Imagen que contiene Logotipo

Descripción generada automáticamente**Trabajo Practico N° 3**

Nombre: Ramiro Nahuel Flores

Fecha:4/2/2024

Trabajo Practico: n°3

Titulo: Mejoras de Script

Tabla de contenido

[**Informe de Mejoras en el Script informe\_del\_sistema.sh** 2](#_Toc189767687)

[**Introducción** 2](#_Toc189767688)

[**Análisis del Código Original** 2](#_Toc189767689)

[**Mejoras Implementadas** 2](#_Toc189767690)

[**Modularización del Código con Funciones** 3](#_Toc189767691)

[**Implementación de Manejo de Errores Robustecido** 3](#_Toc189767692)

[**Mejora en los Comentarios y Documentación** 3](#_Toc189767693)

[**Implementación de main() como Orquestador del Script** 4](#_Toc189767694)

[**Conclusión** 4](#_Toc189767695)

**Informe de Mejoras en el Script informe\_del\_sistema.sh**

**Introducción**

El presente informe detalla las mejoras realizadas sobre el script informe\_del\_sistema.sh, con el objetivo de optimizar su estructura, mejorar la reutilización del código, fortalecer el manejo de errores y documentar correctamente cada sección. Estas mejoras permiten una ejecución más eficiente y robusta, facilitando el mantenimiento y futuras ampliaciones.

**Análisis del Código Original**

El script original tenía las siguientes características:  
✅ Extraía información del sistema, incluyendo:

* Nombre del host
* Fecha y hora
* Espacio en disco disponible
* Usuarios conectados
* Búsqueda de procesos  
  ✅ Funcionaba correctamente, pero presentaba áreas de mejora:  
  ❌ No tenía una estructura modular (funciones reutilizables).  
  ❌ Falta de manejo de errores robusto.  
  ❌ Comentarios escasos o poco explicativos.  
  ❌ Código lineal, lo que dificultaba su mantenimiento.

**Mejoras Implementadas**

**Modularización del Código con Funciones**

El código fue refactorizado para **agrupar la lógica en funciones reutilizables**, mejorando la estructura y facilitando la reutilización.

🔹 **Antes:** Todo el código estaba en un solo bloque sin reutilización de lógica.  
🔹 **Después:** Se crearon funciones para cada tarea específica:

✔️ print\_header(): Imprime el encabezado con el nombre del sistema.  
✔️ get\_root\_space(): Muestra el espacio libre en disco.  
✔️ get\_logged\_users(): Lista los usuarios conectados.  
✔️ search\_process(): Permite buscar procesos en ejecución.  
✔️ main(): Función principal que coordina la ejecución.

**Implementación de Manejo de Errores Robustecido**

Se agregaron controles de errores para evitar fallos inesperados:

✔️ **Uso de set -euo pipefail**:

* set -e: Detiene el script si un comando falla.
* set -u: Evita el uso de variables no definidas.
* set -o pipefail: Detecta errores en comandos encadenados con |.

✔️ **Validaciones en funciones**:

* get\_root\_space() verifica si el punto de montaje existe.
* search\_process() maneja correctamente cuando no se encuentra el proceso.

**Mejora en los Comentarios y Documentación**

Se agregaron comentarios detallados en cada función, explicando su propósito y cómo funciona. Ahora el código es **autodocumentado**, facilitando su comprensión por otros desarrolladores.

Ejemplo de mejora en los comentarios:

🔹 **Antes:**

bash

CopiarEditar

# imprimo el espacio disponible de la unidad root (/)

echo "Uso del Disco $(df -h $point | awk 'NR > 1 {print $7}')"

🔹 **Después:**

bash

CopiarEditar

# Función que obtiene el espacio libre en el sistema de archivos raíz (/)

# Valida que el punto de montaje sea válido antes de ejecutar el comando df

get\_root\_space() {

local point=$1

if [[ -d "$point" ]]; then

echo "Uso del Disco en $point:"

df -h "$point" | awk 'NR==2 {print $4, "disponibles"}'

else

echo "Error: El punto de montaje $point no existe."

exit 1

fi

}

**Implementación de main() como Orquestador del Script**

Se agregó una función main() que organiza la ejecución del script de manera estructurada.

🔹 **Antes:**  
El código se ejecutaba en secuencia, sin un punto de entrada definido.

🔹 **Después:**  
Ahora main() controla el flujo del programa, garantizando que todas las funciones se ejecuten correctamente y en orden lógico.

bash

CopiarEditar

main() {

print\_header

get\_root\_space "/"

get\_logged\_users

search\_process

echo "✅ Informe generado correctamente."

}

El script ahora es más **legible, mantenible y escalable**.

**Resultados y Beneficios de las Mejoras**

🔹 **Código más estructurado**: La modularización facilita la reutilización y mantenimiento.  
🔹 **Mayor robustez**: Implementación de set -euo pipefail y validaciones en cada función.  
🔹 **Documentación mejorada**: Comentarios detallados para cada función.  
🔹 **Mejor experiencia de usuario**: Salida más clara y con validaciones de errores.

**Conclusión**

Las mejoras realizadas en informe\_del\_sistema.sh han optimizado su rendimiento y confiabilidad, permitiendo que sea una herramienta más eficiente para la extracción de información del sistema. La modularización del código y el refuerzo en el manejo de errores aseguran que el script pueda evolucionar y adaptarse a futuras necesidades sin comprometer su estabilidad.

Imágenes del resultado del script en funcionamiento

Texto

El contenido generado por IA puede ser incorrecto.