



UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN
ESCUELA PROFESIONAL DE INGENIERÍA EN
INFORMÁTICA Y SISTEMAS



IMPLEMENTACIÓN DE UN INTÉRPRETE DE COMANDOS

Mini Shell POSIX

PRESENTADO POR:

- LUZ MARINA FLORES CARBAJAL_2025-11901C
- MAYKOL DAVID ESPINOZA KQUERARE_2025-11902C

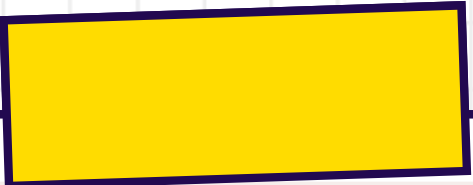
CURSO: SISTEMAS OPERATIVOS


OBJETIVOS

Desarrollar una mini shell en C++ basada en el estándar POSIX, capaz de ejecutar comandos internos y externos, manejar procesos, señales, redirecciones, concurrencia con hilos y medir el uso de memoria.

- Aplicar las funciones POSIX: fork, execvp, waitpid, pipe, dup2, open, close, sigaction, pthread_create, pthread_join.
- Implementar redirecciones, pipes, ejecución en segundo plano (&) y hilos paralelos.
- Añadir comandos internos: cd, pwd, echo, alias, history, meminfo.
- Instrumentar la memoria dinámica (new/delete) para detectar fugas.

ALCANCE

- 
- Ejecución de comandos externos y built-ins.
 - Redirección de entrada/salida.
 - Pipes y procesos en segundo plano.
 - Concurrencia mediante hilos.
 - Monitoreo de memoria.

- 
- Pipes múltiples encadenados.
 - Scripts o macros interactivos.

ARQUITECTURA

Archivo	Función
main.cpp	Bucle principal del shell y manejo de señales.
ShellCore.cpp/h	Ejecución de comandos, pipes, redirecciones y built-ins.
MemoryManager.cpp/h	Instrumentación de memoria.
Globals.cpp/h	Variables globales y manejadores de señales.

El sistema gira en torno a un bucle principal interactivo que muestra el prompt, tokeniza el comando, ejecuta instrucciones internas o externas y coordina procesos e hilos.

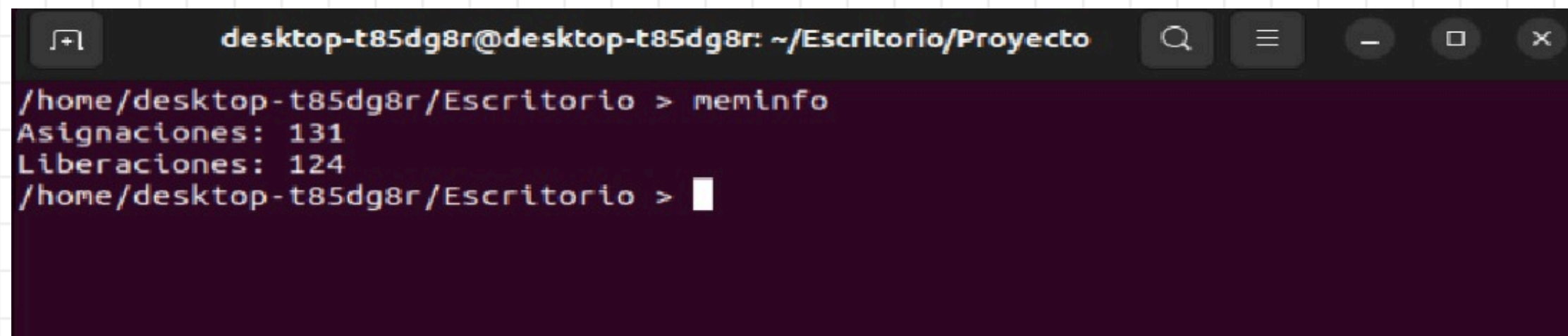
POSIX

Función	Propósito
fork()	Crear proceso hijo
execvp()	Reemplazar el proceso por un nuevo programa
waitpid()	Esperar finalización del hijo
pipe()	Comunicar cmd1 → cmd2
dup2()	Redirigir entradas/salidas
sigaction()	Manejar señales (SIGINT, SIGCHLD)
pthread_create()	Crear hilos para ejecución paralela



GESTIÓN DE MEMORIA

- Sobrecarga global de new y delete.
- Contadores atómicos g_total_alloc y g_total_free.
- Comando meminfo muestra estadísticas en tiempo real.



```
desktop-t85dg8r@desktop-t85dg8r: ~/Escritorio/Proyecto
/home/desktop-t85dg8r/Escritorio > meminfo
Asignaciones: 131
Liberaciones: 124
/home/desktop-t85dg8r/Escritorio > 
```

PRUEBAS Y RESULTADOS

Comando	Descripción	Resultado
pwd	Muestra el directorio actual	Correcto
echo hola > prueba.txt	Redirección de salida	Crea archivo
cat < prueba.txt	Redirección de entrada	Muestra 'hola'
sleep 2 &	Generea proceso en segundo plano	Prompt no bloqueado
alias ll = ls -l	Alias funcional	Correcto
paralelo "sleep 2" "echo termino"	Hilos concurrentes	Ejecución en paralelo
meminfo	Muestra métricas de memoria	Correcto


```
desktop-t85dg8r@desktop-t85dg8r: ~/Escritorio/Proyecto
/home/desktop-t85dg8r/Escritorio/Proyecto > echo "Hola mundo" > /ruta/inexistente/archivo.txt
Abriendo /ruta/inexistente/archivo.txt: No such file or directory
/home/desktop-t85dg8r/Escritorio/Proyecto > █
```

```
desktop-t85dg8r@desktop-t85dg8r: ~/Escritorio/Proyecto
/home/desktop-t85dg8r/Escritorio/Proyecto > fecha
mar 14 oct 2025 18:26:44 -05
/home/desktop-t85dg8r/Escritorio/Proyecto > █
```

```
desktop-t85dg8r@desktop-t85dg8r: ~/Escritorio/Proyecto
/home/desktop-t85dg8r/Escritorio/Proyecto > cd ..
/home/desktop-t85dg8r/Escritorio > pwd
/home/desktop-t85dg8r/Escritorio
/home/desktop-t85dg8r/Escritorio > █
```

```
desktop-t85dg8r@desktop-t85dg8r: ~/Escritorio/Proyecto
/home/desktop-t85dg8r/Escritorio/Proyecto > ls |
DEBUG: Iniciando pipe con: ls |
Error: sintaxis incorrecta en el pipe.
/home/desktop-t85dg8r/Escritorio/Proyecto > █
```




CONCLUSIÓN

Se implementó un intérprete de comandos funcional en C++ conforme a los estándares POSIX, . Gracias a una arquitectura modular, fue posible integrar nuevas funcionalidades sin comprometer la estabilidad del sistema.

