

NOMBRE y APELLIDOS:

MODELO A

DNI:

Duración: 1 hora y 45 minutos.

Calificación:

1.- **[1,5 puntos]** Se pide implementar una función que recibe un grafo y devuelve si el grafo es completo. También se deberá indicar el tipo de dato del grafo (solo el tipo de dato, no sus métodos).

2.- **[4,5 puntos]** Se quiere gestionar una colección digital guardada en distintos dispositivos de almacenamiento (discos duros, SSD y similares). Todo dispositivo de almacenamiento está identificado por un nombre y almacena su contenido empezando en una carpeta raíz llamada "/". Dada una carpeta, esta puede contener una cantidad arbitraria de otras carpetas y ficheros. El contenido guardado en la colección es dispar, hay desde ficheros multimedia (vídeos, audios, imágenes), documentos en varios formatos (docx, pdf, txt, etc.), ficheros de código fuente en varios lenguajes y otro tipo de ficheros. De cada fichero se desea conocer su nombre, su extensión (.avi, .mp4, .bmp, .cpp, etc.) de tipo string, el tamaño que ocupa en bytes y fecha de modificación. Por su parte, de un directorio se desea conocer lo mismo, el tamaño indicará lo que ocupa su contenido y la extensión será una etiqueta sobre el contenido que almacena. Si un directorio está etiquetado como "multimedia", en su contenido solo habrá ficheros multimedia (.mp4, .mp3, .wav, etc.) y otros directorios multimedia. Si un directorio tiene una mezcla de ficheros, estará etiquetado como "varios". Se puede diferenciar un fichero de un directorio por su extensión, ya que todo fichero empieza por punto "." y la etiqueta de los directorios no puede empezar por ".".

Se pide:

a) **[0.25 puntos]** Definir las estructuras de datos usadas para poder almacenar esta información.

b) **[0.25 puntos]** Implementar las operaciones necesarias para añadir un directorio a un disco duro dado su nombre, su etiqueta y su fecha. El tamaño de todo directorio cuando está vacío es 0. Se proporcionará el nombre del dispositivo de almacenamiento y un "enlace" al directorio donde guardar el directorio creado. Todo dispositivo de almacenamiento, al menos, tiene el directorio raíz "/". Se asume que el directorio padre proporcionado existe en el dispositivo del almacenamiento.

c) **[0.5 puntos]** Similar al caso anterior, pero para ficheros. Implementar las operaciones necesarias para añadir un fichero a un disco duro dado su nombre, su tamaño, su fecha y su extensión. Se proporcionará el nombre del dispositivo de almacenamiento y un "enlace" al directorio donde guardar el fichero. Todo dispositivo de almacenamiento, al menos, tiene el directorio raíz "/". Se asume que el directorio padre proporcionado existe en el dispositivo del almacenamiento. El insertar un fichero al directorio padre aumenta de tamaño y su directorio padre también, habrá que actualizar el tamaño que ocupa el nuevo fichero. Nunca hay que cambiar la etiqueta de los directorios padre al insertar un fichero.

d) **[1 punto]** Implementar la eliminación de un directorio. Solo pueden eliminarse directorios vacíos. Esta función recibirá un "enlace" al directorio a borrar y si está vacío lo borrará, si no

está vacío dará un mensaje de error. Implementar además una función que recorra toda la estructura de directorios y elimine todos los directorios vacíos que encuentre.

e) **[1 punto]** Como los dispositivos de almacenamiento pueden fallar se quiere implementar un mecanismo que permita añadir el contenido de un dispositivo a otro. Esta operación recibirá el nombre de dos dispositivos de almacenamiento, el origen y el destino. La operación debe copiar el contenido de un dispositivo en el otro. El destino siempre será una carpeta que cuelga de la raíz "/" que se llama "bak_" + nombre del dispositivo original. El resto de la estructura se conservará la del original. Esto provocará que se actualice el tamaño de la carpeta raíz "/".

f) **[1.5 punto]** Se pide implementar una función de búsqueda de fichero inteligente. Esta operación, dado un nombre de fichero y una extensión deberá buscar dicho fichero en la estructura devolviendo un "enlace" al fichero si lo encuentra o un "enlace" nulo en caso contrario. Aunque esta estructura de almacenamiento no tiene un orden, se desea que la búsqueda sea algo más eficiente que buscar por todos los elementos, por lo tanto, si se está buscando un fichero .avi (multimedia) y se encuentra un directorio etiquetado como "multimedia", se explorará, pero si se encuentra un directorio etiquetado como "documentos", no se explorará. Los directorios etiquetados como "varios" siempre se exploran. Para implementar esta operación se aconseja el uso de una estructura auxiliar que permita saber cuándo hay que explorar un directorio en función de la extensión del archivo a buscar. Para valorar esta estructura se pide definir el tipo de dato y mostrar la inserción de los valores que relacionan las extensiones ".avi" y ".mp4" con la etiqueta "multimedia" y las extensiones ".py" y ".cpp" con la etiqueta "codigo".