

EJERCICIOS. Ficheros**Relación Nº: 8**

- Haz un programa que use la clase `Path` y sus métodos estáticos para obtener y preparar información de acceso a un fichero que está en `c:/directorio1/directorio2/directorio3/texto.txt`
- Haz un programa que copie el contenido de un fichero en otro. Ambos ficheros se pasan desde la línea de comandos según el siguiente formato:

NombrePrograma FicheroOrigen FicheroDestino

FicheroOrigen y FicheroDestino implican una ruta completa y si no se indica es el directorio actual.

- Haz un programa que copie el contenido de un fichero en otro sin incluir un carácter pasado desde la consola. Ambos ficheros y el carácter a excluir se pasan desde la línea de comandos según el siguiente formato:

NombrePrograma FicheroDestino FicheroFuente Caracter

- Haz un programa que muestre por pantalla en contenido de un fichero de texto pasado por la consola.

NombrePrograma [Unidad:\Ruta\]Fichero_de_Texto

- Crea una función que reciba como parámetros dos nombres de ficheros de texto y los concatene dejando el resultado guardado en el primero.
- Escribe un programa que contenga dos funciones: `CrearFichero(string fi)` y `ListarFichero(string fi)` donde:
 - La primera función almacena en un fichero de texto (fi) los nombres y apellidos de los alumnos. En nombre del archivo se le pasa a la función al llamarla.
 - Muestra por pantalla el contenido del fichero (fi) pasado.
- Escribe un programa que tome todos los ficheros dados como argumentos en la línea de comandos y muestre por pantalla su contenido uno tras otro, separando cada contenido por una línea horizontal y el nombre del fichero que se mostrará.
- Haz un programa que muestre por pantalla el contenido de un fichero de texto al revés, desde el final hasta el principio.
- Haz un programa que gestione un fichero en el que vaya guardando los datos: Nombre usuario, fecha, hora y tiempo de conexión, para cada una de las veces que se ha ejecutado.
- Realizar un programa de gestión completo que permita la manipulación de datos de un almacén. Constará de un MENÚ PRINCIPAL como el siguiente. Usar la macro `ESC#` para salir, en caso de hacerlo con componentes gráficos.

MENÚ PRINCIPAL

=====

- | | |
|---|---|
| 1. Alta de artículos. | No se permiten repeticiones |
| 2. Baja de artículos. | Mostrará el artículo a borrar y pedirá confirmación. |
| 3. Consultar un artículo. | Usando el código del artículo |
| 4. Modificar un artículo. | Usando el código del artículo |
| 5. Listar artículos ordenados por código. | |
| 6. Listar artículos ordenados por nombre. | |
| 7. Imprimir | |
| 8. Generar documento HTML | |
| 9. Crear fichero | Crearé el fichero sin datos, si existe lo borrará. Poner un mensaje para que avise antes de borrar los datos y pida confirmación. |
| 0 – Para salir del programa. | |

Los datos de la ficha de artículos son:

cod_art	AutoNumérico	Código del artículo de longitud 3
nombre_art	Char(15)	Nombre del artículo
precio	real	Precio de coste del artículo
pvp	real	Precio de venta al público del artículo.
existencias	numérico	Nº de unidades en el almacén
comentario	char(50)	Campo para comentarios

NOTAS:

- El campo pvp se rellenará de forma automática, será un incremento del 25% sobre el precio.
- En los listados no debe de aparecer el campo comentario, pero en las consultas si lo hará.
- Usar la última línea de la pantalla para mostrar los mensajes de ayuda.
- Los mensajes de ayuda no podrán ser de más de 60 caracteres y una vez que han cumplido su misión se deben borrar.
- Si queremos mejorar el interfaz con el usuario podemos utilizar la función que dibuja una ventana o marco para incluir dentro las opciones del menú principal y los datos de los registros.