Acceso a Datos: UD 1 - Ejercicios

# 1. Ejercicios Ficheros (File)

**1.1 Instancia un fichero y determina si existe o no existe. Muestra su nombre, longitud y ruta absoluta**

**1.2 Instancia un directorio y determina si existe o no existe. Muestra su nombre, longitud y ruta absoluta**

**1.3 Crea un fichero y determina si se ha creado o si ya existía.**

Versión con scanner

Versión con paso de parámetros en tiempo de ejecución

**1.4 Crea una carpeta y determina si se ha creado o si ya existía.**

1.5 Elimina un fichero y determina si se ha eliminado o no

1.6 Elimina un directorio y determina si se ha eliminado o no (sólo se puede eliminar un directorio si esta vacío)

1.7 Dado un directorio muestra el nombre de todos los archivos que contenga, o si está vacío

**1.8 Lista todos los archivos de una carpeta y sus subcarpetas, con su ruta desde la ubicación actual. No deben aparecer las carpetas**

# 2. Ejercicios Streams de caracteres (FileReader y FileWriter)

2.1 Crea un fichero de texto con algún editor de texto y después realiza una clase de Java que visualice su contenido. El programa recibe el nombre del archivo en tiempo de ejecución por línea de comandos.

2.2 Crea una clase de Java que permita buscar una vocal en un fichero de texto y muestre el número de veces que se repite.

2.3 Crea una clase de Java que cree ‘n’ archivos, nombre1.txt, nombre2.txt,.... nombreN.txt. El contenido de cada archivo contendrá la frase “Este es el fichero nombreN.txt”.

2.4 Escribe una clase Java que pida una serie de frases por teclado hasta que se inserte como frase la palabra “fin”. Dichas frases deberán guardarse en un fichero de texto. A continuación, el programa visualizará el contenido del fichero, frase por frase.

2.5 Una clase que dado un archivo .java cambie los tabuladores por 2 espacios (\t es un tabulador)

2.6 Calcular la media de los números almacenados en un fichero. Cada número está en una línea. El nombre del fichero se pasa como argumento de la línea de comandos

# 3. Ejercicios Streams de caracteres (Scanner y PrintWriter)

3.1 Escribir en un fichero utilizando PrintWriter

3.2 Leer un fichero utilizando un Scanner

3.3 Escribir y leer un fichero utilizando PrintWriter y Scanner respectivamente

# 4. Ejercicios Streams de caracteres (BufferedReader y BufferedWriter)

4.1 Crea un método que reciba el nombre de un archivo, y un entero. El método deberá crear el archivo y escribir ‘n’ líneas dentro. Cada línea deberá tener escrito “Esta es la línea n”, sustituyendo ‘n’ por el número de la línea. Utiliza un buffer para realizar la escritura de forma más eficiente.

4.2 Crea un método que reciba un archivo y devuelva la cantidad de palabras de dicho archivo. Utiliza un buffer para realizar la lectura de forma más eficiente.

Se puede usar split()

4.3 Crea un método que permita buscar palabras en un fichero de texto. Se debe mostrar el número de ocurrencias de dicha palabra. Utiliza un búfer para la lectura.

4.4 Crea un método que encripta y otro que desencripta el contenido de un fichero de texto utilizando el código césar. El cifrado César es un tipo de cifrado de sustitución en el que cada letra en el texto se desplaza un cierto número de lugares en el alfabeto. Por ejemplo, con un desplazamiento de 2, 'A' se reemplazaría por 'C', 'B' se convertiría en 'D'. Con desplazamiento 5, 'C' se reemplazaría por 'H', 'E' se convertiría en 'J', etc

Si lo grabais caracter a caracter como INT en UNICODE podéis sumarle o restarle el desplazamiento

4.5 Crea un método que reciba 2 archivos de texto y combine el contenido de los 2 archivos.

# 5. Ejercicios Streams binarios

5.1 Leemos un archivo byte a byte

5.2 Escribimos y leemos un archivo con con DataInputStream y distintos tipos de datos

5.3 Partiendo de una clase Persona con los atributos Nombre, Edad, CiudadNacimiento, generá un menú que muestre estas 2 opciones.

1. Escribir tus datos en el objeto y guardarlos en un archivo binario.
2. Recuperar tus datos de un archivo binario y cargarlos en el objeto Persona y mostrar datos del objeto Persona.

5.4 Una clase que dado el nombre de un archivo binario, y un nuevo nombre, copie ese archivo con el nombre dado al archivo con el nuevo nombre. (Un programa que duplique un archivo binario).

5.5 (No realizar: Pendiente de investigar si es posible, o suministrar más información) pero se puede intentar los siguientes ejemplos:

* Leer etiquetas idtag de mp3
* Información de un archivo BMP

# 6. Ejercicios Acceso aleatorio

6.1 Hacer ejemplo de los apuntes, y de momento no mando ejercicio de clase

Durante el repaso de cara al primer trimestre de esta unidad hemos realizado un ejercicio.

# 7. Ejercicios Serialización

7.1 Creo el objeto alumno y serializo varios alumnos, luego deserializar e imprime por pantalla para comprobar que se han guardado correctamente.

7.2 Realizo la misma acción pero con el objeto Aula que contiene un ArrayList de Alumnos

7.3 Escribe un programa Java que pida al usuario los datos de una serie de empleados y que los guarde en un fichero binario. El usuario introducirá datos de empleados hasta que indique que no quiere añadir ninguno más. Los empleados se almacenarán en un fichero binario como objetos de una clase Empleado que deberá contener nombre, sueldo, año de nacimiento y antigüedad.

* Leer un objeto serializado en un archivo binario
* Guardar en un binario una colección de objetos serializado
* Leer una colección de objetos serializados en un archivo binario

# 8. Ejercicios XML

**8.1 Leer un archivo XML**

**8.2 Guardar una colección en un archivo XML**