



<b>Nombre:</b>			
<b>DNI/NIE:</b>		<b>Equipo:</b>	

El departamento de lenguajes y ciencias de la computación nos ha pedido que desarrollemos un programa capaz de gestionar peticiones de reserva de laboratorios para la realización de exámenes. Para ello el sistema deberá mantener una lista con las diferentes peticiones de reservas, cada una de las cuales estará formada por el nombre de la asignatura, número de alumnos estimado que se van a presentar, el día de realización del examen (el sistema únicamente almacena las reservas de un mes por lo que no se especificará ni el año ni el mes, sino únicamente el número del día) y si es en turno de mañana, tarde o de día completo, y finalmente como se pretende que el programa no sólo sirva para almacenar peticiones de reservas sino que también sirva para realizar las asignaciones de laboratorios vamos a almacenar, junto a cada petición, la lista de laboratorios asignados a dicho examen. De cada laboratorio almacenamos su nombre y su capacidad (número de puestos disponibles). La lista se mantendrá ordenada por el día de realización del examen.

El programa mostrará el siguiente menú:

Elaborado por: <Nombre> <Apellidos>      Curso 1º Gestión  
Fecha: 22/06/2011      Ordenador: <Ordenador>

MENU

- A. Introducir una nueva petición.
- B. Mostrar peticiones de reserva.
- C. Eliminar una petición de reserva.
- D. Añadir una asignación de laboratorio.
- E. Leer estado de fichero.
- X. Salir.

La **opción A**: solicita los datos de una nueva petición y la inserta en el sistema. La lista de asignación de laboratorios se solicita preguntando sucesivamente por nombre de laboratorio y capacidad del mismo y finalizando al introducir un nombre de laboratorio vacío.

La **opción B**: esta opción muestra las peticiones por pantalla.

La **opción C**: solicita por teclado el nombre de una asignatura, si no hay constancia de su solicitud se muestra por pantalla el mensaje "No hay ninguna petición para esta asignatura"; en caso contrario se elimina la solicitud correspondiente.

La **opción D**: nos pide el nombre de la asignatura y si no existe constancia de la petición se muestra el mensaje: "No hay ninguna petición para esta asignatura", en caso contrario nos pide el nombre y capacidad de un laboratorio (sólo uno) y lo añade a la lista de reservas de dicho laboratorio.

La **opción E**: solicita el nombre de un fichero de texto que va a contener una lista de peticiones. Si el fichero existe se borra la lista actual de peticiones y se carga la contenida en el fichero. En caso de no existir el fichero no se hace nada (la lista original de peticiones debe quedar intacta). El formato es el siguiente:

ASIGNATURA@ESTIMADOS@DIA@TURNO@NUMASIGNADOS@LAB1@CAPACIDAD1@...@LAB  
N@CAPACIDADN<ENTER>

Cada reserva irá en una línea y estará finalizada con un retorno de carro. TURNO puede ser MAÑANA, TARDE o COMPLETO. NUMASIGNADOS indica el número de laboratorios que siguen. Si el número de asignados es 0 seguirá el formato:

ASIGNATURA@ESTIMADOS@DIA@TURNO@0<ENTER>

La **opción X**: Finaliza adecuadamente el programa.

El programa principal se denominará reservas.cpp y además se incluirán los siguientes módulos:

- **MLaboratorio** definirá:
  - El tipo **Tlaboratorio**.
- **MListaLaboratorios** definirá:
  - El tipo **TListaLaboratorios**.
  - El tipo **TErrorListaLaboratorios**.
  - **CrearListaLaboratorios**: creará una lista vacía.
  - **ListaLaboratoriosVacía**: devuelve cierto si la lista está vacía.
  - **InsertarListaLaboratorios**: insertará un laboratorio al final de la lista.
  - **DestruirListaLaboratorios**: borra la lista y la deja vacía.
  - **EscribirLaboratorios**: muestra la lista de laboratorios por pantalla.
  - **LeerLaboratoriosFichero**: lee (creándola) una lista de laboratorios de un fichero.
  - **PrimerLaboratorio**: extrae (eliminándola) y devuelve el primer laboratorio de la lista.
- **MReserva** definirá:
  - El tipo **Treserva**
- **MListaReservas** definirá:
  - El tipo **TListaReservas**.
  - El tipo **TListaReservasError**.
  - **CrearListaReservas**: creará una lista vacía.
  - **ListaReservasVacía**: devuelve cierto si la lista está vacía.
  - **InsertarListaReservas**: insertará una reserva en la lista.
  - **DestruirListaReservas**: borra la lista y la deja vacía.
  - **EliminarReserva**: dado el nombre de una asignatura elimina la reserva.
  - **LeerReservas**: lee una lista de reservas de teclado.
  - **EscribirReservas**: escribe la lista de reservas por pantalla.
  - **LeerReservasFichero**: lee una nueva lista de reservas de un fichero. Este procedimiento recibirá como argumento una lista que deberá destruir y generar partiendo de los datos del fichero, si el fichero no puede leerse la lista original deberá permanecer intacta. Al procedimiento se le pasará el nombre del fichero.
  - **PrimeraReserva**: extrae (eliminándola) y devuelve la primera reserva de la lista.

## NOTAS:

- **SE PODRÁN IMPLEMENTAR TODOS LOS PROCEDIMIENTOS Y FUNCIONES AUXILIARES QUE SE CONSIDEREN NECESARIOS DENTRO DEL PROGRAMA PRINCIPAL O EN LA PARTE DE IMPLEMENTACIÓN DE ALGUNO DE LOS MÓDULOS**
- **NO SE PODRÁ:**
  - AÑADIR OPERACIONES EN LA PARTE DE DEFINICIÓN DE UN MÓDULO
  - EL USO DE DETALLES DE IMPLEMENTACIÓN DE LAS LISTAS FUERA DE LA IMPLEMENTACIÓN DE SU MÓDULO es decir, no se podrá hacer nada del tipo `tabla=NULL`, `tabla-> sig =...` etc., fuera de la parte de implementación de los módulos de listas.
  - Suponer que los nombres no llevan espacios.
- **SE DEBE HACER UNA PAUSA DESPUÉS DE EJECUTAR CADA OPCIÓN DE MENÚ, Y BORRAR LA PANTALLA TRAS LA PAUSA.**
- **RECORDAD QUE TODO FICHERO DE CABECERA DEBE LLEVAR SU `#ifndef .... #endif`**

Ejemplo: Fichero `cabecera.h`

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