



**Maestría en Bioinformática**  
**Principios y Lógica de Programación**  
**Docente: Ing. María Gabriela Echeverría**

## **Actividad 2: Procesamiento básico de metadatos en GNU/Linux**

**Estudiante: Christian Robles**

Fecha de entrega: 20 de Enero de 2026

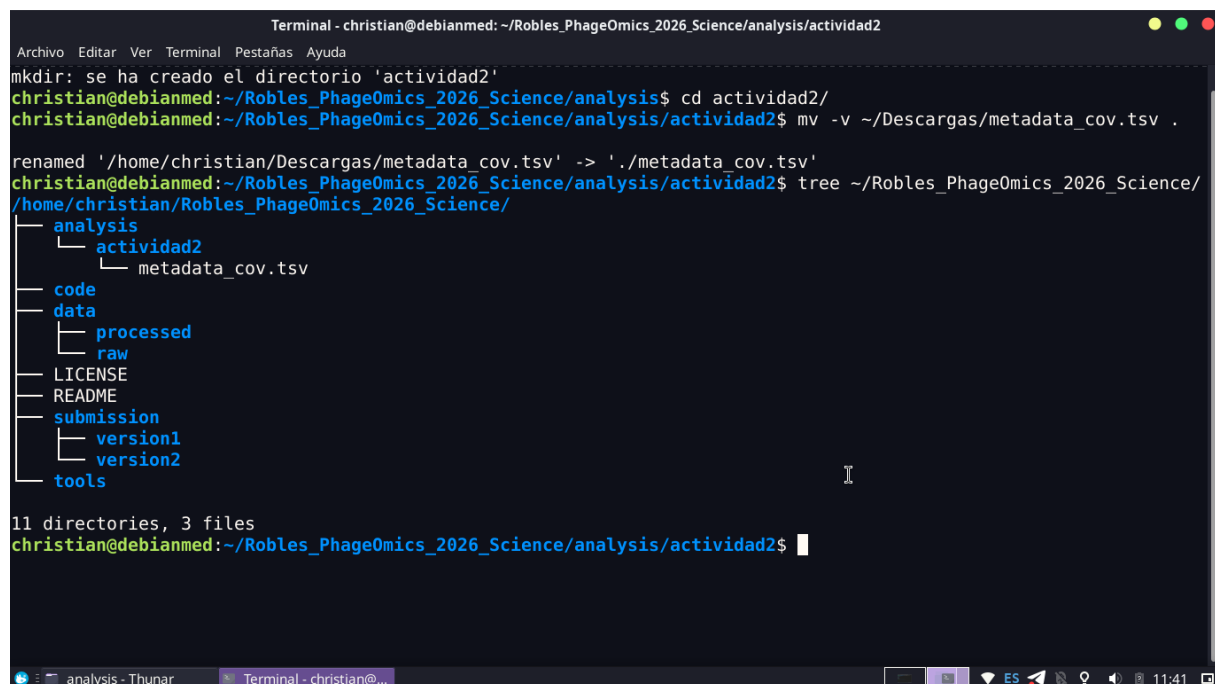
# 1. Introducción

En esta segunda actividad trabajamos con el archivo `metadata_cov.tsv`, descargado desde el aula virtual, con el objetivo principal de practicar la extracción de información relevante y la generación de reportes. La tarea se realizó en **Debian 13 Trixie**, utilizando el entorno de escritorio **XFCE**, **xfce4-terminal** y el gestor de archivos **Thunar**.

# 2. Creación del directorio de trabajo y movimiento del archivo inicial

El primer paso fue crear un nuevo directorio dentro de `analysis` llamado `actividad2`, entrar en él y mover el archivo `metadata_cov.tsv` desde la carpeta de descargas hacia este nuevo espacio de trabajo. Todo esto se realizó de la siguiente manera:

```
mkdir actividad2
cd actividad2
mv -v ~/Descargas/metadata_cov.tsv .
```



**Figura 1:** Mover el archivo `metadata_cov.tsv` al directorio de trabajo.

## Comandos utilizados:

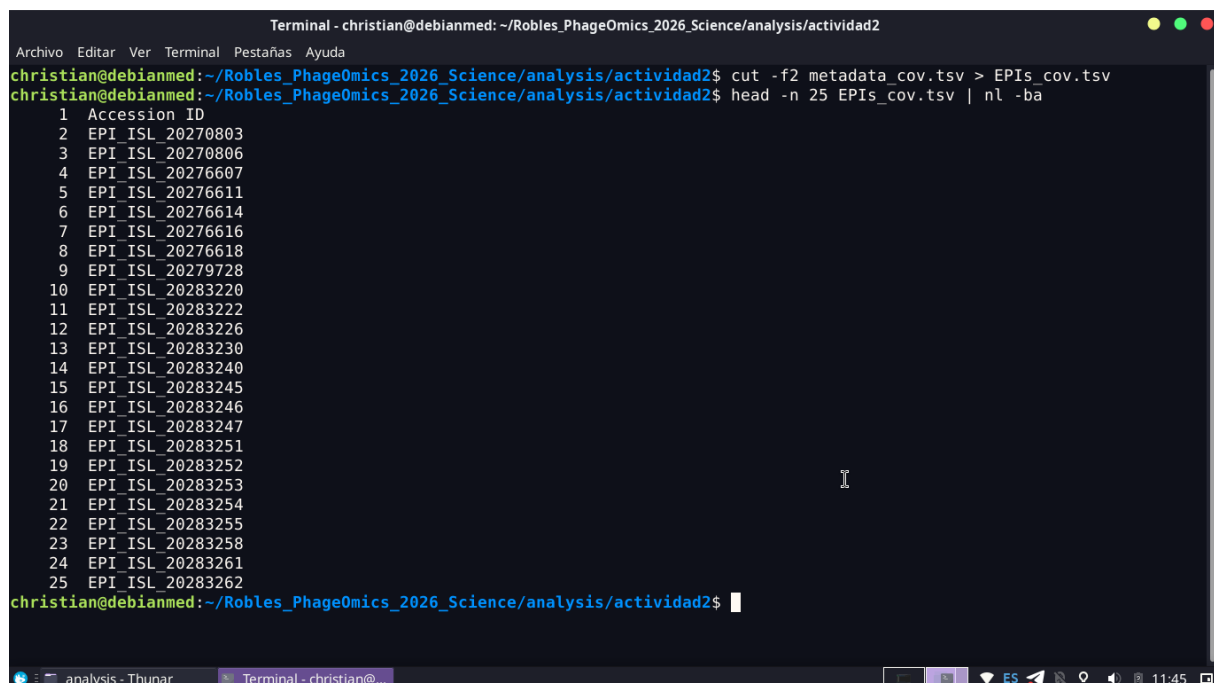
- `mkdir actividad2` → crea la carpeta donde trabajaremos.
- `cd actividad2` → nos mueve directamente dentro de esa carpeta para no tener que escribir rutas largas después.
- `mv -v ~/Descargas/metadata_cov.tsv .` → mueve el archivo desde la carpeta de descargas al directorio actual (`.`), mostrando un mensaje de confirmación gracias a la opción `-v` (*verbose*).

**Uso de la virgulilla:** La virgulilla en Linux representa el directorio personal del usuario (/home/usuario). En este caso, /Descargas apunta directamente a la carpeta de descargas dentro de nuestro home. Usarla hace que los comandos sean más cortos y portables, ya que no necesitamos escribir la ruta completa.

### 3. Extracción de IDs y creación del archivo `EPIs_cov.tsv`

El siguiente paso fue trabajar con el archivo `metadata_cov.tsv` para obtener únicamente la columna de los identificadores de secuencias (Accession ID). Esto se hizo con el comando:

```
cut -f2 metadata_cov.tsv > EPIs_cov.tsv
head -n 25 EPIs_cov.tsv | nl -ba
```



```
Terminal - christian@debianmed: ~/Robles_PhageOmics_2026_Science/analysis/actividad2
Archivo Editar Ver Terminal Pestañas Ayuda
christian@debianmed:~/Robles_PhageOmics_2026_Science/analysis/actividad2$ cut -f2 metadata_cov.tsv > EPIs_cov.tsv
christian@debianmed:~/Robles_PhageOmics_2026_Science/analysis/actividad2$ head -n 25 EPIs_cov.tsv | nl -ba
1 Accession ID
2 EPI_ISL_20270803
3 EPI_ISL_20270806
4 EPI_ISL_20276607
5 EPI_ISL_20276611
6 EPI_ISL_20276614
7 EPI_ISL_20276616
8 EPI_ISL_20276618
9 EPI_ISL_20279728
10 EPI_ISL_20283220
11 EPI_ISL_20283222
12 EPI_ISL_20283226
13 EPI_ISL_20283230
14 EPI_ISL_20283240
15 EPI_ISL_20283245
16 EPI_ISL_20283246
17 EPI_ISL_20283247
18 EPI_ISL_20283251
19 EPI_ISL_20283252
20 EPI_ISL_20283253
21 EPI_ISL_20283254
22 EPI_ISL_20283255
23 EPI_ISL_20283258
24 EPI_ISL_20283261
25 EPI_ISL_20283262
christian@debianmed:~/Robles_PhageOmics_2026_Science/analysis/actividad2$
```

**Figura 2:** Extracción de IDs con el comando `cut` y visualización de las primeras 25 líneas del archivo con el comando `head`

#### Comandos utilizados:

- `cut -f2 metadata_cov.tsv >EPIs_cov.tsv` → con la opción `-f2` extrae la segunda columna del archivo original y la guarda en un nuevo archivo llamado `EPIs_cov.tsv`.
- `head -n 25 EPIs_cov.tsv | nl -ba` → muestra las primeras 25 líneas del archivo generado y las pasa por un pipe o tubería (`|`) hacia el comando `nl -ba` que numera todas las líneas, incluyendo las líneas en blanco si las hubiera, para facilitar la visualización.

#### Comandos `cut` y `head`:

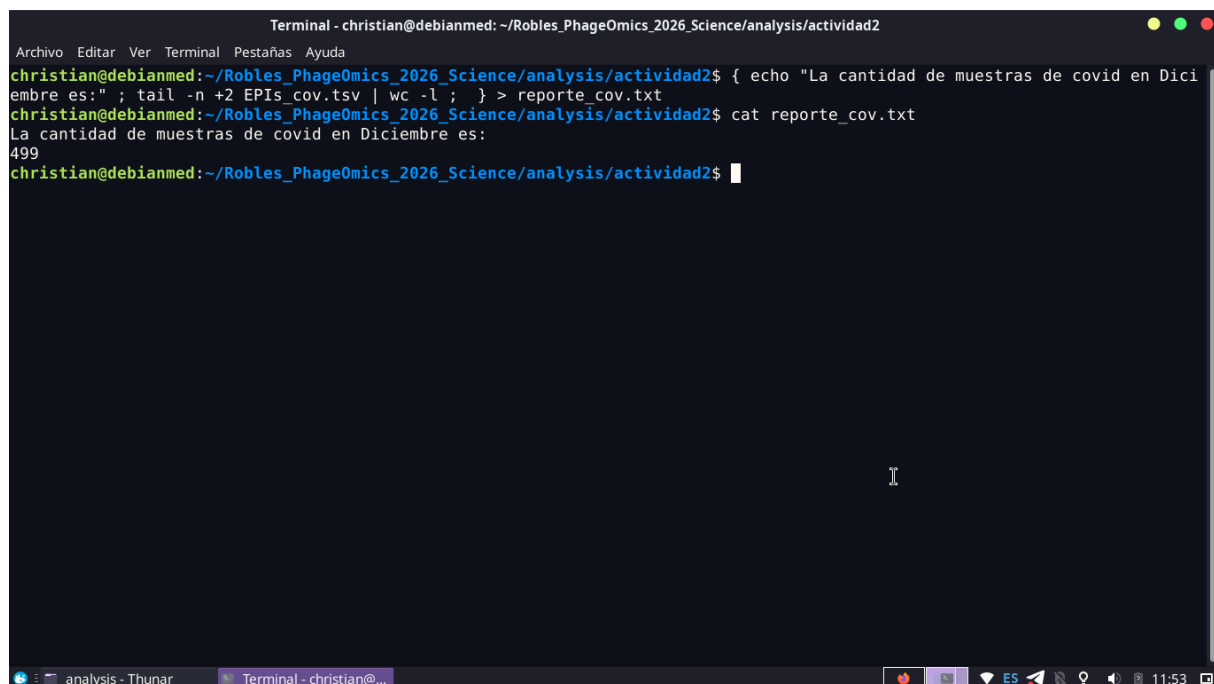
- `cut` es ideal para extraer columnas específicas de archivos delimitados por tabulaciones, como los `.tsv`.
- `head` permite revisar solo una parte del archivo sin necesidad de abrirlo completo.

- `nl -ba` añade numeración a todas las líneas, lo que facilita la referencia y comprobación de los registros.

## 4. Cantidad de muestras de covid en diciembre

En este paso se trabajó con el archivo `EPIs_cov.tsv` para calcular cuántas muestras de covid se registraron en diciembre. Para ello se utilizó un bloque de comandos encadenados que primero añade un encabezado descriptivo y luego cuenta las líneas del archivo (excluyendo la cabecera) que corresponden al número total de muestras:

```
{ echo "La cantidad de muestras de covid en Diciembre es:" ;
  tail -n +2 EPIs_cov.tsv | wc -l ; } > reporte_cov.txt
cat reporte_cov.txt
```



```
Terminal - christian@debianmed: ~/Robles_PhageOmics_2026_Science/analysis/actividad2
Archivo Editar Ver Terminal Pestañas Ayuda
christian@debianmed:~/Robles_PhageOmics_2026_Science/analysis/actividad2$ { echo "La cantidad de muestras de covid en Diciembre es:" ; tail -n +2 EPIs_cov.tsv | wc -l ; } > reporte_cov.txt
christian@debianmed:~/Robles_PhageOmics_2026_Science/analysis/actividad2$ cat reporte_cov.txt
La cantidad de muestras de covid en Diciembre es:
499
christian@debianmed:~/Robles_PhageOmics_2026_Science/analysis/actividad2$
```

Figura 3: Cantidad de muestras de covid en diciembre.

### Comandos utilizados:

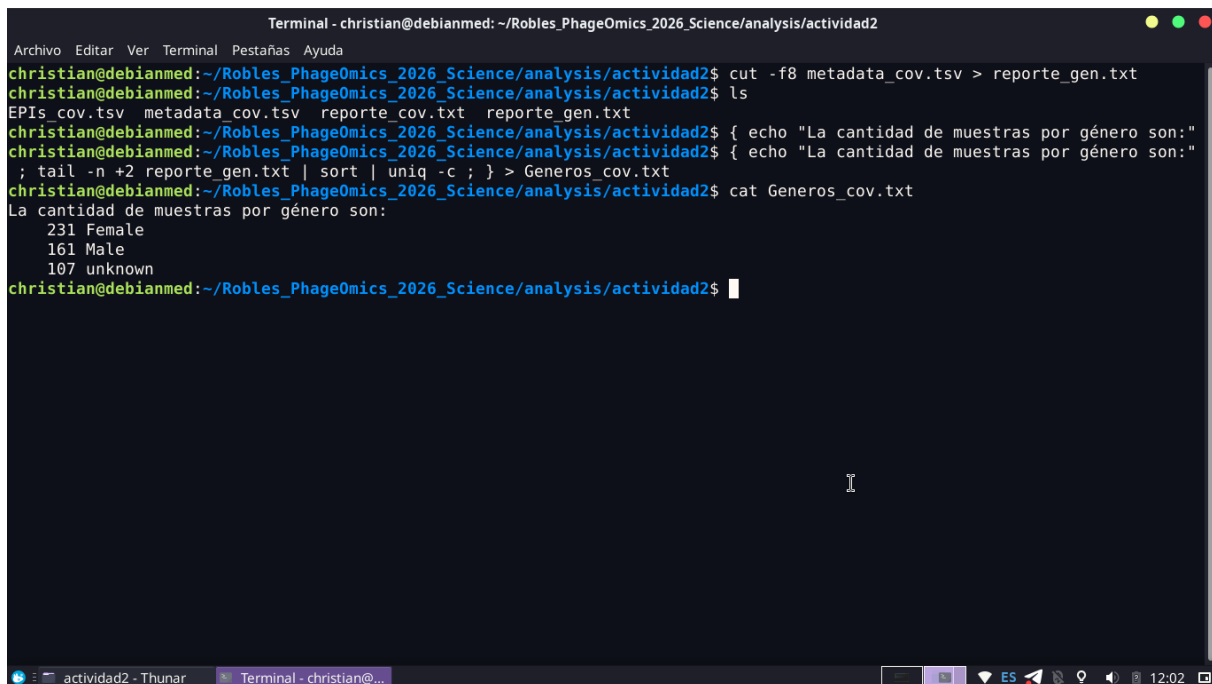
- `echo "La cantidad de muestras de covid en Diciembre es:"` → imprime un mensaje que sirve como encabezado del reporte.
- `tail -n +2 EPIs_cov.tsv | wc -l` → **tail -n +2** muestra a partir de la segunda fila el contenido del archivo de esta forma se elimina la primera línea (cabecera) y **wc -l** cuenta el número de líneas que corresponde al número de muestras. En casos donde la lógica es más compleja (muchos encabezados en el mismo archivo) se pueden usar herramientas mas potentes como `grep`, `awk` o `sed`.
- Todo el bloque entre llaves `{ ... }` se redirige con `>` al archivo `reporte_cov.txt`.
- Finalmente, `cat reporte_cov.txt` muestra el contenido del reporte en pantalla.

Se decidió agrupar los comandos dentro de llaves para generar un reporte completo en un solo paso: primero el encabezado, luego el conteo y redirigir la salida estándar del bloque de comandos dentro de las llaves hacia el archivo del reporte. Al encadenarlos, se asegura que el archivo `reporte_cov.txt` quede listo con toda la información sin necesidad de editarlo manualmente después, o en varios pasos. Además, el comando `cat` al final permite verificar que el resultado es correcto.

## 5. Distribución de muestras por género

En este paso se trabajó nuevamente con el archivo `metadata_cov.tsv`, para obtener esta vez la información sobre el género de las muestras. Se utilizó un bloque de comandos encadenados que primero genera el encabezado del reporte, extrae la columna de género y finalmente genera un reporte con los conteos:

```
cut -f8 metadata_cov.tsv > reporte_gen.txt
{ echo "La cantidad de muestras por género son:" ;
tail -n +2 reporte_gen.txt | sort | uniq -c ; } > Generos_cov.txt
cat Generos_cov.txt
```



```
Terminal - christian@debianmed: ~/Robles_PhageOmics_2026_Science/analysis/actividad2
Archivo Editar Ver Terminal Pestañas Ayuda
christian@debianmed:~/Robles_PhageOmics_2026_Science/analysis/actividad2$ cut -f8 metadata_cov.tsv > reporte_gen.txt
christian@debianmed:~/Robles_PhageOmics_2026_Science/analysis/actividad2$ ls
EPIs_cov.tsv  metadata_cov.tsv  reporte_cov.txt  reporte_gen.txt
christian@debianmed:~/Robles_PhageOmics_2026_Science/analysis/actividad2$ { echo "La cantidad de muestras por género son:"
christian@debianmed:~/Robles_PhageOmics_2026_Science/analysis/actividad2$ { echo "La cantidad de muestras por género son:"
; tail -n +2 reporte_gen.txt | sort | uniq -c ; } > Generos_cov.txt
christian@debianmed:~/Robles_PhageOmics_2026_Science/analysis/actividad2$ cat Generos_cov.txt
La cantidad de muestras por género son:
  231 Female
  161 Male
  107 unknown
christian@debianmed:~/Robles_PhageOmics_2026_Science/analysis/actividad2$
```

Figura 4: Ejemplo con múltiples opciones.png

### Comandos utilizados:

- `cut -f8 metadata_cov.tsv > reporte_gen.txt` → extrae la octava columna del archivo original y la guarda en un archivo intermedio.
- Dentro de las llaves `{ ... }` se ejecutan dos acciones:
  1. `echo "La cantidad de muestras por género son:"` → añade un encabezado descriptivo.

2. `tail -n +2 reporte_gen.txt | sort | uniq -c` → muestra a partir de la segunda fila el contenido del archivo de esta forma se elimina la primera línea que corresponde al encabezado, ordena los valores y cuenta cuántas veces aparece cada categoría de género Female, Male, y unknown.

- Todo este bloque se redirige al archivo `Generos_cov.txt`.
- Finalmente, `cat Generos_cov.txt` muestra el resultado en pantalla.

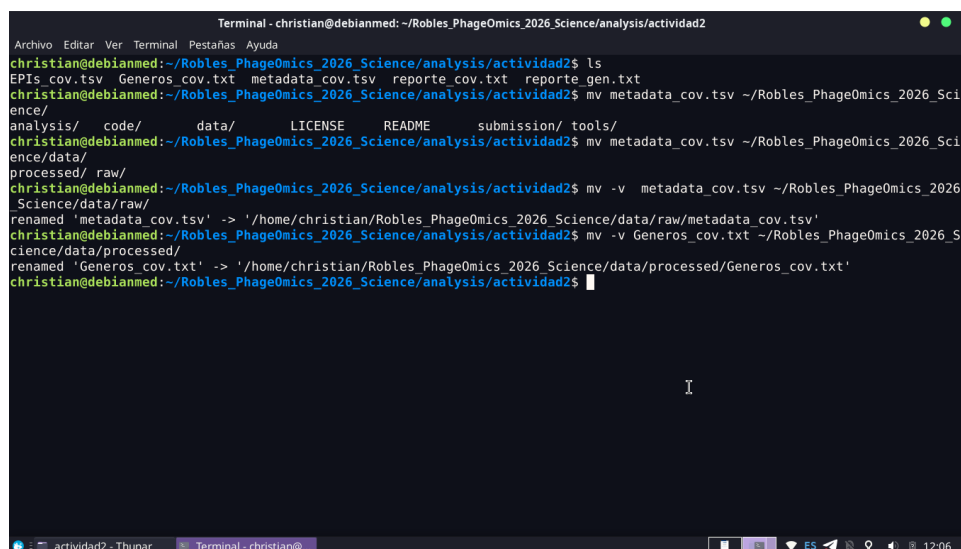
### El comando sort y unique:

**sort | uniq -c**: permite agrupar y contar las categorías de manera sencilla. Primero, `sort` ordena las líneas del archivo, lo que es necesario para que `uniq` pueda identificar las líneas idénticas consecutivas. Luego, `uniq -c` cuenta cuántas veces aparece cada línea única, proporcionando un conteo claro de las muestras por género.

## 6. Organización de archivos en subdirectorios

Una vez generados los reportes, el siguiente paso fue organizar los archivos dentro de la estructura del proyecto. Para ello se movieron los archivos hacia las carpetas `raw` y `processed`, según su tipo:

```
mv -v metadata_cov.tsv ~/Robles_PhageOmic2026_Science/data/raw/
mv -v Generos_cov.txt ~/Robles_PhageOmic2026_Science/data/processed/
```



**Figura 5:** Movimiento de archivos a carpetas correspondientes

## 7. Estructura final del proyecto

Para verificar que todos los archivos quedaron correctamente organizados en sus carpetas, se utilizó el comando `tree` sobre el directorio principal del proyecto. De esta forma se obtiene una vista jerárquica de la estructura completa:

```
tree ~/Robles_PhageOmics_2026_Science/
```

```

Terminal - christian@debianmed: ~/Robles_PhageOmics_2026_Science/analysis/actividad2
christian@debianmed:~/Robles_PhageOmics_2026_Science/analysis/actividad2$ tree ~/Robles_PhageOmics_2026_Science/
/home/christian/Robles_PhageOmics_2026_Science/
├── analysis
│   ├── actividad2
│   │   ├── EPIs_cov.tsv
│   │   ├── reporte_cov.txt
│   │   └── reporte_gen.txt
│   ├── code
│   ├── data
│   │   ├── processed
│   │   │   └── Generos_cov.txt
│   │   └── raw
│   │       └── metadata_cov.tsv
│   ├── LICENSE
│   ├── README
│   ├── submission
│   │   ├── version1
│   │   └── version2
│   └── tools
├── code
├── data
├── LICENSE
├── README
├── submission
├── version1
├── version2
└── tools
11 directories, 7 files
christian@debianmed:~/Robles_PhageOmics_2026_Science/analysis/actividad2$

```

**Figura 6:** Estructura final de directorios

El comando `tree` recorre de manera recursiva los directorios y muestra su contenido en forma de árbol. Esto permite visualizar tanto las carpetas como los archivos, y comprobar que cada elemento está en el lugar correcto.

## 8. Conclusión

La actividad permitió practicar la extracción de datos desde el archivo `metadata_cov.tsv` y la generación de reportes claros y organizados. Mediante el uso de comandos encadenados, redirecciones de la salida estándar y pipes para filtrar información relevante, contar muestras y clasificarlas por género.

## 9. Bibliografía

Fox, B., & Ramey, C. (2025). *GNU Bash Reference Manual* [Consultado en enero de 2026]. <https://www.gnu.org/software/bash/manual/>

Newham, C., & Robbins, A. (2005). *Learning the Bash Shell* (3rd). O'Reilly Media.

Shotts, W. E. (2019). *The Linux Command Line: A Complete Introduction* (2nd). No Starch Press.