

# Semana 1: El Entorno del Ingeniero

Linux | Bash | Pipes | awk | Git

Curso IA Agroindustria

Repositorio: curso-ia-agroindustria-2

Enero 2026

# Agenda

- Objetivo de la semana y criterio de éxito.
- Terminal: navegar, organizar proyectos, inspeccionar datos.
- Pipes y awk: mini-pipelines para CSV.
- Git: bitácora científica (commits + .gitignore).
- Entregable A1: script reproducible + repo en GitHub.

# Objetivo (en 1 frase)

Al final de la semana podrás tomar un dataset en CSV, inspeccionarlo, validar calidad básica, automatizar un reporte y versionarlo con Git.

# Estructura mínima de proyecto

## Convención recomendada

- data/raw: datos originales (no se versionan).
- data/processed: datos listos para análisis.
- scripts: automatización (bash/python).
- reports: salidas (txt, tablas, figuras).

```
mkdir -p proyecto_ia/{data/raw,data/processed,scripts,reports}
```

# Terminal esencial (mínimo viable)

- `pwd`, `ls -lh`, `cd`
- `mkdir -p`, `cp`, `mv`, `rm -i`
- `find`, `du -sh`, `df -h`

```
pwd
ls -lh
cd proyecto_ia
mkdir -p data/raw
find . -name "*.csv"
du -sh data/*
```

# Inspección rápida de CSV

- Ver encabezado y muestra: head.
- Ver últimas filas: tail.
- Contar registros: wc -l.
- Navegar sin abrir Excel: less.

```
head -n 5 data/raw/sensores.csv  
tail -n 5 data/raw/sensores.csv  
wc -l data/raw/sensores.csv  
less data/raw/sensores.csv
```

# Señales de mala calidad (rápidas)

- Faltantes típicos: ,,
- Codificación rara / caracteres no ASCII.
- Columnas inconsistentes (separador incorrecto).

```
grep ",," data/raw/sensores.csv | head  
grep -P "[^\x00-\x7F]" data/raw/sensores.csv | head
```

# Pipes: entrada → proceso → salida

- El pipe (|) conecta comandos: salida de A → entrada de B.
- Construye pipelines sencillos y potentes sin abrir Python.

```
# Frecuencia por sensor (col 1)
cut -d',' -f1 data/raw/sensores.csv | sort | uniq -c | sort -nr | head
```

# awk: estadísticas rápidas en columnas

```
# Promedio temperatura (col 3)
awk -F',' '{s+=$3} END {print "Temp_prom:", s/NR}' data/raw/sensores.csv

# Max humedad (col 4) ignorando header
awk -F',' 'NR>1 {if($4>max) max=$4} END {print "Hum_max:", max}' \
    data/raw/sensores.csv
```

## Idea

awk es ideal para validaciones rápidas antes de cargar datos en pandas.

# Git como bitácora científica

- Un commit es un “fotograma” del proyecto.
- Commits pequeños + mensajes claros = reproducibilidad y auditoría.
- .gitignore evita subir datos brutos y basura.

```
git init
git add .
git commit -m "feat: estructura inicial"
git status
git log --oneline --graph
```

# .gitignore mínimo para la semana 1

```
cat > .gitignore << EOF
data/raw/
__pycache__/
.ipynb_checkpoints/
*.log
*.tmp
EOF
```

# Entregable A1 (qué se entrega)

- ① Repo con estructura data/ scripts/ reports/.
- ② scripts/setup\_proyecto.sh idempotente.
- ③ reports/validacion\_inicial.txt generado por el script.
- ④ Al menos 3 commits con mensajes tipo feat:/docs:/fix:.

# Checklist (si esto funciona, pasaste la semana)

- Puedes navegar y organizar proyectos solo con terminal.
- Puedes inspeccionar CSV y detectar problemas básicos.
- Puedes construir un pipeline con cut | sort | uniq o awk.
- Puedes crear commits claros y usar .gitignore.

# Siguiente semana

Semana 2: NumPy para alto rendimiento (vectorización, profiling, memoria).