

UNIDAD 11

FICHEROS

EJERCICIOS (B)

PROGRAMACIÓN
CFGS DAW

UD11. FICHEROS

EJERCICIOS (parte B) - LECTURA Y ESCRITURA DE FICHEROS

Para probar estos ejercicios utilizar el archivo “Documentos.zip”.

Ejercicio B1 - Máximo y mínimo

Implementa un programa que muestre por pantalla los valores máximos y mínimos del archivo ‘numeros.txt’.

Ejercicio B2 - Notas de alumnos

El archivo ‘alumnos_notas.txt’ contiene una lista de 10 alumnos y las notas que han obtenido en cada asignatura. El número de asignaturas de cada alumno es variable. Implementa un programa que muestre por pantalla la nota media de cada alumno junto a su nombre y apellido, ordenado por nota media de mayor a menor.

Ejercicio B3 - Ordenando archivos

Implementa un programa que pida al usuario un nombre de archivo A para lectura y otro nombre de archivo B para escritura. Leerá el contenido del archivo A (por ejemplo ‘usa_personas.txt’) y lo escribirá ordenado alfabéticamente en B (por ejemplo ‘usa_personas_sorted.txt’).

Ejercicio B4 - Nombre y apellidos

Implementa un programa que genere aleatoriamente nombres de persona (combinando nombres y apellidos de ‘usa_nombres.txt’ y ‘usa_apellidos.txt’). Se le pedirá al usuario cuántos nombres de persona desea generar y a qué archivo **añadirlos** (por ejemplo ‘usa_personas.txt’).

Ejercicio B5 - Diccionario

Implementa un programa que cree la carpeta ‘Diccionario’ con tantos archivos como letras del abecedario (A.txt, B.txt... Z.txt). Introducirá en cada archivo las palabras de ‘diccionario.txt’ que comiencen por dicha letra.

Ejercicio B6 - Búsqueda en PI

Implementa un programa que pida al usuario un número de cualquier longitud, como por ejemplo “1234”, y le diga al usuario si dicho número aparece en el primer millón de decimales del nº pi (están en el archivo ‘pi-million.txt’). No está permitido utilizar ninguna librería ni clase ni método que realice la búsqueda. Debes implementar el algoritmo de búsqueda tú.

Ejercicio B7 - Estadísticas

Implementa un programa que lea un documento de texto y muestre por pantalla algunos datos estadísticos: nº de líneas, nº de palabras, nº de caracteres y cuáles son las 10 palabras más comunes (y cuántas veces aparecen). Prueba el programa con los archivos de la carpeta ‘Libros’.

NOTA: Para llevar la cuenta de cuántas veces aparece cada palabra puedes utilizar una [HashTable](#). Una tabla hash es una estructura de datos tipo colección (como el ArrayList), que [permite almacenar pares clave-valor](#). Por ejemplo {“elefante”, 5} o {“casa”, 10} son pares <String,Integer> que asocian una palabra (clave) con un nº entero (valor).