

Supuesto Didáctico


2.º DAM – Programación de Servicios y Procesos

“Monitor y gestor de procesos multiplataforma”

2.1 ¿Qué supuesto queremos trabajar?

Desarrollar una aplicación que simule tareas reales de:

- **Administración de procesos del sistema operativo**
- **Monitorización y control de recursos**

 Mediante programación práctica, el alumnado aplicará conceptos clave sobre procesos, señales, y uso de herramientas del sistema.

2.2 Contextualización del alumnado

- Curso: 2.º DAM
- Asignatura: Programación de Servicios y Procesos
- Alumnado con conocimientos de programación en varios lenguajes
- Familiarizado con sistemas operativos (Linux, Windows) y herramientas CLI
- Nivel intermedio-avanzado

2.3 Conocimientos previos requeridos

- Fundamentos de programación estructurada y orientada a objetos
- Uso de la consola del sistema operativo (comandos básicos de procesos)
- Lectura/escritura de ficheros
- Uso de estructuras de control, funciones y módulos
- Experiencia básica en Python, Java o Bash

2.4 Objetivos de aprendizaje

- Simular tareas reales de administración de procesos
- Usar herramientas y librerías para consultar procesos del sistema
- Enviar señales de control a procesos activos
- Registrar eventos en ficheros de log
- Desarrollar aplicaciones multiplataforma con orientación práctica



2.5 Metodología






- Enfoque por proyectos (ABP)
- Desarrollo guiado e incremental
- Pruebas frecuentes y revisión de código
- Presentación oral del funcionamiento
- Tutorías técnicas durante la implementación

2.6 Material didáctico (DUA)

Aplicación del **Diseño Universal para el Aprendizaje**:

- Código base por lenguaje
- Diagrama de flujo de procesos
- Esquema funcional de señales y logs
- Interfaz de consola como punto de partida
- Alternativa CLI o GUI, según el ritmo del alumnado





2.7 Secuencia de acciones formativas

1.  Revisión de comandos del sistema operativo (`ps` , `kill` , `top` , `htop`)
2.  Análisis de funcionamiento del sistema real
3.  Desarrollo incremental de la aplicación
4.  Pruebas, validación y depuración
5.  Presentación y defensa del proyecto final

2.8 Actividad principal

“MONITOR Y GESTOR DE PROCESOS MULTIPLATAFORMA”

El alumnado debe desarrollar una aplicación que:

-  Muestra procesos activos:
 - PID, uso de CPU, memoria, usuario
-  Permite enviar señales:
 - kill, nice, suspend
-  Registra todas las acciones en un **fichero de log**
-  Lenguajes posibles:
 - Python (con `psutil`)
 - Java (`ProcessHandle`)
 - Bash + comandos del sistema

Opcionales para ampliar

- Exportación a **CSV** o **JSON**
- Interfaz gráfica (Tkinter, JavaFX, Zenity)
- Integración con API REST para consultar procesos remotamente
- Monitorización extendida: red, disco, puertos







2.9 Evaluación: Instrumentos y criterios

Criterios

- Funcionalidad básica y ampliaciones
- Uso correcto de librerías y señales
- Estructura del código y modularidad
- Registro completo y veraz de acciones
- Presentación clara del proyecto

Instrumentos

-  Rúbrica detallada
-  Pruebas en ejecución
-  Defensa oral
-  Observación del desarrollo



2.10 Inclusión y atención a la diversidad

- Código base disponible en varios lenguajes
- Posibilidad de hacer una versión solo-lectura
- Plantillas de log e interfaz predefinidas
- Acompañamiento técnico durante todo el proceso
- Entregas parciales y revisión adaptada al ritmo individual


2.11 Actividades de ampliación

Para el alumnado con ritmo avanzado:

- Monitorizar tráfico de red o uso de disco
- Implementar un servicio en segundo plano
- Crear panel interactivo con GUI
- Enviar alertas si un proceso supera uso de CPU/memoria
- Integración con webhook o API externa

Conclusión

Este supuesto permite al alumnado:

- ✓ Comprender procesos del sistema desde la programación
 - ✓ Trabajar con señales, logs, y monitorización real
 - ✓ Aplicar conocimientos en una simulación profesional
 - ✓ Usar Python, Java o Bash en un contexto práctico
-  ¡Convierte tu programa en una herramienta real de administración!