



ESCOLA TÈCNICA SUPERIOR
D'ENGINYERIA
Universitat Rovira i Virgili



Departament d'Enginyeria Informàtica i Matemàtiques

Pràctica 1: Busca un fitxer redundant

Fonaments de Sistemes Operatius

EQUIP:

Magdalena Tolstoy Guerra

Antonio Torres Cabero

ASSIGNATURA:

Fonaments de Sistemes Operatius

Data: 18 / 03 / 2018

Índex

Resum de l'enunciat	3
Pseudocodi	4
Funcions.py	4
FitxerIguals.py	6
softLink	8
vimdiff	8
Funcions i mètodes utilitzats	9
Inteficie.py	9
Funcions.py	10
FitxersIguals.py	12
vimdiff	13
Jocs de proves	14
Joc de proves 1	15
Joc de proves 2	25
Joc de proves 3	26
Joc de proves Classe fitxersIguals	27

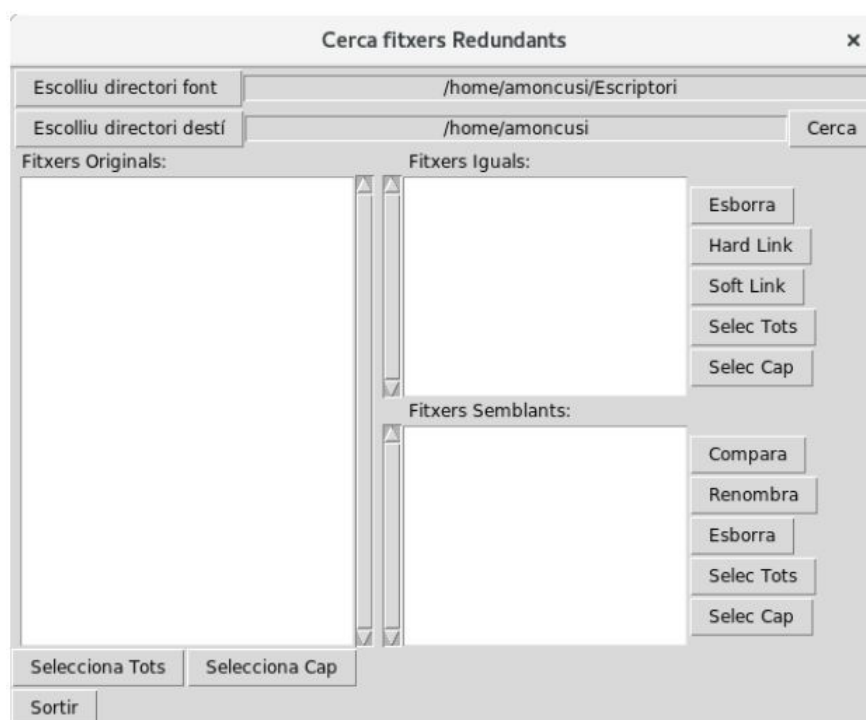
Resum de l'enunciat

L'objectiu d'aquesta pràctica és realitzar una aplicació gràfica que donats els fitxers d'un directori font busqui fitxers amb el mateix nom a un directori destí i als seus subdirectoris. A més, l'aplicació ha de comparar el contingut dels fitxers per a saber si els fitxers són iguals o no.

Per als fitxers iguals permetrem a l'usuari mantenir les dues còpies, esborrar el fitxer i crear un soft-link o hard-link al fitxer original o directament esborrar el fitxer destí. En canvi, per als fitxers amb el mateix nom però amb diferent contingut l'usuari podrà comparar-los, canviar-los el nom o esborrar-los.

A l'hora de cercar les rèpliques s'ha de tenir en compte que el directori font pot estar inclòs dins del directori destí i que per tant sempre trobarem els fitxers almenys una vegada i no esborrarem els fitxers font. En cas de trobar-nos amb soft links o que el directori destí inclou la paperera (on hi poden haver còpies) tampoc haurem de fer res.

L'estructura gràfica a implementar és la següent:



Les accions dels diferents botons fan referència a la llistes d'elements que tenen més propietats i tan sols als elements seleccionats de la llista.

S'ha de permetre a l'usuari escollir un subconjunt dels fitxers originals i llavors tan sols li haurem de mostrar els fitxers iguals i semblants d'aquests fitxers.

Finalment, el botó de compara ha d'obrir una nova finestra que mostrarà l'inode i els path relatiu dels fitxers iguals i el nombre de línies diferents. Opcionalment es pot obrir una finestra per a comparar o modificar els fitxers de dos en dos.

Pseudocodi

Funcions.py

```
/* Mètode que donats un directori i una llista amb els paths relatius (respecte al directori) d'una sèrie de fitxers elimina els fitxers. */
```

```
acció eliminarFitxers(directori: Cadena, fitxers: Cadena[])
```

```
    si (fitxers.estaBuit()) llavors  
        llancem llistaBuida("La llista està buida")
```

```
    fsi
```

```
    per (i:=0; i<fitxers.longitud(); i++) fer  
        path := unir_paths(directori, fitxers[i])  
        eliminar_fitxer(path)
```

```
    fper
```

```
facció
```

```
/* Mètode que donats un directori origen, un destí i una llista amb els paths relatius (respecte al directori destí) crea un hard link dels fitxers originals al directori destí */
```

```
acció hardLink(dir_origen: Cadena, dir_desti: Cadena, fitxers: Cadena[])
```

```
    si (fitxers.estaBuit()) llavors  
        llancem llistaBuida("La llista està buida")
```

```
    fsi
```

```
    eliminarFitxers(dir_desti, fitxers)  
    per (i:=0; i<fitxers.longitud(); i++) fer  
        fitxer := base_del_path(p_relatiu)  
        path_origen := unir_paths(dir_origen, fitxer)  
        path_desti := unir_paths(dir_desti, fitxers[i])  
        crea_hardlink(path_origen, path_desti)
```

```
    fper
```

```
facció
```

```
/* Mètode que donats un directori origen, un destí i una llista amb els paths relatius (respecte al directori destí) crea un soft link dels fitxers originals al directori destí. Implementat amb un script, ja que el codi en python hagués sigut pràcticament igual a hardLink */
```

```
acció softLink(dir_origen: Cadena, dir_desti: Cadena, fitxers: Cadena[])
```

```
    si (fitxers.estaBuit()) llavors  
        llancem llistaBuida("La llista està buida")
```

```
    fsi
```

```
    per (i:=0; i<fitxers.longitud(); i++) fer  
        executa_script('./sofftLink', dir_origen, dir_desti, fitxer[i])
```

```
    fper
```

```
facció
```

```
/* Mètode que donat un directori, un path relatiu d'un fitxer i un string amb un nom nou per al fitxer ens retorna el path relatiu del fitxer i canvia el nom del fitxer */
```

```
funció renombraFitxer(directori: Cadena, p_relatiu: Cadena, nomnou:Cadena)
```

```

    si (nomnou="") llavors
        llancem ValueError("String buida")
    sino
        p_relatiu := directori_del_path(p_relatiu)
        path_nou := unir_paths(p_relatiu, nomnou)
        p_abs := unir_paths(directori, p_relatiu)
        renombrar(p_abs, nomnou)
        retorna path_nou
    fsi
ffunció

/* Funció que retorna l'inode del fitxer destí, el path relatiu del fitxer destí i
el nombre de línies diferents entre fitxer destí i original. A més obre el vimdiff
per a veure les diferències dels fitxers */
funció comparaFitxer(dir_font: Cadena, dir_desti: Cadena, p_relatiu: Cadena)
    fitxer = base_del_path(p_relatiu)
    path_font = unir_paths(dir_font, fitxer)
    path_desti = unir_paths(dir_desti, p_relatiu)

    executa_script('./vimdiff', path_font, path_desti)

    dif := 0

    amb obrir(path_font) com a f1, obrir(path_desti) com f2
        per linia1, linia2 en zip(f1, f2) fer
            si (linia1<>linia2) llavors
                dif = dif +1
            fsi
        fper
    famb
    len1 := longitud_fitxer(path_font)
    len2 := longitud_fitxer(path_desti)
    si (len1<len2) llavors
        dif = dif+(len2-len1)
    sino si (len1>len2) llavors
        dif = dif + (len1-len2)
    fsi

    inode := obtenir_inode(path_desti)
    retorna inode+path_desti+dif
ffunció

```



```

        per x = 0 fins longitud(llista_font)
            per y = 0 fins longitud(llista_desti)

                nom1 = obtindreNomArxiu(llista_font[x])
                nom2 = obtindreNomArxiu(llista_desti[y])

                si(nom1 = nom2) llavors
                    llista_fitxers.afegir(llista_desti[y])

    retorna llista_fitxers

/* Funcio encarregada de llistar els fitxers amb el mateix contingut del directori
font i desti sense tenir en compte els fitxers de la paperera ni els soft i hard
links*/

funcio llistaFitxersIguals(llista listaFiles) retorna Llista

    llista_fitxers = llistaFitxersOriginals()
    Llista llista_iguals

    per x=0 fins longitud(listaFiles) fer

        archivo1 = obrir(directori_font + listaFiles[x], 'llegir')
        archivo2 = obrir(directori_desti + llista_fitxers[x], 'llegir')
        archivo1.llegir()
        archivo2.llegir()

        si (archivo1 = archivo2) llavors
            llista_iguals.afegir(llista_fitxers[x])

    retorna llista_iguals

/* Funcio encarregada de llistar els fitxers amb un contingut diferent del
directori font i desti sense tenir en compte els fitxers de la paperera ni els
soft i hard links*/

funcio llistaFitxersSemblants(llista listaFiles) retorna Llista

    Llista llista_fitxers = llistaFitxersOriginals()
    Llista llista_semlants

    per x=0 fins longitud(listaFiles) fer

        archivo1 = obrir(directori_font + listaFiles[x], 'llegir')
        archivo2 = obrir(directori_desti + llista_fitxers[x], 'llegir')
        archivo1.llegir()
        archivo2.llegir()

        si (archivo1 != archivo2) llavors
            llista_semlants.afegir(llista_fitxers[x])

    retorna llista_semlants

```

softLink

```
# Shell script que rep el directori font, el directori destí i el path relatiu
# d'un fitxer respecte al directori destí i crea un soft link del fitxer destí cap
# al fitxer original
#src_path := $1
#dst_path := $2
#file_path := $3

file=$(nom_base "$3")
src_full_path := "$1/$file"

dst_full_path := "$2/$3"
eliminar $dst_full_path

crear_soft_link $src_full_path $dst_full_path
```

vimdiff

```
# Shell script que rep el dos paths de fitxers i els obre a vim per a comparar-los
obrir_en_nova_terminal vimdiff $1 $2
```


Funcions i mètodes utilitzats

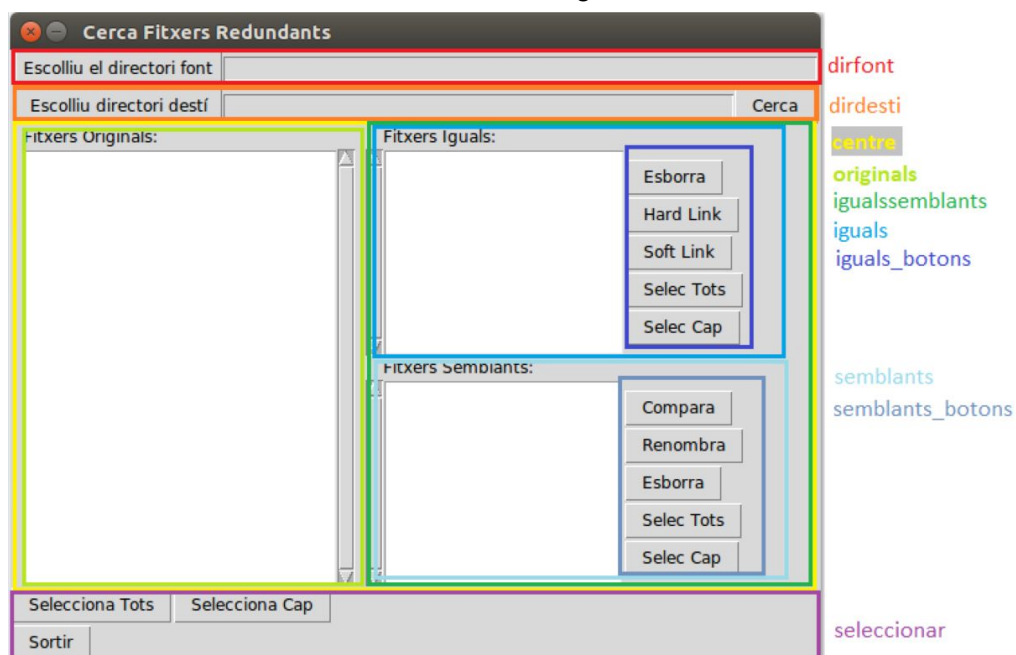
Interficie.py

En aquest fitxer s'implementen dos classes diferents i el codi principal.

La primera classe implementada s'anomena Interficie i implementa la interfície gràfica de la pràctica. La segona classe s'anomena Subfinestra i crea una finestra filla de la finestra principal, on es mostra a l'usuari la informació del "Compara".

Interficie:

L'estructura de frames de la interfície ha estat la següent:



En general, els widgets emprats han estat els explicats al laboratori, a excepció de dos:

- *tkFileDialog*: aquest widget es genera al fer clic en "Escolliu directori font" i "Escolliu directori destí". Aquest mòdul permet obrir i guardar fitxers o obrir directoris.

La funcionalitat que ens ha interessat ha estat la de obrir directoris, ja que crea una finestra on es mostren els directoris de l'ordinador i permet a l'usuari escollir-ne un.

Font de la informació: <https://pythonspot.com/tk-file-dialogs/>.

Consultat el 7/03/2018.

- *tkSimpleDialog*: mòdul que crea finestres de dialog. Aquest mòdul pot crear una finestra que demani a l'usuari un enter, un decimal o un string. Nosaltres hem utilitzat aquesta última finestra, per a que quan l'usuari faci clic a renombrar li aparegui una finestra petita on es pugui introduir el nou nom del fitxer.

Font de la informació:

<http://epydoc.sourceforge.net/stdlib/tkSimpleDialog-module.html>.

Consultat el 13/03/2018.

Subfinestra:

En aquesta classe no fem servir el “.pack” per a organitzar els diferents elements de la finestra, sinó que fem servir el .grid:

- *grid*: permet col·locar els widgets en una taula bidimensional, que està dividida en files i columnes. En el cas de voler que un widget ocupi més d'una cel·la s'ha de fer servir el “columnspan” o el “rowspan”.

Font de la informació: <http://effbot.org/tkinterbook/grid.htm>.

Consultat el 17/03/2018.

Funcions.py

El fitxer funcions.py implementa part de les funcions que es criden des de la interfície per a editar o esborrar els fitxers.

Les funcions utilitzades en aquests mètodes han sigut:

- *os.remove(path)*: mètode que permet eliminar un fitxer, a partir del seu path. Si el path es correspon a un directori salta una exception.

Font de la informació: https://www.tutorialspoint.com/python/os_remove.htm.

Consultat el 14/03/2018.

- *os.path.basename(path)*: donat un path aquesta funció retorna la cua del path. Internament la funció fa servir *os.path.split* per a dividir el path en dos, cua i cap. Per exemple, si el path passat és '/exemple/directori/fitxer' la funció retornaria 'fitxer'.

Font de la informació: <https://docs.python.org/2/library/os.path.html>.

Consultat el 15/03/2018.

- *os.path.join(path, *paths)*: la funció uneix un o més components d'un path de manera intel·ligent. El resultat d'aquesta funció és la concatenació de cadascun de path i de qualsevol *paths passat per paràmetre amb un únic separador de path. Si un dels *paths és considerat un path absolut (és a dir, que inicia amb '/'), tots els components anteriors són descartats i la concatenació continua a partir d'aquell element. És per això que no fem servir aquesta funció per a concatenar tots els paths, ja que els paths relatius al directori destí són considerats paths absoluts.

Font de la informació: <https://docs.python.org/2/library/os.path.html>.

Consultat el 15/03/2018.

- *os.link(src, dst)*: el mètode *os.link* crea un hard link que apunta a *src* anomenat *dst*.

Font de la informació: https://www.tutorialspoint.com/python/os_link.htm.

Consultat el 16/03/2018.

- `subprocess.call(args, *)`: aquesta funció executa la comanda continguda a `args`, espera a que es completi la comanda i retorna l'atribut `returncode`.
Per a executar una comanda amb arguments s'ha de passar en una llista de la següent manera: `subprocess.call(["ls", "-l"])`.

Font de la informació: <https://docs.python.org/2/library/subprocess.html>.

Consultat el 17/03/2018

- `os.path.dirname(path)`: donat un path aquesta funció retorna el cap del path. Tal i com fa `os.path.basename`, internament la funció fa servir `os.path.split` per a dividir el path en dos, cua i cap. Per exemple, si el path passat és `'/exemple/directori/fitxer'` la funció retornaria `'/exemple/directori'`.

Font de la informació: <https://docs.python.org/2/library/os.path.html>.

Consultat el 15/03/2018.

- `os.rename(src, dst)`: el mètode renombre el fitxer o directori `src` amb el nom `dst`. Si `dst` ja existeix en aquest path s'ataca una excepció.

Font de la informació: https://www.tutorialspoint.com/python/os_rename.htm.

Consultat el 16/03/2018.

- `zip(*iterables)`: la funció agafa els paràmetres o paràmetre d'entrada i crea un iterador (llista, strings, etc.) que afegeix els elements basats en els iterables passats i retorna un iterador de tuples.

Si passem tres llistes al `zip`, la funció ens retornarà una llista on a cada posició tindrem una tupla de les tres llistes.

```
> a = b = c = range(5)
> print zip(a, b, c)
[(0, 0, 0),
 (1, 1, 1),
 ...]
```

Font de la informació:

<https://www.programiz.com/python-programming/methods/built-in/zip>.

<https://stackoverflow.com/questions/13704860/zip-lists-in-python>.

Consultat el 17/03/2018.

- `sum(iterable, start)`: la funció afegeix `start` i els elements de l'iterable passat. `Start` és un valor opcional i per defecte és 0.

Font de la informació:

<https://www.programiz.com/python-programming/methods/built-in/sum>.

Consultat el 17/03/2018.

- `os.stat(path).st_ino`: `os.stat()` fa una crida de l'estat del sistema a la ruta indicada. Aquesta funció pot retornar diferents valors, però en el nostre cas com ens interessa l'inode del fitxer afegim `"st_ino"`

Font de la informació: https://www.tutorialspoint.com/python/os_stat.htm.

Consultat el 15/03/2018.

FitxersIguals.py

Fitxers iguals es una classe que s'encarrega de gestionar la informació dels dos path retornant llistes dels fitxers que tenen el mateix nom, els que son semblants i els que son iguals.

Aquesta classe compta amb:

Constructor: s'encarrega de inicialitzar una nova instància de la classe, passant-li per paràmetre la direcció font i la direcció destí.

```
def __init__(self, font, desti):
```

Setters: s'utilitzen per a guardar un nou directori destí o directori font.

```
def nombrarFont(self, n):
```

```
def nombrarDesti(self, n):
```

Funcions de la classe: s'encarreguen de extreure la informació necessària dels dos directoris que hem demanat al usuari. Per a obtindre la informació necessària he utilitzat la llibreria “**os** — Miscellaneous operating system interfaces” que ens ajuda en la gestió de informació de fitxers, mes endavant explicaré les funcions utilitzades d'aquesta llibreria.

```
def llistaFitxersMateixNom(self):
```

Aquesta funció s'encarrega de obtindre tots els fitxers amb el mateix nom i retornar-los en una llista. crea una llista amb el nom dels arxius del directori font i una altre amb els del directori destí, després els compara i els guarda tots en una nova llista.

Per a la implementació d'aquesta he utilitzat:

os.walk(directori): s'utilitza per a recórrer tot l'arbre de directoris del directori escollit, et separa el directori escollit en una tupla de tres elements, (directori font, directori relatiu, arxiu).

Font de la informació: <https://docs.python.org/2/library/os.path.html>

os.sep: retorna una cadena que serà el separador de directoris que es diferent depenent del sistema operatiu que estem utilitzant en aquell moment.

Font de la informació: <https://docs.python.org/2/library/os.path.html>

os.path.split(directori): et separa el directori del nom del arxiu.

Font de la informació: <https://docs.python.org/2/library/os.path.html>

def llistaFitxersOriginals(self)

S'encarrega de llistar tots els fitxers del directori font i retornar-los en forma de llista.

os.listdir(directori): retorna els noms dels arxius que conté aquell directori.

Font de la informació: https://www.tutorialspoint.com/python/os_listdir.htm

os.isfile(directori): retorna un booleà, cert si el directori es un arxiu i falç si no ho es.
join(cadena, cadena): uneix les dues cadenas en una.

Font de la informació: <https://docs.python.org/2/library/os.path.html>

def llistaFitxersIguals(self, listaFiles):

Aquesta funció utilitza les dues funcions anteriors ja que utilitzant les llistes que ens retorna i el nom dels fitxers del directori font que ens passen per paràmetre obrirem els dos arxius per a comprovar si contenen la mateixa informació.

open(directori, modo): obres un arxiu en mode lectura, escritura, etc.

Font de la informació: http://librosweb.es/libro/algoritmos_python/capitulo_11.html

os.path.split(directori): parteix el directori en directori i nom del arxiu del final del directori.

Font de la informació: <https://docs.python.org/2/library/os.path.html>

list.append(element): afageix un element a una llista creada previament.

Font de la informació: https://www.tutorialspoint.com/python/list_list.htm

def llistaFitxersSemblants(self, listaFiles):

Aquesta funció es exactament igual que la anterior amb la única diferencia que retorna el fitxers que no son exactament igual.

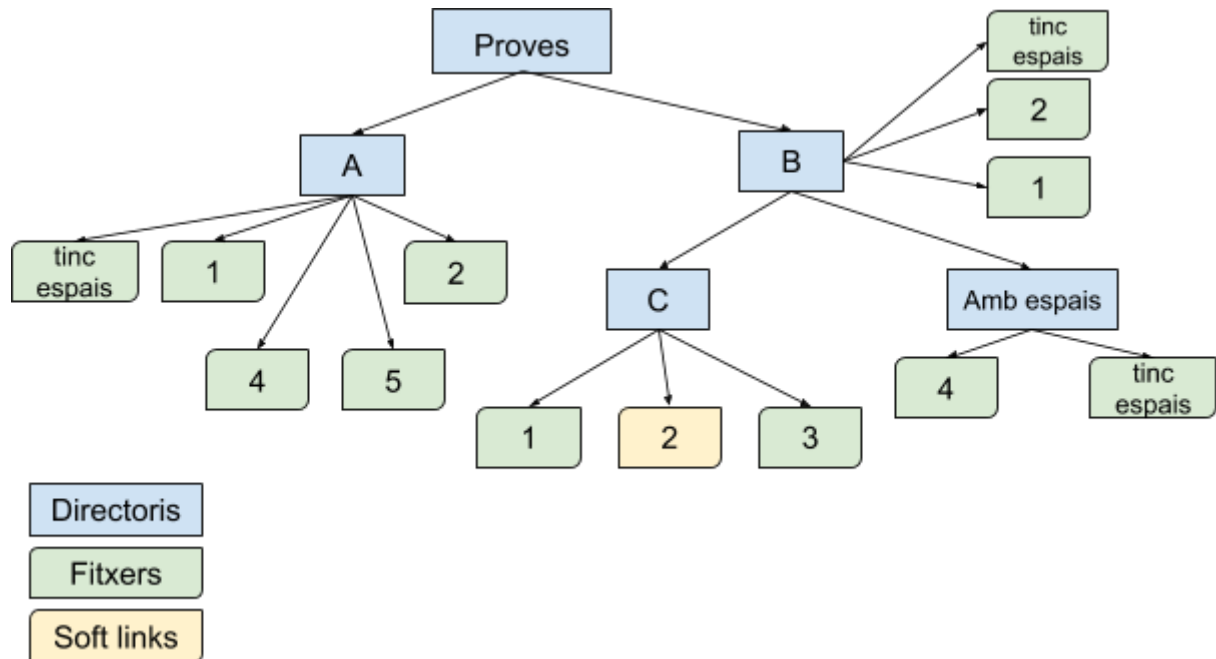
vimdiff

La comanda per a obrir el fitxer ja l'hem vist a classe, però no com obrir-lo en una nova terminal:

- `gnome-terminal -e "vimdiff '$1' '$2'" & disown:`

Jocs de proves

L'arbre de directoris que hem fet servir per a realitzar la pràctica ha estat el següent:



On el soft link Proves/B/C/2 apunta a Proves/A/2 i la relació entre fitxers és la següent:

Fitxer	Iguals	Semblants
1		/A/1
		/B/1
		/B/C/1
2	/A/2	
	/B/2	
3	Únic fitxer, sense iguals ni semblants	
4	/A/4	
	/B/Amb espais/4	
5	Únic fitxer, sense iguals ni semblants	
tinc espais	/A/tinc espais	
	/B/tinc espais	
		/B/Amb espais/tinc espais

Els nostres objectius a l'hora de crear els directoris i els fitxers han estat els següents:

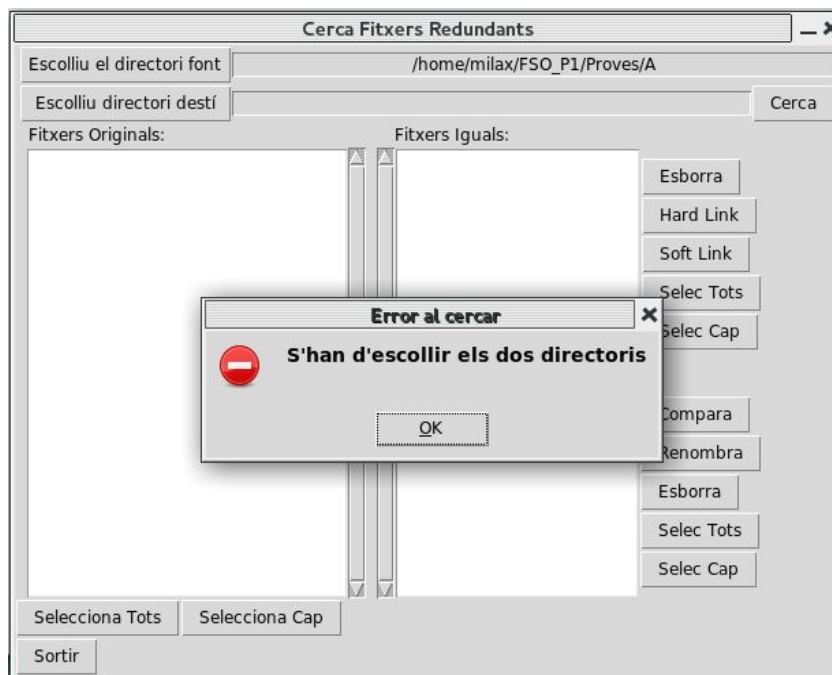
- Tenir fitxers que no tinguin ni iguals ni semblants.
- Tenir fitxers que tinguin alhora fitxers semblants i iguals.
- Tenir directoris i fitxers amb espais al nom.
- Tenir soft links a algun fitxer.
- Tenir fitxers semblants de dos tipus:
 - Amb la mateixa quantitat de caràcters i línies, pero canviant algun caràcter.
 - Amb llargades diferents, per a textos casi iguals (a excepció de la llargada).

Joc de proves 1

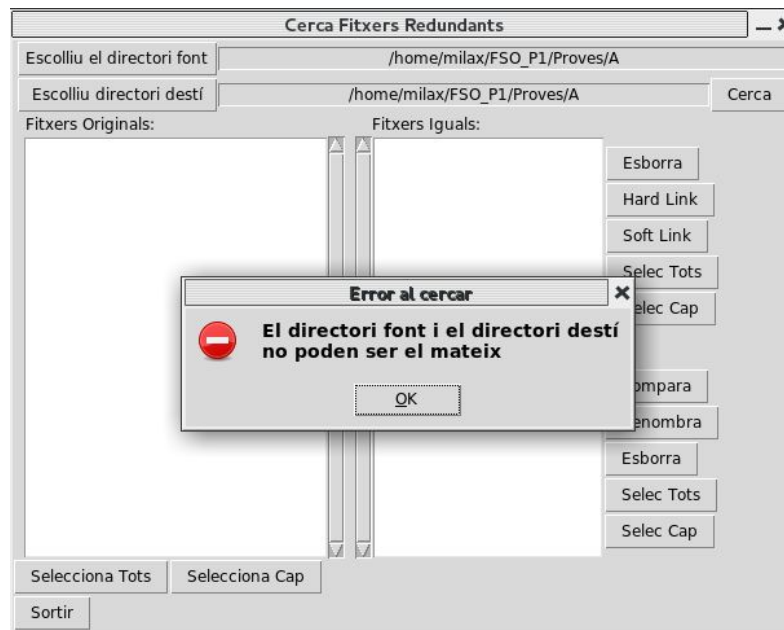
En aquest joc de proves hem agafat el directori origen com a /Proves/A i el directori destí com a /Proves/B. Aquest és el joc de proves més complet, ja que hem testejat totes les funcionalitats de la interfície gràfica.

Cerca

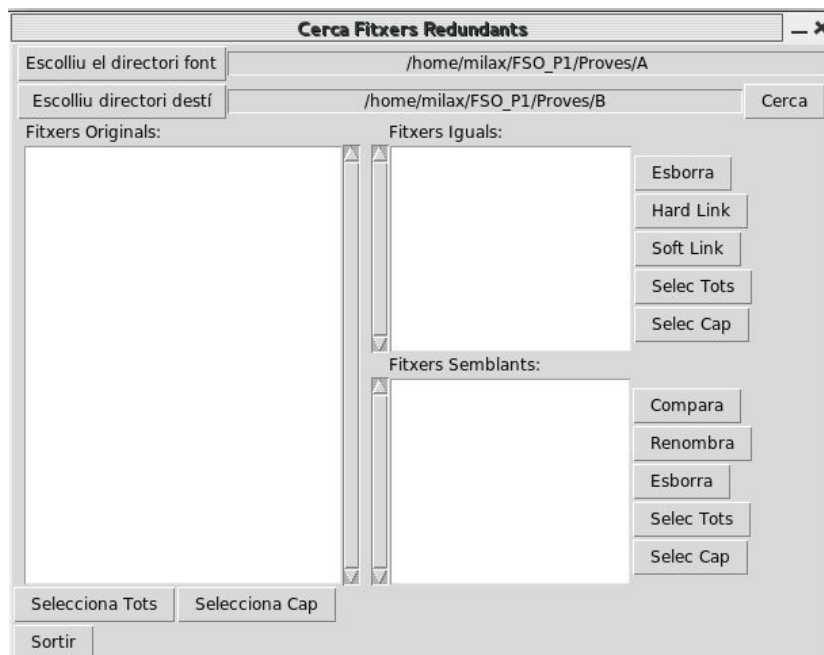
- Comprovació de que si no s'ha escollit un directori i es prem cerca surt un missatge d'error:



- Comprovació de que si escollim el mateix directori destí i origen surt i es prem cerca un missatge d'error:

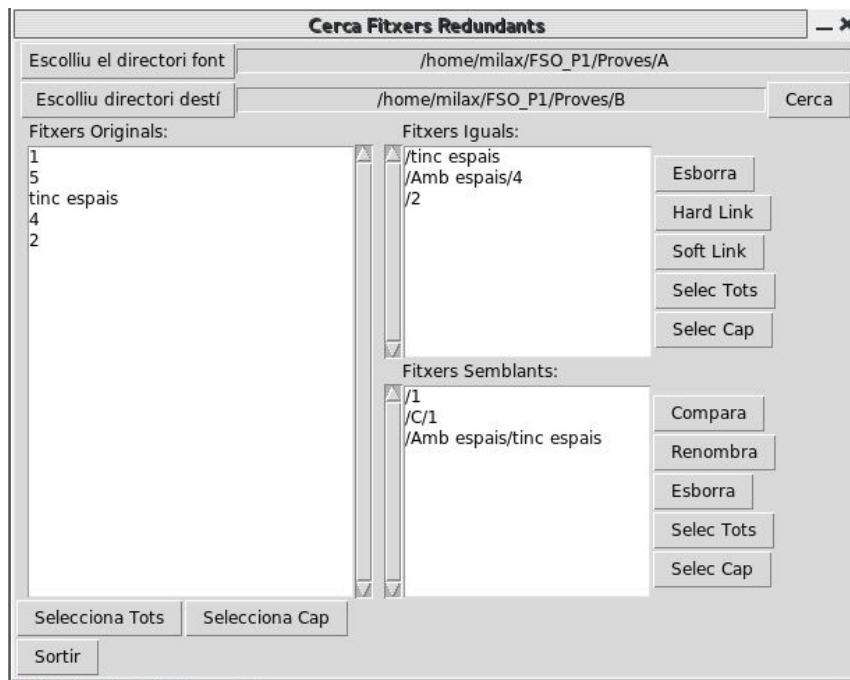


- Comprovació de que si es prem algun dels botons per a cercar directoris i després es prem cancela, no es canvia el path.



Abans d'aquesta imatge hem premut "Escolliu directori destí" i després hem fet clic a cancel·lar. Com es pot veure el path destí no ha variat.

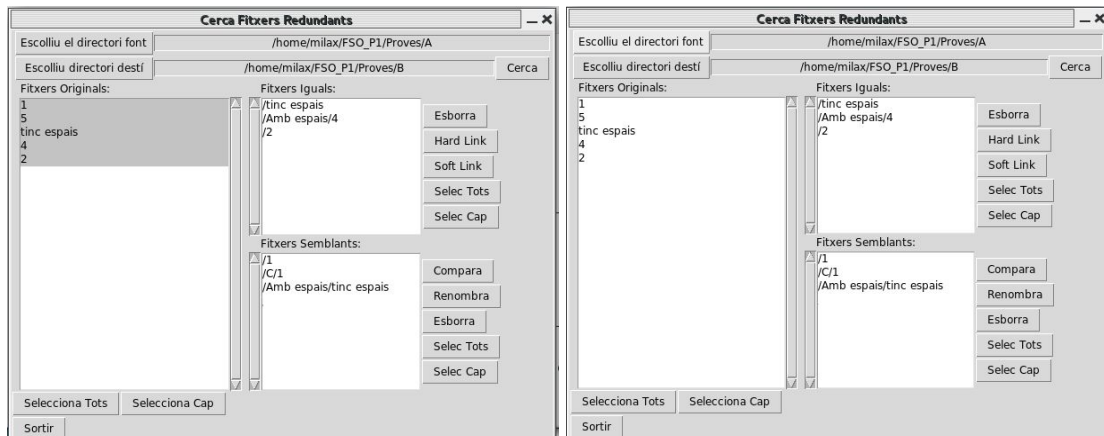
- Comprovació de que les llistes són correctes quan premem cerca:

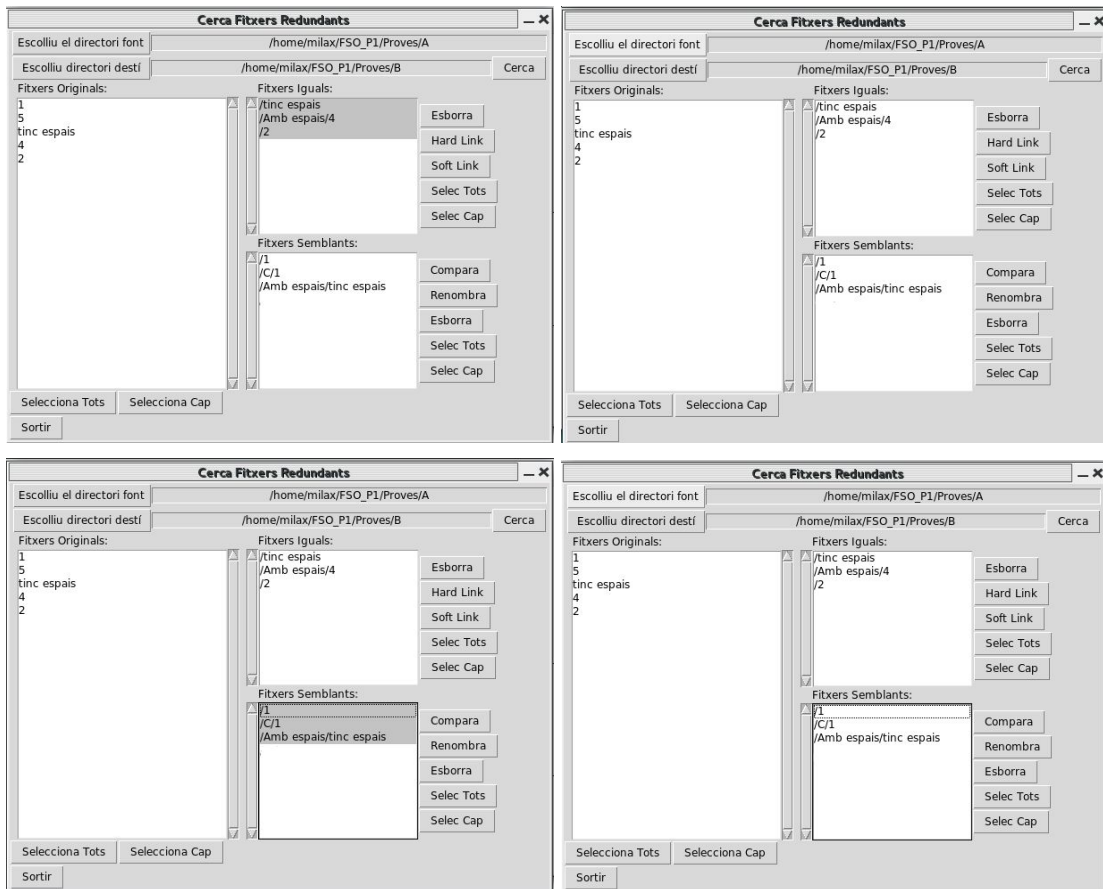


Com es pot veure a la imatge i tenint en compte la distribució i relació dels fitxers, comentada anteriorment, les llistes de fitxers són correctes. Cal remarcar a més, que el soft link /C/2 no apareix.

Selecció

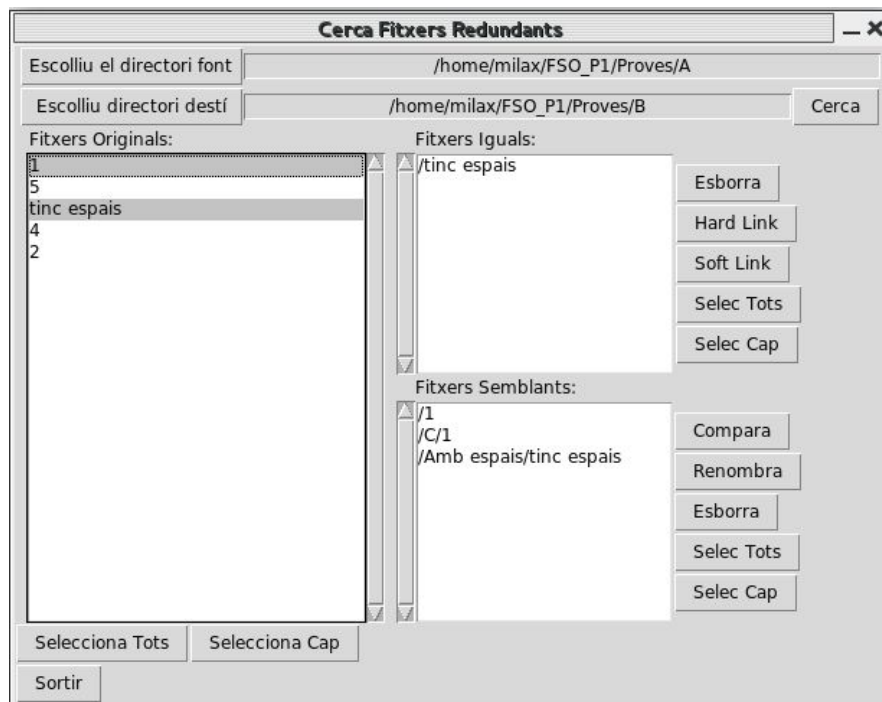
- Correcte funcionament dels botons Seleccionar tots i Seleccionar cap de totes les llistes:





Tots els botons funcionen perfectament, i en el cas dels fitxers originals al fer clic en els dos botons se'ns mostra tota la llista de fitxers semblants i iguals.

- Comprovació de que al seleccionar alguns elements de la llista de fitxers originals es carreguen els seus respectius fitxers iguals i semblants:

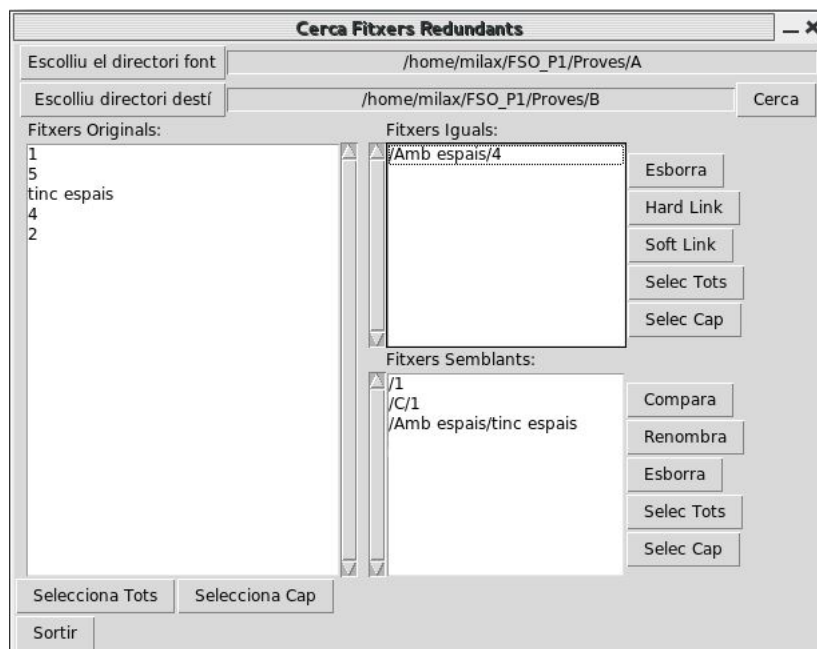


- Comprovació de que els botons de les llistes de fitxers iguals i semblants mostren un missatge d'error si es premen quan hi ha elements a les llistes però no s'ha seleccionat cap:



Esborrar

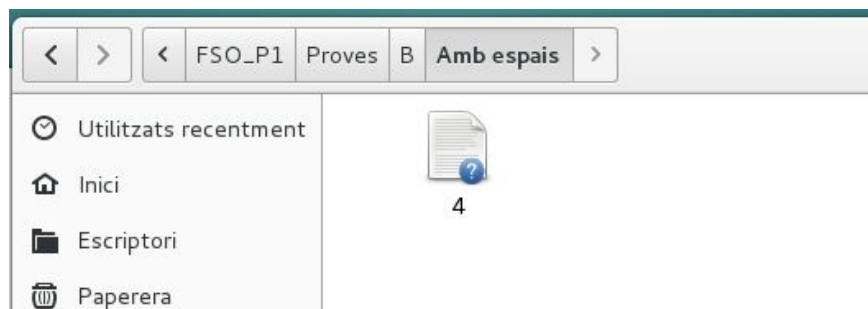
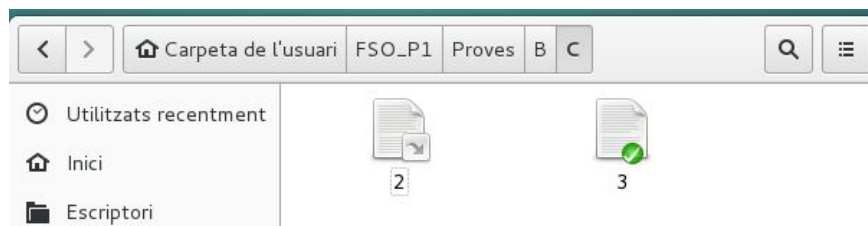
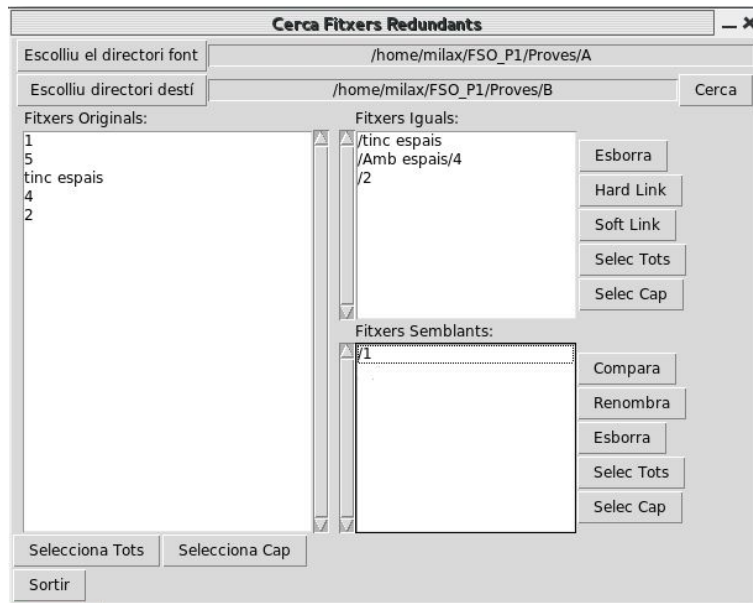
- Comprovació de correcte funcionament del botó "Esborra" de la llista de fitxers iguals:
Eliminació de /tinc espais i de /2.





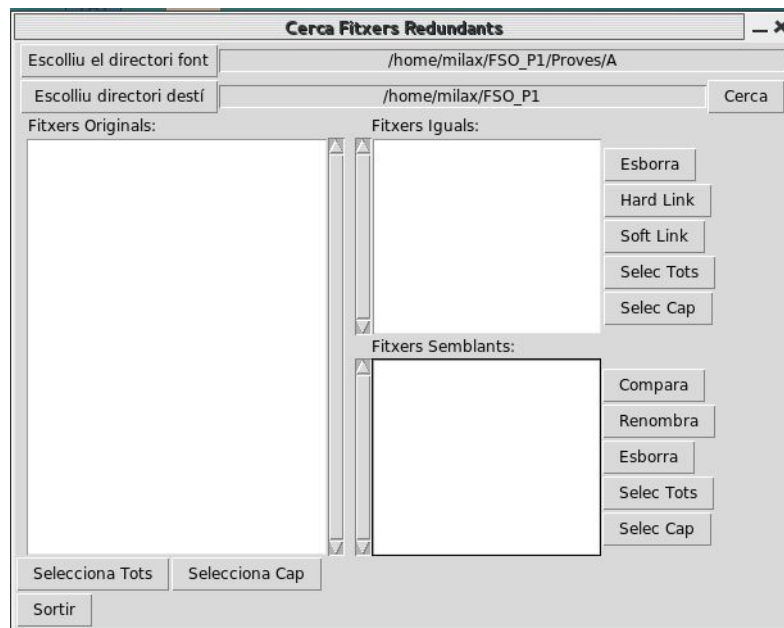
- Comprovació de correcte funcionament del botó “Esborra” de la llista de fitxers semblants:

Eliminació de /C/1 i /Amb espais/tinc espais.



Canvi de directori

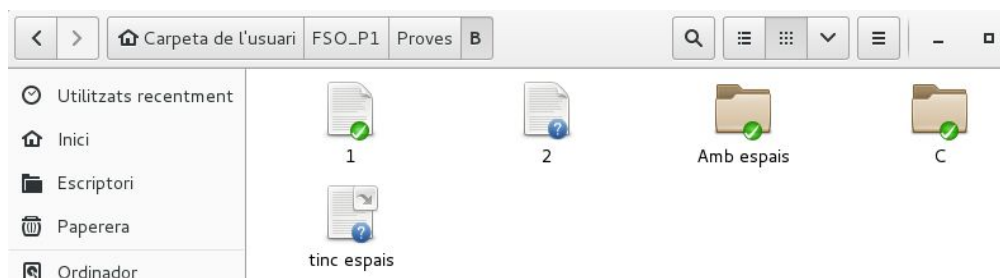
- Comprovació de que no es mostren elements a les llistes al canviar el directori font o destí:

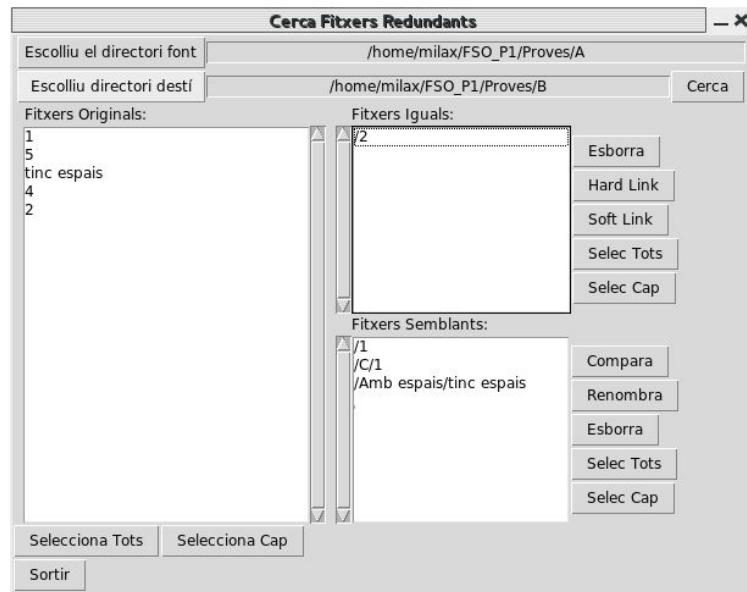


Hard i soft links

- Correcta creació de soft links:

Creació de soft links de /tinc espais i /Amb espais/4. Els links porten al fitxer que correcte.



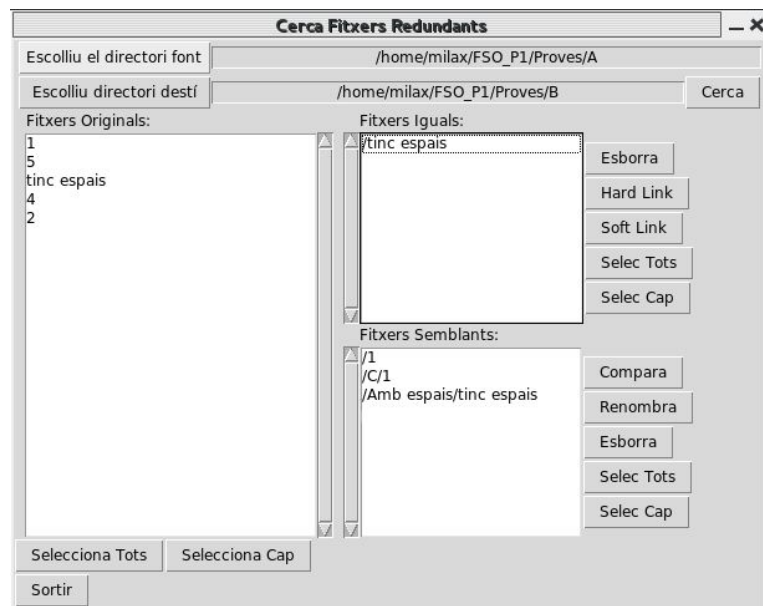


- Correcta creació de hard links:

Creació de hard link de /2 i de /Amb espais/4.

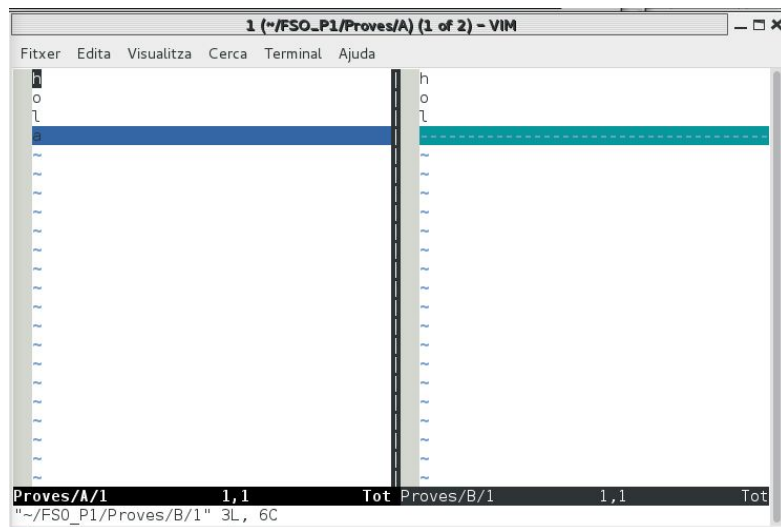
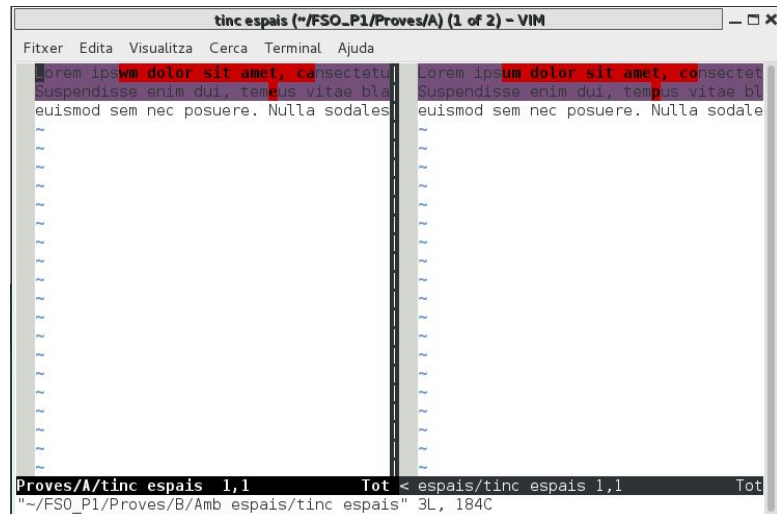
```
milax@sense-nom:~/FSO_P1/Proves/B$ ls -l
total 20
-rw-r--r-- 1 milax milax 6 mar 18 03:34 1
-rw-r--r-- 2 milax milax 287 mar 17 20:21 2

milax@sense-nom:~/FSO_P1/Proves/B/Amb espais$ ls -l
total 8
-rw-r--r-- 2 milax milax 368 mar 18 02:16 4
-rw-r--r-- 1 milax milax 184 mar 17 20:21 tinc espais
```



Compara

- Comprovació de que s'obre el vimdiff en un nou terminal i que la finestra que mostra les dades dels fitxers mostri les dades corresponents:
Comparació de /1 i de /Amb espais/tinc espais.

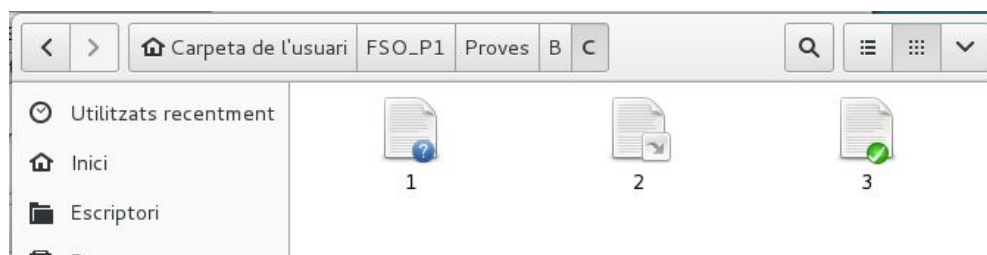
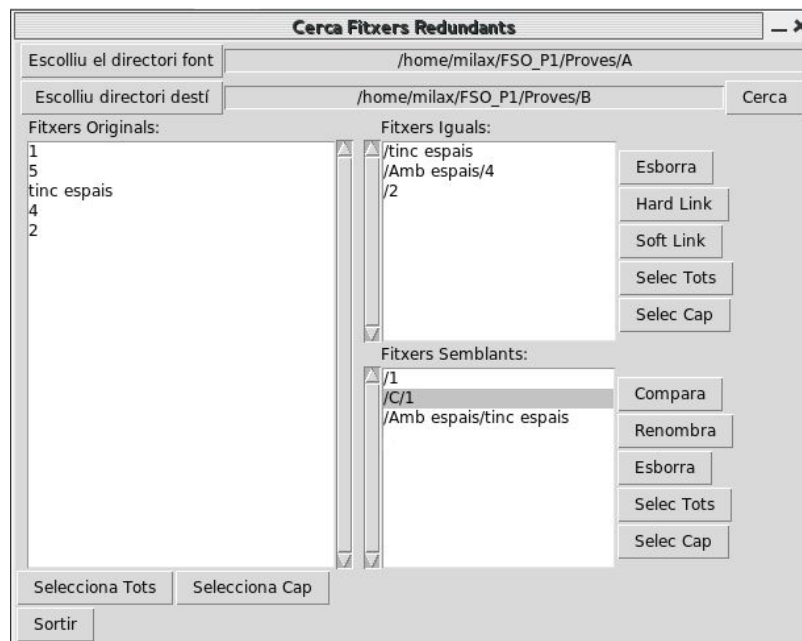


Comparació fitxers		
Inode	Path relatiu al directori destí	Número de línies diferents
1571140	/home/milax/FSO_P1/Proves/B/1	1
1571137	/home/milax/FSO_P1/Proves/B/Amb espais/tinc espais	2

El nombre de línies diferents és correcte.

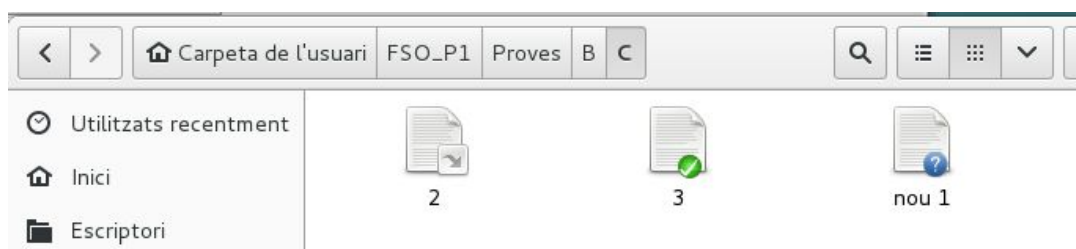
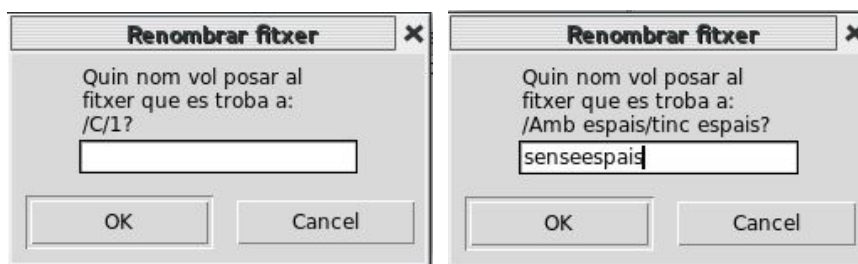
Renombra

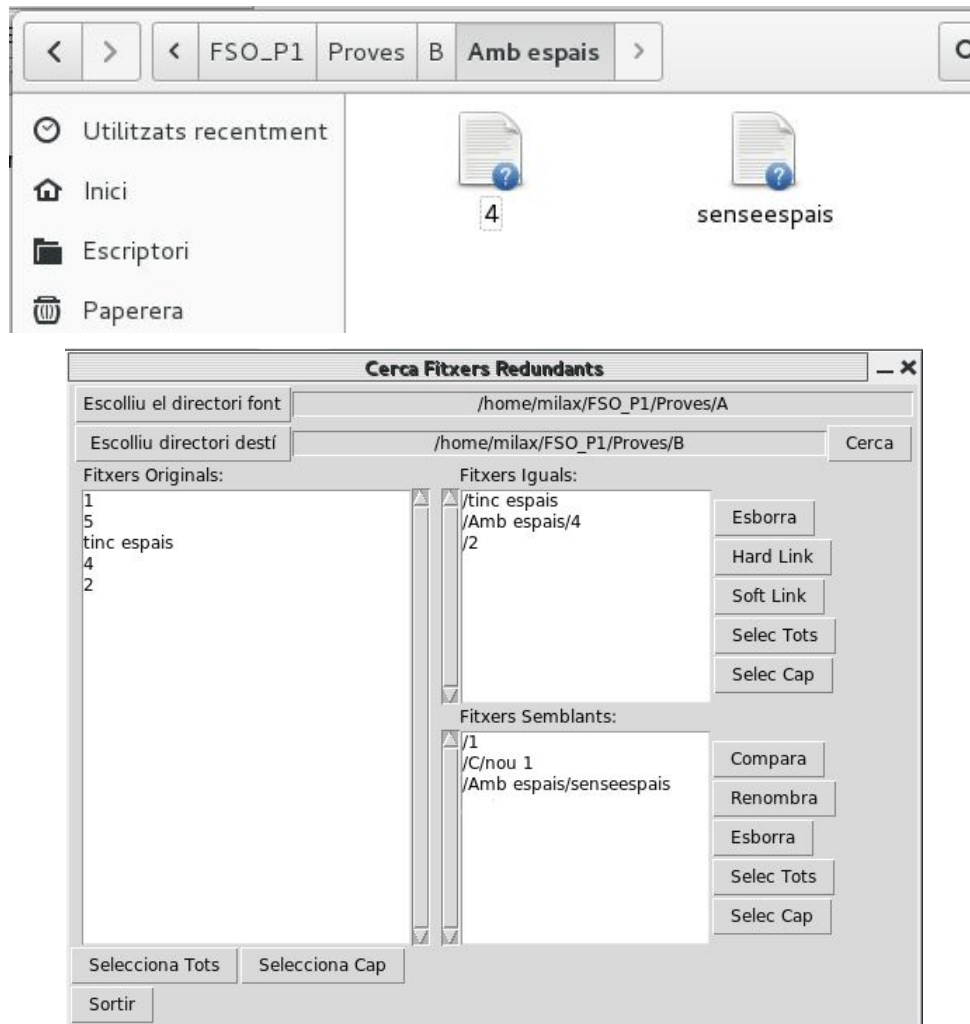
- Comprovació de que si es prem cancela no es canvia el nom del fitxer corresponent: Intentem canviar el nom de /C/1 però després fem clic a cancel·lar.



Com es pot veure no canvia ni el nom a la llista ni el nom del fitxer.

- Comprovació de que el canvi de nom es realitza correctament:
Canviem el nom a /C/1 i a /Amb espais/tinc espais.

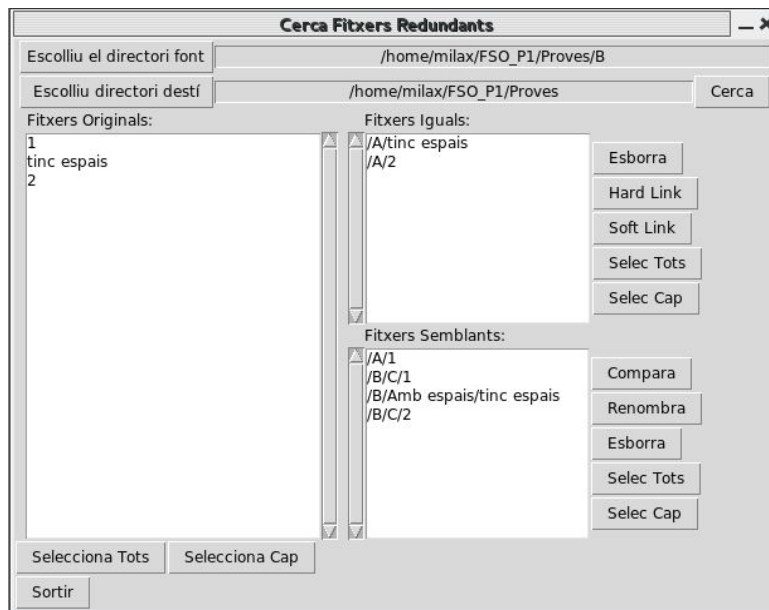




Joc de proves 2

En aquest joc de proves el que volíem comprovar era que si el directori origen es troba dins del directori destí la cerca de fitxers es realitza correctament. Això significa que tots els fitxers que hi hagin a la carpeta origen no s'han de mostrar com a fitxers de la carpeta destí ja que són el mateix fitxer d'origen, però sí els fitxers dels possibles subdirectoris de la carpeta origen.

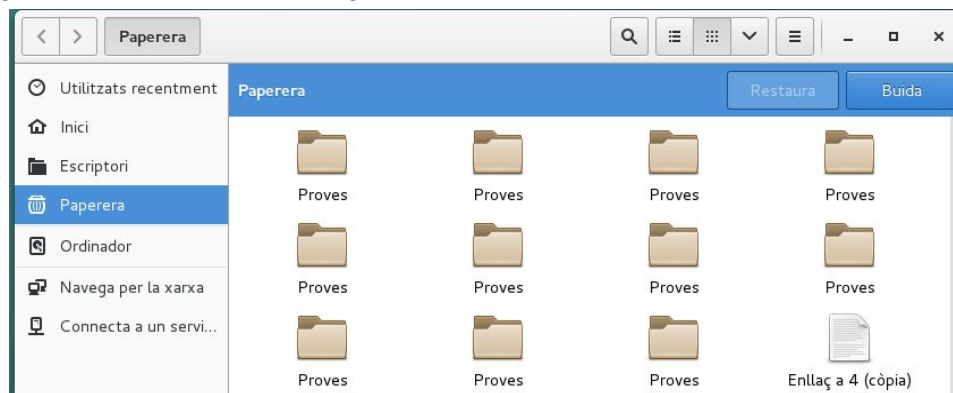
El directori origen que hem utilitzat per a aquest joc de proves és /proves/B i el destí /Proves.

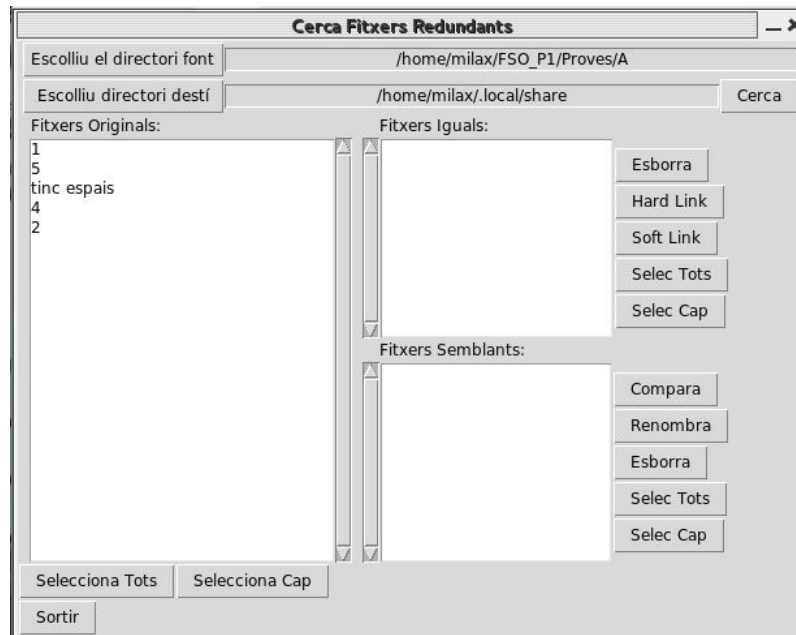


Joc de proves 3

En l'últim joc de proves hem volgut escollir una carpeta destí que continguéss la carpeta de paperera. L'objectiu és comprovar que no s'afegeixen els fitxers de la paperera a les llistes. El directori origen utilitzat ha sigut /Proves/A i el destí home/milax/.local/share.

El contingut de la paperera és el següent:





Joc de proves Classe fitxersIguals

Funcio llistaFitxersOriginals(self):

Director font = ""	Retornem una llista buida
No hi ha cap arxiu al directori	Retornem una llista buida
Només hi ha carpetas al directori	Retornem una llista buida

Funcio llistaFitxersMateixNom(self):

No hi ha cap arxiu amb el mateix nom	Retornem una llista buida
Hi ha la paperera en un subdirectori	La obviem
Trobem soft o hard links	No els incluïm a la llista