

# Video2 (1)

- **Propósito del video:** Enseñar cómo compilar programas en C usando la terminal de Linux y trabajar con ejecutables.
- **Entorno de trabajo:**
  - Se recomienda **Linux** y la terminal para todo el curso.
  - Comandos básicos de navegación:
    - **ls** → lista archivos en el directorio actual.
    - **cd** → cambiar de directorio.
    - **/** → separar carpetas en la ruta.
    - **Ctrl + L** → limpiar pantalla de la terminal.
- **Editor de código:**
  - El profesor usa **JBam**, pero se pueden usar otros como **Geany** o **Code::Blocks**.
  - Recomendación: mantenerlo simple para enfocarse en aprender C y no complicarse con IDEs complejos.
- **Entrada de datos:**
  - **scanf**: equivalente a **input** en Python.
  - Sintaxis: **scanf("%d", &variable);**
    - El **ampersand (&)** indica que se pasa la dirección de memoria de la variable.
    - Se requiere que la variable ya esté definida.
  - Importancia: entender el manejo de memoria y cómo C trabaja con direcciones.
- **Documentación interna:**
  - Siempre comentar el código para facilitar lectura y seguimiento.
  - Ejemplo: incluir comentarios dentro del **main** y funciones para describir su funcionamiento.
- **Compilación con GCC:**
  - GCC no viene instalado por defecto en Linux; se debe instalar según la distribución.
  - Comando básico:

```
gcc archivo.c
```

    - Crea un ejecutable por defecto llamado **a.out**.
    - Para verificar el archivo: **file a.out** → muestra que es un ejecutable de Linux.
  - Ejecutar el programa:

```
./a.out
```
  - Opción de salida personalizada:

```
gcc archivo.c -o nombreEjecutable
./nombreEjecutable
```

- Recomiendo **borrar ejecutables antiguos** antes de recompilar para evitar confusiones con errores previos.
- **Atajos y trucos en terminal:**
  - Autocompletado con **tab** para nombres de archivos y comandos.
  - Flecha arriba para acceder a comandos previos sin reescribirlos.
- **Recomendaciones finales:**
  - Practicar compilación y ejecución de los ejercicios del video anterior.
  - Mantener documentación interna y estructura ordenada del código.
  - Familiarizarse con el manejo de terminal y GCC para futuras prácticas del curso.