



01 – Bash scripting

TRABAJO DE:
Sofía López del Sastre

Índice

APUNTES DE BASH	3
EJERCICIO 1:	4
EJERCICIO 2:	5
EJERCICIO 3:	5
EJERCICIO 4:	6
EJERCICIO 5:	6
EJERCICIO 6:	7
EJERCICIO 7:	8
EJERCICIO 8:	8
EJERCICIO 9:	8
EJERCICIO 10:	9
EJERCICIO 11:	9
EJERCICIO 12:	9
EJERCICIO 13:	10
EJERCICIO 14:	11
EJERCICIO 15:	12
EJERCICIO 16:	13
EJERCICIO 17:	13
EJERCICIO 18:	14
EJERCICIO 19:	15
EJERCICIO 20:	15

APUNTES DE BASH

- Para encadenar un comando con otro = (primer comando) | (segundo comando)
- Para meter en un archivo aquello que has hecho = (comando) (archivo en dónde lo quieres hacer) > (documento en donde quieres guardarlo)
- Manual de comandos = man “comando”
- rm *.c = borra todos los archivos que acaben en .c
- LS:
 - ls (te muestra todos los archivos que hay)
 - ls *.c (te muestra todos los archivos acabados en .c)
 - ls -l (recoge todos los archivos de la carpeta)
- GREP logs.txt:
 - grep “palabra” (busca las palabras en el archivo e imprime la frase entera)
 - grep -v “palabra” (busca todas las palabras excepto la dada)
 - grep -c “palabra” (cuenta el número de palabras que hay)
 - grep -o “palabra” (busca solo las palabras que hay en el archivo)
 - grep “^palabra” (busca las palabras que estén al principio de la frase en el archivo e imprime la frase entera)
 - grep “palabra\$” (busca las palabras que estén al final de la frase en el archivo e imprime la frase entera)
 - grep “^palabra\$” (busca las palabras que estén al principio y al final de la frase en el archivo e imprime la frase entera)
 - grep “^\$” (frases en blanco)
 - grep -E “[buscar caracteres]”
 - grep ‘\b[0-9]*[02468]\b’ (la \b es para buscar cosas en el grep, pero el número entero, no a la mitad, es decir, si lo buscas normal por ejemplo un 2, te puede buscar un 21, que sería impar entonces si lo haces con el \b, te busca solo los números que tienen 2)
 - grep -r “palabra” . (el -r busca recursivamente en los directorios y el punto indica que la búsqueda comienza en el directorio actual)
- HEAD / TAIL logs.txt:
 - head (imprime las primeras frases del archivo)
 - head -n 2 (imprime las 2 primeras frases del archivo)
 - tail (igual pero en vez de por el principio, por el final)
- CAT logs.txt (muestra lo que hay dentro del archivo)
- SORT logs.txt (para ordenar el texto)
- UNIQU (te quita las frases repetidas varias veces seguidas)
- EDITORES DE TEXTO:
 - LESS logs.txt (editor de texto)
 - NANO logs.txt (para cambiar el archivo)
- SED
 - sed ‘s/palabra_buscada/palabra_reemplazar/g’ archivo
 - Pero, a su vez, por ejemplo en el ejercicio 16 pongo en primer lugar la palabra que quiero buscar en el código, y se me cogen las frases que tenga esa palabra, luego se me va a buscar (/s es de search), al principio de la frase (/^), se pone un modificado (sed '/error/s/^/MODIFICADO: /' file1.txt > output 16.txt)
- WC
 - wc -l (cuenta numéricamente todos los archivos del directorio en el que estás)
 - find | wc -l (este comando se encarga de encontrar archivos y directorios dentro de un directorio)

EJERCICIO 1:

Encuentra todas las líneas que terminen con "error" en el archivo `file1.txt` y guarda el resultado en `output1.txt`.

Código: `grep "error$" file1.txt > output1.txt`

```
c-fgs@Info1-Pc15:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ grep "error$" file1.txt
línea 1 con error
línea 2 con error
línea 3 con error
línea 4 con error
línea 5 con error
línea 6 con error
línea 7 con error
línea 8 con error
línea 9 con error
línea 11 con error
línea 12 con error
línea 13 con error
línea 14 con error
línea 15 con error
línea 16 con error
línea 17 con error
línea 18 con error
línea 19 con error
línea 21 con error
línea 22 con error
línea 23 con error
línea 24 con error
línea 25 con error
línea 26 con error
línea 27 con error
línea 28 con error
línea 29 con error
línea 31 con error
línea 32 con error
línea 33 con error
línea 34 con error
línea 35 con error
línea 36 con error
línea 37 con error
línea 38 con error
línea 39 con error
línea 41 con error
línea 42 con error
línea 43 con error
línea 44 con error
línea 45 con error
línea 46 con error
línea 47 con error
línea 48 con error
línea 49 con error
```

EJERCICIO 2:

Reemplaza todas las ocurrencias de "DEBUG" por "INFO" en el archivo `file1.txt` y guarda el resultado en `output2.txt`.

Código: `sed 's/DEBUG/INFO/g' file1.txt > output2.txt`

```
cfigs@Info1-Pc15:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ sed 's/DEBUG/INFO/g' file1.txt
Línea 1 con error
Línea 2 con error
Línea 3 con error
Línea 4 con error
Línea 5 con error
Línea 6 con error
Línea 7 con error
Línea 8 con error
Línea 9 con error
INFO: Línea de prueba 10
Línea 11 con error
Línea 12 con error
Línea 13 con error
Línea 14 con error
Línea 15 con error
Línea 16 con error
Línea 17 con error
Línea 18 con error
Línea 19 con error
INFO: Línea de prueba 20
Línea 21 con error
Línea 22 con error
Línea 23 con error
Línea 24 con error
Línea 25 con error
Línea 26 con error
Línea 27 con error
Línea 28 con error
Línea 29 con error
INFO: Línea de prueba 30
Línea 31 con error
Línea 32 con error
Línea 33 con error
Línea 34 con error
Línea 35 con error
Línea 36 con error
Línea 37 con error
Línea 38 con error
Línea 39 con error
INFO: Línea de prueba 40
Línea 41 con error
Línea 42 con error
Línea 43 con error
Línea 44 con error
Línea 45 con error
Línea 46 con error
Línea 47 con error
Línea 48 con error
Línea 49 con error
cfigs@Info1-Pc15:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ sed 's/DEBUG/INFO/g' file1.txt > output2.txt
```

EJERCICIO 3:

Muestra las primeras 5 líneas de `file1.txt` que contengan "ERROR" y guarda el resultado en `output3.txt`.

Código: `head -n 5 file1.txt | grep "error" > output3.txt`

```
cfigs@Info1-Pc15:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ head -n 5 file1.txt | grep "error"
Línea 1 con error
Línea 2 con error
Línea 3 con error
Línea 4 con error
Línea 5 con error
cfigs@Info1-Pc15:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ head -n 5 file1.txt | grep "error" > output3.txt
```

EJERCICIO 4:

Cuenta el número de archivos `.txt` en el directorio actual y guarda el resultado en `output4.txt`.

Código: `ls *.txt | wc -l > output4.txt`

```
sofia@Tairn:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ ls *.txt | wc -l
25
sofia@Tairn:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ ls *.txt | wc -l > output4.txt
```

EJERCICIO 5:

Ordena las líneas del archivo `file1.txt` alfabéticamente, elimina duplicados, y guarda el resultado en `output5.txt`.

Código: `sort file1.txt | uniq > output5.txt`

```
cfgs@Info1-Pc15:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ sort file1.txt | uniq
DEBUG: Línea de prueba 10
DEBUG: Línea de prueba 20
DEBUG: Línea de prueba 30
DEBUG: Línea de prueba 40
DEBUG: Línea de prueba 50
Línea 1 con error
Línea 11 con error
Línea 12 con error
Línea 13 con error
Línea 14 con error
Línea 15 con error
Línea 16 con error
Línea 17 con error
Línea 18 con error
Línea 19 con error
Línea 2 con error
Línea 21 con error
Línea 22 con error
Línea 23 con error
Línea 24 con error
Línea 25 con error
Línea 26 con error
Línea 27 con error
Línea 28 con error
Línea 29 con error
Línea 3 con error
Línea 31 con error
Línea 32 con error
Línea 33 con error
Línea 34 con error
Línea 35 con error
Línea 36 con error
Línea 37 con error
Línea 38 con error
Línea 39 con error
Línea 4 con error
Línea 41 con error
Línea 42 con error
Línea 43 con error
Línea 44 con error
Línea 45 con error
Línea 46 con error
Línea 47 con error
Línea 48 con error
Línea 49 con error
Línea 5 con error
Línea 6 con error
Línea 7 con error
Línea 8 con error
Línea 9 con error
```

EJERCICIO 6:

Divide el archivo `file6_large.txt` en dos archivos: uno con las primeras 100 líneas (`out6.1.txt`) y otro con el resto (`out.6.txt`).

Código:

```
head -n 100 file6_large.txt > out6.1.txt
```

```
tail -n 100 file6_large.txt > out6.txt
```

```
sofia@Tairn:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ head -n 100 file6_large.txt
Línea 1
Línea 2
Línea 3
Línea 4
Línea 5
Línea 6
Línea 7
Línea 8
Línea 9
Línea 10
Línea 11
Línea 12
Línea 13
Línea 14
Línea 15
Línea 16
Línea 17
Línea 18
Línea 19
Línea 20
Línea 21
Línea 22
Línea 23
Línea 24
Línea 25
Línea 26
Línea 27
Línea 28
Línea 29
Línea 30
Línea 31
Línea 32
Línea 33
Línea 34
Línea 35
```

```
Línea 84
Línea 85
Línea 86
Línea 87
Línea 88
Línea 89
Línea 90
Línea 91
Línea 92
Línea 93
Línea 94
Línea 95
Línea 96
Línea 97
Línea 98
Línea 99
Línea 100
cfigs@Info1-Pc15:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ tail -n 100 file6_large.txt > out6
.txt
cfigs@Info1-Pc15:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ tail -n 100 file6_large.txt
Línea 101
Línea 102
Línea 103
Línea 104
Línea 105
Línea 106
Línea 107
Línea 108
Línea 109
Línea 110
Línea 111
Línea 112
Línea 113
Línea 114
Línea 115
Línea 116
Línea 117
Línea 118
Línea 119
Línea 120
Línea 121
```

EJERCICIO 7:

Elimina todas las líneas en blanco del archivo `file4.txt` y guarda el resultado en `output7.txt`.

Código: `grep "^$" file4.txt | uniq > output7.txt`

```
cfgs@Info1-Pc15:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ grep "^$" file4.txt | uniq
cfgs@Info1-Pc15:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ grep "^$" file4.txt | uniq > output7.txt
```

EJERCICIO 8:

Reemplaza todas las comas por tabulaciones en el archivo `file2.csv` y guarda el resultado en `output8.csv`.

Código: `sed 's/,/\t/g' file2.csv > output8.csv`

```
cfgs@Info1-Pc15:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ sed 's/,/\t/g' file2.csv
nombre edad      ciudad
Persona1      21      Ciudad1
Persona2      22      Ciudad2
Persona3      23      Ciudad3
Persona4      24      Ciudad4
Persona5      25      Ciudad5
Persona6      26      Ciudad6
Persona7      27      Ciudad7
Persona8      28      Ciudad8
Persona9      29      Ciudad9
Persona10     30      Ciudad10
Persona11     31      Ciudad11
Persona12     32      Ciudad12
Persona13     33      Ciudad13
Persona14     34      Ciudad14
Persona15     35      Ciudad15
Persona16     36      Ciudad16
Persona17     37      Ciudad17
Persona18     38      Ciudad18
Persona19     39      Ciudad19
Persona20     40      Ciudad20
Persona21     41      Ciudad21
Persona22     42      Ciudad22
Persona23     43      Ciudad23
Persona24     44      Ciudad24
Persona25     45      Ciudad25
Persona26     46      Ciudad26
Persona27     47      Ciudad27
Persona28     48      Ciudad28
Persona29     49      Ciudad29
cfgs@Info1-Pc15:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ sed 's/,/\t/g' file2.csv > output8.csv
```

EJERCICIO 9:

Inserta el prefijo "INICIO:" antes de cada línea que comience con una vocal en el archivo `file5.txt` y guarda el resultado en `output9.txt`.

Código: `sed '/^[aeiouAEIOU]/s/^/INICIO:/' file5.txt > output9.txt`

```
cfgs@Info1-Pc15:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ sed '/^[AEIOUaeiou]/ s/^/INICIO: /' file5.txt
INICIO: Una línea 1
INICIO: Otra línea 2
INICIO: Ejemplo línea 3
INICIO: Otra línea 4cfgs@Info1-Pc15:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ sed '/^[AEIOUaeiou]/ s/^/INICIO: /'
' file5.txt > output9.txt
```


EJERCICIO 10:

Cuenta cuántas líneas en el archivo `file1.txt` contienen la palabra "ERROR" y guarda el resultado en `output10.txt`.

Código: `grep -c "error" file1.txt > output10.txt`

```
cfgs@Info1-Pc15:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ grep -c "error" file1.txt
45
cfgs@Info1-Pc15:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ grep -c "error" file1.txt > output10.txt
```

EJERCICIO 11:

Crea un script que muestre el número total de archivos en el directorio actual y escribe el resultado en `output11.txt`.

Código: `ls | wc -l > output11.txt`

```
cfgs@Info1-Pc15:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ ls | wc -l
21
```

EJERCICIO 12:

Encuentra líneas duplicadas en el archivo `file7_duplicates.txt`, muestra cuántas veces se repiten, y guarda el resultado en `output12.txt`.

Código: `sort file7_duplicates.txt | uniq -c | sort -n > output12.txt`

```
cfgs@Info1-Pc15:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ sort file7_duplicates.txt | uniq
-c | sort -n
  1 Línea única
 15 Línea repetida
```

EJERCICIO 13:

Extrae las primeras 100 líneas del archivo `file6_large.txt` y guarda el resultado en `output13.txt`.

Código: `head -n 100 file6_large.txt > output13.txt`

```
cfigs@Info1-Pc15:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ head -n 100 file6_large.txt
Línea 1
Línea 2
Línea 3
Línea 4
Línea 5
Línea 6
Línea 7
Línea 8
Línea 9
Línea 10
Línea 11
Línea 12
Línea 13
Línea 14
Línea 15
Línea 16
Línea 17
Línea 18
Línea 19
Línea 20
Línea 21
Línea 22
Línea 23
Línea 24
Línea 25
Línea 26
Línea 27
Línea 28
Línea 29
Línea 30
Línea 31
Línea 32
Línea 33
Línea 34
Línea 35
Línea 36
Línea 37
Línea 38
Línea 39
Línea 40
```

EJERCICIO 14:

Busca todas las líneas que contengan números en el archivo `file1.txt` y guarda el resultado en `output14.txt`.

Código: `grep ["0-9"] file1.txt > output14.txt`

```
cfigs@Info1-Pc15:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ grep ["0-9"] file1.txt
Línea 1 con error
Línea 2 con error
Línea 3 con error
Línea 4 con error
Línea 5 con error
Línea 6 con error
Línea 7 con error
Línea 8 con error
Línea 9 con error
DEBUG: Línea de prueba 10
Línea 11 con error
Línea 12 con error
Línea 13 con error
Línea 14 con error
Línea 15 con error
Línea 16 con error
Línea 17 con error
Línea 18 con error
Línea 19 con error
DEBUG: Línea de prueba 20
Línea 21 con error
Línea 22 con error
Línea 23 con error
Línea 24 con error
Línea 25 con error
Línea 26 con error
Línea 27 con error
Línea 28 con error
Línea 29 con error
DEBUG: Línea de prueba 30
Línea 31 con error
Línea 32 con error
Línea 33 con error
Línea 34 con error
Línea 35 con error
Línea 36 con error
Línea 37 con error
Línea 38 con error
Línea 39 con error
DEBUG: Línea de prueba 40
Línea 41 con error
Línea 42 con error
Línea 43 con error
Línea 44 con error
Línea 45 con error
Línea 46 con error
Línea 47 con error
Línea 48 con error
Línea 49 con error
DEBUG: Línea de prueba 50
```

EJERCICIO 15:

Convierte el archivo `file2.csv` en un archivo delimitado por punto y coma y guarda el resultado en `output15.csv`.

Código: `sed 's/,;/g' file2.csv > output15.csv`

```
sofia@Tairn:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ sed 's/,;/g' file2.csv
nombre;edad;ciudad
Persona1;21;Ciudad1
Persona2;22;Ciudad2
Persona3;23;Ciudad3
Persona4;24;Ciudad4
Persona5;25;Ciudad5
Persona6;26;Ciudad6
Persona7;27;Ciudad7
Persona8;28;Ciudad8
Persona9;29;Ciudad9
Persona10;30;Ciudad10
Persona11;31;Ciudad11
Persona12;32;Ciudad12
Persona13;33;Ciudad13
Persona14;34;Ciudad14
Persona15;35;Ciudad15
Persona16;36;Ciudad16
Persona17;37;Ciudad17
Persona18;38;Ciudad18
Persona19;39;Ciudad19
Persona20;40;Ciudad20
Persona21;41;Ciudad21
Persona22;42;Ciudad22
Persona23;43;Ciudad23
Persona24;44;Ciudad24
Persona25;45;Ciudad25
Persona26;46;Ciudad26
Persona27;47;Ciudad27
Persona28;48;Ciudad28
Persona29;49;Ciudad29
Persona30;50;Ciudad30
```

EJERCICIO 16:

Inserta un prefijo "MODIFICADO:" antes de las líneas que contengan la palabra "ERROR" en el archivo `file1.txt` y guarda el resultado en `output16.txt`.

Código: `sed '/error/s/^/MODIFICADO: /' file1.txt > output16.txt`

```
cfigs@Info1-Pc15:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ sed '/error/s/^/MODIFICADO: /' file1.txt
MODIFICADO: Línea 1 con error
MODIFICADO: Línea 2 con error
MODIFICADO: Línea 3 con error
MODIFICADO: Línea 4 con error
MODIFICADO: Línea 5 con error
MODIFICADO: Línea 6 con error
MODIFICADO: Línea 7 con error
MODIFICADO: Línea 8 con error
MODIFICADO: Línea 9 con error
DEBUG: Línea de prueba 10
MODIFICADO: Línea 11 con error
MODIFICADO: Línea 12 con error
MODIFICADO: Línea 13 con error
MODIFICADO: Línea 14 con error
MODIFICADO: Línea 15 con error
MODIFICADO: Línea 16 con error
MODIFICADO: Línea 17 con error
MODIFICADO: Línea 18 con error
MODIFICADO: Línea 19 con error
DEBUG: Línea de prueba 20
MODIFICADO: Línea 21 con error
MODIFICADO: Línea 22 con error
MODIFICADO: Línea 23 con error
MODIFICADO: Línea 24 con error
MODIFICADO: Línea 25 con error
MODIFICADO: Línea 26 con error
MODIFICADO: Línea 27 con error
MODIFICADO: Línea 28 con error
MODIFICADO: Línea 29 con error
DEBUG: Línea de prueba 30
MODIFICADO: Línea 31 con error
MODIFICADO: Línea 32 con error
MODIFICADO: Línea 33 con error
MODIFICADO: Línea 34 con error
MODIFICADO: Línea 35 con error
MODIFICADO: Línea 36 con error
MODIFICADO: Línea 37 con error
MODIFICADO: Línea 38 con error
MODIFICADO: Línea 39 con error
DEBUG: Línea de prueba 40
MODIFICADO: Línea 41 con error
MODIFICADO: Línea 42 con error
MODIFICADO: Línea 43 con error
MODIFICADO: Línea 44 con error
MODIFICADO: Línea 45 con error
MODIFICADO: Línea 46 con error
MODIFICADO: Línea 47 con error
MODIFICADO: Línea 48 con error
MODIFICADO: Línea 49 con error
cfigs@Info1-Pc15:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ |
```

EJERCICIO 17:

Extrae las líneas del archivo `file3.txt` que contengan números pares y guarda el resultado en `output17.txt`.

Código: `grep '\b[0-9]*[02468]\b' file3.txt > output.txt`

```
cfigs@Info1-Pc15:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ grep '\b[0-9]*[02468]\b' file3.txt
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
cfigs@Info1-Pc15:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ grep '\b[0-9]*[02468]\b' file3.txt > output17.txt
```

EJERCICIO 18:

Busca recursivamente en todos los subdirectorios del directorio actual archivos que contengan la palabra "ERROR" y guarda las coincidencias en `output18.txt`.

Código: `grep -r "error" . > output18.txt`

```
sofia@Tairn:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ grep -r "error" .
./output14.txt:Línea 1 con error
./output14.txt:Línea 2 con error
./output14.txt:Línea 3 con error
./output14.txt:Línea 4 con error
./output14.txt:Línea 5 con error
./output14.txt:Línea 6 con error
./output14.txt:Línea 7 con error
./output14.txt:Línea 8 con error
./output14.txt:Línea 9 con error
./output14.txt:Línea 11 con error
./output14.txt:Línea 12 con error
./output14.txt:Línea 13 con error
./output14.txt:Línea 14 con error
./output14.txt:Línea 15 con error
./output14.txt:Línea 16 con error
./output14.txt:Línea 17 con error
./output14.txt:Línea 18 con error
./output14.txt:Línea 19 con error
./output14.txt:Línea 21 con error
./output14.txt:Línea 22 con error
./output14.txt:Línea 23 con error
./output14.txt:Línea 24 con error
./output14.txt:Línea 25 con error
./output14.txt:Línea 26 con error
./output14.txt:Línea 27 con error
./output14.txt:Línea 28 con error
./output14.txt:Línea 29 con error
./output14.txt:Línea 31 con error
./output14.txt:Línea 32 con error
./output14.txt:Línea 33 con error
./output14.txt:Línea 34 con error
./output14.txt:Línea 35 con error
./output14.txt:Línea 36 con error
./output14.txt:Línea 37 con error
./output14.txt:Línea 38 con error
./output14.txt:Línea 39 con error
./output14.txt:Línea 41 con error
./output14.txt:Línea 42 con error
./output14.txt:Línea 43 con error
./output14.txt:Línea 44 con error
./output14.txt:Línea 45 con error
./output14.txt:Línea 46 con error
./output14.txt:Línea 47 con error
./output14.txt:Línea 48 con error
./output14.txt:Línea 49 con error
./file1.txt:Línea 1 con error
./file1.txt:Línea 2 con error
./file1.txt:Línea 3 con error
./file1.txt:Línea 4 con error
./file1.txt:Línea 5 con error
./file1.txt:Línea 6 con error
./file1.txt:Línea 7 con error
./file1.txt:Línea 8 con error
./file1.txt:Línea 9 con error
./file1.txt:Línea 11 con error
```

EJERCICIO 19:

Extrae las últimas 50 líneas del archivo `file6_large.txt` y guarda el resultado en `output19.txt`.

Código: `tail -n 50 file6_large.txt > output19.txt`

```
sofia@Tairn:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ tail -n 50 file6_large.txt
Línea 151
Línea 152
Línea 153
Línea 154
Línea 155
Línea 156
Línea 157
Línea 158
Línea 159
Línea 160
Línea 161
Línea 162
Línea 163
Línea 164
Línea 165
Línea 166
Línea 167
Línea 168
Línea 169
Línea 170
Línea 171
Línea 172
Línea 173
Línea 174
Línea 175
Línea 176
Línea 177
Línea 178
Línea 179
Línea 180
Línea 181
Línea 182
Línea 183
Línea 184
Línea 185
Línea 186
Línea 187
Línea 188
Línea 189
Línea 190
Línea 191
Línea 192
Línea 193
Línea 194
Línea 195
Línea 196
Línea 197
Línea 198
Línea 199
```

EJERCICIO 20:

Lista todos los archivos, incluidos los ocultos, de forma recursiva desde el directorio actual y guarda el resultado en `output20.txt`.

Código: `find | wc -l > output20.txt`

```
sofia@Tairn:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ find | wc -l
39
sofia@Tairn:~/DAM/Sistemas/Bash/Ejercicios/Ejercicios a entregar$ find | wc -l > output20.txt
```