

UNITAT 11

FITXERS

EXERCICIS (A)

PROGRAMACIÓ
CFGS DAW

Autors:

Carlos Cacho y Raquel Torres

Revisat per:

Lionel Tarazon - lionel.tarazon@ceedcv.es

Fco. Javier Valero – franciscojavier.valero@ceedcv.es

José Manuel Martí - josemanuel.marti@ceedcv.es

2021/2022

Llicència



[CC BY-NC-SA 3.0 ES](https://creativecommons.org/licenses/by-nc-sa/3.0/es/) Reconeixement – No Comercial – Compartir Igual (by-nc-sa) No

es permet un ús comercial de l'obra original ni de les possibles obres derivades, la distribució de les quals s'ha de fer amb una llicència igual a la que regula l'obra original. Aquesta és una obra derivada de l'obra original de Carlos Cacho i Raquel Torres.

UF11. FITXERS

EXERCICIS (A) - GESTIÓ DE FITXERS

Per a provar aquests exercicis utilitzar l'arxiu "Documents.zip". Descarrega-ho de l'aula virtual i descomprimeix-ho en la carpeta de cada projecte que crees.

Exercici A1 - Mostrar informació de fitxers

Implementa un programa que demane a l'usuari introduir per teclat una ruta del sistema d'arxius (per exemple, "C:/Windows" o "Documents") i mostre informació sobre aquesta ruta (veure funció més a baix). El procés es repetirà una vegada i una altra fins que l'usuari introduïska una ruta buida (tecla intro). Haurà de manejar les possibles excepcions.

Necessitaràs crear la funció **void mostraInfoRuta(File ruta)** que donada una ruta de tipus File faci el següent:

- Si és un arxiu, mostrarà per pantalla el nom de l'arxiu.
- Si és un directori, mostrarà per pantalla la llista de directoris i arxius que conté (els seus noms). Haurà de mostrar primer els directoris i després els arxius.
- En qualsevol cas, afegir davant del nom l'etiqueta [*] o [A] per a indicar si és un directori o un arxiu respectivament.
- Si el path no existeix llançarà un FileNotFoundException.

Exercici A2 - Mostrar informació de fitxers (v2)

Partint d'una còpia del programa anterior, modifica la funció **mostraInfoRuta**:

- En el cas d'un directori, mostrarà la llista de directoris i arxius en ordre alfabètic. És a dir, primer els directoris en ordre alfabètic i després els arxius en ordre alfabètic. Et serà útil `Arrays.sort()`.
- Afegir un segon argument 'boolean info' que quan siga 'true' mostrarà, al costat de la informació de cada directori o arxiu, la seua grandària en bytes i la data de l'última modificació. Quan 'info' siga 'false' mostrarà la informació com en l'exercici anterior.

Exercici A3 - Canviant de nom directoris i fitxers

Implementa un programa que faci el següent:

- Canviar el nom de la carpeta 'Documents' a 'DOCS', el de la carpeta 'Fotografias' a 'FOTOS' i el de la carpeta 'Libros' a 'LECTURAS'.
- Canviar el nom de tots els arxius de les carpetes FOTOS i LECTURES llevant-li l'extensió. Per exemple, 'astronauta.jpg' passarà a dir-se 'astronauta'.

Exercici A4 - Creant (i movent) carpetes

Implementa un programa que cree, dins de 'Documents', dues noves carpetes: 'Les meues Coses' i 'Alfabet'. Mou les carpetes 'Fotografias' i 'Libros' dins de 'Les meues Coses'. Després crea dins de 'Alfabet' una carpeta per cada lletra de l'alfabet (en majúscules): 'A', 'B', 'C'... 'Z'. Et seran d'ajuda els codis numèrics ASCII: <https://elcodigoascii.com.ar>

Exercici A5 - Esborrant arxius

Implementa un programa amb una funció **boolean esborraTot(File f)** que esborre f: Si no existeix llançarà una excepció. Si és un arxiu l'esborrarà. Si és un directori, esborrarà primer els seus arxius i després el propi directori (recorda que per a poder esborrar un directori ha d'estar buit). Retornarà 'true' si va poder esborrar el 'File f' ('false' si no ha sigut possible).

Prova la funció esborrant les carpetes: 'Documents/Fotografias', 'Documents/Libros' i 'Documents' (és a dir, tres anomenades a la funció, en aqueix ordre).

Super extra challenge: Aquesta funció, tal com està definida, no esborrarà les subcarpetes que estiguen dins d'una carpeta (per a això caldria esborrar primer el contingut d'aquestes subcarpetes). Se t'ocorre com podria fer-se? Intenta-ho si t'animes ;-)

CAS PRÀCTIC A - MiniTerminal & MiniFileManager

Implementa un programa que funcione com una xicoteta terminal Linux amb alguns comandos (simplificats) que permeten a l'usuari realitzar diferents operacions de gestió d'arxius. Els comandos que l'usuari podrà executar són:

- **pwd:** Mostra com és la carpeta actual.
- **cd <DIR>:** Canvia la carpeta actual a 'DIR'. Amb .. canvia a la carpeta superior.
- **ls:** Mostra la llista de directoris i arxius de la carpeta actual (primer directoris, després arxius, tots dos ordenats alfabèticament).
- **ll:** Com ls però mostra també la grandària i la data d'última modificació.
- **mkdir <DIR>:** Crea el directori 'DIR' en la carpeta actual.
- **rm <FILE>:** Esborra 'FILE'. Si és una carpeta, esborrarà primer els seus arxius i després la carpeta. Si té subcarpetes, les deixarà intactes i mostrarà un avís a l'usuari.
- **mv <FILE1> <FILE2>:** Mou o canvia el nom a 'FILE1' a 'FILE2'.
- **help:** Mostra una breu ajuda amb els comandos disponibles.
- **exit:** Acaba el programa.

Classe MiniTerminal: Classe principal (amb funció main) que s'encarregarà d'interactuar amb l'usuari i interpretar els comandos (quin comando es demana, arguments, etc.). Utilitzarà la segona classe per a realitzar les operacions de gestió d'arxius. Manejarà totes les possibles excepcions.

Classe MiniFileManager: Tindrà els atributs i mètodes que necessites per a realitzar les diferents operacions relacionades amb la gestió d'arxius. Necessitaràs almenys un mètode per cada operació. Es llançarà una excepció si es produeix un error o l'operació sol·licitada no és possible. Alguns exemples que podries implementar:

- **String getPWD():** Retorna una cadena de text amb la carpeta actual.
- **boolean changeDir(String dir):** Canvia la carpeta actual a dir . Retorna 'true' si va ser possible.
- **void printList(boolean info):** Mostra una llista amb els directoris i arxius de la carpeta actual. Si info és 'true' mostrarà també la seua grandària i data de modificació.
- etc.