UNIVERSIDAD NACIONAL DE COLOMBIA

Manual de Ingeniería de Software I

Fundamentos de Terminal, Linux, Automatización y Git

John Jairo Leal Gómez
Curso de IA Aplicada al Agro

11 de enero de 2026

Índice

1. Introducción: La Filosofía CLI

La Interfaz de Línea de Comandos (CLI) no es una herramienta antigua; es una herramienta **precisa**. En la era de la IA, la terminal es esencial porque vive en:

- **Servidores en la nube** (AWS, Google Cloud) donde se entrenan modelos.
- **Dispositivos IoT** en campo (Raspberry Pi, sensores).
- **Entornos de desarrollo** como GitHub Codespaces o Docker.

2. Introducción: £Por qué Linux es el corazón de la IA moderna?

La Inteligencia Artificial no se construye en vacío. Detrás de cada modelo que predice cosechas, clasifica frutas o detecta plagas, hay un **sistema operativo robusto, eficiente y abierto**: Linux.

Linux en la cadena de valor de la IA

- **Entrenamiento**: Los clusters de GPUs en la nube (NVIDIA DGX, AWS EC2) corren Ubuntu o CentOS.
- **Despliegue en campo**: Dispositivos IoT (Raspberry Pi, NVIDIA Jetson) usan distribuciones Linux ligeras para ejecutar modelos en tiempo real.
- **Herramientas clave**: Docker, Git, Python, TensorFlow, FastAPI todas nacieron y se optimizan en entornos Linux.

En este curso, no aprenderás Linux por nostalgia. Lo aprenderás porque es la **infraestructura invisible** que sostiene la IA aplicada. Dominar la terminal es dominar el control sobre tus propios modelos, datos y decisiones técnicas un paso esencial hacia la **soberanía tecnológica**.

Principio Fundamental: Todo es un archivo

En Linux, *todo es un archivo*. Un sensor es un archivo, el disco duro es un archivo, y los procesos son archivos. Manipular archivos es manipular el sistema.

3. Módulo 1: Navegación y Reconocimiento

Antes de dar órdenes, debemos saber dónde estamos.

3.1. El Sistema de Archivos

- **/** (Raíz): El inicio de todo.
- **~** (Home): Tu carpeta personal (/home/usuario).
- **.** (Punto): El directorio actual.
- **..** (Doble punto): El directorio padre (atrás).

3.2. Comandos de Exploración

Comando	Descripción Técnica
`pwd`	Muestra la ruta absoluta actual.
`ls`	Lista los archivos en la carpeta actual.
`ls -l`	Formato largo (permisos, dueño, tamaño y fecha).
`ls -a`	Muestra archivos ocultos (configuraciones como .git).
`ls -R`	Recursivo. Muestra el contenido de todas las subcarpetas.
`ls -lh`	*Human-readable*. Muestra tamaños en KB, MB, GB.
`history`	Muestra la lista de comandos recientes.
`clear`	Limpia la pantalla (pero no borra el historial).

Consejo Profesional

Usa la tecla `Tab` para autocompletar nombres de archivos y carpetas. ¡Ahorra tiempo y evita errores!

4. Módulo 2: Manipulación de Infraestructura

Como ingenieros, creamos y destruimos estructuras de datos y directorios.

4.1. Creación y Destrucción

- `mkdir nombre`: Crea una carpeta.
- `mkdir -p a/b/c`: Crea una ruta completa de carpetas anidadas.
- `touch archivo.txt`: Crea un archivo vacío o actualiza la fecha.
- `rm archivo`: Borra un archivo.
- `rm -r carpeta`: Borra una carpeta y su contenido (Recursivo).
- `rm -rf carpeta`: Borra forzadamente sin preguntar (Force).

△ Cuidado con rm

El comando `rm` no envía a la papelera. **Elimina permanentemente**. Nunca uses `rm -rf /`.

4.2. Movimiento y Copiado

- `cp origen destino`: Copia un archivo.
- `cp -r origen destino`: Copia una carpeta entera.
- `mv origen destino`: Mueve un archivo. También se usa para **renombrar**.

5. Módulo 3: Visualización y Edición de Datos

Un Ingeniero de Datos a menudo necesita inspeccionar CSVs de gigabytes sin abrir Excel.

5.1. Lectura Eficiente

- `cat archivo`: Concatena e imprime todo el archivo en pantalla.
- `head -n 5 archivo`: Muestra solo las primeras 5 líneas.
- `tail -n 5 archivo`: Muestra las últimas 5 líneas (útil para logs en tiempo real).
- `less archivo`: Permite navegar el archivo con flechas (salir con 'q').

5.2. El Editor Nano

Nano es un editor de texto en terminal, fundamental para ediciones rápidas en servidores.

- **Ctrl + O**: Guardar (Write Out).
- **Ctrl + X**: Salir.
- **Ctrl + K**: Cortar línea.
- **Ctrl + U**: Pegar línea.

6. Módulo 4: Scripting y Automatización (Bash)

Bash es un lenguaje interpretado que nos permite automatizar tareas repetitivas.

6.1. Estructura y Ejemplo Real

Todo script debe iniciar con el *Shebang*: `#!/bin/bash`. El siguiente ejemplo automatiza la creación de zonas de monitoreo:

```
#!/bin/bash
# Script para desplegar zonas de monitoreo

echo ">>> Iniciando despliegue automatico..."

# Bucle FOR para crear 3 zonas
for i in {1..3}; do
    echo "Configurando Zona $i"
    mkdir -p "zona_$i/sensores"

    # Generar script de Python dinamicamente (Heredoc)
    cat << EOF > "zona_$i/sensores/main.py"
print(f"Zona $i: Reporte simulado")
EOF
done

echo "[OK] Despliegue completado."
```

Listing 1: deploy.sh: Automatización de Infraestructura

6.2. Permisos de Ejecución

Linux protege el sistema impidiendo que cualquier archivo se ejecute como programa. Para correr tu script debes otorgar permisos:

- `chmod +x deploy.sh`: Otorga permiso de execution (ejecución).

Importancia de los scripts!

Automatización en flujos de IA

En producción, un script Bash puede:

- Descargar nuevos datos de sensores cada hora.
- Preprocesarlos con Python.
- Ejecutar un modelo y enviar alertas si detecta riesgo de sequía.

ąTu primer script es el embrión de un sistema de monitoreo inteligente!

7. Módulo 5: Git (Control de Versiones)

Git gestiona la historia de tu código, permitiendo "viajar en el tiempo"si algo sale mal.

1. **git init**: Crea un repositorio en la carpeta actual.
2. **git status**: £Qué ha cambiado desde la última "foto¿.
3. **git add .**: Prepara todos los cambios para la foto.
4. **git commit -m “mensaje”**: Toma la foto y la guarda permanentemente.
5. **git log**: Muestra el historial de commits.
6. **git push**: Sube tu código a la nube (GitHub/GitLab).

Git en Ciencia de Datos

Los científicos de datos usan Git para:

- Versionar datasets pequeños (con DVC o Git LFS).
- Documentar cambios en el preprocesamiento.
- Colaborar en notebooks de Jupyter sin sobrescribir el trabajo del otro.

En tu proyecto final, tu repositorio será tu portafolio técnico.