

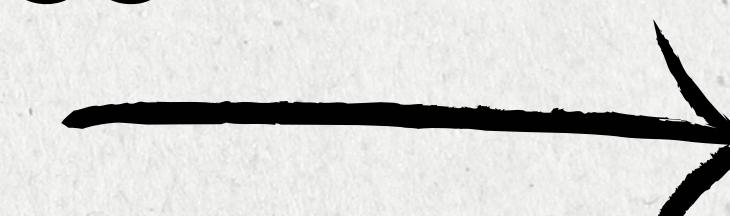
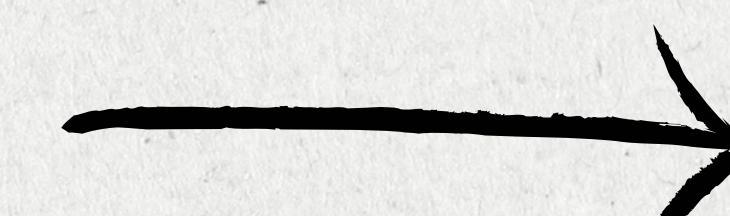


# **UNIDAD IO Y II: FICHEROS**

## **CLASE 2: LECTURA POR CARACTERES**

# FLUJOS O STREAMS

## TIPOS DE FLUJOS

- Flujo de bytes  
(8 bits)  InputStream y OutputStream
- Flujo de caracteres  
(16 bits)  Reader y Writer

# FORMAS DE ACCESO A UN FICHERO

## Acceso secuencial

- Los datos se escriben y se leen en orden
- Para acceder a registros concretos hay que atravesar todo el archivo
- La escritura de datos se hace desde el último

## Acceso aleatorio

- Se puede acceder a cualquier registro para modificarlo o leerlo
- Se puede acceder a registros concretos si conocemos el tamaño
- Se permite la modificación de datos intermedios

# OPERACIONES SOBRE FICHEROS

## Operación

Consultas

Altas

## Secuencial

Es necesario comenzar desde el principio y leer todos hasta encontrar el registro buscado

Se añadiría tras el último registro al final del documento

## Aleatorio

Debemos conocer el identificador y aplicar la fórmula:  
 $\text{tamaño\_del\_registro} * \text{id}$

De la misma manera que una consulta, se buscaría su posición y se insertarían los datos en la posición, si ya está ocupada por otros datos, al final del documento

# OPERACIONES SOBRE FICHEROS

## Operación

Bajas

Modificacio  
nes

## Secuencial

Se escribirían todos los registros en otro fichero menos el que queramos borrar

Similar a bajas, se deberá escribir otro fichero con el registro modificado

## Aleatorio

$\text{tamaño\_del\_registro} * \text{id}$  para encontrarlo y lo “desactivamos” con un campo switch on/off

$\text{tamaño\_del\_registro} * \text{id}$  para encontrarlo, modificamos los datos necesarios y reescribimos

# ACCESO SECUENCIAL

## VENTAJAS

- Aprovechan bien el espacio
- Útiles al tratar con datos que se leen secuencialmente (backups)
- Rápido acceso al siguiente registro

## DESVENTAJAS

- No se puede acceder a un registro concreto
- El proceso de actualización puede ser muy tedioso y consumir tiempo de ejecución

# ACCESO ALEATORIO

## VENTAJAS

- Rápido acceso a cualquier registro para su escritura, lectura o modificación

## DESVENTAJAS

- Posibles colisiones entre registros a la hora de colocarlos en su posición o en la zona de excedentes
- Desaprovecha el espacio al dejar huecos en blanco

# ESCRITURA Y LECTURA DE FICHEROS

- Fichero de texto por caracteres  
(FileReader y FileWriter)
- Fichero de texto por buffer  
(BufferedReader y BufferedWriter)
- Fichero binario (primitivas)  
(DataInputStream y DataOutputStream)
- Fichero binario (objetos)  
(ObjectInputStream y ObjectOutputStream)
- Fichero aleatorio

## LECTURA POR CARACTERES

Métodos:

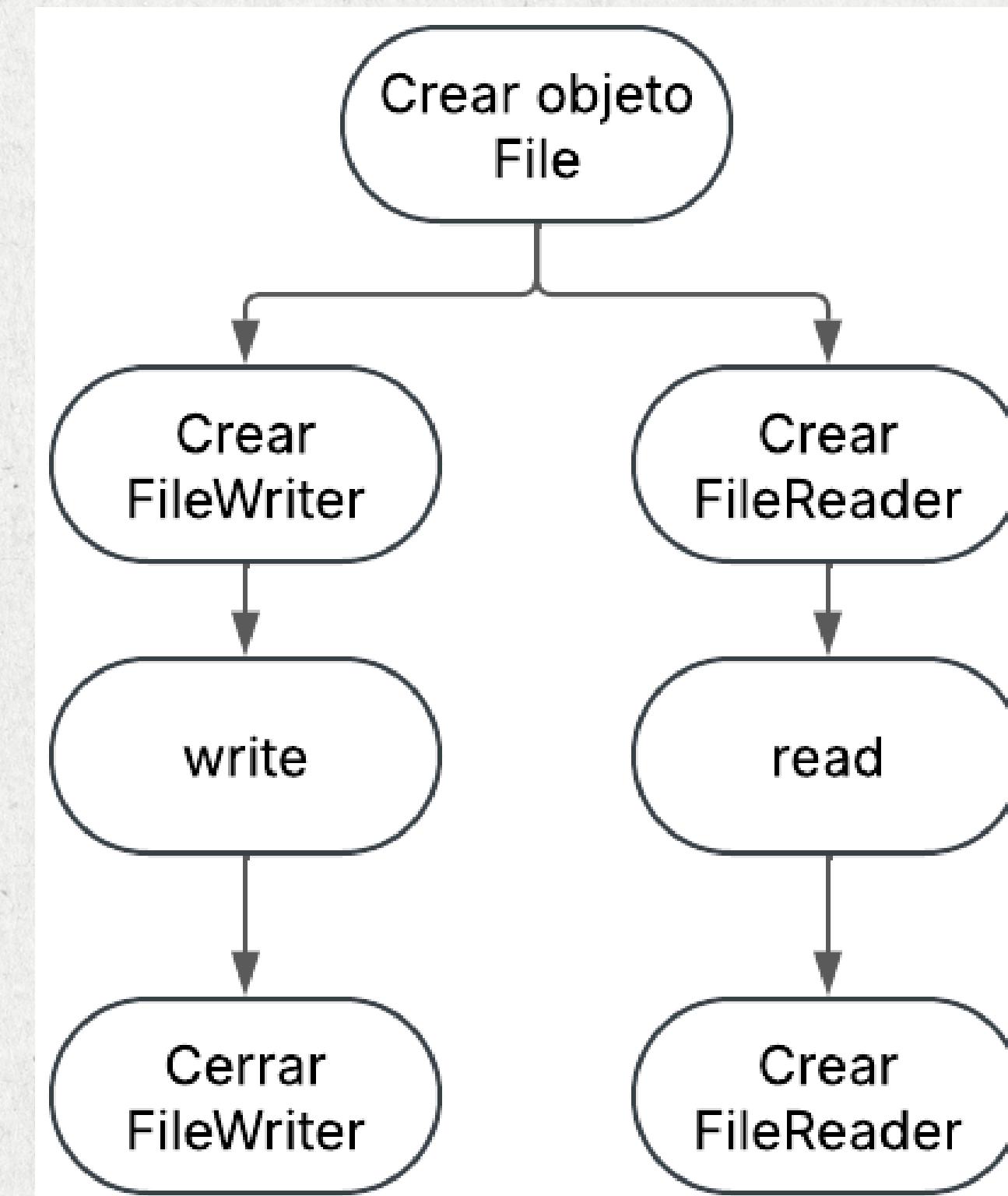
- Constructor: FileReader (Objeto\_File\_a\_leer)
- int read()
- int read(char[ ] buf)
- int read(char[ ] buf, int desplazamiento, int n)

# ESCRITURA POR CARACTERES

Métodos:

- Constructor: FileWriter (Objeto\_File\_a\_escribir)
- void write(int c)
- void write(char[] buf)
- void write(char[] buf, int desplazamiento, int n)
- void write(String str)
- void append(char c)

# LECTURA / ESCRITURA POR CARACTERES



# SECCIÓN DE LA EXCEPCIÓN

- Abrir ficheros con la clase FileReader puede generar la excepción FileNotFoundException
- Usar FileWriter puede generar la excepción IOException (In/Out)

**AMBAS DEBEN MANEJARSE CON  
TRY/CATCH O THROW**

# EJERCICIOS

**Ejercicio 1:**  
Escribe una frase en  
un archivo mediante  
FileWriter para,  
después, leer el fichero  
y mostrar su  
información por  
pantalla

## LECTURA POR BUFFER

Métodos:

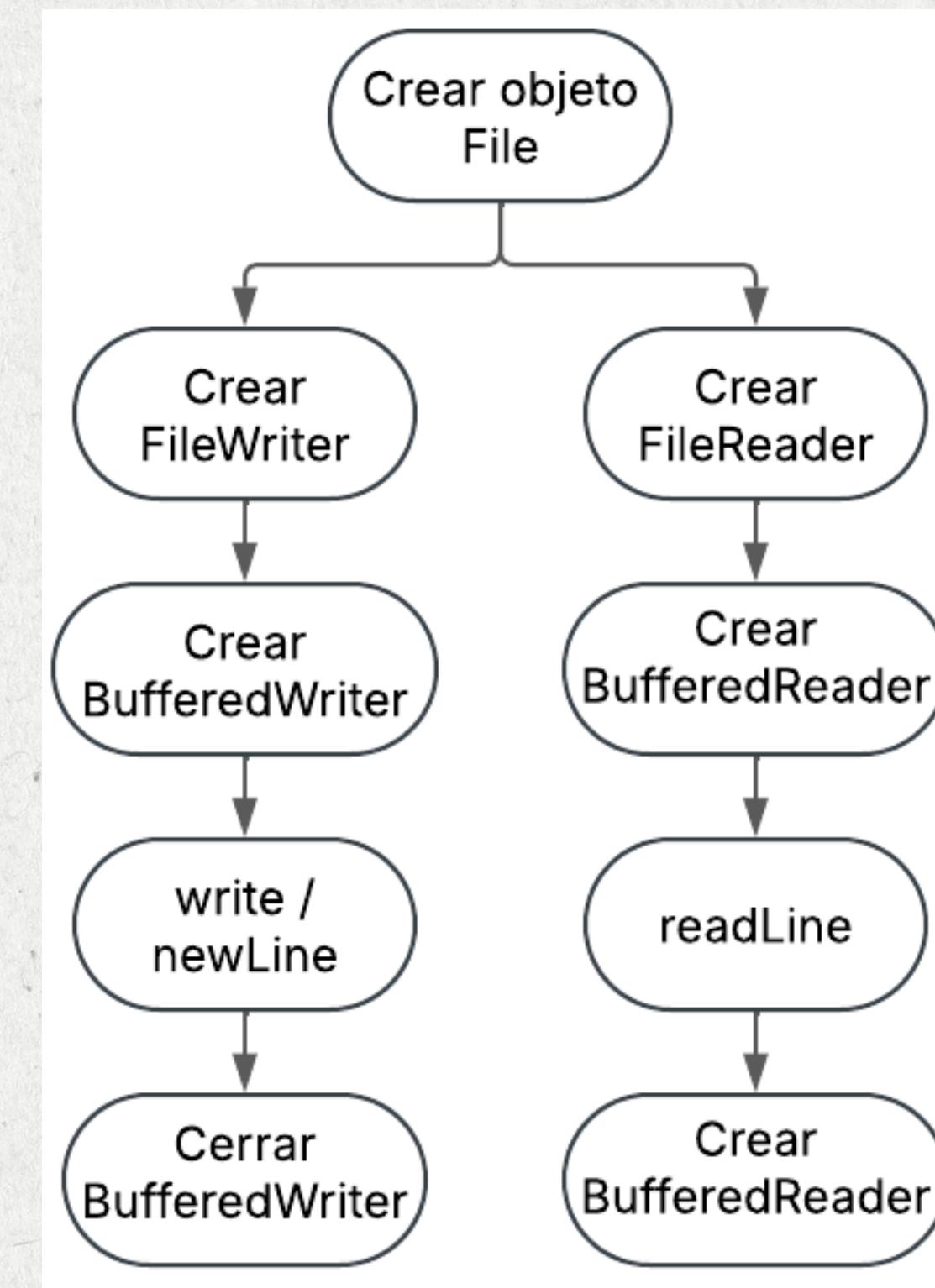
- Constructor: BufferedReader  
(Objeto\_FileReader)
- readLine()

## ESCRITURA POR BUFFER

Métodos:

- Constructor: BufferedWriter (Objeto\_FileWriter)
- void write(int c)
- void write(String str)
- void newLine()

## LECTURA/ESCRITURA POR BUFFER



# EJERCICIOS

**Ejercicio 2:**  
Escribe un párrafo de varias líneas en un archivo para, después, leer el fichero y mostrar su información por pantalla.

# EJERCICIOS

## Ejercicio 3:

Se os ha proporcionado un fichero de texto “libros.txt” en el que cada línea corresponde con el isbn, el nombre de un libro, su autor, la editorial y el año de publicación separados por comas. Lee el fichero desde Java y escribe toda la información en otro nuevo fichero con el formato:

-----  
ISBN

Título (Autor)

Editorial, Año de publicación

-----