

Parcial III. Algoritmos de Ordenamiento 2024.

Sistema de Consola de Comandos con Gestión de Ficheros

Diseñar y desarrollar un sistema de consola de comandos utilizando Programación Orientada a Objetos (POO). El sistema simulará un entorno de sistema de archivos, permitiendo al usuario interactuar con ficheros, carpetas, y unidades de almacenamiento. Además, el sistema deberá gestionar usuarios y roles, ofreciendo diferentes niveles de acceso y comandos basados en el rol del usuario.

Un fichero es una colección de datos que se almacena en un dispositivo de almacenamiento, como un disco duro, SSD, o unidad de memoria flash. Los ficheros pueden contener diversos tipos de información, como texto, imágenes, audio, video, o programas. Los ficheros deberán contener

- ID: Un identificador único para cada fichero.
- Nombre: El nombre del fichero.
- Tamaño: Tamaño del fichero en kilobytes.
- Extensión: Tipo de fichero (por ejemplo, .txt, .jpg).
- FechaCreación: Fecha en la que el fichero fue creado.
- FechaModificación: Fecha de la última modificación del fichero YYYY-MM-DD.
- Contenido: Datos contenidos en el fichero.

Una carpeta también conocida como directorio, es una entidad en un sistema de archivos que se utiliza para organizar y almacenar ficheros. Las carpetas permiten agrupar ficheros de manera lógica y jerárquica, facilitando la gestión y el acceso a los datos. Una carpeta puede contener fichero o carpetas. Los datos de cada carpeta deberán ser:

- ID: Identificador único de la carpeta.
- Nombre: Nombre de la carpeta.
- ListaFicheros: Lista de ficheros contenidos en la carpeta.
- FechaCreación: Fecha en la que la carpeta fue creada.
- Lista carpetas.
- TamañoTotal: Tamaño total de todos los ficheros en la carpeta.

Una unidad, en el sentido de almacenamiento, es un dispositivo o partición de almacenamiento donde se guardan los ficheros y carpetas. Cada unidad deberá manejar los respectivos datos de:

- ID: Identificador único de la unidad.
- Nombre: Nombre de la unidad (por ejemplo, C:, D:).
- CapacidadTotal: Capacidad total de la unidad de almacenamiento.
- EspacioDisponible: Espacio disponible actual en la unidad.
- ListaCarpetas: Lista de carpetas en la unidad.

- TipoUnidad: Tipo de unidad (por ejemplo, disco duro, unidad USB).

Un comando es una palabra clave o frase que, cuando se ingresa en la consola, es interpretada por el sistema para realizar una acción específica. Estos comandos pueden variar en complejidad desde operaciones simples hasta comandos que requieren una serie de argumentos o parámetros. Los comandos deben guardar:

Clase Comando

- ID: Identificador único del comando.
- NombreComando: Nombre del comando (por ejemplo, "crearFichero", "eliminarCarpeta").
- Descripción: Descripción breve de lo que hace el comando.
- RolRequerido: Rol mínimo necesario para ejecutar el comando.

Los datos deberán estar guardados en archivos xml, csv, json o serializados según sea seleccionado por el equipo.

Cuando el programa se inicie, todos los datos tendrán que ser cargados validando errores mediante manejo de excepciones.

Programar el comando **dir** para visualizar de manera rápida y eficiente el contenido de directorios (carpetas) específicos, incluyendo subdirectorios y ficheros contenidos en ellos. El comando deberá ser capaz de presentar información relevante sobre cada elemento listado, como el nombre, tipo, tamaño y fecha de última modificación. Con el comando dir realizar las siguientes búsquedas implementando algoritmos de ordenamiento:

1. **(3 pts)** Listar ficheros y carpetas ordenados por tamaño. Además proporcionar opciones para ordenar de menor a mayor tamaño (ascendente) o de mayor a menor (descendente). Utilizar quicksort

dir nombreUnidad ubicación asc|desc

Ejemplo: dir C: C:/Documentos asc

2. **(2 pts)** Mostrar ficheros y carpetas en orden de fecha de última modificación, desde los más recientes a los más antiguos y viceversa. Utilizar mergesort

dir -LastUpdate nombreUnidad ubicacion asc|desc

Ejemplo:

dir -LastUpdate C: C:/Documentos asc

3. **(2 pts)** Permitir al usuario especificar un rango de tamaño (por ejemplo, de 100KB a 500KB) y extensión a buscar listar todos los ficheros y carpetas que caigan dentro de ese rango y extensión. Si no se especifica la extensión, se buscarán todos los ficheros. Usar shellsort

dir -range inicio-fin -ext extensión nombreUnidad ubicacion asc|desc

Ejemplo: dir -range 100-500 -ext txt C: C:/Documentos asc

4. **(2 pto)** Listar todos los ficheros y carpetas que sean mayores o menores que un tamaño específico dado por el usuario. Usar algoritmo heapsort. Por ejemplo, listar todos los ficheros mayores de 1MB.
dir -range tamaño>|<|= -ext extensión nombreUnidad ubicación asc|desc
Ejemplo: dir -range 1024> -ext jpg C: C:/Documentos desc
5. **(1 pto)** Ordenar por fecha de creación carpetas y ficheros según una ubicación de forma ascendente o descendente.
dir -create nombreUnidad ubicación asc|desc
Ejemplo: dir -create C: C:/Documentos asc

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. La carga de los datos tiene un valor de 2 pts.
6. Identificar y modelar todas las clases descritas en el enunciado tiene un valor de 3 puntos
7. Los códigos iguales tendrán una penalización de puntos menos.
8. La entrega y defensa se realizará de forma presencial en hora de clases.
9. Realizar validaciones de datos introducidos por el usuario en los comandos y datos cargados.
10. El código deberá estar comentado.
11. Tener datos por defectos para tomarlos como prueba.
12. 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.