

IMPLANTACIÓN DE SISTEMAS

UNIVERSIDAD DE SONSONATE

JUEVES 08 DE AGOSTO DE 2025

SINAPSIS Y WHATSAPP



CLAVE
IMP-SIS-G2



Implantación de Sistemas - G02
Grupo de WhatsApp



MONITOREO DE PROCESOS Y PERFORMANCE

OBJETIVOS



**IDENTIFICAR PROCESOS Y ENTENDER
SU IMPACTO EN EL RENDIMIENTO DEL
SISTEMA.**

**DOMINAR HERRAMIENTAS BÁSICAS Y
AVANZADAS PARA ANÁLISIS DE
PERFORMANCE.**

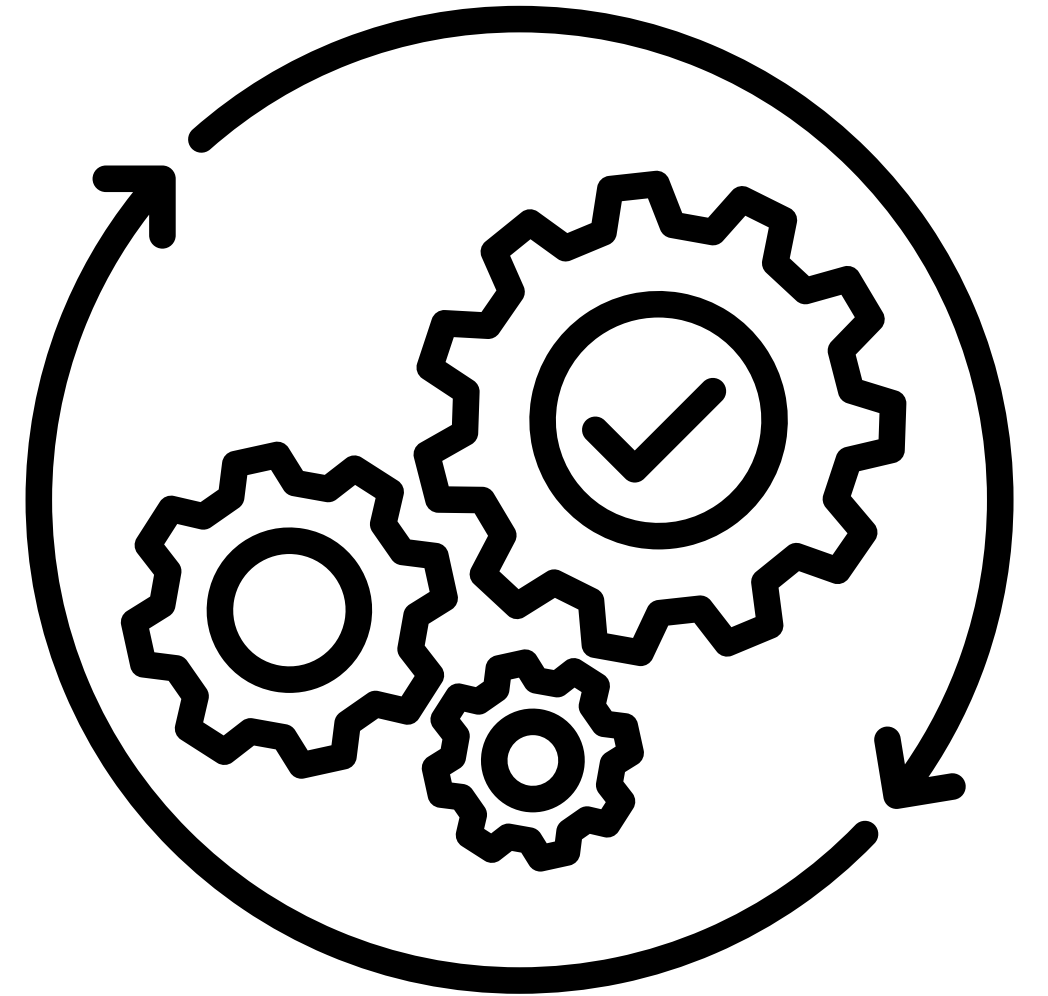
**APLICAR SOLUCIONES PRÁCTICAS A
PROBLEMAS COMUNES.**

IDENTIFICACIÓN Y CONTROL DE PROCESOS

¿QUÉ ES UN PROCESO?

DEFINICIÓN

Un proceso es una instancia de un programa en ejecución. Cada vez que ejecutas una aplicación, se crea al menos un proceso.



CARACTERÍSTICAS DE LOS PROCESOS

- **PID (PROCESS ID): IDENTIFICADOR ÚNICO NUMÉRICO.**
- **PPID (PARENT PROCESS ID): ID DEL PROCESO PADRE.**
- **ESTADO: RUNNING, SLEEPING, ZOMBIE, STOPPED**
- **PRIORIDAD: NIVEL DE IMPORTANCIA PARA EL PLANIFICADOR DE TAREAS.**
- **RECURSOS: CPU, MEMORIA, ARCHIVOS ABIERTOS**

COMANDOS PARA IDENTIFICAR PROCESOS

Comando **ps** (Process Status)

```
bash
ps                # Procesos del usuario actual en terminal
ps aux            # Todos los procesos del sistema
ps -ef            # Formato estándar Unix
ps -u username    # Procesos de un usuario específico
```

Comando **pstree**

```
bash
pstree            # Árbol de procesos
pstree -p        # Con PIDs
pstree username  # Procesos de un usuario
```

Listado de procesos específicos

```
bash
ps aux | grep nginx    = Buscar procesos de nginx
ps aux | grep -v grep  = Excluir el proceso grep
pgrep nginx            = Obtener solo PIDs de nginx
pidof nginx            = PIDs de un proceso específico
```

Interpretando la salida de **ps aux**

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.1	225316	9876	?	Ss	10:30	0:01	systemd
www	1234	2.5	1.2	123456	45678	?	S	10:31	0:30	nginx

- USER: Usuario propietario del proceso.
- PID: Process ID.
- %CPU: Porcentaje de CPU utilizado.
- %MEM: Memoria virtual en KB.
- VSZ: Memoria virtual en KB.
- RSS: Memoria física (residente) en KB.
- TTY: Terminal asociado.
- STAT: Estado del proceso.
- START: Hora de inicio.
- TIME: Tiempo total de CPU usado.
- COMMAND: Comando ejecutado.

COMANDOS PARA CONTROLAR PROCESOS

Terminar procesos

```
bash

kill PID           # Terminar proceso gracefully (SIGTERM)
kill -9 PID        # Forzar terminación (SIGKILL)
kill -15 PID       # Terminación graceful (equivale a kill)
killall process_name # Terminar todos los procesos con ese nombre
pkill pattern      # Terminar procesos que coincidan con patrón
```

Control de trabajos (Jobs)

```
bash

command &          # Ejecutar en background
jobs               # Listar trabajos en background
fg %1              # Traer trabajo 1 a foreground
bg %1              # Enviar trabajo 1 a background
nohup command &    # Ejecutar independiente de terminal
```

Modificar prioridades

```
bash

nice -n 10 command # Ejecutar con menor prioridad
renice 5 PID        # Cambiar prioridad de proceso existente
# Valores: -20 (máxima prioridad) a +19 (mínima prioridad)
```

Señales importantes

```
bash

kill -l            # Listar todas las señales disponibles

# Señales más comunes:
# SIGTERM (15) - Terminación graceful
# SIGKILL (9)  - Terminación forzada (no puede ser ignorada)
# SIGSTOP (19) - Pausar proceso
# SIGCONT (18) - Continuar proceso pausado
# SIGHUP (1)   - Hang up (recargar configuración)
```

HERRAMIENTAS PARA ANÁLISIS DEL RENDIMIENTO

HERRAMIENTAS DE MONITOREO EN TIEMPO REAL

TOP - Monitoreo Clásico

```
bash

top           # Monitor básico
top -u username # Procesos de un usuario
top -p PID1,PID2 # Procesos específicos
```

Interpretando TOP

```
top - 14:30:25 up 5 days, 3:45, 2 users, load average: 0.15, 0.25, 0.30
Tasks: 187 total, 1 running, 186 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2.3 us, 1.1 sy, 0.0 ni, 96.5 id, 0.1 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 8192.0 total, 3456.2 free, 2234.8 used, 2501.0 buff/cache
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used. 5654.3 avail Mem
```

Comandos dentro de TOP

- **1**: Ver CPUs individuales
- **M**: Ordenar por memoria
- **P**: Ordenar por CPU
- **k**: Terminar proceso
- **r**: Cambiar prioridad
- **q**: Salir

HTOP - Monitoreo Avanzado

```
bash
```

```
htop
```

```
# Monitor interactivo mejorado
```

Interpretando TOP

CPU[5.3%] Tasks: 74, 152 thr; 1 running											
Mem[677M/1.95G] Load average: 0.00 0.04 0.05											
Swp[0K/0K] Uptime: 8 days, 23:50:29											
PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
1	root	20	0	202M	3540	2160	S	0.0	0.2	1:09.73	/usr/lib/systemd/systemd --switched-
10114	root	20	0	169M	23872	2140	S	0.0	1.2	0:25.62	└─ lfd - sleeping
7096	root	20	0	273M	63180	3604	S	0.0	3.1	0:01.74	└─ /usr/bin/spamd --pidfile /var/run
7100	root	20	0	273M	60284	708	S	0.0	2.9	0:00.00	└─ spamd child
7099	root	20	0	273M	60284	708	S	0.0	2.9	0:00.00	└─ spamd child
1145	root	20	0	15692	1488	1180	S	0.0	0.1	0:01.77	└─ /usr/sbin/dovecot -F
28880	dovecot	20	0	62280	3328	2472	S	0.0	0.2	0:00.00	└─ dovecot/dict
20393	dovecot	20	0	77968	4352	3180	S	0.0	0.2	0:00.01	└─ dovecot/auth
13378	root	20	0	15948	1268	956	S	0.0	0.1	0:00.00	└─ dovecot/ssl-params
1178	root	20	0	9488	1180	860	S	0.0	0.1	0:00.25	└─ dovecot/log
1177	dovecot	20	0	9360	1004	852	S	0.0	0.0	0:00.32	└─ dovecot/anvil
1134	root	20	0	107M	808	676	S	0.0	0.0	0:00.00	└─ /sbin/agetty --noclear tty1 linux
1133	root	20	0	123M	1548	920	S	0.0	0.1	0:04.16	└─ /usr/sbin/crond -n
14031	root	20	0	176M	2480	1560	S	0.0	0.1	0:00.00	└─ /usr/sbin/CROND -n
14033	root	20	0	110M	1452	1264	S	0.0	0.1	0:00.00	└─ /bin/bash /usr/share/clamav
14041	root	20	0	105M	344	276	S	0.0	0.0	0:00.00	└─ sleep 7328
1131	root	20	0	25904	932	732	S	0.0	0.0	0:00.00	└─ /usr/sbin/atd -f
1122	root	20	0	107M	796	668	S	0.0	0.0	0:00.00	└─ /sbin/agetty --keep-baud 115200 3
1116	root	18	-2	309M	8820	5284	S	0.0	0.4	1:44.84	└─ /usr/sbin/rsyslogd -n
1142	root	20	0	309M	8820	5284	S	0.0	0.4	0:22.24	└─ /usr/sbin/rsyslogd -n
1141	root	20	0	309M	8820	5284	S	0.0	0.4	1:22.55	└─ /usr/sbin/rsyslogd -n
1115	root	20	0	110M	4012	2980	S	0.0	0.2	0:00.67	└─ /usr/sbin/sshd -D
25784	root	20	0	157M	6480	5000	S	0.0	0.3	0:00.05	└─ sshd: allanb [priv]
25786	allanb	20	0	157M	2672	1192	S	0.0	0.1	0:00.09	└─ sshd: allanb@pts/1
25787	allanb	20	0	112M	2024	1644	S	0.0	0.1	0:00.01	└─ -bash
F1Help F2Setup F3Search F4Filter F5Sorted F6Collap F7Nice -F8Nice +F9Kill F10Quit											

Ventajas de HTOP

- Interfaz más colorida e intuitiva.
- Navegación con mouse.
- Árbol de procesos integrado.
- Filtrado más fácil.
- Barras graficas CPU y memoria.

HERRAMIENTAS DE MONITOREO DE RECURSOS

MEMORIA

```
bash

free -h           # Uso de memoria en formato humano
free -m           # En megabytes
cat /proc/meminfo  # Información detallada de memoria
vmstat 1          # Estadísticas de memoria virtual cada segundo
```

CPU

```
bash

lscpu              # Información de CPU
cat /proc/cpuinfo  # Detalles completos de CPU
mpstat 1           # Estadísticas de CPU cada segundo
sar -u 1 10        # CPU usage cada segundo, 10 veces
```

DISCO

```
bash

df -h              # Espacio en disco por filesystem
du -sh directory/  # Tamaño de directorio
iostat 1           # Estadísticas de E/O cada segundo
iotop              # Procesos ordenados por E/O
```

RED

```
bash

netstat -tulpn     # Conexiones de red activas
ss -tulpn          # Versión moderna de netstat
iftop              # Tráfico de red en tiempo real
nethogs            # Uso de red por proceso
```

HERRAMIENTAS DE DIAGNÓSTICO ESPECIALIZADO

Análisis de Rendimiento

```
bash

# Verificar carga del sistema
uptime          # Load average
w               # Usuarios conectados y carga

# Procesos que más recursos consumen
ps aux --sort=-%cpu | head -10    # Top CPU
ps aux --sort=-%mem | head -10    # Top memoria
```

Debugging de Procesos

```
bash

strace -p PID      # Rastrear system calls de un proceso
ltrace -p PID      # Rastrear library calls
lsof -p PID        # Archivos abiertos por proceso
lsof -i :8080      # Qué proceso usa puerto 8080
```

Análisis de Logs

```
bash

dmesg              # Mensajes del kernel
journalctl -f      # Log del sistema en tiempo real
tail -f /var/log/syslog # Seguir log del sistema
```

¿SABÍAS QUE?

"¿Sabías que cuando Netflix detecta que un servidor está consumiendo más del 80% de CPU por más de 5 minutos, automáticamente redistribuye la carga a otros servidores? Esto es posible gracias al monitoreo continuo."