

Parcial III y IV. Proyecto. Tipos de Datos Abstracto.

Programar un Sistema de Ficheros.

Se requiere programar un sistema de ficheros. Un sistema de ficheros es una estructura jerárquica de archivos y directorio que controla la forma en cómo se guardan los datos en los dispositivos

Una consola de línea de comando es un programa que permite a las personas introducir instrucciones en la computadora a través de una línea de texto

Este sistema deberá permitir la gestión de ficheros y directorio a través de una consola de línea de comandos el cual estará constantemente ejecutando en la espera de que el usuario introduzca un comando.

El programa deberá contar con los siguientes módulos:

1. **Módulo de línea de comandos:** Este módulo es el encargado de estar ejecutándose de forma constante a la espera de que el usuario introduzca algún comando para después proceder con su ejecución.

Se debe contar con un directorio raíz C: por defecto y además deberá mostrar la dirección actual. Ejemplo:

```
C:\>
```

Se debe validar si un comando introducido por el usuario es correcto o incorrecto, en caso de ser incorrecto, entonces mostrar un mensaje de error.

```
C:\>dfg
"dfg" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.
C:\>
```

2. **Módulo de Gestión de Directorios:**

Este módulo debe permitir la creación, modificación y eliminar directorios. En este módulo se deberá programar los siguientes comandos.

- **Comando dir:** Listar los directorios y ficheros contenidos en la posición actual del usuario junto a la cantidad de directorios existentes. El comando debe tener lo siguiente:

```
C:\Users\Lab20>dir
```

```

27/06/2018  03:45 p.m.    <DIR>      .
27/06/2018  03:45 p.m.    <DIR>      ..
27/06/2018  03:45 p.m.    <DIR>      .idlerc
03/07/2019  06:35 p.m.    <DIR>      Contacts
10/10/2019  03:13 p.m.    <DIR>      Desktop
11/10/2019  05:00 p.m.    <DIR>      Documents
03/07/2019  06:35 p.m.    <DIR>      Downloads
03/07/2019  06:35 p.m.    <DIR>      Favorites
03/07/2019  06:35 p.m.    <DIR>      Links
03/07/2019  06:35 p.m.    <DIR>      Music
03/07/2019  06:35 p.m.    <DIR>      Pictures
03/07/2019  06:35 p.m.    <DIR>      Saved Games
03/07/2019  06:35 p.m.    <DIR>      Searches
03/07/2019  06:35 p.m.    <DIR>      Videos

```

- **Comando cd:** Este comando permitirá desplazarse entre directorios y pasando como parámetro el nombre del directorio y la dirección destino

```
C:\Users\Lab20>cd documents
```

```
C:\Users\Lab20\Documents>
```

```

C:\Users\Lab20>cd documents
C:\Users\Lab20\Documents>cd ..
C:\Users\Lab20>

```

```

C:\Users\Lab20>cd C:\Users\lab20.LAB20-PROFESOR\Downloads
C:\Users\lab20.LAB20-PROFESOR\Downloads>

```

- **Comando mkdir:** Este comando permite crear un directorio pasando como parámetro el nombre del nuevo directorio
 - **Comando rmdir:** Este comando permite eliminar un directorio pasando como parámetro el nombre del directorio o la dirección en la que se encuentra el directorio a eliminar
 - **Comando rename:** Este comando deberá permitir renombrar un directorio.
3. **Módulo de Gestión de Ficheros:** Este modulo debe permitir la gestión de ficheros (crear, modificar, consultar, listar y modificar). Se deberá programar los siguientes comandos.
- **Comando create:** Permitirá la creación de ficheros recibiendo como parámetro el nombre y extensión del fichero a crear. Una vez ingresado, este debe validar que en ese directorio ese fichero sea único y en caso contrario, mostrar un mensaje de error. En caso de que el nombre sea único y sea válido, entonces se le solicitara al usuario introducir el contenido del fichero.
 - **Comando remove:** Permitirá eliminar un archivo pasando como parámetro el nombre o la dirección donde se encuentra. En caso de que este fichero no exista, entonces deberá mostrar un mensaje de error. Si este fichero existe, entonces deberá mostrar un mensaje de confirmación para empezar a borrar.

- **Comando open:** Permitirá visualizar en la consola el contenido de un determinado fichero según su nombre o su dirección. En caso de que este fichero no exista mostrar un mensaje de error
- 4. **Módulo de Exportar e Importar datos:** Este módulo permitirá guardar en archivos de texto todos los datos correspondientes a los ficheros y directorios que se manejan en el usuario y es el encargado además de cargar dichos datos a los momentos de iniciar el programa. Los datos deben guardarse de forma automática en tiempo de ejecución. Y deberán ser cargados al iniciar el programa. El estudiante es libre de seleccionar la estrategia en la cual va a guardar y procesar los datos (archivos binarios, json, csv, etc...)
- 5. **Módulo de archivo de configuración y log:** Este módulo deberá permitir el archivo de configuración al momento de iniciar el programa
 - **Comando log:** Este comando permite visualizar el registro de errores y comandos ejecutados en el sistema junto a su respectiva fecha y hora de ejecución. Para esto se debe programar un registro de historial donde se guarde en archivos de texto el historial de la ejecución de comandos, así como también la ocurrencia de errores durante el tiempo de ejecución.
 - **Comando config:** Este comando permitirá mostrar en pantalla los datos del archivo de configuración

Especificaciones Técnicas

1. Desarrollar el proyecto utilizando paradigma de programación orientada a objetos.
2. Usar listas enlazadas: Pilas y Colas.
3. La unidad C: Es el puntero que siempre apuntará hacia el primer directorio.
4. Se debe realizar todas las validaciones necesarias mediante un adecuado control de errores y manejo de excepciones.
5. Los directorios serán manejados en una estructura de listas enlazadas. Cada directorio contiene uno o más ficheros. Cada fichero debe almacenarse en una estructura de listas enlazadas tipo Cola
6. El sistema debe realizar validaciones a cada uno de los datos introducidos por el usuario, así como también sus datos internos y generar mensajes de error..
7. Se debe programar un comando **help** que contiene una guía de utilización de los comandos del sistema de ficheros. El contenido debe almacenarse en un archivo de texto que será cargado al momento de iniciar el programa. Ejemplo:

```

C:\Users\argenis>help
For more information on a specific command, type HELP command-name
ASSOC          Displays or modifies file extension associations.
ATTRIB         Displays or changes file attributes.
BREAK          Sets or clears extended CTRL+C checking.
BCDEDIT        Sets properties in boot database to control boot loading.
CACLS          Displays or modifies access control lists (ACLs) of files.
CALL           Calls one batch program from another.
CD             Displays the name of or changes the current directory.
CHCP           Displays or sets the active code page number.
CHDIR          Displays the name of or changes the current directory.
CHKDSK         Checks a disk and displays a status report.
CHKNTFS        Displays or modifies the checking of disk at boot time.
CLS            Clears the screen.
CMD            Starts a new instance of the Windows command interpreter.
COLOR          Sets the default console foreground and background colors.
COMP           Compares the contents of two files or sets of files.
COMPACT        Displays or alters the compression of files on NTFS partitions.
CONVERT        Converts FAT volumes to NTFS. You cannot convert the
               current drive.
COPY           Copies one or more files to another location.
DATE           Displays or sets the date.
DEL            Deletes one or more files.
DIR            Displays a list of files and subdirectories in a directory.
DISKPART       Displays or configures Disk Partition properties.
DOSKEY         Edits command lines, recalls Windows commands, and
               creates macros.
DRIVERQUERY    Displays current device driver status and properties.
ECHO           Displays messages, or turns command echoing on or off.
ENDLOCAL       Ends localization of environment changes in a batch file.
ERASE          Deletes one or more files.
EXIT           Quits the CMD.EXE program (command interpreter).
FC             Compares two files or sets of files, and displays the
               differences between them.
FIND           Searches for a text string in a file or files.
FINDSTR        Searches for strings in files.
FOR            Runs a specified command for each file in a set of files.
FORMAT         Formats a disk for use with Windows.
FSUTIL         Displays or configures the file system properties.
FTYPE          Displays or modifies file types used in file extension
               associations.

```

Pautas de Evaluación.

1. La evaluación es individual o en equipo de dos personas.
2. Cada módulo tiene un valor de 4 puntos.
3. Proyectos iguales o con códigos iguales tendrán una penalización de puntos menos
4. Durante la defensa se probarán todos los casos en los que el sistema puede caerse, por cada caída se descontará 1 pto sobre el total de la nota final del proyecto.
5. La entrega y defensa se realizará de forma presencial en hora de clases.
6. Entregables:
 - a. Diagrama UML del proyecto
 - b. Código del proyecto.
 - c. Archivos de texto
7. Realizar validaciones de datos introducidos por el usuario
8. Tener datos por defectos para tomarlos como prueba.
9. En cada módulo se evaluará los siguiente:
 - a. Funcionamiento del módulo(errones, resultados correctos, independencia).
 - b. Legibilidad del código(nombres de variables, código comentado correctamente)
 - c. El alumno aplicó elementos conceptuales en la programación del módulo
 - d. Módulo entregado puntualmente.
 - e. Defensa del proyecto

