Gestión de información
Búsquedas y tratamiento

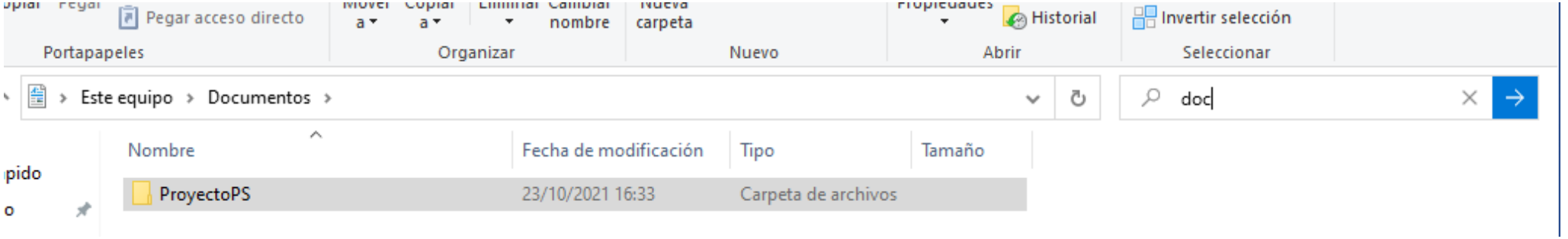
El tratamiento de la información

- Veremos a continuación las funcionalidades proporcionadas por los sistemas para la búsqueda de información.
- En el caso de Linux, profundizaremos sobre las herramientas disponibles por el sistema para su tratamiento.

Windows

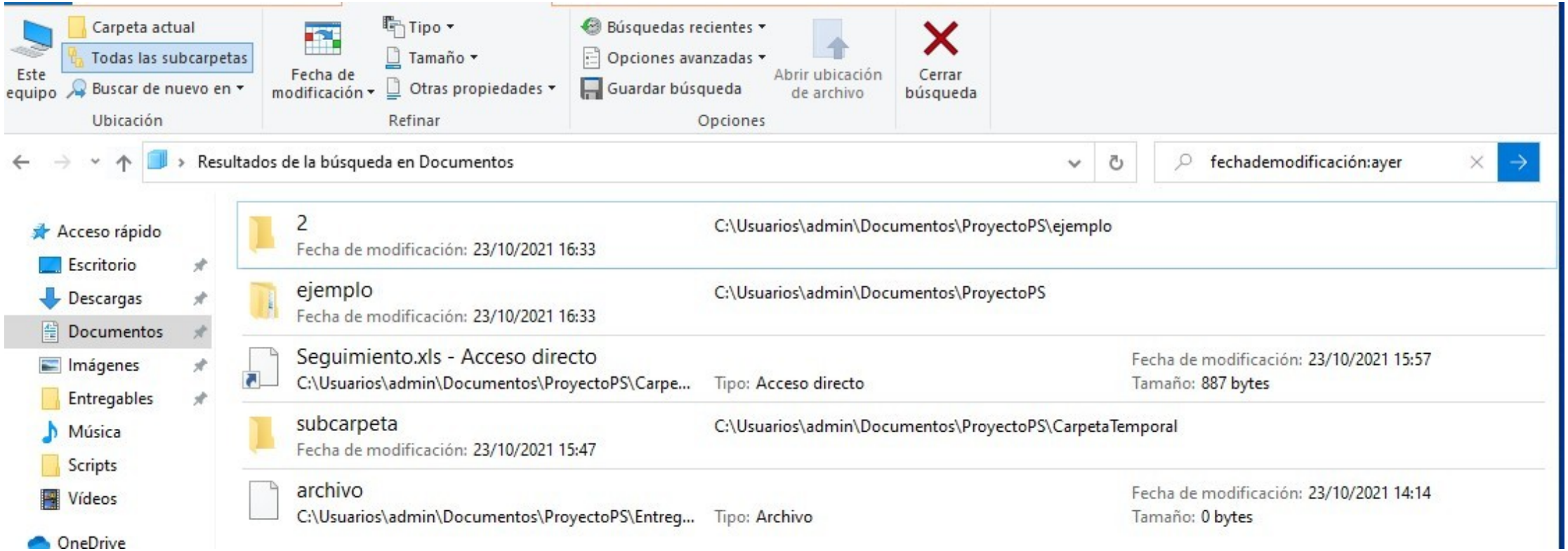
Windows - Búsquedas gráficas

- Windows dispone de un interfaz gráfico muy potente para realizar búsquedas de información en el sistema.
- Por una parte, si queremos buscar por nombre de archivo, y sabemos su nombre aproximado, podemos utilizar el buscador inferior.
- Otra alternativa, más potente es el propio **Explorador de archivos**



Windows - Búsquedas gráficas (II)

- Al realizar una primera búsqueda, se abre la herramienta de búsqueda para refinarla



Windows - Búsquedas gráficas (III)

- Se trata de una herramienta muy potente que permite refinar búsquedas por:
  - Fecha modificación, creación
  - Tamaño
  - Tipo de archivo
  - Contenido del archivo
- Las condiciones se pueden combinar mediante NOT, OR y/o AND.

Windows - Búsquedas línea de comandos

- Podemos utilizar el comando DIR para buscar por línea de comandos archivos de los que conozcamos parte de su nombre.
- Por ejemplo, para buscar desde la carpeta de trabajo los archivos de texto que comiencen por “fi” y que pagine los resultados:
  - DIR “tr*.pdf” /S /P

```
C:\Users\admin>dir "fi*.txt" /S /P
 El volumen de la unidad C no tiene etiqueta.
 El número de serie del volumen es: 08D0-C1E3

 Directorio de C:\Users\admin\Documents\ProyectoPS\Entregables

24/10/2021  12:22                38 fichero1.txt
               1 archivos             38 bytes

    Total de archivos en la lista:
               1 archivos             38 bytes
               0 dirs  24.127.098.880 bytes libres
```

Windows - Búsquedas línea de comandos

- Windows dispone también de comandos de búsqueda para buscar texto dentro de archivos:
  - Buscar cadenas de texto dentro de uno o más archivos

    FIND [/V] [/C] [/N] [/I] [/OFF[LINE]] "cadena" [[unidad:][ruta]archivo[...]]
  - Buscar cadenas en archivos empleando expresiones regulares

    FINDSTR [/B] [/E] [/L] [/R] [/S] [/I] [/X] [/V] [/N] [/M] [/O] [/P] [/F:archivo] [/C:cadena] [/G:archivo] [/D:lista_directorios] [/A:atrib_color] [/OFF[LINE]] cadenas [[unidad:][ruta]archivo[...]]

Linux

Linux - Búsquedas

- En Linux el comando más potente para realizar búsquedas de ficheros es *find*.

  find [ruta] [criterio] [acción]

- *ruta*: opcional. Si no se especifica, se busca recursivamente desde el directorio de trabajo
- *criterio*: opcional. Se pueden combinar tantos como se quiera.
- *acción*: también opcional, permite ejecutar un comando sobre los ficheros encontrados.
- Se trata de un comando con multitud de opciones, veremos algunas.

Linux - Criterios find – nombre archivo

- *name*: especifica patrones que deben cumplir los nombres de archivo
  - *find . -name fichero.txt* → Busca desde el directorio actual, recursivamente hacia abajo, un fichero con ese nombre
  - *find . -name fi*.txt* → Busca recursivamente ficheros que comiencen por "fi" y acaben en ".txt"
- *iname* funciona igual, pero no distingue mayúsculas y minúsculas.

```
oper@UbuntuDeskIESTeis:~/Documentos/proyecto1$ find . -name fi*
./tarea3/fichero3.txt
oper@UbuntuDeskIESTeis:~/Documentos/proyecto1$ find . -iname fi*
./tarea3/fichero3.txt
./tarea2/Fichero1.txt
```

Linux - Criterios find – profundidad

- Por defecto, find busca recursivamente desde el directorio de trabajo.
- Se puede limitar desde qué nivel comenzamos la búsqueda y cuántos niveles de profundidad:
  - *mindepth n*: la búsqueda comienza n niveles más abajo del directorio en el que nos encontramos
  - *maxdepth n*: hasta qué nivel se buscará respecto del actual

*find . -maxdepth 1 -name fi** → solo busca en directorio actual

*find . -maxdepth 2 -name fi** → busca en actual y subdirectorios directos (sin bajar más) *find . -mindepth 2 -name fi** → comienza en los subdirectorios y recorre recursivamente *find . -maxdepth 2 -mindepth 2 -name fi** → solo busca en

Linux - Criterios find – profundidad

subdirectorios

Linux - Criterios find – fechas

- Se puede filtrar por:
  - c: fecha/hora de creación
  - m: fecha/hora de última modificación
  - a: fecha/hora de último acceso
- Se puede especificar en minutos (*min*) o días (*time*)
- Delante del valor puede haber signo:
  - "+": mayor que
  - "-": menor que
  - Sin signo: igualdad

-amin -1: accedidos en el último minuto

-amin +1: accedidos hace más de 1 minuto

-amin 1: accedidos hace exactamente 1 minuto

Linux - Criterios find – tamaños

- Opción *-size [+-]n[bckMG]* → Ojo, único valor obligatorio n
  - "+n"→ archivos con tamaño mayor que n
  - "-n" → archivos con tamaño menor que n
  - b: bloques de 512 bytes
  - c: bytes
  - k: kilobytes
  - M: megabytes
  - G: gigabytes

Ejemplo:

find . -size +1M

Linux - Criterios find – tipo

- Opción *-type <tipo>*
- Siendo *tipo*:
  - d: directorio
  - f: fichero regular
  - l: enlace simbólico
  - b: dispositivo de bloque
  - c: dispositivo de caracter

Ejemplo:

find . -type d

Linux - Criterios find – otros criterios

- Permisos: *-perm* → no los hemos visto, los veremos más adelante
- Propietario: *-user*
- Grupo: *-group*
- Por id de usuario: *-uid*
- Por id de grupo: *-gid*
- Se pueden combinar criterios con “-and” (opcional), “-or” o “!”. Para asignar preferencia, se utilizan los paréntesis.

Linux - Acciones find

- Es posible ejecutar una acción para cada fichero devuelto por la búsqueda.
- Para ello, al final del comando se incluye:

  -exec <comando_a_ejecutar> {} \;

  {}: representa los resultados de la búsqueda.
- Ejemplo:

```
oper@UbuntuDeskIESTeis:~/Documentos/proyecto1$ find . -iname fi*
./tarea3/fichero3.txt
./tarea2/Fichero1.txt
```

```
oper@UbuntuDeskIESTeis:~/Documentos/proyecto1$ find . -iname fi* -exec ls -l {} \;
-rw-rw-r-- 1 oper oper 0 oct 24 12:57 ./tarea3/fichero3.txt
-rw-rw-r-- 1 oper oper 0 oct 24 12:57 ./tarea2/Fichero1.txt
```

Linux - grep: Búsquedas de patrones de texto

- Se trata de un comando muy potente para buscar patrones de texto en ficheros (recuerda que stdin es también un fichero).

  grep [opciones] <patrón> [fichero]

- Patrones:
  - el comando devolverá las líneas de los ficheros que contienen el patrón indicado (hay una opción para forzar identidad exacta).
  - El patrón puede ser una expresión regular.

Linux - grep: expresiones regulares (reducido)

- *^cadena* → la línea comienza por el texto “cadena”
- *cadena$* → la línea finaliza por el texto “cadena”
- . → cualquier carácter
- .* → 0 o más caracteres
- *[]* → rango de caracteres
- ¿Cual es el resultado de?:
  - *grep daemon.*nologin /etc/passwd*

```
oper@UbuntuDeskIESTeis:~/Documentos/proyecto1$ grep daemon.*nologin /etc/passwd
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
avahi-autoipd:x:109:116:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:110:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
avahi:x:115:121:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
colord:x:121:126:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
pulse:x:123:128:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
```

- Con el resultado anterior, ¿cuál sería el resultado de?
  - *grep ^daemon.*nologin /etc/passwd*

Linux - grep: algunas opciones

- -i → no distingue mayúsculas/minúsculas
- -c → no devolverá las líneas que coinciden, solo la cuenta
- -r → busca de forma recursiva
- -n → además de mostrar la línea coincidente, indica el número
- -v → devuelve las líneas que NO cumplen el patrón
- -l → solo devuelve los ficheros que contienen el patrón
- -w → al menos una palabra debe cumplir el patrón de forma exacta
- Opciones completas en man

Linux - editor de texto sed (I)

sed [-n] <programa_sed> <fichero/s>

- Se trata de uno de los comandos más empleados en scripts para realizar edición de ficheros.
- No modifica el fichero tratado, procesa las instrucciones.
- Trata cada línea del fichero aplicando la instrucción dada
- Teniendo el fichero prueba.txt con contenido:

```
oper@UbuntuDeskIESTeis:~/Documentos/proyecto1/tarea1$ cat prueba.txt
Esta línea es la primera línea
Esta línea es la segunda línea
Esta es otra
Esta línea es la cuarta línea
Esta línea es la quinta línea
Esta línea es la última línea
```

Linux - editor de texto sed (II)

- Prueba lo siguiente:
  - sed 's/línea/Línea/' prueba.txt
  - sed 's/línea/Línea/g' prueba.txt
  - sed '1,3 s/línea/Línea/g' prueba.txt
  - sed '3d' prueba.txt
- Podrás comprobar que el fichero prueba.txt no ha sido alterado

Linux - awk

- Se trata de un lenguaje de procesamiento de texto empleado en scripts.

  awk '{<programa>}'
- awk trata cada línea y la devuelve procesada de acuerdo al programa indicado.
- Para awk las líneas están divididas en campos, numerados de $1..$n
- El carácter de separación de campo por defecto es el espacio, pero se puede cambiar. Por ejemplo, con opción -F

Linux - awk ejemplo

```
oper@UbuntuDeskIESTeis:~/Documentos/proyecto1/tarea1$ ls -l *
-rw-rw-r-- 1 oper oper 176 oct 24 17:38 prueba.txt
-rw-rw-r-- 1 oper oper 106 oct 24 17:56 salida
```

- Para quedarnos solo con el usuario propietario (tercer valor) y nombre del fichero (noveno valor)

```
oper@UbuntuDeskIESTeis:~/Documentos/proyecto1/tarea1$ ls -l *| awk '{print $3 " " $9}'
oper prueba.txt
oper salida
```

- Para obtener el tamaño total de los archivos en el directorio que comienzan por “p”:

```
oper@UbuntuDeskIESTeis:~/Documentos/proyecto1/tarea1$ ls -l | awk '$9 ~ /p/ { sum += $5 } END { print sum }'
176
```

- awk incorpora un lenguaje extremadamente potente para analizar textos y procesarlos. El ejemplo superior es el uso más trivial.

Linux - cut

cut [opciones] [fichero]

- Se trata de un comando útil para quedarnos con ciertas partes de las líneas.
- Las opciones más útiles:

  -d: establece qué carácter separa cada campo

  -f: indica qué campos queremos obtener

- Para realizar la operación equivalente al primer ejemplo de awk:

```
oper@UbuntuDeskIESTeis:~/Documentos/proyecto1/tarea1$ ls -l *| cut -d" " -f3,9
oper prueba.txt
oper salida
```

Linux - sort

sort [opciones] [fichero]

- Devuelve las líneas del fichero ordenadas.
- opciones más útiles:

  -n: la clave de ordenación es numérica

  -r: ordenar de forma inversa

  -k: columna por la que ordenar

  -t: indica el separador de campos (espacio por defecto)
- Ejemplos en:

  https://sanchezcorbalan.es/ordenar-con-el-comando-sort/