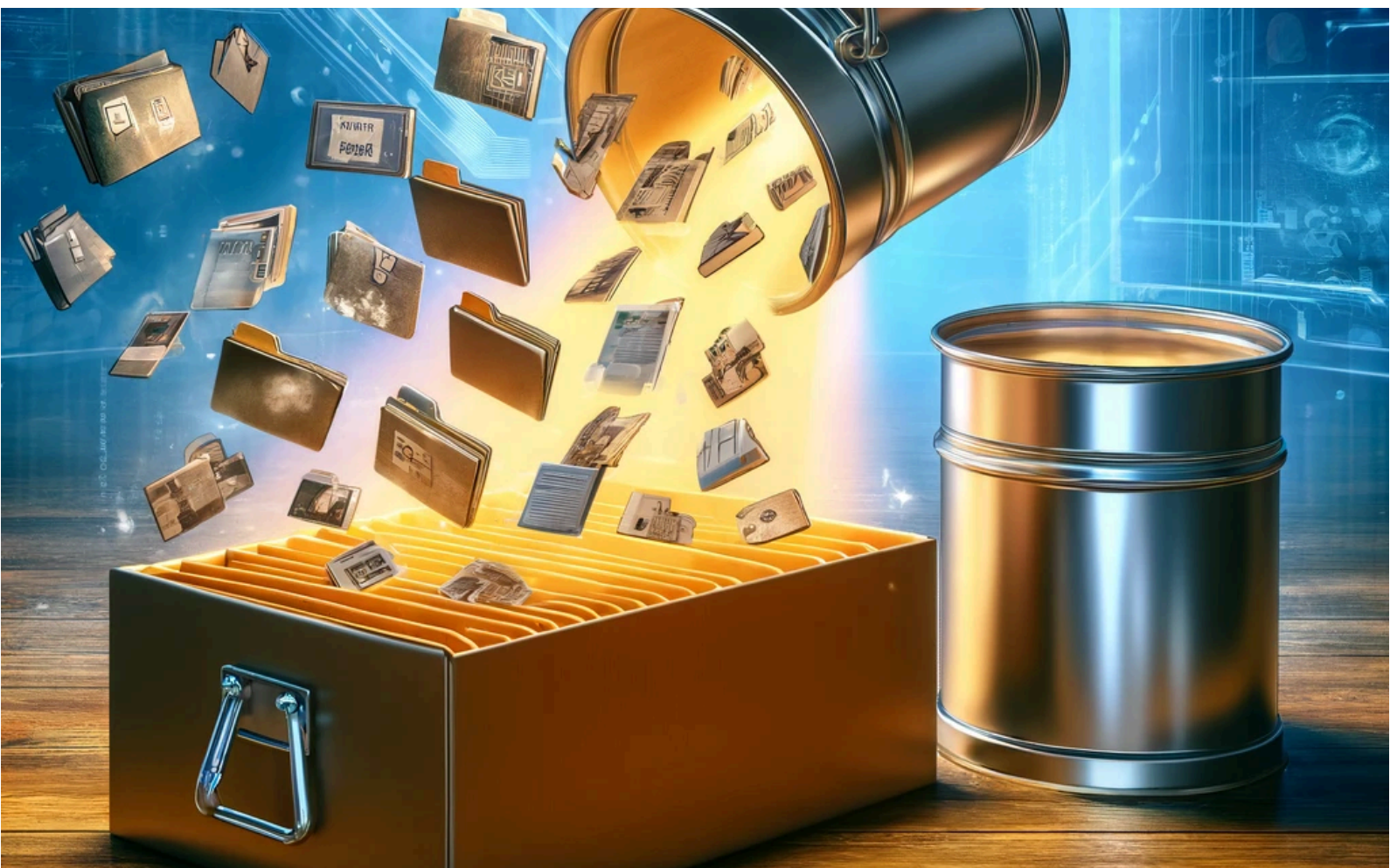


app

ndupcopy



### Objectius/Conceptes

L'objectiu d'aquesta pràctica és que l'alumnat sigui capaç de fer una eina per línia d'ordres.

Dificultat	Tipus d'activitat
Mitjans de comunicació	Desenvolupament

## Introducción

La pràctica tracta de fer una còpia de fitxers des d'unes carpetes a una altra però sense copiar els fitxers duplicats..

## Especificaciones

S'ha de crear una eina per línia d'ordres a la qual se us passarà una sèrie de directoris d'origen, i un directori de destinació. L'eina haurà d'explorar tots els fitxers de les carpetes (i subcarpetes) d'origen i els copiarà a la carpeta de destinació. En cas que es detecte que un fitxer ja existeix a la carpeta de destinació, no es copiarà. D'aquesta manera, l'efecte serà que tots els fitxers s'han copiat ignorant els duplicats.

Les carpetes d'entrada poden tenir subcarpetes que cal recórrer recursivament a la recerca de tots els fitxers.

Per passar a l'aplicació les rutes de les carpetes d'origen i de destinació, es pot optar per una de les opcions següents, però se'n poden proposar d'altres:

- Passar les rutes de les carpetes com a arguments de la línia d'ordres.
- Crear fitxers que el programa llegirà i interpretarà. D'aquesta manera, en un fitxer de text, per exemple, hi pot haver una sèrie de línies que siguin les rutes de les carpetes d'origen.

Com a punt mínim, si es véu que recórrer diverses carpetes d'origen fos una tasca massa complexa, es pot fer servir una sola carpeta d'origen i una de destinació, amb la consegüent reducció de la nota (3 punts).

## Restricciones

El programa a desenvolupar compta amb una sèrie de requisits. En cas de **incompliment** d'algun d'ells, la pràctica **no es corregirà**.

- Abans de presentar la pràctica, cal provar-ho. Per demostrar-ho, s'ha de lliurar un zip amb diverses carpetes per provar el programa. També s'ha de proporcionar la línia d'ordres que cal utilitzar per provar el programa. Abans de la correcció, s'executarà aquesta línia d'ordres proporcionada amb les carpetes que hi ha dins del zip (el qual es descomprimirà, evidentment).
- El programa ha d'estar ben estructurat, usant funcions i variables amb noms que facin entendre per a què serveixen.
- Per llegir un fitxer no es poden fer servir instruccions de nivell superior que el facin ja directament, s'ha de fer una funció que, amb un bucle, recorre un stream i l'emmagatzene en un array de bytes.
- Per comparar dos fitxers, primer cal comparar els seus hashes. En cas que els hashes coincideixen, llavors sí, es compara el contingut del fitxer. El hash que es pot fer servir, ha de ser un conegut, per exemple MD5, o SHA256.

## Ampliaciones

Es pot, de manera opcional, realitzar alguna de les millores següents:

- Es pot fer servir el tamany del fitxer abans que el hash per poder dir si són el mateix o no.
- Es pot utilitzar un hash de tipus "sencer" extret del propi fitxer com a pas previ a la comparació de hashes.
- Podeu replicar l'estructura de carpetes que hi ha dins de les carpetes d'origen a la carpeta de destinació.
- Es pot generar el hash de manera seqüencial sense necessitat de tenir tot el fitxer a la memòria RAM. Això, juntament amb el fet de no fer servir molta memòria RAM.
- Si el fitxer no té format de data, i és de tipus vídeo o imatge, intentar posar-li al nom del fitxer alguna cosa que indique la data en què es va fer la possible foto o vídeo.
- Es pot fer una funció per comparar dos fitxers que no necessite que el contingut sencer dels fitxers estigui a la RAM per poder comparar-los.