

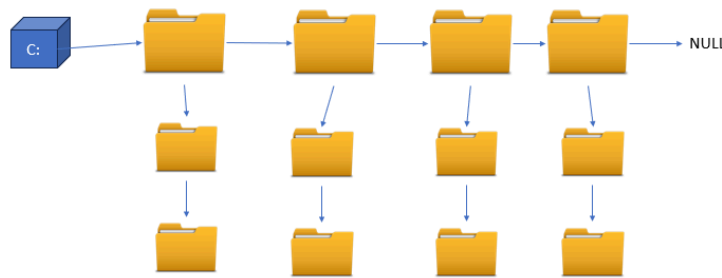
Parcial III.TDA: Pilas y Colas.

Sistema de Consola de Comandos con Gestión de Ficheros

Un sistema de archivos es una estructura organizativa y un conjunto de reglas utilizadas por un sistema operativo para gestionar y almacenar archivos en dispositivos de almacenamiento, como discos duros, unidades de estado sólido (SSD), unidades USB y otros medios de almacenamiento. Proporciona una interfaz para que los usuarios y las aplicaciones puedan acceder, organizar y manipular datos almacenados en estos dispositivos.

El proyecto busca imitar un sistema de archivos de forma lógica y se utilizará una sola unidad por defecto (C:) que contendrá la estructura de archivos y carpetas en listas enlazadas tipo pilas y colas. Una carpeta puede contener una o más carpetas o archivos. Los archivos se guardan en listas enlazadas.(3 pts)

Ejemplo:



A la consola de comandos antes programada, se requiere programar e implementar los siguientes módulos:

comando CD:(2 pts)

La función de este comando es cambiar el directorio de trabajo actual en el sistema de consola de comandos. El comando "cd" permite al usuario navegar a través de la estructura jerárquica de carpetas y establecer el directorio activo para futuras operaciones.

Este comando debe tener los siguientes argumentos:

cd C:/Documentos/Proyectos

cd .. (Regresa al directorio padre)

cd /usuario/nombre_de_usuario (Cambia a una ruta absoluta)

comando MKDIR(2 ptos)

Este comando tiene la finalidad de crear una nueva carpeta en el sistema de archivos. La creación de carpetas es esencial para organizar y estructurar la información de manera lógica y jerárquica.

Cuando se crea un nuevo directorio, este deberá ser añadido a la estructura de listas enlazada tipo cola y la lista resultando serán respaldado automáticamente en archivos json, txt o serializado según sea elegido por el equipo.

Este comando deberá tener los siguientes argumentos

```
mkdir NuevaCarpeta
```

```
mkdir direccion_destino/Proyectos
```

comando TYPE:(2 ptos)

El propósito de este comando es facilitar la creación de nuevos archivos en el sistema de archivos. Permite al usuario especificar el contenido del archivo, contribuyendo así a la manipulación eficiente de datos.

Cuando se crea un nuevo archivo, este deberá ser añadido a la estructura de listas enlazada tipo cola y la lista resultando serán respaldado automáticamente en archivos json, txt o serializado según sea elegido por el equipo.

Para esta entrega solo se manejan archivos de texto

Este comando deberá tener los siguientes argumentos:

```
type NuevoArchivo.txt "Este es el contenido del archivo."
```

```
type /direccion/Documentos/Notas.txt "Notas importantes."
```

comando RMDIR:(2 ptos)

El propósito de este comando es permitir al usuario eliminar una carpeta específica y su contenido en el sistema de archivos. La eliminación de carpetas es una operación fundamental para gestionar la organización de datos

Cuando se borra un directorio, los datos de la lista resultante debe ser respaldados automáticamente

```
rmdir /s /q CarpetaEliminar
```

```
rmdir /s /q /direccion/ViejosProyectos
```

Módulo de carga y configuración(3 ptos)

Al realizar operaciones sobre archivos o directorios, guardar logs, los datos resultantes deberán ser respaldados automáticamente en archivos de texto, json, o serializados. Al iniciar el programa, estos datos deberán ser cargados.

Se deberá programar un archivo de configuración donde se indique dirección de los archivos de respaldos y también la activación o desactivación de comandos.

El programa deberá llevar un registro de los errores ocurridos y operaciones realizadas que se almacenarán en una lista enlazada tipo pila con los siguientes comandos a utilizar:

clear log: Cambiar el log de errores. Para vaciar el log de errores, se realiza como pilas.

log: Mostrar una lista de errores y historial de operaciones, el recorrido se realiza como pilas

comando DIR:(1 ptos)

Se utilizará el mismo comando implementado anteriormente para mostrar los archivos o carpetas de un directorio en una ubicación específica o la actual.

Tips:

Tener en cuenta para el próximo proyecto: múltiples unidades, rm, mismos comandos con estructura de árboles, manejar múltiples unidades, comando search

Pautas de Evaluación.

1. La evaluación es individual o en equipo de un máximo de tres personas.
2. Usar paradigma de programación orientada a objetos
3. La defensa del proyecto tiene un valor de 4 puntos.
4. Utilizar repositorios github tiene un valor de 1 punto.
5. Los códigos iguales tendrán una penalización de puntos menos.
6. La entrega y defensa se realizará de forma presencial en hora de clases.

7. Realizar validaciones de datos introducidos por el usuario en los comandos y datos cargados.
8. El código deberá estar comentado.
9. Tener datos por defectos para tomarlos como prueba.
10. En cada módulo se evaluará lo siguiente:
 - Funcionamiento del módulo (errores, resultados correctos, independencia).
 - Legibilidad del código (nombres de variables, código comentado correctamente)
 - El alumno aplicó elementos conceptuales en la programación del módulo
 - Módulo entregado puntualmente.