



Departamento de Lenguajes  
y Ciencias de la Computación  
UNIVERSIDAD DE MÁLAGA

# LABORATORIO DE PROGRAMACIÓN

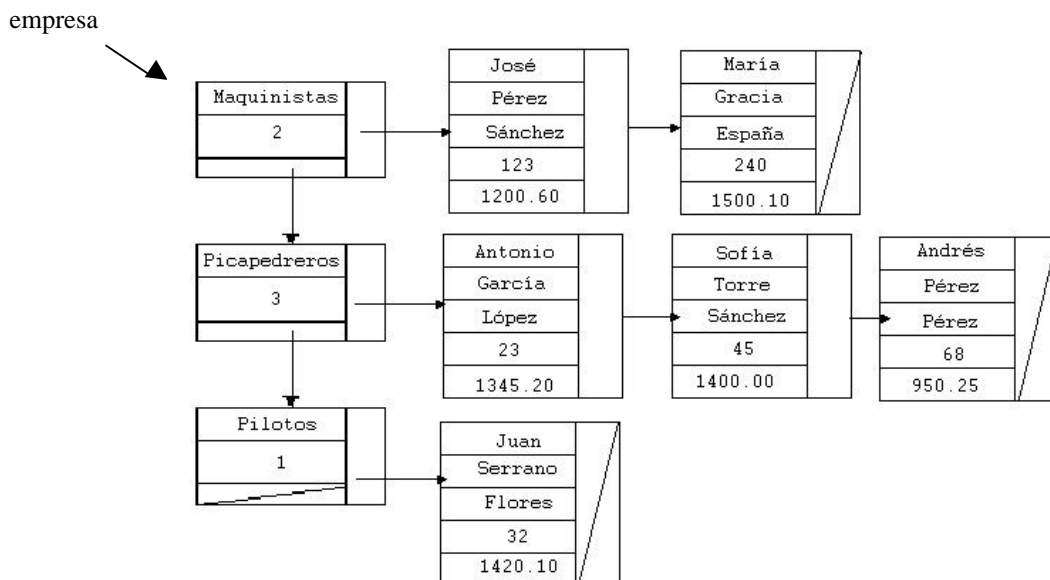
E.T.S.I.Informática- GESTIÓN. 13/09/2002

Duración del Examen: 4 horas

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

ORDENADOR \_\_\_\_\_

La empresa *Demoliciones Twin Towers, S.A.* para conmemorar el primer aniversario de su creación ha decidido informatizar la gestión de su personal. De cada trabajador, debemos almacenar su nombre y dos apellidos (todas cadenas de un máximo de 20 de caracteres), su número de trabajador (número natural) y su salario en euros (número real). Los trabajadores están agrupados por departamentos (existiendo un número indeterminado de departamentos). Por cada departamento se almacenará su nombre, número de empleados que hay en él y una lista enlazada con los empleados que pertenecen a dicho departamento ordenados por su número de trabajador (no puede haber un empleado que esté en más de un departamento). Por tanto, la estructura que almacene la información de la empresa (tipo *TEmpresa*) tendrá la estructura siguiente:



Se pide el desarrollo de un módulo llamado *MEmpresa* (cuya definición se llamará *MEmpresa.h* y su implementación se llamará *MEmpresa.cpp*) para el manejo de la estructura de datos especificada con anterioridad y que tendrá las siguientes operaciones **Y NINGUNA OTRA OPERACIÓN** :

- **CrearEmpresa**: Creará un empresa vacía, es decir, sin trabajadores ni departamentos.
- **EmpresaVacía**: Nos dirá si una empresa está vacía, es decir, sin trabajadores ni departamentos.
- **InsertarEmpleado**: Dados los datos del empleado y el nombre del departamento, se insertará de forma ordenada y atendiendo al número de trabajador, el empleado en el departamento que corresponda. Se supone que no hay dos trabajadores con el mismo número de trabajador. Si dicho departamento no existe, se creará. Recuerde que hay que actualizar el número de empleados del departamento.
- **BorrarEmpleado**: Dado el número de trabajador, se borrará de la estructura el empleado correspondiente. Si el trabajador no existe, no se hará nada ni se informará de nada. Si el departamento queda sin trabajadores tras la eliminación, también debe ser eliminado de la estructura.
- **BuscarEmpleado**: Dado el número de trabajador de un empleado, se devolverá si existe o no dicho empleado y en caso de que exista en la estructura, todos los datos del empleado, así como el nombre del departamento al que pertenece.
- **RangoEmpleados**: Devolverá el menor y el mayor número de empleado almacenado en la estructura. En caso de estar vacía devolverá 0 y 0 y en caso de haber sólo 1 empleado, su número será el mayor y el menor.
- **DestruirEmpresa**: Destruirá toda la estructura, libreando todos los recursos y memoria que hayan sido reservados.

El programa principal (towers.cpp), deberá presentar el siguiente menú, cuyas opciones se especifican a continuación:

Nombre: (Apellidos, Nombre)	Curso: 1º
Especialidad: Gestión	Grupo: A/B/C
Puesto: Número de ordenador	Fecha: 13/09/02

Demoliciones Twin Towers S.A.  
=====

A. Insertar Empleado.  
B. Buscar Empleado.  
C. Visualizar Empresa.  
D. Salvar Empresa a Fichero  
E. Borrar Empleado.  
F. Cargar Empresa desde Fichero

X. Salir del Programa

Introduzca Opción:

- A. Insertar Empleado.** Se pedirá por pantalla el número de trabajador y se comprobará que no existe un trabajador ya almacenado con ese número (en caso de existir se informará del error y se volverá a pedir el número). Posteriormente, se pedirán los datos del trabajador, así como el nombre del departamento en que va a trabajar.
- B. Buscar Empleado.** Se solicitará el número de empleado y se mostrará por pantalla todos los datos del empleado, así como el nombre del departamento a que pertenece. Si el trabajador no existe se informará de ello.
- C. Visualizar Empresa.** Se mostrará por pantalla todos los datos de todos los empleados que pertenecen a la empresa, así como el nombre del departamento a que pertenecen. Los trabajadores deben aparecer ordenados por su número de trabajador.
- D. Salvar Empresa a Fichero.** Se pedirá el nombre de un fichero y se salvarán con el siguiente formato de texto (SP = ESPACIO):  
<n\_emp><SP><nombre><SP><apl><SP><ape2><SP><salario><SP><depto><SP><ENTER>  
Por Ejemplo, la estructura del dibujo se salvaría: empresa.txt  
23 Antonio García López 1345.20 Picapedreros  
32 Juan Serrano Flores 1420.10 Pilotos  
45 Sofía Torre Sánchez 1400.0 Picapedreros  
68 Andrés Pérez Pérez 950.25 Picapedreros  
123 José Pérez Sánchez 1200.60 Maquinistas  
240 María Gracia España 1500.10 Maquinistas
- E. Borrar Empleado.** Se solicitará el número de empleado y se mostrará por pantalla todos los datos del empleado, así como el nombre del departamento a que pertenece. Si el trabajador no existe se informará de ello. Si existe, se pedirá confirmación de borrado y en caso de ser afirmativa, se eliminará.
- F. Cargar Empresa desde Fichero.** Se solicitará el nombre de un fichero de texto con el formato anterior y se insertarán los trabajadores en la empresa. (Puede suponerse que no hay números de empleado duplicados).
- X. Salir del Programa.** Se pedirá confirmación de salida y en caso de ser afirmativa, se saldrá del programa mostrando el nombre, apellidos y curso del alumno.

#### Notas:

- SE RECUERDA QUE EL USO DE DETALLES DE IMPLEMENTACIÓN DE UN MÓDULO, DENTRO DE LA IMPLEMENTACIÓN DE OTRO MÓDULO O DEL PROGRAMA PRINCIPAL, SERÁ CAUSA DE SUSPENSO.**
- Se debe hacer una pausa después de ejecutar cada opción de menú, y borrar la pantalla tras la pausa.
- Es obligatorio trabajar en el directorio **C:\LPGSEP02**. Si no existe se creará.
- Para **Aprobar** deberá ser correcta la definición de tipos, la modularización y funcionar **CORRECTAMENTE** las Opciones A,B,C,D y X del menú.
- 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