrar dónde está el archivo `inventario.txt` dentro de nuestra carpeta personal.
- **Paso a paso:** Ejecuta `find ~ -name "inventario.txt"`
- **Explicación:** `find` busca en la jerarquía de directorios. El símbolo `~` indica que busque en el *home* y `-name` especifica el nombre exacto. 🧐

Ejercicio 7: Búsqueda por tamaño (find -size)

- **Objetivo:** Localizar archivos grandes que puedan estar llenando el disco.
- **Paso a paso:** Ejecuta `sudo find /var/log -size +10M`
- **Explicación:** Este comando busca en la carpeta de logs archivos que pesen más de 10 Megabytes. Es vital para el mantenimiento del sistema. 🐘

Ejercicio 8: Edición de flujo para sustitución (sed)

- **Objetivo:** Cambiar la palabra "Hardware" por "Componentes" solo en la salida de pantalla.
- **Paso a paso:** Ejecuta `sed 's/Hardware/Componentes/g' inventario.txt`
- **Explicación:** `sed` es un editor de flujo. La sintaxis `'s/buscar/reemplazar/g'` realiza una sustitución global sin modificar el archivo original (a menos que uses la opción -i). 🖋️

Ejercicio 9: Extracción y formateo de columnas (awk)

- **Objetivo:** Mostrar solo el nombre del producto y su precio, ignorando el ID y la categoría.
- **Paso a paso:** Ejecuta `awk -F ":" '{print "Producto: " $2 " - Precio: " $4}' inventario.txt`
- **Explicación:** `awk` es un lenguaje de procesamiento. `-F ":"` define el separador. `$2` y `$4` representan la segunda y cuarta columna respectivamente. 🏗️

Ejercicio 10: Filtrado avanzado con condiciones (awk con lógica)

- **Objetivo:** Mostrar solo los productos que tengan un precio superior a 100 euros.
- **Paso a paso:** Ejecuta `awk -F ":" '$4 > 100 {print $2 " cuesta más de 100€"}' inventario.txt`

- **Explicación:** `awk` permite evaluar condiciones. Aquí le decimos: "si la cuarta columna es mayor a 100, imprime la segunda columna y el mensaje". ¡Puro poder de programación en una línea! 💰