

## **11.11 LABORATORIO 11: ARCHIVOS**

### **Objetivos**

- ✓ Familiarizarse con la escritura de archivos,
- ✓ Practicar la lectura y escritura de estructuras de datos sobre archivos

#### **1. Escribir frases**

El ejercicio consiste en implementar una función que le pregunte al usuario 10 frases distintas y las escriba en un archivo.

El archivo debe encontrarse en la misma ruta que el script que posee la función.

#### **2. Biblioteca e historial**

Una biblioteca requiere de un programa que controle sus préstamos y que pueda a partir del historial de ellos obtener información importante.

La biblioteca posee dos archivos de texto: uno posee todos los nombres de los libros y el otro posee los autores. Ambos archivos se encuentran ordenados, por lo que el libro que se encuentra en la línea 56 posee su autor en la línea 56 del otro archivo.

La función básica para el manejo de préstamos consiste en solicitarle al usuario el nombre del libro que va a solicitar, más el autor del mismo, el nombre del día de la semana en que está sacando el libro (lunes, martes, ...) y la fecha. Se debe asegurar que el libro pedido junto con su autor correspondan a uno que realmente se encuentre

dentro de los archivos (ambos datos deben coincidir en la misma línea).

La información de cada préstamo a la vez debe ir siendo colocada en otro archivo de texto (cada préstamo debe corresponder a una línea por lo que se deben utilizar tabs para que sea fácil de leer.)

Después de realizar todos los préstamos de varios meses se pueden realizar las siguientes consultas:

- Imprimir todos los libros que hayan de un autor respectivo.
- ¿Cuál es el día de la semana que llegan más personas a pedir préstamos de libros?
- ¿Cuál es el autor sobre el cual se hacen más préstamos?
- ¿Cuál es el libro sobre el cual se hacen más préstamos?

Toda la información y estructuras de datos requeridas para realizar la consulta se pueden mantener en memoria para no tener que obtenerlas a partir del archivo con el historial de los préstamos.

Implemente un menú sobre consola para correr el programa fácilmente.

Los archivos necesarios se anexan a la práctica, sus nombres son libros.txt y autores.txt.

Otros ejercicios

► 451 Diseña un programa que cuente el número de caracteres de un fichero de texto, incluyendo los saltos de línea. (El nombre del fichero se pide al usuario por teclado.)

► 452 Haz un programa que, dada una palabra y un nombre de fichero, diga si la palabra aparece o no en el fichero. (El nombre del fichero y la palabra se pedirán al usuario por teclado.)

► 453 Haz un programa que, dado un nombre de fichero, muestre cada una de sus líneas precedida por su número de línea. (El nombre del fichero se pedirá al usuario por teclado.)

► 454 Haz una *función* que, dadas la ruta de un fichero y una palabra, devuelva una lista con las líneas que contienen a dicha palabra.

Diseña a continuación un programa que lea el nombre de un fichero y tantas palabras como el usuario desee (utiliza un bucle que pregunte al usuario si desea seguir introduciendo palabras). Para cada palabra, el programa mostrará las líneas que contienen dicha palabra en el fichero.

► 455 Haz un programa que muestre por pantalla la línea más larga de un fichero. Si hay más de una línea con la longitud de la más larga, el programa mostrará únicamente la primera de ellas. (El nombre del fichero se pedirá al usuario por teclado.)

► 456 Haz un programa que muestre por pantalla todas las líneas más largas de un fichero. (El nombre del fichero se pedirá al usuario por teclado.) ¿Eres capaz de hacer que el programa lea una sola vez el fichero?

► 457 La orden **head** (‘cabeceras’, en inglés) de Unix muestra las 10 primeras líneas de un fichero. Haz un programa **head.py** que muestre por pantalla las 10 primeras líneas de un fichero. (El nombre del fichero se pedirá al usuario por teclado.)

► 458 En realidad, la orden **head** de Unix muestra las  $n$  primeras líneas de un fichero, donde  $n$  es un número suministrado por el usuario. Modifica **head.py** para que también pida el valor de  $n$  y muestre por pantalla las  $n$  primeras líneas del fichero.

► 465 Haz un programa que lea un fichero de texto que puede contener vocales acentuadas y muestre por pantalla una versión del mismo en el que cada vocal acentuada ha sido sustituida por la misma vocal sin acentuar.

► 468 Diseña un programa que obtenga los 100 primeros números primos y los almacene en un fichero de texto llamado **primos.txt**.