

Introducción a la Terminal y Git

Clase 1

Profesor Yazmin Rubiela Cote

Universidad Industrial de Santander

1 de febrero de 2026

¿Por qué la "Pantalla Negra"?

GUI (Interfaz Gráfica)

- Es como un **Restaurante**.
- Solo puedes pedir lo que hay en el menú.
- Fácil, pero limitado.

CLI (Línea de Comandos)

- Es como la **Cocina**.
- Tienes los ingredientes crudos.
- Puedes automatizar, crear y controlar todo.
- Es fundamental para Data Science.

Paso 0: Instalación de Git

¡Atención!

Necesitamos instalar el intérprete de comandos. Sin esto, no podemos hablar con el computador.



Windows

- 1 Ir a: git-scm.com
- 2 Descargar "Download for Windows".
- 3 **Instalación:** Dar Siguiente a todo, PERO...
- 4 Asegurarse de marcar:
"Git from the command line and also from 3rd-party software".



Mac / Linux

- **Mac:** Abrir la terminal y escribir git. Si no está, pedirá instalar "Xcode Tools". Aceptar.
- **Linux:** Generalmente ya viene instalado. Si no:
`sudo apt install git`

Paso 1: ¿Cómo acceder a la consola?

Una vez instalado, vamos a abrir la "Puerta de la Cocina".

Usuarios Windows (IMPORTANTE)

No usaremos el CMD ni PowerShell. Usaremos una terminal basada en Linux.

- 1 Presiona la tecla **Inicio**.
- 2 Escribe: **Git Bash**.
- 3 Verás un icono de dos diamantes de colores.
¡Ábrelo!

Usuarios Mac / Linux

- **Mac:** Presiona Cmd + Espacio, escribe **Terminal** y dale Enter.
- **Linux:** Presiona Ctrl + Alt + T.

Si ven una ventana negra (o blanca) con un cursor parpadeando...
¡Ya están dentro!

Comandos de Supervivencia (1/3)

Imagina tu computador como un **Árbol** invertido de carpetas.

1. ¿Dónde estoy? (Print Working Directory)

```
$ pwd  
/c/Users/Estudiante/Escritorio
```

Comandos de Supervivencia (1/3)

Imagina tu computador como un **Árbol** invertido de carpetas.

1. ¿Dónde estoy? (Print Working Directory)

```
$ pwd  
/c/Users/Estudiante/Escritorio
```

2. ¿Qué hay aquí? (List)

```
$ ls          # Listar simple  
$ ls -l      # Listar con detalles (tamaño, fecha)  
$ ls -a      # Mostrar ocultos (como .git)
```

3. Cambiar de directorio (Change Directory)

```
$ cd Escritorio      # Entrar a una carpeta
$ cd ..              # SALIR (Ir al padre)
$ cd ~                # Ir al Home (Casa)
```

¡Cuidado!

Si la carpeta tiene espacios, usa comillas:

```
cd "Mi Carpeta"
```

4. Crear carpetas (Make Directory)

```
$ mkdir CursoMath
```


Comandos de Supervivencia (3/3)

4. Crear carpetas (Make Directory)

```
$ mkdir CursoMath
```

5. Crear archivos vacíos

```
$ touch script.py
```

Comandos de Supervivencia (3/3)

4. Crear carpetas (Make Directory)

```
$ mkdir CursoMath
```

5. Crear archivos vacíos

```
$ touch script.py
```

6. Borrar (PELIGRO: No hay papelera)

```
$ rm archivo.txt          # Borrar archivo  
$ rm -rf carpeta          # Borrar carpeta y todo su contenido
```

Automatización: Crear y Editar

1. Creación Masiva (Llaves {}) ¡No escribas 10 veces el mismo comando! Usa patrones.

```
# Crear una secuencia num rica
$ touch clase{1..5}.py
> clase1.py, clase2.py ...
```

```
# Crear combinaciones
$ touch exp_{A,B}_{2024}.txt
> exp_A_2024.txt, exp_B_2024.txt
```

2. Editar sin salir (Nano) Editor de texto dentro de la terminal.

```
# Abrir editor
$ nano script.py
```

Controles de Nano:

- **Guardar:** Ctrl + O (Enter)
- **Salir:** Ctrl + X

Interactuar con Archivos

Dependiendo de qué quieras hacer, el comando cambia:

[HTML]EFEFEF Objetivo	Comando	Ejemplo
Leer rápido (Solo imprimir en pantalla)	cat	\$ cat notas.txt
Abrir en Windows (Excel, PDF, Bloc de Notas)	start	\$ start reporte.pdf
Programar (IDE) (Abrir en VS Code)	code	\$ code script.py \$ code . (Carpeta actual)

*Tip: Recuerda usar la tecla **TAB** para autocompletar los nombres.*

STOP

- **Git:** Es el software instalado en tu PC (La máquina del tiempo).
- **GitHub:** Es la red social donde subimos el código (La nube).

Actividad 1: Crear la Identidad

Instrucción Inmediata

Todos por favor ingresen a: <https://github.com/>

Pasos:

- 1 Click en **Sign Up**.
- 2 Usen su correo personal (recomendado).
- 3 Elijan un *username* profesional (ej. `jperes_math`, no `dragon_slayer99`).
- 4 Resuelvan el puzzle de verificación.
- 5 ¡**Confirmen el correo electrónico ahora mismo!**

Actividad 2: Instalación

Usuarios Windows:

- 1 Descargar **Git Bash** desde `git-scm.com`.
- 2 **Importante:** Durante la instalación, seleccionar siempre "Use Git from Git Bash only".

Usuarios Mac:

- 1 Abrir terminal y escribir `git --version`.
- 2 Aceptar la instalación de "Command Line Tools".

Linux: `sudo apt install git`

Actividad 3: Presentarse ante Git

Una vez instalado, abran la terminal (Git Bash en Windows) y ejecuten:

```
# Configura tu nombre (aparecer en el historial)
$ git config --global user.name "Tu_Nombre_Real"

# Configura tu email (EL MISMO de GitHub)
$ git config --global user.email "tu@correo.edu.co"
```


Actividad 3: Presentarse ante Git

Una vez instalado, abran la terminal (Git Bash en Windows) y ejecuten:

```
# Configura tu nombre (aparecer en el historial)
$ git config --global user.name "Tu_Nombre_Real"

# Configura tu email (EL MISMO de GitHub)
$ git config --global user.email "tu@correo.edu.co"
```

verificar que quedó bien:

```
$ git config --list
```

Reto Final: .^{El} Arquitecto de Datos"

Objetivo: Estructurar el curso, generar datos masivos y limpiar errores, todo sin tocar el mouse.

Ejecuten esta secuencia con precisión (cuidado con los espacios):

```
$ cd ~ # 1. Volver al origen
$ mkdir MathComp2026 # 2. Crear carpeta del curso
$ cd MathComp2026 # 3. Entrar
$ mkdir Semana{1..4} # 4. Crear 4 carpetas de golpe (
    Batch)
$ cd Semana1 # 5. Entrar a la primera
$ touch simulacion_{A,B,C}.csv # 6. Crear 3 archivos de datos a la
    vez
$ touch archivo_borrar.tmp # 7. Crear un archivo "basura"
$ rm archivo_borrar.tmp # 8. ELIMINAR el archivo (Limpieza)
$ ls -F # 9. Verificar el triunfo
```

Reto Final: .^{El} Arquitecto de Datos"

Objetivo: Estructurar el curso, generar datos masivos y limpiar errores, todo sin tocar el mouse.

Ejecuten esta secuencia con precisión (cuidado con los espacios):

```
$ cd ~ # 1. Volver al origen
$ mkdir MathComp2026 # 2. Crear carpeta del curso
$ cd MathComp2026 # 3. Entrar
$ mkdir Semana{1..4} # 4. Crear 4 carpetas de golpe (
    Batch)
$ cd Semana1 # 5. Entrar a la primera
$ touch simulacion_{A,B,C}.csv # 6. Crear 3 archivos de datos a la
    vez
$ touch archivo_borrar.tmp # 7. Crear un archivo "basura"
$ rm archivo_borrar.tmp # 8. ELIMINAR el archivo (Limpieza)
$ ls -F # 9. Verificar el triunfo
```

Resultado Esperado

¡Misión Cumplida!