

Sistemas operativos en red

UD 04. Actividades Evaluables 01



Autor: Sergi García, Gloria Muñoz

Actualizado Septiembre 2025

Licencia



Reconocimiento - No comercial - CompartirIgual (BY-NC-SA): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se ha de hacer con una licencia igual a la que regula la obra original.

Nomenclatura

A lo largo de este tema se utilizarán diferentes símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

Importante

Atención

Interesante

A entregar

ÍNDICE

1. Fecha de entrega	3
2. Observaciones previas a la realización de tareas evaluables	3
3. Actividad 01	3
4. Actividad 02	3
5. Actividad 03	3
6. Actividad 04	3
7. Actividad 05	4
8. Actividad 06	4
9. Actividad 07	4
10. Actividad 08	4
11. Actividad 09	4
12. Actividad 10	4
13. Actividad 11	4
14. Actividad 12	5
15. Actividad 13	5
16. Actividad 14	5
17. Actividad 15	5

UD04. ACTIVIDADES EVALUABLES 01

1. FECHA DE ENTREGA

Fecha límite de entrega: Martes 14 de octubre a las 23:55.

La actividad será evaluada cuando haya pasado la fecha límite de la entrega.

! Atención: la fecha de entrega no es prorrogable. Si no la entregas en tiempo y forma, la calificación de la actividad será 0.

2. OBSERVACIONES PREVIAS A LA REALIZACIÓN DE TAREAS EVALUABLES

- Salvo excepciones que lo indique en cada actividad, deberás generar un único documento para todo el boletín y en ese documento incluir la respuesta a cada actividad.
 - Si para la entrega se requiere entregar varios ficheros, entrégalos comprimido en un único fichero con extensión ".zip".
- Las actividades deben realizarse en la lengua indicada en cada actividad. La gramática y ortografía tenéis que intentar hacerla bien.
- **Se deben incluir comentarios del código en cada fichero entregado.**
- **NO SE PUEDE UTILIZAR CHATGPT NI SIMILARES, ya que el objetivo es que vosotros intentéis pensar como escribir el código.**

! Atención: no cumplir estas consideraciones puede reducir la nota hasta 3 puntos.

3. ACTIVIDAD 01

Visita esta página con más de 500 scripts en PowerShell <https://github.com/fleschutz/PowerShell> . Elige 5 de ellos (**no vale elegir algunos muy parecidos**) y coméntalos con máximo detalle.

■ A entregar: ficheros de los 5 scripts solicitados y comentados, con nombre NumEjer-nombre.ps1

4. ACTIVIDAD 02

Realiza un script que pida un nombre de usuario y cree dicho usuario local sin contraseña. Puedes utilizar “New-LocalUser”.

■ A entregar: fichero con script solicitado, con comentarios, con nombre NumEjer-nombre.ps1

5. ACTIVIDAD 03

Realiza un script que pida un nombre de usuario y una contraseña. Tras ello debe crear dicho usuario local con esa contraseña. Puedes utilizar “New-LocalUser”.

■ A entregar: fichero con script solicitado, con comentarios, con nombre NumEjer-nombre.ps1

6. ACTIVIDAD 04

Realiza un script que pida un nombre de usuario local e indique si existe o no dicho usuario. Puedes usar para ello “Get-LocalUser”.

 A entregar: fichero con script solicitado, con comentarios, con nombre NumEjer-nombre.ps1

7. ACTIVIDAD 05

Realiza un script que pida un nombre de usuario local y si no existe de un error. En caso de existir, deberá desactivar la cuenta. Puedes usar para ello “Get-LocalUser” y “Disable-LocalUser”.

 A entregar: fichero con script solicitado, con comentarios, con nombre NumEjer-nombre.ps1

8. ACTIVIDAD 06

Realizar un script que pregunte cuantos directorios quieras crear (entre 0 y 999). Deberá crear los directorios con una estructura similar a “000”, “001”, “002”, etc. (con los ceros a la izquierda que sean necesarios). Los directorios los creará dentro de un subdirectorio “tmp” de la ruta donde ejecutes el PowerShell, para ser más fácil hacer pruebas, borrarlos, etc. Puedes utilizar “New-Item”.

 A entregar: fichero con script solicitado, con comentarios, con nombre NumEjer-nombre.ps1

9. ACTIVIDAD 07

Haz el ejercicio anterior, usando “do-while”.

 A entregar: fichero con script solicitado, con comentarios, con nombre NumEjer-nombre.ps1

10. ACTIVIDAD 08

Realizar un script que pregunte cuantos exámenes ha realizado el usuario.

Tras ello, pedirá las notas de esos exámenes. Una vez introducidas las notas, mostrará la nota media de los exámenes y también mostrará cuantos exámenes se han aprobado.

 A entregar: fichero con script solicitado, con comentarios, con nombre NumEjer-nombre.ps1

11. ACTIVIDAD 09

Realiza un script que mediante “Get-Process” nos permita saber el programa que más CPU está utilizando y el programa que más memoria está utilizando.

 A entregar: fichero con script solicitado, con comentarios, con nombre NumEjer-nombre.ps1

12. ACTIVIDAD 10

Realiza un script que mediante “Get-Process” nos permita saber el número de hilos del programa que más CPU está utilizando y el número de hilos del programa que más memoria está utilizando.

 A entregar: fichero con script solicitado, con comentarios, con nombre NumEjer-nombre.ps1

13. ACTIVIDAD 11

Realiza un script que mediante “Read-Host” y “Test-Connection” nos permita escanear una subred

(con máscara 255.255.255.0) para ver que IPs están en marcha en la red y cuáles no. Para funcionar se le indicará desde qué IP empieza y en qué IP acaba.

 **A entregar:** fichero con script solicitado, con comentarios, con nombre NumEjer-nombre.ps1

14. ACTIVIDAD 12

Crea un menú que funcione en bucle (es decir, que tras elegir una opción y terminarla, vuelve al principio). Este menú deberá implementar las siguientes opciones:

1. Ver listado de procesos.
2. Ver información detallada de un proceso.
3. Parar un proceso en marcha.
4. Ver el proceso que más CPU consume.
5. Ver el proceso que más memoria ocupa.
6. Salir.

 **A entregar:** fichero con script solicitado, con comentarios, con nombre NumEjer-nombre.ps1

15. ACTIVIDAD 13

Muestra como llenar un vector de 100 elementos en PowerShell con números aleatorios. Puede ayudarte “Get-Random”.

 **A entregar:** fichero con script solicitado, con comentarios, con nombre NumEjer-nombre.ps1

16. ACTIVIDAD 14

Siguiendo este tutorial y adaptándolo a Powershell:

<https://ugeek.github.io/blog/post/2019-03-14-crea-un-bot-de-telegram-con-bash-y-una-sola-linea-de-terminal.html> y sabiendo que el comando “curl” que se nombra en el tutorial está disponible en Windows 10 (y si no lo tiene, se puede instalar como aquí se indica <https://www.pchardwarepro.com/como-instalar-curl-en-windows-10/>).

Realiza el siguiente ejercicio:

Usa las tareas programadas de Windows para que cada 5 minutos envíe al bot de Telegram un mensaje con la RAM libre del sistema.

Puede ser útil utilizar:

`Get-CIMInstance Win32_OperatingSystem | Select FreePhysicalMemory`

 **A entregar:** fichero con script solicitado, con comentarios, con nombre NumEjer-nombre.ps1 y enseñar el bot funcionando al profesorado.

17. ACTIVIDAD 15

Siguiendo este tutorial del ejercicio anterior para crear bots de Telegram y complementando lo aprendido con

<https://ugeek.github.io/blog/post/2019-03-28-enviando-imagenes-audios-archivos-con-mi-bot-de-telegram-de-una-sola-linea-de-terminal.html>

Realiza el siguiente ejercicio:

Tienes un directorio con 10 imágenes llamadas “1.jpg”, “2.jpg”, ..., “10.jpg”. Haz un bot de Telegram que cada minuto envíe una imagen aleatoria de ese directorio. Puede ayudarte “Get-Random”.

 **A entregar:** fichero con script solicitado, con comentarios, con nombre NumEjer-nombre.ps1 y enseñar el bot funcionando al profesorado.