

Programa multiplataforma

Definición Programa Java

Vamos a definir una serie de opciones de menú que se listarán en un programa java. Si el sistema operativo tiene UI tendrá que hacerse haciendo uso de Swing según se quiera en caso contrario se tendrá que hacer mediante consola.

Una vez se elija una opción se deberá de llamar a un script (compatible con el S.O.) en el que nos encontremos para terminar la acción.

Opción 1 del programa. Contador de archivos.

Crear un programa multiplataforma en Java en el que se llame a un programa externo que cuente cuántos archivos hay en una carpeta de manera recursiva o no, recoja este número de archivos y compruebe si al contarlos en java de el mismo número.

También tendrá que contar los bytes que ocupan dichos archivos.

```
javac -d . src\programamultiplataforma\FileCounter.java  
java programamultiplataforma.FileCounter "C:\Users\kgv17\Desktop\CarpetaEjemplo" true
```

Opción 2 del programa. Renombrar.

Llamará a un script que cambie todos los nombres de los archivos de una carpeta recursivamente que cumplan una expresión regular por otro nombre seguido del número de archivos modificados. El programa Java deberá de verificar que se ha hecho correctamente.

Además el script deberá de ver que un archivo de esa carpeta no tenga ya ese nombre.

```
javac programamultiplataforma/FileRenamer.java  
java programamultiplataforma.FileRenamer "C:\\Users\\kgv17\\Desktop\\CarpetaEjemplo"  
".*\\.txt" renamed_file
```

Opción 3 del programa. Reemplazar.

Reemplazar en los nombres de todos los archivos lo que cumpla una expresión regular introducida en java por el usuario por lo que introduzca el usuario en el programa JAVA.

Similar a lo que se hace al principio de este video:

<https://www.youtube.com/watch?v=EPKnnG-n6-Q>

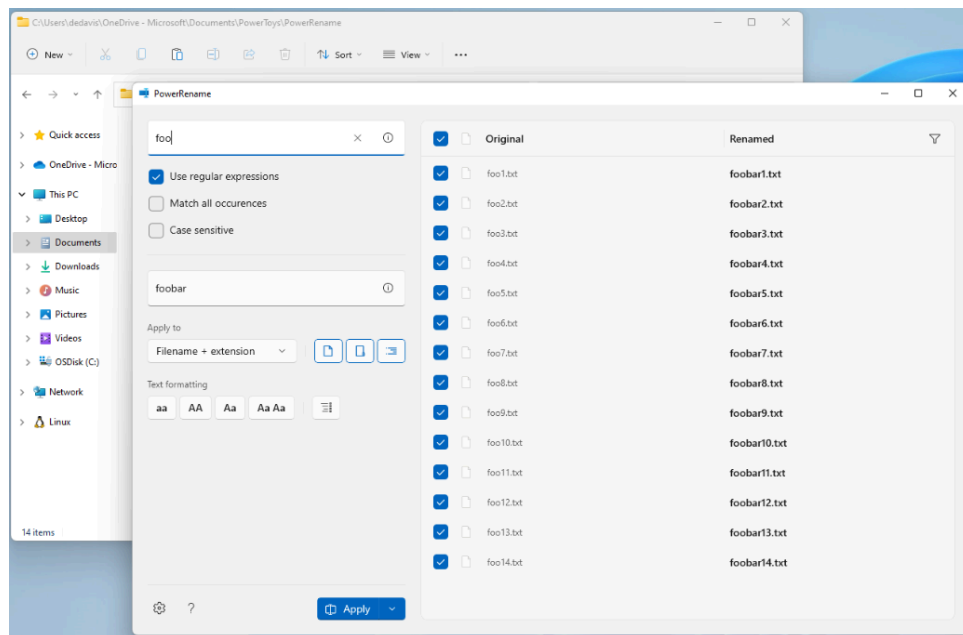
Si dos archivos se van a llamar igual deberá de ponerse un número a la derecha para que no se llamen igual. Este número será incremental según el nombre del archivo.

Por ejemplo se renombra el archivo hola.txt y ahora se encuentra otro que debe de renombrarse en la misma carpeta como hola.txt, habría que llamarle hola1.txt, en caso de encontrar adios.txt y luego otro adios.txt a crear se le podría adios1.txt pero en caso de tener que poner otro hola.txt se nombraría hola2.txt, por lo que los índices son independientes por el nombre del archivo.

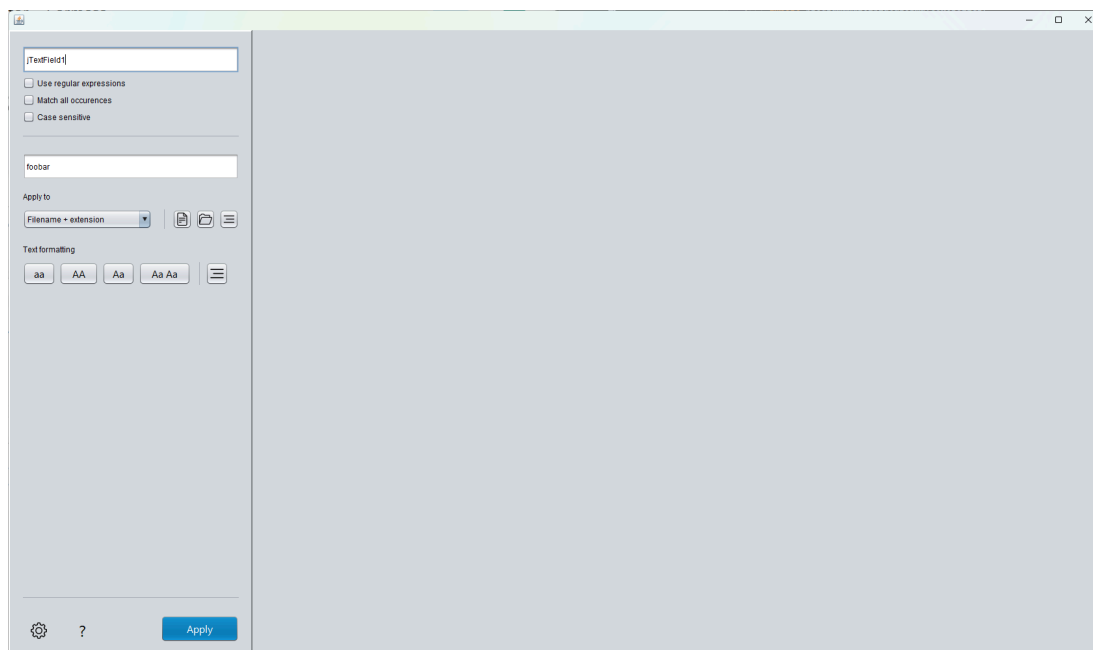
Opción 4 PowerRename

Aparecerá un menú como

<https://learn.microsoft.com/es-es/windows/powertoys/powerrename>



Con el cuál se podrá hacer todo lo que se puede hacer con PowerRename. De PowerToys de Microsoft Windows pero multiplataforma y llamando a script como en el resto de opciones.

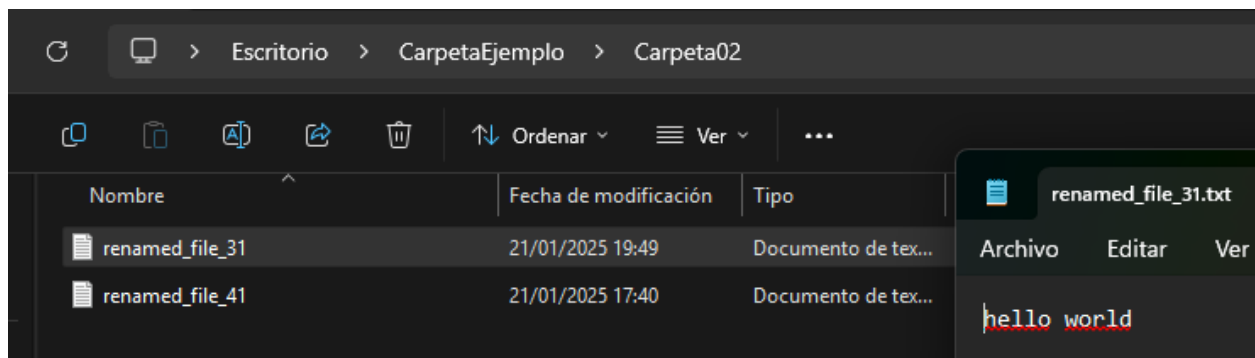
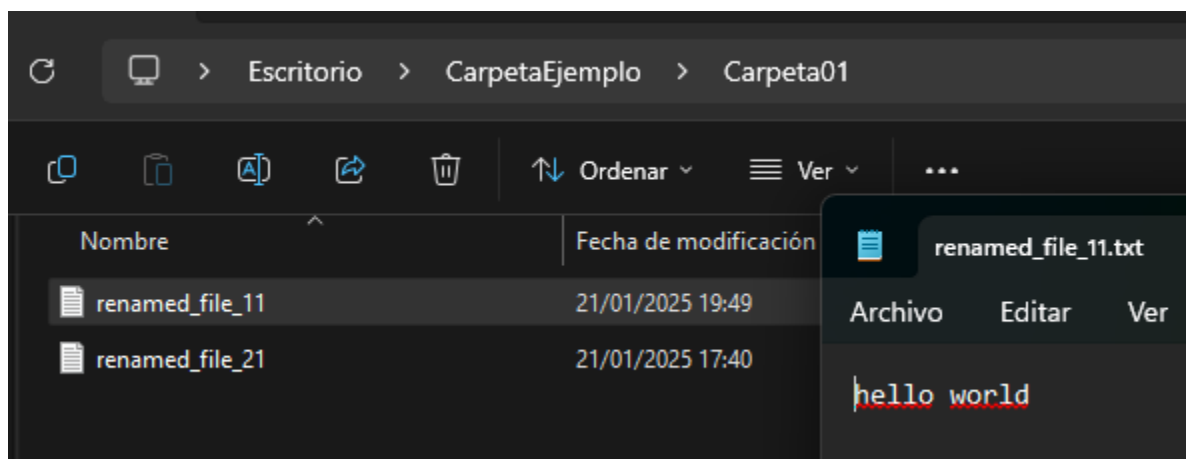


Opción 5 del programa. Concatenador.

Llamará a un script que concatene todos los textos que se contengan los archivos de una carpeta de manera recursiva. Los nombres de los documentos deberán de cumplir una expresión regular al igual que el texto a concatenar que tendrá otra expresión regular. Ambas expresiones regulares estarán definidas por el usuario en el programa Java.

Estos textos se guardarán en un archivo de un nombre introducido por el usuario y con extensión txt.

El proceso Java deberá de verificar que este archivo .txt creado por el script externo es correcto.



```
run.
Enter the directory path:
C:\Users\kgv17\Desktop\CarpetaEjemplo
Enter the regex pattern for file names:
.*\.txt
Enter the regex pattern for content:
.*
Enter the name of the output file (with .txt extension):
output.txt
Created new output file: C:\Users\kgv17\Documents\NetBeansProjects\ProgramaMultiplataforma\output.txt

Searching in directory: C:\Users\kgv17\Desktop\CarpetaEjemplo
File pattern: .*\.txt
Content pattern: .*

Files in directory:
Found file: renamed_file_11.txt
Found file: renamed_file_21.txt
Found file: renamed_file_31.txt
Found file: renamed_file_41.txt

Checking file: renamed_file_11.txt
File matches pattern: renamed_file_11.txt
Reading 1 lines from renamed_file_11.txt
Added matching line: hello world

Checking file: renamed_file_21.txt
File matches pattern: renamed_file_21.txt
Reading 0 lines from renamed_file_21.txt
No matching content found in: renamed_file_21.txt

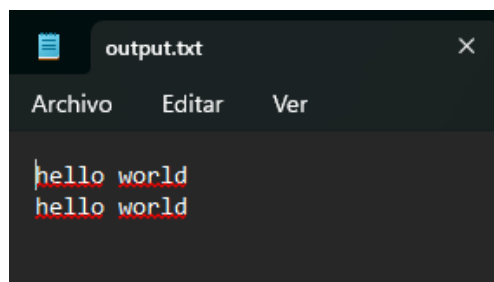
Checking file: renamed_file_31.txt
File matches pattern: renamed_file_31.txt
Reading 1 lines from renamed_file_31.txt
Added matching line: hello world

Checking file: renamed_file_41.txt
File matches pattern: renamed_file_41.txt
Reading 0 lines from renamed_file_41.txt
No matching content found in: renamed_file_41.txt

Summary:
Total files found: 4
Files matching name pattern: 4
File concatenation completed. Processed 4 files.
The output file was created successfully with 2 lines.

Output file contents:
hello world
hello world
BUILD SUCCESSFUL (total time: 31 seconds)
|
```

Activar Windows

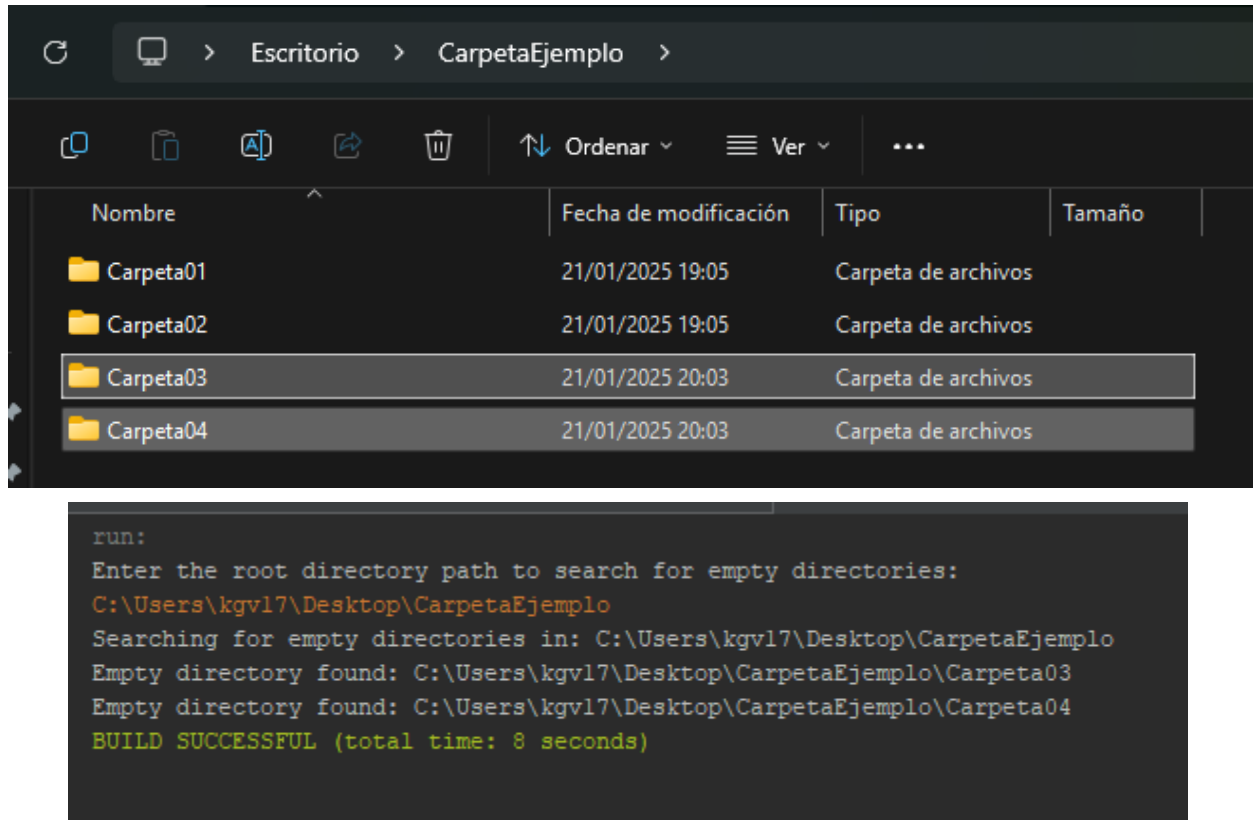


The screenshot shows a text editor window with the title 'output.txt'. The window has a menu bar with 'Archivo', 'Editar', and 'Ver'. The text area contains two lines of text: 'hello world' and 'hello world', each on a new line. The text is highlighted with a red selection box.

Opción 6 del programa

Siguiendo la lógica del resto del programa define una utilidad interesante, documéntala e implementala.

Mostrar carpetas vacías:



Entrega

- Se entregará un Google docs con capturas de las ejecuciones de las distintas opciones haciendo como un manual de usuario de cómo se usa la aplicación creada y con qué sistemas es compatible.
- En el Google Docs se pondrá un enlace a un zip con el código fuente del programa.
- Se pondrá en el Google Docs otro enlace a un zip con el programa ejecutable con todos los scripts adicionales.
- Se entregará la URL del repositorio donde se ha estado trabajando.
 - Es importante que las distintas versiones ejecutables también estén en el repositorio subidas a main.
- El profesor debe de poder tener acceso a todos los ficheros pedidos, en caso de no tener acceso en el momento de la corrección se contarán como no entregados.