**Tema 3, Actividad 7**

# COMANDOS LINUX

**Ejercicios**

**1.- Preparar la estructura básica para el desarrollo de los ejercicios.**

A) Crear la misma estructura de directorios y archivos de la **Actividad 1** de las diapositivas.

B) Crear en el directorio prueba un archivo llamado archivo05.txt que contendrá el siguiente texto:

|  |
| --- |
| Tamara  Mar  Emma  Ava  Olivia  Mía  Sandra  Rosa  Blanca  Paloma  Natalia  Kelly  Maya  Esther  Ángela  Cecilia  Aurora  Rut  Teresa  Diana |

C) Crear en el directorio prueba un archivo llamado archivo06.txt que contendrá el siguiente texto:

|  |
| --- |
| 15. Ángela  17. Aurora  4. Ava  9. Blanca  16. Cecilia  20. Diana  3. Emma  14. Esther  12. Kelly  2. Mar  13. Maya  6. Mía  11. Natalia  5. Olivia  10. Paloma  8. Rosa  18. Rut  7. Sandra  1. Tamara  19. Teresa |

D) Crear en el directorio prueba un archivo llamado archivo07.txt que contendrá el siguiente texto:

Año,Marca,Modelo,Descripción,Precio

1997,Ford,E350,moon,3000.00

1999,Chevyr,Venture,Extended Edition,4900.00

1999,Chevy,Venture,Very Large,5000.00

1996,Jeep,Grand Cherokee,moon roof,4799.00

**2.- Comando sort - Escribe el comando adecuado para cada uno de los siguientes ejercicios:**

1. Ordenar el archivo05.txt por orden alfabético

|  |
| --- |
|  |

1. Ordenar el archivo05.txt por orden alfabético y guardar el resultado en un archivo llamado archivo05ordenado.txt

|  |
| --- |
|  |

1. Ordenar el archivo05.txt por orden alfabético inverso y guardar el resultado en el mismo archivo del ejercicio anterior, sin eliminar lo que tenía.

|  |
| --- |
|  |

1. Ordenar el archivo06.txt por orden numérico y guardar el resultado en un archivo llamado archivo06ordenado.txt

|  |
| --- |
|  |

1. Ordenar el archivo05ordenado.txt y eliminar los duplicados.

|  |
| --- |
|  |

1. Comprobar si el archivo06ordenado.txt está ordenado

|  |
| --- |
|  |

1. Ordenar el archivo07.txt por año

|  |
| --- |
|  |

1. Ordenar el archivo07.txt por orden alfabético de marca

|  |
| --- |
|  |

1. Ordenar el archivo07.txt por precio

|  |
| --- |
|  |

**3.- Comando cut - Escribe el comando adecuado para cada uno de los siguientes ejercicios:**

1. Extraer la 1ª letra de cada uno de los nombres del archivo05.txt

|  |
| --- |
|  |

1. Extraer los 3 primeros caracteres de cada uno de los nombres del archivo05.txt

|  |
| --- |
|  |

1. Extraer la columna marca del archivo07.txt

|  |
| --- |
|  |

1. Extraer la columna precio del archivo07.txt

|  |
| --- |
|  |

1. Extraer la columna precio del archivo07.txt y ordena por precio de menor a mayor

|  |
| --- |
|  |

1. Extraer la columna precio del archivo07.txt y ordena por precio de mayor a menor y guarda el resultado en el fichero archivo07precio.txt

|  |
| --- |
|  |

1. Extraer la marca y el precio del archivo07.txt

|  |
| --- |
|  |

1. Extraer la marca y el precio del archivo07.txt y ordenar el resultado por precio

|  |
| --- |
|  |

1. Extraer el nombre, el teléfono y el importe del archivo03.txt

|  |
| --- |
|  |

1. Extraer el nombre, el teléfono y el importe del archivo03.txt, ordenar el resultado por importe y guarda el resultado en el archivo03importe.txt

|  |
| --- |
|  |

1. Crear un nuevo archivo llamado: 'empleados.txt' cuyo contenido sean los campos: apellidos y nombre del archivo03.txt

|  |
| --- |
|  |

1. A partir del archivo: 'empleados.txt', obtener el archivo: 'lista\_emp.txt' cuyo contenido debe estar clasificado alfabéticamente por apellidos.

|  |
| --- |
|  |

1. Obtener una lista ordenada alfabéticamente de todos los usuarios del sistema (usa el archivo /etc/passwd)

|  |
| --- |
|  |

1. Obtener una lista ordenada alfabéticamente de todos los usuarios del sistema y el tipo de Shell que usan (1ª y última columna del archivo /etc/passwd)

|  |
| --- |
|  |

1. Obtener el nombre de los 6 primeros usuarios del sistema. (Ayuda: usa el comando cut y head)

|  |
| --- |
|  |

1. Obtener el nombre de los 6 últimos usuarios del sistema. (Ayuda: usa el comando cut y tail)

|  |
| --- |
|  |

# SCRIPTS LINUX

1. Realizar un script llamado **comparalineas.sh** al que se le pasarán dos ficheros como argumentos y se comparará el nº de líneas del primer fichero y el segundo fichero:

* Si el nº de líneas del 1er fichero es **menor** que el nº de líneas del 2º fichero dirá:

“El fichero1.txt con 7 líneas es **menor** que el fichero2.txt con 43 líneas”

* Si el nº de líneas del 1er fichero es **mayor** que el nº de líneas del 2º fichero dirá:

“El fichero3.txt con 56 líneas es **mayor** que el fichero4.txt con 10 líneas”

* Si el nº de líneas del 1er fichero es **igual** que el nº de líneas del 2º fichero dirá:

“Los ficheros fichero5.txt y fichero6.txt son **iguales** y tienen 43 líneas”

**Comprobaciones previas:**

* Comprobar si hay 2 argumentos
* Comprobar si los 2 ficheros existen. Si alguno de los dos no existe dará un mensaje de error.

1. Realizar un script llamado **calculadora1.sh** para realizar una calculadora de sumas, restas, multiplicaciones y divisiones. El script pedirá por pantalla 2 números y la operación a realizar. Se realizará la operación y se mostrará por pantalla.
2. Realizar un script llamado **calculadora2.sh** modificar la calculadora del ejercicio 2 para que pregunte si quieres hacer más operaciones: **¿Quiere hacer más operaciones? (s para continuar) 🡪 Continuará si pulsa s y saldrá en cualquier otro caso.**