

Esta serie se centra en el **procesamiento de texto y filtros**, herramientas esenciales para cualquier administrador que necesite extraer información de archivos de configuración o registros (logs) masivos.

Para que estos ejercicios funcionen, primero vamos a crear un archivo de pruebas con datos estructurados.

---

## Preparación del entorno

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

```
cat << EOF > usuarios.txt
1:Juan:Administrador:Madrid
2:Maria:Sistemas:Barcelona
3:Pedro:Ventas:Sevilla
4:Ana:Sistemas:Valencia
5:Luis:Ventas:Madrid
EOF
```

---

## Taller Práctico: Procesamiento de Texto y Filtros

### Ejercicio 1: Visualización con numeración (**cat**)

- **Objetivo:** Leer el contenido de un archivo visualizando el número de cada línea.
- **Paso a paso:** Ejecuta `cat -n usuarios.txt`
- **Explicación:** El comando `cat` muestra todo el archivo. La opción `-n` añade numeración, lo cual es muy útil para depurar scripts o referenciar líneas específicas. 

### Ejercicio 2: Inversión de orden vertical (**tac**)

- **Objetivo:** Leer el archivo empezando por la última línea y terminando por la primera.
- **Paso a paso:** Ejecuta `tac usuarios.txt`
- **Explicación:** `tac` es literalmente `cat` al revés. Se utiliza mucho para leer logs donde lo más reciente (que suele estar al final) es lo que queremos ver primero. 

### Ejercicio 3: Inspección de cabeceras (**head**)

- **Objetivo:** Mostrar solo el inicio de un archivo para no saturar la terminal.
- **Paso a paso:** Ejecuta `head -n 3 usuarios.txt`
- **Explicación:** `head` extrae las primeras líneas. Por defecto muestra 10, pero con `-n` podemos especificar el número exacto. 

### Ejercicio 4: Inspección de colas (**tail**)

- **Objetivo:** Ver el final de un archivo.
- **Paso a paso:** Ejecuta `tail -n 2 usuarios.txt`

- **Explicación:** `tail` es fundamental para ver las últimas entradas de archivos que crecen constantemente. Al igual que `head`, acepta el parámetro `-n`. 

### Ejercicio 5: Estadísticas de archivo (`wc`)

- **Objetivo:** Contar cuántas líneas, palabras y caracteres tiene un documento.
- **Paso a paso:** Ejecuta `wc usuarios.txt`
- **Explicación:** `wc` (Word Count) devuelve tres números: líneas, palabras y bytes/caracteres. Si solo quieres las líneas, usa `wc -l`. 

### Ejercicio 6: Transformación de caracteres (`tr`)

- **Objetivo:** Convertir todo el texto de minúsculas a mayúsculas.
- **Paso a paso:** Ejecuta `cat usuarios.txt | tr 'a-z' 'A-Z'`
- **Explicación:** `tr` (translate) intercambia o borra caracteres. En este caso, mapea el rango de minúsculas al de mayúsculas usando una "tubería" (`|`) para recibir el texto de `cat`. 

### Ejercicio 7: Extracción de columnas (`cut`)

- **Objetivo:** Extraer solo los nombres de los usuarios (segunda columna).
- **Paso a paso:** Ejecuta `cut -d ':' -f 2 usuarios.txt`
- **Explicación:** `cut` divide las líneas. `-d ':'` define que el separador de columnas es el dos puntos, y `-f 2` indica que queremos la segunda columna (el campo "Field"). 

### Ejercicio 8: Limpieza de caracteres con `tr`

- **Objetivo:** Eliminar los dos puntos para que el texto sea más legible.
- **Paso a paso:** Ejecuta `tr -d ':' < usuarios.txt`
- **Explicación:** La opción `-d` (delete) de `tr` elimina el carácter indicado. Usamos `<` para redirigir el archivo hacia el comando. 

### Ejercicio 9: Monitoreo en tiempo real (`tail -f`)

- **Objetivo:** Ver cómo un archivo crece en vivo (ideal para logs de servidores).
- **Paso a paso:** 1. Ejecuta `tail -f usuarios.txt` 2. (Abre otra terminal o minimiza) y ejecuta: `echo "6:Jose:Sistemas:Bilbao" >> usuarios.txt`
- **Explicación:** La opción `-f` (follow) mantiene el comando abierto y actualiza la pantalla en cuanto el archivo recibe nuevos datos. Para salir, pulsa `Ctrl + C`. 

### Ejercicio 10: Combinación avanzada (Tuberías)

- **Objetivo:** Obtener solo el nombre del último usuario añadido en mayúsculas.
- **Paso a paso:** Ejecuta `tail -n 1 usuarios.txt | cut -d ':' -f 2 | tr 'a-z' 'A-Z'`

- **Explicación:** Aquí unimos todo: `tail` coge la última línea, `cut` extrae el nombre y `tr` lo pasa a mayúsculas. Es la esencia del poder de la terminal Linux. ☺