



# LABORATORIO DE PROGRAMACIÓN

E.T.S.I. Informática Gestión

Examen de Septiembre: 14-09-2007 Examen Tradicional

Duración del Examen: 4 horas

APELLIDOS \_\_\_\_\_ NOMBRE \_\_\_\_\_

ORDENADOR \_\_\_\_\_

La revista de economía *MisEuros*, desea que le hagamos una aplicación informática para gestionar la lista de los empresarios más ricos en el ramo de los supermercados. Sabemos que la riqueza de un empresario se mide por el número de acciones que posee en cada una de las 4 empresas más importantes del país en el sector (*HiperCont*, *CorreFur*, *ElRoski*, *MercaPoma*) multiplicado por el valor de cada acción. Se pide:

- a.) Elaborar un módulo llamado *MCadena* (ficheros *MCadena.h* y *MCadena.cpp*) que :
- Exporte el tipo *TCadena* que será un **array de caracteres**.
  - Exporte las constantes *MAXCAD* (40), *FINCAD* (carácter 0) y *ENTER*(' \n' )..
  - Exporte una función que nos permita copiar 2 cadenas.
  - Exporte una función que nos permita saber si 2 cadenas son iguales.
- b.) Elaborar un módulo llamado *MEmpresa* (ficheros *MEmpresa.h* y *MEmpresa.cpp*) que :
- Exporte el tipo *TEmpresa* que será un **enumerado** con los valores de las empresas.
  - Exporta la función *SucEmpresa* que retorna el sucesor del enumerado.
  - Exporte una función/procedimiento para convertir un dato de tipo *TCadena* en el valor de tipo *TEmpresa* que le corresponde.
  - Exporte una función/procedimiento para convertir un dato de tipo *TEmpresa* a la cadena de caracteres (*TCadena*) que lo representa.
- c.) Elaborar un módulo llamado *MListaEmpresarios* (ficheros *MListaEmpresarios.h* y *MListaEmpresarios.cpp*) que :
- Exporte el tipo *TEmpresario* que será un registro con la siguiente información de cada empresario: nombre, edad y número de acciones que posee de cada una de las empresas (un array *TAcciones* cuyo tipo también será exportado).
  - Exporte el tipo *TError* que será un enumerado con los posibles valores de error.
  - Exporte el tipo *TListaEmpresarios* que será una **lista enlazada y ordenada crecientemente (de menor a mayor) por la edad del empresario**.
  - Exporte las funciones:
    - *CrearLista*: Crea una lista vacía.
    - *DestruirLista*: Destruye una lista.
    - *ListaLlena*: Nos dice si una lista está llena.
    - *ListaVacía*: Nos dice si una lista está vacía.
    - *InsertarEmpresario*: Inserta un empresario en la lista.
    - *ExtraerEmpresario*: Nos devuelve la información y elimina de la lista el empresario más joven de la lista.
- d.) Realizar una aplicación llamada *MisEuros.cpp* que usando los módulos anteriores implemente las siguientes opciones de menú:

Elaborado por: <Nombre>                      < Apellidos>  
Fecha: <Fecha>                                      Ordenador : <Ordenador>

M E N U            P R I N C I P A L  
=====

- A.- Introducir un empresario nuevo desde Teclado.
- B.- Mostrar el empresario más Rico.
- C.- Mostrar por pantalla la Información de todos los empresarios.
- D.- Borrar un empresario.
- E.- Salvar los Datos en un fichero Binario.
- F.- Cargar Datos desde un Fichero Binario.
- X.- Salir del Programa.

Introduzca Opción: \_

**A. Introducir un empresario nuevo desde Teclado.**

Se pedirá al usuario todos los datos del empresario y se insertará en la lista.

**B. Mostrar empresario más Rico.**

Se pedirá al usuario el nombre de un fichero de texto donde se encuentre el valor de las acciones de cada empresa con el siguiente formato: <nombre\_empresa> <valor>, donde el valor es un número real que indica el valor en euros de 1 acción de esa empresa. Y se mostrará por pantalla los datos del empresario más rico (SIN QUE DEJE DE ESTAR EN LA LISTA). Ejemplo de Fichero:

```
HiperCont 2.67
CorreFur 1.2
ElRoski 0.98
MercaPoma 5.3
```

**C. Mostrar por pantalla la Información de todos los Empresarios.**

Se mostrará por pantalla toda la información de todos los empresarios ordenada **decrecientemente** (de mayor a menor) por edad y sin que se modifique el contenido de la lista.

**D. Borrar un Empresario.**

Se pedirá al usuario el nombre del empresario y se borrará toda su información de la lista. Si no existiera dicho empresario no se hará nada y se informará del error.

**E. Salvar los Datos en un fichero Binario.**

Se pedirá al usuario el nombre de un fichero y se grabará todo el contenido de la lista en él y sin que se modifique el contenido de la lista. Si existiera previamente un fichero con dicho nombre se borrará su contenido y si no existiera se creará.

**F. Cargar Datos desde un Fichero Binario.**

Se pedirá al usuario el nombre del fichero y se insertará su contenido en la lista (la lista puede o no estar vacía). Si el fichero no existiera no se insertará nada.

**X. Salir del Programa.** En esta opción se pedirá confirmación de salida. Si es afirmativa, se terminará el programa, liberando todos los recursos que hayan sido reservados y si es negativa, se volverá al menú principal.

**EL AÑADIR OPERACIONES EN LA PARTE DE DEFINICIÓN DE ALGÚN MÓDULO, ASÍ COMO EL USO DE DETALLES DE IMPLEMENTACIÓN DE LOS MODULOS DENTRO DEL PROGRAMA PRINCIPAL SERÁ CAUSA DE SUSPENSO (ES DECIR, ESTÁ PROHIBIDO USAR 'I->', NULL, ETC FUERA DE LA IMPLEMENTACIÓN DEL MÓDULO DE LISTAS).**

**Notas:**

1. **Para Aprobar** será necesario el correcto funcionamiento de las opciones A, B, E, F y X.
2. Las opciones C y D servirán para obtener una mayor puntuación.
3. Todas las cadenas de caracteres tendrán un máximo de 40 caracteres.
4. Es obligatorio trabajar en el directorio **C:\LPGSEP07**. Si no existe se creará
5. No olvide escribir su nombre y el código de ordenador en el menú del programa principal.
6. Se recomienda y valora la buena descomposición del programa principal en procedimientos y funciones, así como el uso de procedimientos y funciones auxiliares dentro de la implementación de los módulos cuando estas sean necesarias.
7. Se deberá hacer un tratamiento de errores con Tipos Enumerados, donde los mensajes de error se muestren desde el programa principal.
8. Las opciones del menú han de ser letras. No se distinguirá entre mayúsculas y minúsculas.
9. No se admitirán efectos laterales.
10. Recordad que todo fichero de cabecera debe llevar su pragma `#ifndef ... #endif`

Ejemplo: Fichero: Cabecera.h

```
#ifndef _Cabecera_h_
#define _Cabecera_h_
    // Contenido del archivo de Cabecera
#endif
```