Descripción de algoritmos

* Main

① Inicializar sistema

Se instancian las clases necesarias para que el sistema pueda funcionar, una vez cargadas las dependencias y librerías, se lee el archivo JSON donde están almacenadas cada una de las carpetas y archivos, a partir de esta información ahora en memoria, se construye el grafo. Luego, se genera un ciclo infinito y se espera la entrada del usuario.

② Captura de parámetros

Cuando el usuario ingresa una o varias instrucciones, la cual viene primeramente como una cadena, se valida, para ello se convierte en un arreglo, cuyos índices serán las subcadenas separadas por punto y coma (en el caso de ser varios comandos) luego se separa nuevamente en otro arreglo por espacios, el comando es válido si el índice cero de este último coincide con una de las instrucciones válidas del programa, dependiendo si el programa necesita parámetros, deberán ser validados también, para su posterior ejecución. Si el comando no es válido, se le notificará al usuario.

carpetas y archivos, a partir de esta información ahora en memoria, se construye el grafo. Luego, se genera un ciclo infinito y se espera la entrada del usuario.

② Captura de parámetros

Cuando el usuario ingresa una o varias instrucciones, la cual viene primeramente como una cadena, se valida, para ello se convierte en un arreglo, cuyos índices serán las subcadenas separadas por punto y coma (en el caso de ser varios comandos) luego se separa nuevamente en otro arreglo por espacios, el comando es válido si el índice cero de este último coincide con una de las instrucciones válidas del programa, dependiendo si el programa necesita parámetros, deberán ser validados también, para su posterior ejecución. Si el comando no es válido, se le notificará al usuario.

✱ TDA Graph

③ Método add

Para agregar al grafo un elemento ya sea directorio, archivo o un enlace a archivo, el sistema pedirá como entrada al usuario; el comando que identifica el tipo de elemento a guardar, un identificador del mismo (si es un enlace se pedirá una ruta completa de un archivo ya existente), luego el sistema capturará la ubicación actual (la ruta por defecto es el root "C:/") y el tipo de componente. Por último, si la instrucción es válida, el método llamará a una función interna que se encargará de almacenar el registro en la lista enlazada.

que se encargará de almacenar el registro en la lista enlazada.

④ Método Search

Retornará un nodo de tipo directorio, el cual buscará en profundidad (recursividad), dentro del árbol, empezando por el root "C:/". La búsqueda se realizará comparando cada identificador del árbol con el parámetro recibido del usuario, que es el elemento a buscar, si se encuentra dentro del árbol, se romperá la búsqueda con un break, si no, seguirá comparando hasta la totalidad del árbol, sino encuentra ninguna coincidencia se retornará null.

⑤ Método remove

Su funcionalidad consiste en eliminar un elemento del árbol, pero también hará un respaldo de dicho elemento con cada componente en su interior ya sean subdirectorios, archivos o enlaces en otra lista llamada "trash" (papelera), de igual manera, en la papelera, también se guarda la fecha en la que este fue eliminado y su identificador que representará el elemento borrado.

⑥ Método navigation

Busca en el árbol, por medio del método search, la carpeta actual donde se encuentra el usuario en la ejecución del sistema, el parametro de búsqueda lo recibe del usuario, una vez encontrado retorna un nodo que contiene el elemento, sino es encontrado, notificará al usuario.

subdirectorios, archivos o enlaces en otra lista llamada "trash" (papelera), de igual manera, en la papelera, también se guarda la fecha en la que este fue eliminado y su identificador que representará el elemento borrado.

⑥ Método navigation

Busca en el árbol, por medio del método search, la carpeta actual donde se encuentra el usuario en la ejecución del sistema, el parametro de búsqueda lo recibe del usuario, una vez encontrado retorna un nodo que contiene el elemento, sino es encontrado, notificará al usuario.

✱ LinkedList

⑦ Método addList

Agrega nodos a la lista de manera jerárquica y sin duplicados, usando una jerarquía alfabética propia

usuario, una vez encontrado retorna el nodo que contiene el elemento, sino el encontrado, notificará al usuario.

* LinkedList

⑦ Método addList

Agrega nodos a la lista de manera jerárquica y sin duplicados, usando una jerarquía alfabética propia del sistema, siendo directorios primero, luego archivos. Se utilizará una lista de tipo cola, verificando la existencia y agregando cada elemento sin romper el orden establecido por la jerarquía.

Este método también escribe en un archivo JSON y CSV, previamente cargados, los elementos a almacenar para que queden almacenados de forma permanente en disco, así en la próxima ejecución del programa este pueda disponer de los datos y reestructurar el flujo interno del programa.

⑧ Método Search

Método que devuelve un nodo cuyo valor interno coincide con el parámetro de búsqueda proporcionado por el usuario a través dela ejecución de un comando y que es distribuido en el flujo del programa, si el valor se encuentra, retorna su contenedor, sino se retorna null.

⑨ Método searchByExtension

⑨ Método searchBy Extension

Obtendrá un parámetro, el cual será una extensión de archivo, para luego realizar una búsqueda en profundidad, si encuentra coincidencias entre los archivos en el interior del árbol, los guardará en un arreglo que es por último retornado para mostrar sus elementos en consola.

⑩ Método pop

Remueve un elemento (directorio o archivo) de la lista cuyo identificador coincida con el del parámetro de búsqueda requerido por el sistema al usuario, si encuentra el nodo que contiene el valor en cuestión, se elimina de la lista, sino se notificará al usuario.

Este método también escribe en un archivo JSON y CSV, previamente cargados, removiendo los registros en cuestión.

⑪ Método alreadyExist

Método que comprueba si un objeto existe en la lista, recibe el nombre del elemento que se desea encontrar, se retornará verdadero si este se encuentra y falso de ser el caso contrario.

⑫ Método print

Este comando mostrará en consola una lista ya sea horizontal o vertical de los registros de los carpetas o archivos en el árbol, y dependiendo del parámetro que se le proporcione (ls, -l, trash) obtendrá los elementos del árbol o papelera, y los convertirá en una tabla utilizando cadena con formato, para su posterior impresión.

Si el parámetro es "ls -l" o "trash" se

del árbol o papelera, y los convertirá en una tabla utilizando cadena.format, para su posterior impresión.

Si el parámetro es "ls -l" o "trash" se retornará una lista vertical con información detallada de cada objeto, así como el nombre, tipo y fecha de creación o eliminación, de lo contrario se imprimirá una lista horizontal.

* Compare

(13) Método order

Tomando como referencia el alfabeto, y organizado de manera manual por los desarrolladores del sistema, este método comparará cadenas, según el peso de dicha cadena se retorna un entero (-1, 0, 1) que representarán cuando una cadena es mayor, menor o igual que otra.

Este méto será util al momento de agregar nodos a una lista.

(14) Método compare

Método que compara cadenas de texto, cuya finalidad es comprobar si una cadena es mayor que otra. Retornando cualquiera de ellas.

Este método nos ayudará para la búsqueda de directorios en el árbol.

* MultiFunction

(15) Método cd

Se asignará un arreglo donde se almacenarán los [illegible] siempre se trabajará con el

* MultiFunction

(15) Método cd

Se asignará un arreglo donde se almacenará la ruta donde el usuario esté, siempre se trabajará con el último índice de este, el usuario proporcionará una instrucción para moverse entre un directorio a otro, dependiendo de este, si es "cd" el sistema registrará como parámetro el nombre del directorio, el cual será internamente buscado en el árbol, si se encuentra, se almacenará en el arreglo el hash retornado, y se actualizará la ruta en la línea de comandos, de lo contrario, se notificará al usuario; si el parámetro no es un identificador de carpeta, también se enviará un mensaje al usuario, sin embargo, si el parámetro es ".." el sistema interpretará que desea regresar al directorio padre, borrando así el directorio actual del arreglo.

(16) Método pwd

Al momento de que el usuario introduzca la orden "pwd", el sistema buscará la ruta actual, que es el último índice del arreglo de rutas, y la retornará en una cadena para luego imprimirla en la consola.