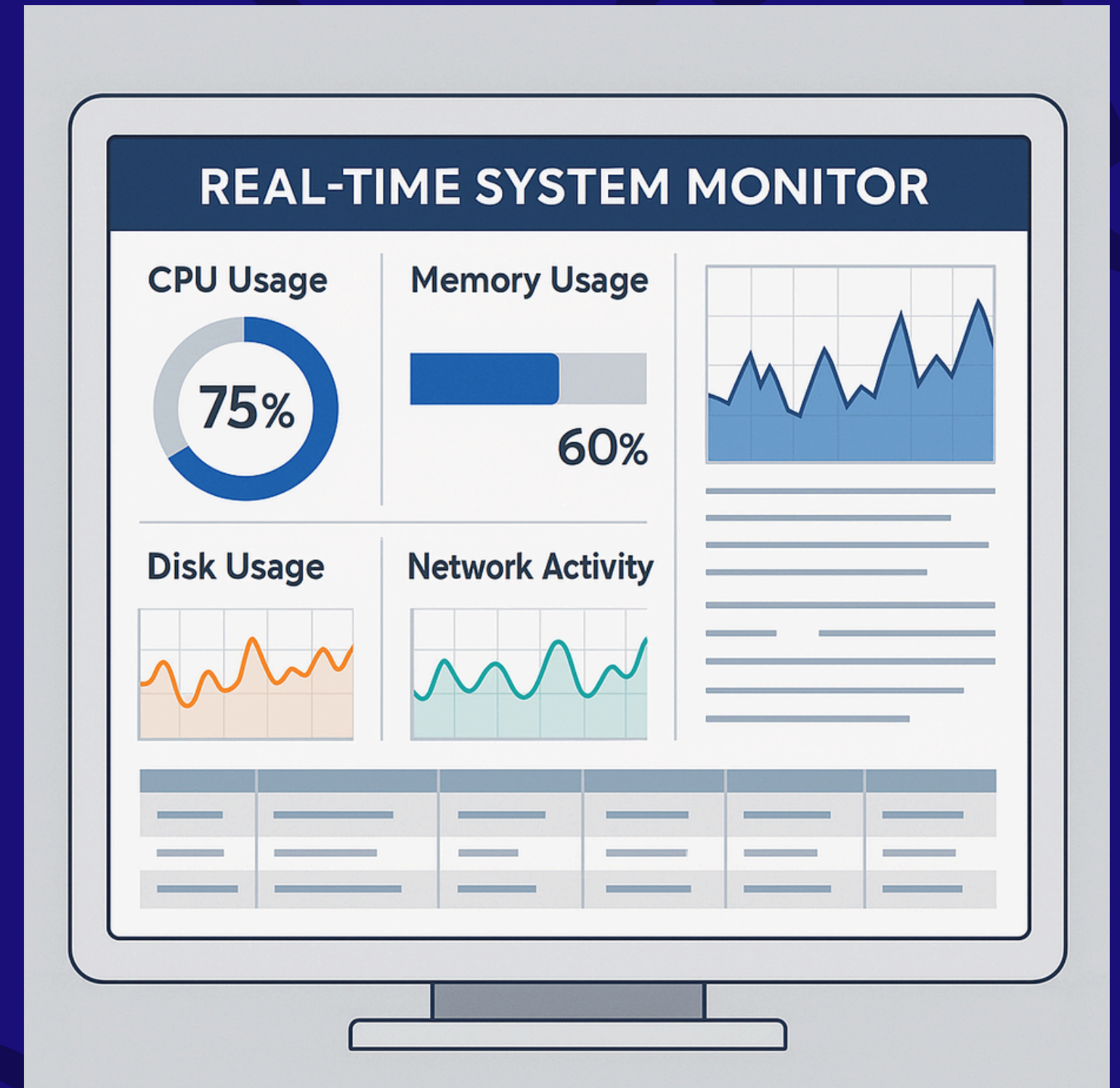


# DESARROLLO DE UN MONITOR DE SISTEMA EN TIEMPO REAL

CON NOTIFICACIONES DE  
UMBRALES DE RECURSOS

Por  
Claudia Lucía Serna Gómez  
Paula Andrea Guarín Guarín  
Duván Ferney Ruiz Ocampo





# INTRODUCCIÓN

Los sistemas operativos deben mantener un equilibrio constante en el uso de los recursos. Es común encontrar procesos que, por errores de diseño o fallos, consumen recursos excesivamente

Este proyecto propone una solución personalizada para enfrentar ese problema: un monitor que identifique y notifique automáticamente cuando un proceso sobrepase los límites definidos.



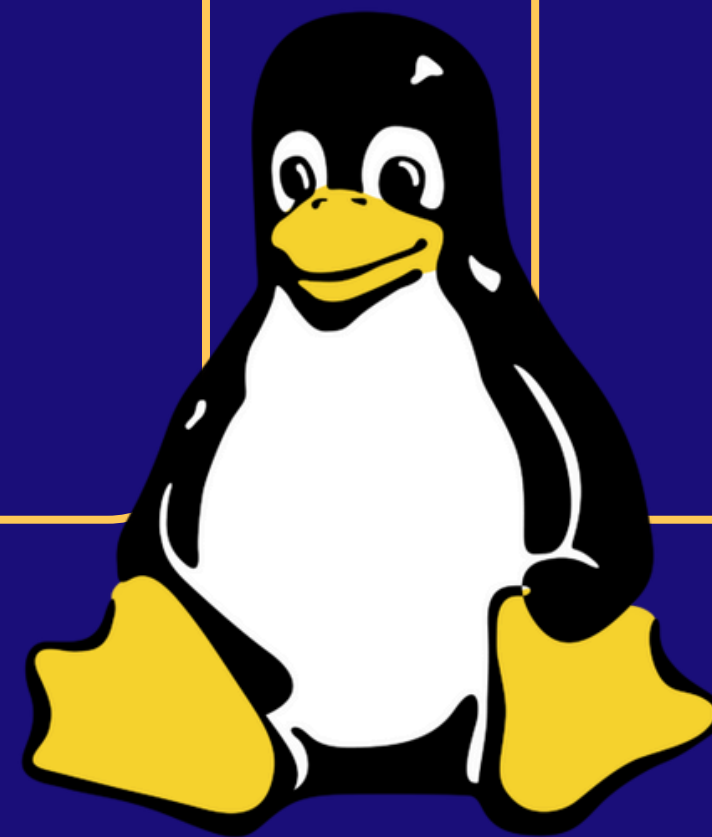
# OBJETIVOS

## OBJETIVO PRINCIPAL

Desarrollar un monitor de sistema que supervise en tiempo real el consumo de recursos (CPU y memoria) de los procesos activos, notificando al administrador cuando se superen umbrales predefinidos, para facilitar la gestión eficiente del sistema.

## OBJETIVOS ESPECÍFICOS

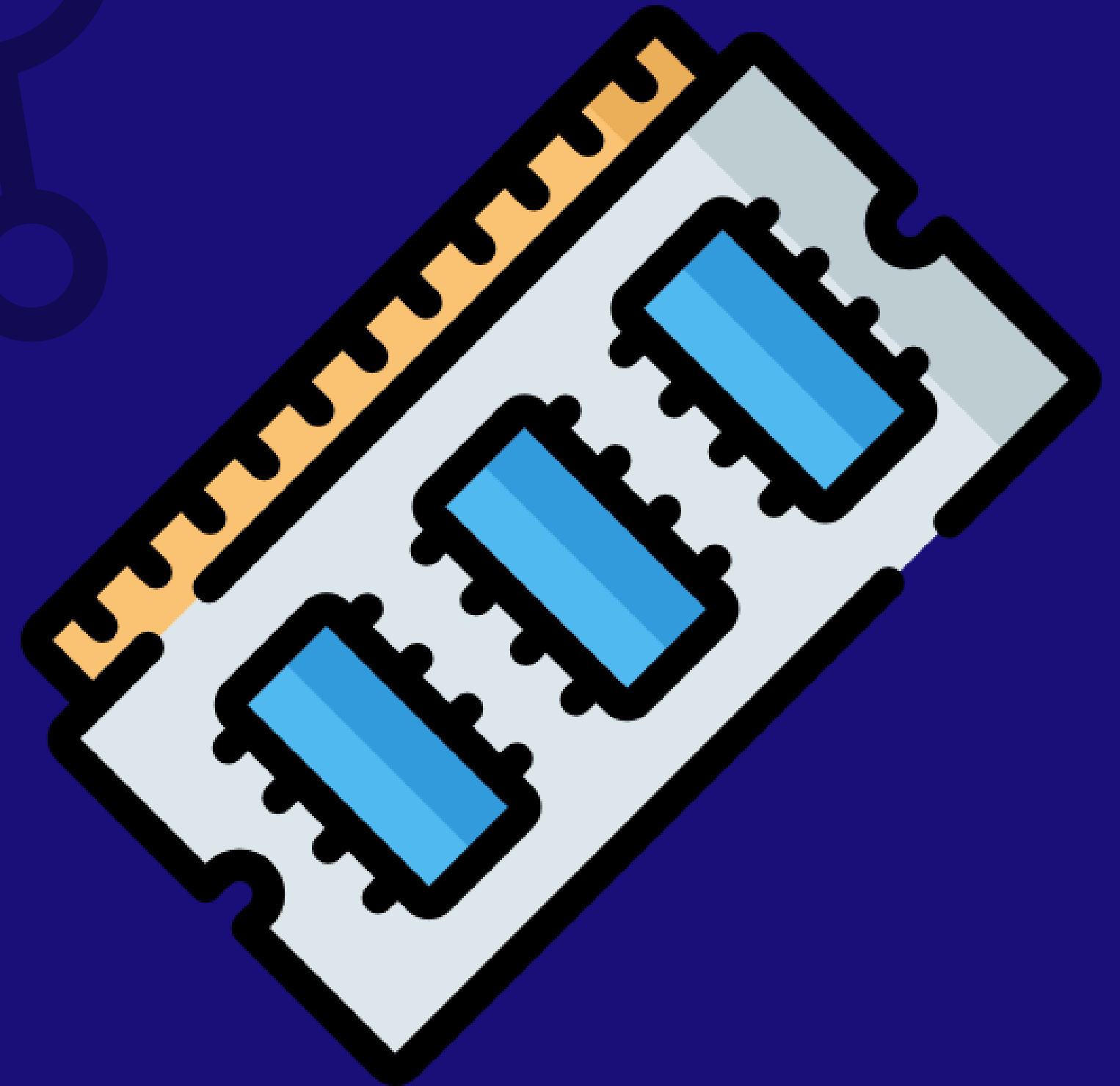
- Implementar un mecanismo de monitoreo en tiempo real que evalúe el consumo de CPU y memoria de cada proceso.
- Desarrollar un sistema de notificación que alerte al administrador cuando un proceso exceda los umbrales establecidos.
- Validar el funcionamiento del monitor mediante pruebas controladas en diferentes escenarios de carga del sistema.



# MARCO TEÓRICO

Se abordan conceptos como:

- La gestión de procesos mediante estructuras como task struct.
- La planificación (scheduling) y asignación de prioridades.
- La comunicación entre procesos (IPC), necesaria para implementar notificaciones mediante señales (SIGALRM), sockets o archivos de log.



# CONCEPTOS CLAVE

## SISTEMA DE ARCHIVOS/PROC

Se utiliza para leer datos como el porcentaje de uso de CPU o memoria disponible.

## GESTIÓN DE PROCESOS

Incluye la identificación, estados y control de los procesos mediante señales.

## PLANIFICACIÓN (SCHEDULING):

Se consideran prioridades (nice, priority) para identificar procesos que podrían estar abusando del sistema.

## COMUNICACIÓN ENTRE PROCESOS (IPC):

Se emplean mecanismos como señales, pipes o sockets para enviar alertas desde el monitor hacia el administrador.

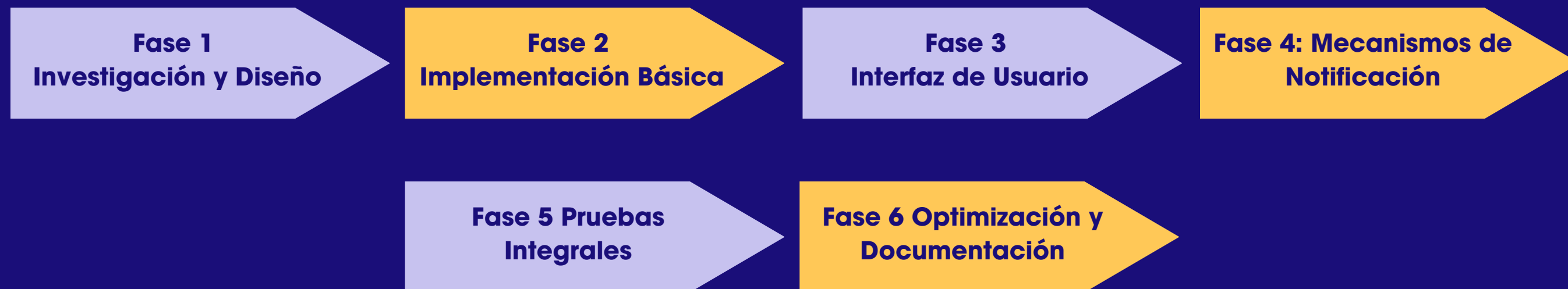






# METODOLOGÍA

- **Enfoque:** Prototipado incremental.
- **Técnicas:** Programación modular, pruebas unitarias.
- Validación por fases.



# HERRAMIENTAS Y TECNOLOGÍAS

## Lenguaje C



Compilado con gcc. Eficiencia y cercanía al sistema operativo.

## Entorno de Desarrollo



WSL/Ubuntu con Visual Studio Code

## Lectura de Métricas

kernel

A través de /proc y sysfs, para obtener datos del kernel.

## Interfaz de Usuario



ncurses, para mostrar la información de forma interactiva.

## Notificaciones



Se utilizarán syslog, señales (signal.h) y sockets para alertar.

## Depuración



Herramientas como gdb y Valgrind para identificar errores y fugas de memoria.

# CRONOGRAMA



Desarrollo de un Monitor de Sistema en Tiempo Real con Notificaciones de Umbrales de Recursos																																					
			Marzo							Abril							Mayo							Junio							Julio						
Actividad	Fecha inicio	Fecha fin	1	5	10	15	20	25	30	1	5	10	15	20	25	30	1	5	10	15	20	25	30	1	5	10	15	20	25	30	1	5	10	15	20	25	30
Investigación de /proc y diseño inicial	1-mar	30-mar																																			
Implementacion de lectura de métricas (CPU, memoria)	1-abr	30-abr																																			
Desarrollo de la interfaz con ncurses	1-may	30-may																																			
Integración de notificaciones y umbrales	1-jun	30-jun																																			
Pruebas y optimizacion	15-jun	5-jul																																			
Documentacion y entrega	1-jul	10-jul																																			

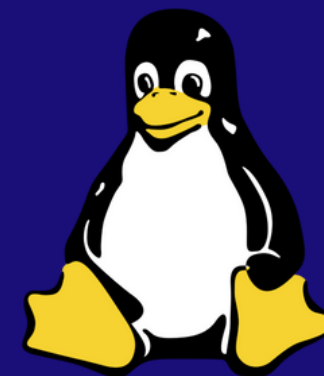
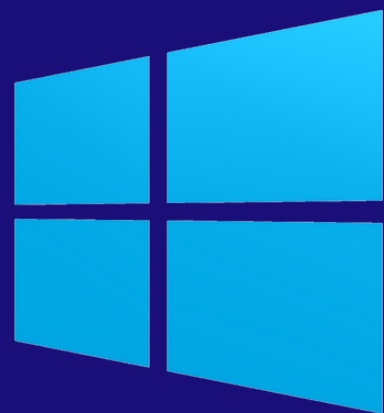


# BIBLIOGRAFÍA

The kernel development community (s.f). The /proc Filesystem. Kernel.  
<https://kernel.org/doc/html/latest/filesystems/proc.html>

Love, R. D. (2008). Linux system programming. Choice Reviews Online, 45(08), 45-4429. <https://doi.org/10.5860/choice.45-4429>

Bovet, D. P., Casetti, M., & Oram, A. (2000). Understanding the Linux Kernel.  
<http://ermak.cs.nstu.ru/Understanding.Linux.Kernel.pdf>



**Gracias**