



Departamento de Lenguajes
y Ciencias de la Computación
UNIVERSIDAD DE MÁLAGA

LABORATORIO DE PROGRAMACIÓN I

E.T.S.I.Informática- GESTIÓN/SISTEMAS . 15-02-2002

Duración del Examen: 3 horas

APELLIDOS _____ NOMBRE _____

ORDENADOR _____

La empresa *Los Planes Reformados S.A.*, ha decidido informatizar la gestión de sus trabajadores. Dicha empresa tiene un máximo de 10 trabajadores en plantilla, debiéndose almacenar la siguiente información de cada trabajador: nombre, apellidos (ambos son cadenas de un máximo de 80 caracteres), y turno (matinal, tarde, noche) que debe implementarse obligatoriamente con un tipo enumerado. La aplicación deberá mostrar en pantalla el siguiente menú:

```
Los Planes Reformados S.A
-----
Autor: <Apellidos> <Nombre>
Fecha: 15/02/02   Ordenador: XXXX

A. Insertar Trabajador.
B. Buscar Trabajador.
C. Listar Trabajadores.
D. Borrar Trabajador
E. Consulta por Campos

X. Salir del Programa.
```

Descripción de las Opciones:

- **Insertar Trabajador.** Se pedirá el número de trabajador, que debe ser un número entre el 1 y el 10 y se comprobará que no hay ningún trabajador almacenado con ese número. Si el número de trabajador no fuera válido (no esté entre 1 y 10) o ya estuviera algún trabajador almacenado con ese número, se informará del pertinente error y se volverá al programa principal. En cambio, si el número es válido, se pedirán el resto de los datos (nombre, apellidos y turno) y se insertará dicho trabajador en la estructura. Ejemplo:

```
Introduzca Número de Trabajador: 3
Nombre: Pepe
Apellidos: García Pérez
Turno: Tarde
```

- **Buscar Trabajador.** Se pedirá por teclado el número de trabajador. Si el número de trabajador no fuera válido (no esté entre 1 y 10) o no hubiera algún trabajador almacenado con ese número, se informará del pertinente error y se volverá al programa principal. En cambio, si el número es válido, se mostrará por pantalla todos los datos (nombre, apellidos y turno) de dicho trabajador. Ejemplo:

```
Introduzca Número de Trabajador: 3
Trabajador N°: 3
Nombre: Pepe
Apellidos: García Pérez
Turno: Tarde
```

- **Listar Trabajadores.** Se mostrará por pantalla todos los datos (nombre, apellidos y turno) de los trabajadores que están almacenados en la estructura precedidos de su número de trabajador. Ejemplo:

Listado de Trabajadores

Trabajador N°: 3
Nombre: Pepe
Apellidos: García Pérez
Turno: Tarde

Trabajador N°: 4
Nombre: Juan
Apellidos: Jiménez Gómez
Turno: Tarde

Trabajador N°: 7
Nombre: Pepe
Apellidos: López Tejada
Turno: Tarde

- **Borrar Trabajador.** Se pedirá por teclado el número de trabajador. Si el número de trabajador no fuera válido (no esté entre 1 y 10) o no hubiera algún trabajador almacenado con ese número, se informará del pertinente error y se volverá al programa principal. En cambio, si el número es válido, se mostrará por pantalla todos los datos (nombre, apellidos y turno) de dicho trabajador y se informará de que va a ser borrado, pasándose a eliminarlo de la estructura (sin pedir conformidad). Ejemplo:

Introduzca Número de Trabajador: 3
Se procede a Borrar al Trabajador:
Trabajador N°: 3
Nombre: Pepe
Apellidos: García Pérez
Turno: Tarde

- **Consulta por Campos.** Se leerá una cadena de caracteres que contendrá 3 palabras separadas por el carácter '#'. Cada una de ellas será el valor con el que hay que comparar cada persona, permitiéndose el uso del asterisco '*' como comodín. El programa deberá mostrar en pantalla a todos los trabajadores que cumplan dicha condición de consulta. Por ejemplo, supongamos que queremos ver todos los trabajadores que se llaman Pepe y trabajan en turno de Tarde.

Introduzca la condición: Pepe##Tarde
Trabajador N°: 3
Nombre: Pepe
Apellidos: García Pérez
Turno: Tarde
Trabajador N°: 7
Nombre: Pepe
Apellidos: López Tejada
Turno: Tarde

- **Salir del Programa.** Se pedirá al usuario una confirmación de abandono del programa, saliendo del mismo en caso de que sea afirmativa y mostrando el nombre, apellidos y curso del autor del programa o volviendo al menú principal si no lo es.

Se valorará:

1. La corrección del programa.
2. Una buena modularización (uso de procedimientos/funciones)
3. Uso de tipos adecuados.
4. La ausencia de efectos laterales.
5. La legibilidad del código.
6. La presencia y claridad de mensajes de salida resultado de cada operación.

Notas:

1. Se debe hacer una pausa después de ejecutar cada opción de menú, y borrar la pantalla tras la pausa.
2. Se puede utilizar cualquier subrutina de los módulos estándares de TOPSPEED EXCEPTO LA LIBRERÍA STR
3. Es obligatorio trabajar en el directorio C:\LP1. Si no existe se creará
4. Para Aprobar deberá ser correcta la definición de tipos, la modularización y funcionar CORRECTAMENTE las Opciones A, B, C, D y X del menú.