

Informe Lab1 Sistemes Operatius

Aran Roig i Biel Palomar

Febrer 2025



1 Resum

Aquesta pràctica ha consistit en el desenvolupament de quatre scripts en Bash per automatitzar tasques de monitoratge i anàlisi de dades mediambientals. S'han tractat problemes com la gestió de temperatures, la classificació de la qualitat de l'aire, l'anàlisi del consum energètic i el càlcul d'emissions de CO^2

Mitjançant la manipulació d'arxius, la lectura d'arguments i el control d'errors, s'han aplicat conceptes fonamentals de Bash scripting. Aquest treball ha permès consolidar habilitats en automatització i anàlisi de dades, destacant la seva rellevància en el context del canvi climàtic i els Objectius de Desenvolupament Sostenible.

2 Exercicis

2.1 Exercici 1

Primer hem comprovat que si no hi ha cap argument aleshores mostri el format d'execució del script.

```
oslab@oslab:/media/sf_shared/SistemasOperatius/lab_p1$ ./ejercicio1.sh
Se requiere el siguiente formato: ./ejercicio1.sh <archivo_ciudad>
```

Un cop hem assegurat que tenim més d'un argument, iterem per a cada argument i comprovem que l'argument és un fitxer, i si ho és, mostrem el tamany del fitxer, agafant la cinquena columna de la sortida de `ls -l` amb una pipe cap a la comanda `awk` i l'afegim a la variable `$total`

Finalment mostrem la mida de tots els fitxers en bytes, mostrant la variable `$total`

```
oslab@oslab:/media/sf_shared/SistemasOperatius/lab_p1$ ./ejercicio1.sh barcelona-temp.txt
Tamaño del archivo barcelona-temp.txt: 806 bytes
Tamaño del archivo combinado: 806 bytes
```

2.2 Exercici 2

En aquest exercici primer hem comprovat que si es passen menys de dos arguments el script mostri el seu format d'execució:

```
oslab@oslab:/media/sf_shared/SistemasOperatius/lab_p1$ ./ejercicio2.sh
Formato de uso: ./ejercicio2.sh <carpeta_datos> <umbral>
oslab@oslab:/media/sf_shared/SistemasOperatius/lab_p1$ ./ejercicio2.sh datos_aire/
Formato de uso: ./ejercicio2.sh <carpeta_datos> <umbral>
```

Després comprovem que efectivament el primer element que hem passat sigui un directori:

```
oslab@oslab:/media/sf_shared/SistemasOperatius/lab_p1$ ./ejercicio2.sh resumen_calidad_aire.txt 3
resumen_calidad_aire.txt no es un directorio
```

I seguidament comprovem que el segon argument sigui efectivament un nombre. Això ho fem veient si la expressió regular `^[+-]?[0-9]+([.][0-9]+)?` accepta el segon argument

```
oslab@oslab:/media/sf_shared/SistemasOperatius/lab_p1$ ./ejercicio2.sh datos_aire/ dsada
dsada no es un nombre real
```

Un cop comprovat tot això el script compara la primera línia de cada fitxer i compara els dos nombres utilitzant `bc`, els va sumant a variables i mou els fitxers segons el resultat de la comparació a les carpetes `normal` i `alerta`. En cas que no existeixin els directoris es creen utilitzant `mkdir -p normal alerta`. L'argument `-p` es per fer que en cas que existeixin no les creïn, ja que només ens interessa moure els fitxers dins de les

carpetes, en cas que existeixin. Finalment mostra l'informe per pantalla i el mateix que mostra acaba escrit també al fitxer `resumen_calidad_aire.txt`

```
oslab@oslab:/media/sf_shared/SistemasOperatius/lab_pi$ ./ejercicio2.sh datos_aire/ 25.0
Procesando archivos de calidad del aire...
Archivos procesados: 3
Alertas detectadas: 2
Archivos normales: 1
Se ha generado el informe en resumen_calidad_aire.txt
```

2.3 Exercici 3

En aquest exercici primer de tot mirem que hi hagi els 8 paràmetres necessaris, si no es així tanquem el programa i avisem l'usuari.

```
oslab@oslab:/media/sf_Dades/SistemasOperatius/lab_pi$ ./ejercicio3.sh 14 245 256 234 267 278 289
Nombre de parametros incorrecto: ./ejercicio3.sh <hora> <dia1> <dia2> <dia3> <dia4> <dia5> <dia6> <dia7>
```

Seguidament mirem que l'hora estigui entre 0 i 24, i si no es així avortem amb code 1.

```
oslab@oslab:/media/sf_Dades/SistemasOperatius/lab_pi$ ./ejercicio3.sh 26 245 256 234 267 278 289 292
La hora ha de estar en format 24h (0-23)
```

Tot seguit guardem l'hora a la variable `$hora` i fem un `shift` per a esborrar el paràmetre. Llavors imprimim la primera línia que diu la hora a l'usuari.

Fem un for per imprimir cada dia, i dins de cada iteració fem un altre bucle per a imprimir les barres. Ho hem imprès amb l'opció `-e` per a que llegeixi ASCII.

Finalment mirem la tendència comparant el primer amb l'últim.

2.4 Exercici 4

A l'exercici4 primer de tot hem tornat a comprovar que hi hagués el nombre de paràmetres correcte. Si no és així avisem a l'usuari.

```
oslab@oslab:/media/sf_Dades/SistemasOperatius/lab_pi$ ./ejercicio4.sh tren
El número de parámetros es incorrecto: ./ejercicio4.sh <medio> <distancia>
```

Seguidament hem mirat que haguessin posat un mètode de transport dins de les seleccions (Cotxe, tren, bus o bici). Això ho fem amb un case, i si no és cap d'aquests sortim amb un exit 1.

```
oslab@oslab:/media/sf_Dades/SistemasOperatius/lab_pi$ ./ejercicio4.sh hola 23
El medio proporcionado no es correcto, opciones: coche bus tren bici
```

De ser tot correcte farem els càlculs de consum del cotxe i el mètode seleccionat i ensenyarem per pantalla el que demana l'enunciat. Per a fer els càlculs amb decimals hem fet servir la comanda `awk` de la següent manera:

```
echo "$emision" | awk '{print $1 /1000}'
```

3 Conclusions i valoracions personals

En conclusió, aquesta pràctica ens ha servit per a interioritzar els fonaments de bash i com fer uns scripts bàsics que siguin d'utilitat.

Personalment trobem que és una pràctica molt útil i de nivell adequat, tot i que el fet de només poder fer servir les comandes donades al tutorial és limitant per fer un millor codi.

4 Contribució

Aran Roig Serra 50%

Biel Palomar González 50%