

ASIGNATURA: LENGUAJES DE PROGRAMACIÓN  
PERÍODO ACADÉMICO: 2023-1  
FECHA: 07/07/2023

NOTA

### LABORATORIO N° 4

CÓDIGO	APELLIDOS Y NOMBRES	SECCIÓN

### ANTES DE INICIAR LA EVALUACIÓN DEBE LEER LAS INSTRUCCIONES

Si la evaluación indica cargar algún archivo en la computadora, recuerde que es responsabilidad del estudiante hacerlo en el tiempo establecido y con las instrucciones dadas. Debe indicar el número de la misma en el recuadro siguiente:

#### INSTRUCCIONES GENERALES:

- El examen consta de 6 preguntas, el puntaje se indica en cada una de ellas.
- Genere un documento solo con sus respuestas y súbalo utilizando Blackboard.
- Debe utilizar los métodos y la forma de programación utilizada durante las clases del curso.
- El procedimiento, el orden, la claridad de las respuestas y el uso apropiado del lenguaje (notaciones, símbolos y unidades), serán considerados como criterios de calificación.
- **Leer detenidamente las situaciones que ocasionarán la anulación de la evaluación, que se encuentran a continuación.**

#### SITUACIONES QUE OCASIONARÁN LA ANULACIÓN DE LA EVALUACIÓN:

- Utilizar funciones y/o forma de programar diferente a lo realizado en clase.
- Consultar material no autorizado o hacer uso de Internet.
- Abrir cualquier aplicativo no autorizado. Solo se puede utilizar VSCode y su editor de texto.
- Entregar respuestas generadas por inteligencia artificial.

*Los profesores de la asignatura*

<div><div></div><div>FIRMA DEL ALUMO (LEYÓ LAS INSTRUCCIONES)</div></div>
---

## **Sistemas de Archivos**

Los sistemas de archivos desempeñan un papel muy importante en la informática moderna, sirviendo como base para organizar, almacenar y acceder a datos en dispositivos de almacenamiento. Proporcionan una estructura jerárquica que permite almacenar archivos en directorios y subdirectorios, lo que permite una organización eficiente y una recuperación de información adecuada.

Los sistemas de archivos se encargan de administrar los metadatos de los archivos, como nombres de archivo, tamaños, permisos y marcas de tiempo. También gestionan la asignación y administración del espacio de almacenamiento, garantizando un uso óptimo y evitando la fragmentación de datos. Sin los sistemas de archivos, sería difícil administrar y localizar archivos, lo que conduciría al caos e ineficiencia en el manejo de datos. Los sistemas de archivos son fundamentales para el funcionamiento de los sistemas operativos y son esenciales para diversas aplicaciones, desde computadoras personales hasta sistemas de servidores a gran escala.

En este laboratorio se utilizarán algunos ejercicios para ilustrar el funcionamiento de estos sistemas de archivos mediante el desarrollo de un “simulador”. Para ello, utilizaremos el paradigma imperativo y el paradigma orientado a objetos.

---

### **PREGUNTA 1 (3 puntos)**

a) Realice la creación de la clase SistemaArchivos, con los siguientes atributos:

- directorioRaiz
- espacioTotal
- espacioUtilizado
- espacioDisponible
- cantidadArchivos
- cantidadCarpetas
- tamanoMaximoArchivo

Realice la creación de un constructor que reciba los parámetros adecuadamente. Tome en cuenta que en principio, el espacioUtilizado será cero. Adicionalmente, el espacioDisponible será igual al espacioTotal. Los valores de cantidadArchivos y cantidadCarpetas serán cero. El resto de valores serán entrada para el constructor.

b) Realice la creación de una función que lea los datos de la clase desde el teclado, realice la creación de una nueva instancia y muestre los datos de la nueva instancia en pantalla (incluyendo todos los atributos).

### **PREGUNTA 2 (3 puntos)**

Implemente una función que reciba un sistema de archivos (objeto) y un array de números enteros. Su función deberá retornar verdadero o falso dependiendo si es posible almacenar todos los archivos del array en el sistema de archivos mencionado. Debe tomar en cuenta, tanto el espacio disponible como el tamaño máximo de archivo.

#### **Ejemplo:**

Para un sistema de archivo con tamanoArchivoMaximo = 1024 y espacioDisponible: 10240.  
Y un array con los siguientes tamaños: [1800, 1200, 9800]

La función deberá retornar falso. En primer lugar, el valor 1800 excede el tamaño máximo permitido. Adicionalmente, la suma de todos los tamaños excede el espacio disponible (10240).

### **PREGUNTA 3 (3 puntos)**

Para cierto sistema de archivos, los identificadores de archivo (números enteros), se encuentran almacenados en un array. Asimismo, los permisos habilitados para los usuarios en general se encuentran almacenados en otro array cuyos elementos toman los siguientes valores: 0 (ningún permiso), 1 (solo lectura) y 2 (lectura y escritura) y los elementos se encuentran en el mismo orden que en el primer array.

Implemente un programa que reciba estos dos arrays, un identificador de archivo y una operación (representada con una cadena que puede ser "READ" – lectura y "WRITE" escritura) y retorne verdadero o falso dependiendo de si la operación se encuentra habilitada o no.

**Ejemplo:**

Para los arrays:

ids = [1200, 1500, 1670, 1900]

permisos = [0, 0, 1, 2]

Podemos tener distintos parámetros de entrada:

- Para id = 1200 y operacion = "WRITE" retorna falso. Como vemos, para el archivo de id = 1200 el valor correspondiente al permiso es cero, por lo tanto no hay permiso ni de lectura ni de escritura.
- Para id = 1900 y operación "READ" retorna true, ya que el valor del permiso es 2 y por lo tanto se puede leer y escribir.

**Nota:** No se requiere que utilice la clase SistemaArchivos para la implementación de este ejercicio, trabaje con las instrucciones específicas de la pregunta.

**PREGUNTA 4 (4 puntos)**

La información sobre un grupo de archivos se encuentra almacenada utilizando 2 listas enlazadas. La primera de ellas contiene la información del identificador de la carpeta asociada al archivo, mientras que la segunda de ellas contiene el tamaño de cada uno de estos archivos.

Por ejemplo, podemos tener:

ids\_carpeta: 123 → 456 → 789 → 123 → 100

tamaños: 1200 → 4500 → 1230 → 7800 → 3600

En este caso, el primer archivo se encuentra en la carpeta 123 y tiene tamaño 1200, el segundo de ellos está en la carpeta 456 y tiene tamaño 4500, y así sucesivamente.

Se le solicita lo siguiente:

- Crear un programa que, dado un id de carpeta, calcule el tamaño total de los archivos almacenados en dicha carpeta.  
Por ejemplo, para el valor 123. Se tendrá un total de  $1200 + 7800 = 9000$ , que corresponden al primer y cuarto elemento de la lista.
- Crear un programa que muestre en pantalla el id de la carpeta con la mayor

**Nota:** No se requiere que utilice la clase SistemaArchivos para la implementación de este ejercicio. Adicionalmente, puede utilizar la implementación de ListaEnlazada trabajada en clase, sea como

### **PREGUNTA 5 (4 puntos)**

Sobre la implementación de las 2 primeras preguntas, se le solicita realizar los siguientes cambios:

- Crear la clase Archivo, con los atributos nombre, tipo, tamaño, propietario, fecha de creación y fecha de modificación.
- Crear la clase Carpeta, con los atributos nombre, ruta completa, tamaño, fecha de creación y fecha de modificación. Asimismo, la carpeta debe tener la capacidad de almacenar el conjunto de archivos que pertenezcan a ella. Utilice la estructura de datos de su preferencia.
- Modifique su clase SistemaArchivos para que el atributo directorioRaiz referencia a una instancia de la clase Carpeta.

Para cada una de estas clases considere métodos adicionales que permitan mostrar el contenido

Finalmente, realice la creación de un Sistema de Archivos con información de carpetas y archivos relacionados. Muestre el contenido en pantalla.

**Observación:** No se están dando los aspectos completos de implementación, se espera que el alumno complete la información con supuestos y los documente.

### **PREGUNTA 6 (3 puntos)**

Dada una carpeta, muestre el listado de todas sus carpetas hijas y los archivos que contienen. Este reporte debe incluir todos los niveles. Es decir, las carpetas hijas, las carpetas hijas de las carpetas hijas, etc.

Por ejemplo, para la estructura siguiente:

```
/
|-- home/
|   |-- user1/
|   |   |-- documents/
|   |   |   |-- file1.txt
|   |   |   |-- file2.txt
|   |   |-- images/
|   |   |   |-- image1.jpg
|   |   |   |-- image2.jpg
|   |-- user2/
|   |   |-- documents/
|   |   |   |-- file1.txt
|   |   |   |-- file2.txt
|   |   |-- images/
|   |   |   |-- image1.jpg
|   |   |   |-- image2.jpg
|-- etc/
|-- var/
|-- usr/
|-- bin/
```

Si la carpeta indicada es "/", se mostrará tanto la información de "home" (carpeta hija) como la de "user1" (hija de "home"). Asimismo, se mostrarán los archivos relacionados.