Descripcion de Algoritmos

TDA Graph:

Metodo Add:

Se agregaran por defecto Nodos de tipo vertex, con su nombre y tipo de archivo a guardar (Carpeta o Archivo) en el nodo referencia que es el root, esta funcion o metodo necesitara el nombre y tipo del archivo a guardar y una referencia que sera el nombre de la carpeta en la que deseamos guardar, si no se da dicha referencia se considerará almacenar dicho archivo en como hijo del root o carpeta Padre, internamente se hará uso de otro metodo que agregara a una Lista enlazada.

Metodo Search:

Metodo que consta en retornar un Nodo tipo carpeta, este funcionará recibiendo el nombre de la carpeta a buscar, dicha busqueda tendra lugar en el arbol de n-nodos de forma recursiva, por defecto se empezara a buscar en la carpeta "root", con un break que utilizara una funcion que comparará lo que buscamos con el nodo del arbol que estemos recorriendo, si lo es lo retornará, caso contrario se hara una navegación en profundidad de cada carpeta en el arbol, recorriendo este en su totalidad, si no se logra encontrar la carpeta a buscar dentro del arbol, se retornara un vacio, que representara, que no se encuentra o no existe dicha carpeta.

Metodo Remove:
Su funcionalidad consiste en eliminar carpeta o archivo dentro del arbol y almacenar en una lista "trash" dicho nodo que contendra el archivo y en dado caso sea una carpeta, tendra todo lo que halla en ésta, la lista "trash" representa o simula nuestro reciclador, en lo que se va almacenado todo lo borrado del arbol, también usa un método interno que elimina de una LinkedList, funciona con el nombre del archivo a eliminar y una referencia que representa el nombre de la carpeta que contiene dicho archivo, con la cual buscaremos su nodo, para poder eliminar de este el archivo a tratar.

Metodo navegation:
Busca en el arbol por medio de search la carpeta en la que estamos navegando, esta resibe como parámetro el nombre de la carpeta en la que navegamos y retorna dicho Nodo que la contiene

Linked List:

Metodo AddList:
Agrega nodos a una lista enlazada de manera ordenada usando un alfabeto de referencia, se agregan de forma jerarquica, siendo carpetas primero y luego archivos, se utiliza una lista tipo cola, y no agrega archivos repetidos, confirmando que lo que se ingresa no existe en dicha lista, se utilizará unos metodos que nos ayudaran a ordenar lo ingresado por peso alfabético y otra para verificar si existe o no el archivo a añadir, esta tendra condicionales que nos facilitara la jerarquia, para que sin importar el orden de ingresar archivos o carpetas, esta lo haga de manera jerárquica.

Metodo Search:
Metodo que buscara un nodo dentro de nuestra lista enlazado, comparando el dato ingresado (nombre del objeto dentro del Nodo) con los nodos existentes de la lista, si se encuentra se retornara dicho nodo encontrado, caso opuesto no se retornara un nodo, sinó, un vacio.

Metodo Pop:
Elimina y retorna un nodo, éste funciona resibiendo un parametro (nombre de archivo) el cual se buscara en nuestra lista y si lo encuentra se guardara dicho nodo en una variable temporal para luego de eliminarlo de la lista, podamos retornarlo.

Metodo AlreadyExist:
Metodo que comprueba si un elemento existe ya en nuestra lista, resibe el nombre del archivo que deseamos confirmar, si se lograra encontrar retornaremos un booleano (True) que significará que dicho archivo ya existe en la lista, caso contrario retornara un False que significa que no existe en nuestra lista.

Metodo Print:
Este metodo tendra un doble papel, el cual, al resibir un parametro, y dicho parámetro cumpla ciertas condiciones, nos imprimirá una lista de forma horizontal ó una lista de forma vertical, dicha lista sera dela carpeta en la que estamos navegando.

Compare:

Metodo Order:
Tomando como base un alfabeto, éste metodo compara cadenas de nombres, según el peso de dicha cadena se retornara un entero (1, -1, 0) que representan cuando una cadena es mayor, menor o igual que la otra, esta función se utilizara al momento de agregar un dodo a una lista.

Metodo Compare:
Metodo que compara cadenas de texto, con la funcionalidad de comprobar si la cadena es igual a la otra, en dicho caso se retornara cualquiera de las dos cadenas, este metodo nos ayudara para buscar carpetas dentro de arbol

(7) Recorrer directorios con comando cd
Si el parametro recivido es ".."
se verifica si está en la raíz si es así
se retorna la raíz nuevamente, sino
se hace uso del pop de los arreglos para
eliminar su último elemento. Sino se
guarda la ubicación de la ubicación actual
si existe se agrega al arreglo de rutas la
ruta encontrada entrando a la misma,
sino se encuentra se manda un mensaje
en consola

(10) Comando ls y ls -1
Si existe un parametro en el comando
y ese parametro es -1, se captura la dirección
actual, se obtienen los nombres de carpetas y
archivos en esa ubicación, se guardan esos
nombres en un arreglo por último se genera
la cadena con formato a imprimir con
saltos de línea y tabulaciones con dichos
nombres en forma de columnas.
Si el parametro es distinto a -1 se le indica
al usuario que introdujo un comando erroneo
Sino existe ningún parametro se realiza
la operación que se hizo con el parametro -1
pero se concatena de forma horizontal, por
último sino se le indica al usuario que
ha introducido un comando erroneo.