



## **PRÁCTICA N° 5: 2 sesiones**

(S5 y S6: de 26, 27, 28 de abril y 6 y 7 de mayo hasta 3, 4, 5, 13 y 14 de mayo 2004)

### **GESTIÓN DE LAS RESERVAS DE UN HOTEL Listas con punto de interés**

#### **0. OBJETIVO**

El objetivo de esta práctica es aprender a implementar y utilizar el TAD Listas. Para ello se propone el desarrollo de un programa de gestión de un hotel.

#### **1 EL HOTEL DE LOS LÍOS**

El alumno deberá desarrollar un programa que permita gestionar este lío de hotel. Nuestro hotel cuenta con un número determinado de habitaciones, cada una de las cuales con una serie de datos estáticos o semi-estáticos:

- Número de habitación
- Habitación tipo no fumador o fumador
- Habitación tipo individual, doble, suite
- Precio de la habitación

Además de los anteriores, podría ser interesante mantener algún dato que indicase si en esos momentos la habitación está ocupada o no.

Nuestro programa deberá permitir gestionar las habitaciones del hotel pudiendo:

- añadirse o quitarse habitaciones por ampliación o re-estructuración del hotel
- ver datos sobre las habitaciones disponibles
- modificar algún dato estático de la habitación, siempre que esta habitación no se encuentre ocupada o haya sido reservada con anterioridad

Además, deberá gestionarse una lista de reservas de habitaciones donde deberá poderse:

- solicitar y asignar una reserva con los datos de fecha de entrada, fecha de salida y nombre del huésped
- mostrar la lista de reservas por habitación
- quitar alguna reserva
- modificar alguna reserva (fecha de salida)

A partir de estos datos cada nuevo día nuestro programa deberá comprobar qué huéspedes van a dejar el hotel, permitiendo sacar la reserva correspondiente de la lista de reservas e imprimiendo la factura correspondiente.

#### **2. DESARROLLO DEL PROGRAMA DE GESTIÓN**

El alumno deberá usar 2 listas distintas y relacionadas entre sí. Por una lado la lista de habitaciones (el hotel propiamente dicho, **ListaH**) y por otro la lista de reservas (**ListaR**). Para ello, se deberán crear las clases sobre las que se apoyan dichas listas, que se diferenciarán únicamente por los datos que se almacenan en cada nodo.



Para la parte privada de la clase se deja libertad al programador, pero exigiéndose la utilización de listas dinámicas.

Clase ListaH	Clase ListaR
<pre>struct Habitacion {     int NumHabita;     bool fumador; //true es fumador     int tipo; //1.-sencilla 2.-doble 3.-suite     float precio; //de habitacion     bool ocupada; //true es ocupada };  class ListaH { public:     typedef Habitacion Valor;     ListaH (void);     ListaH (const ListaH &amp;);     ~ListaH (void);     bool Insertar (Valor x);     bool Eliminar (void);     bool Consulta (Valor &amp; x);     bool ListaVacía (void);     void IrAlInicio (void);     bool Avanzar (void);     bool FinalLista (void);  private:     ??? };</pre>	<pre>struct fecha {     int dia;     int mes;     int anyo; };  struct Reserva {     int NumReserva;     int NumHabitaR;     fecha entrada;     fecha salida;     string nombre; };  class ListaR { public:     typedef Reserva ValorR;     ListaR (void);     ListaR (const ListaR &amp;);     ~ListaR (void);     bool Insertar (ValorR x);     bool Eliminar (void);     bool Consulta (ValorR &amp; x);     bool ListaVacía (void);     void IrAlInicio (void);     bool Avanzar (void);     bool FinalLista (void);  private:     ??? };</pre>

Una vez creadas las clases, se pasará al programa de gestión de hotel. Para que el programa trabaje con datos introducidos con anterioridad se deberá cargar al principio de la ejecución, desde los ficheros "habita.txt" y "reserva.txt", las habitaciones y reservas existentes. Así mismo, al finalizar se deberá guardar en esos mismos ficheros la situación cuando se acabe de usar el programa, para recuperarla en una próxima ejecución.

Se adjunta un ejemplo de cada uno de estos ficheros para que se siga el mismo formato:



habita.txt	reserva.txt
<pre>1 n 1 100 2 n 2 120 3 n 3 150 5 s 1 100</pre> <p>En cada fila está separados por un espacio:</p> <ul style="list-style-type: none"> <li>• numero de habitación</li> <li>• n/s (donde “n” es no fumador y “s” fumador)</li> <li>• un numero desde el 1 hasta el 3 donde 1 es individual, 2 doble y 3 suite</li> <li>• precio de la habitación por día</li> </ul>	<pre>1 1 09/06/04 18/06/04 Juan Martinez 2 1 20/06/04 22/06/04 Jose Garcia 3 2 14/06/04 15/06/04 Maria de la O 5 3 08/06/04 12/06/04 Juan Garcia 6 3 15/06/04 20/06/04 Pepa Rodriguez</pre> <p>En cada fila hay separados por un espacio:</p> <ul style="list-style-type: none"> <li>• número de reserva</li> <li>• número de habitación reservada</li> <li>• fecha de entrada</li> <li>• fecha de salida</li> </ul> <p>y en la siguiente fila el nombre de la persona que ha hecho la reserva.</p>

Para un correcto funcionamiento del programa de gestión será necesario así mismo introducir la fecha del día actual. Tras este paso el programa revisará la lista de reservas y rellenará el campo “ocupado” de la lista de habitaciones. Hecho esto se pasará a un menú o menú con submenús que permita realizar las siguientes tareas:

- Añadir una habitación por ampliación del hotel. El número de habitación será puesto automáticamente por el programa, a partir del número de la última habitación. El usuario deberá introducir los datos estáticos manualmente.
- Mostrar un listado con las habitaciones disponibles incluyendo sus datos estáticos (número de habitación, fumador o no fumador, tipo de habitación, precio y si está ocupada o no en estos momentos).
- Modificar algún dato estático de la habitación. Primero deberá comprobarse que la habitación no está ocupada y que no hay una reserva asociada a esa habitación. Los datos que podrán modificarse son: fumador o no fumador, tipo de habitación y precio.
- Quitar alguna habitación del hotel. Igual que anteriormente, deberá comprobarse antes que la habitación no está ocupada y que no hay una reserva asociada a esa habitación.
- Solicitar y asignar una reserva con los datos de fecha de entrada, fecha de salida, nombre del huésped, tipo de habitación (fumador o no fumador y sencilla, doble, o suite). A partir de estos datos, el programa deberá recorrer la lista de habitaciones y de reservas y deberá asignar la primera habitación disponible que encuentre y un número de reserva. Si no encuentra la habitación deseada deberá informar de ello.
- Mostrar la lista de reservas por habitación. El programa deberá recorrer todas las habitaciones del hotel y en cada habitación informa de las reservas asociadas a esa habitación.
- Quitar una reserva, a partir del número de reserva. Para poder quitarse una reserva deberá comprobarse con anterioridad que no se ha ocupado ya la habitación con esa misma reserva. Es decir que el día de entrada debe ser menor mayor que el día actual. Si se dá este caso debe enviarse al usuario a modificación de reserva.
- Modificar una reserva. Se supone que esto solamente se hace si se está usando en estos momentos la



habitación. Si no se está usando la habitación deberá informarse que primero debe anular la reserva y luego abrir una reserva nueva.

A partir del número de reserva el programa permitirá modificar la fecha de salida (**nunca la de entrada** ya que se supone que ya está usando la habitación). Además, la fecha de salida nunca podrá ser igual al día actual (nunca se podrá salir hoy sino mañana como mínimo). Disminuir los días de reserva siempre será posible, pero para extender los días de reserva deberá antes comprobarse que no interfiere sobre otra reserva de esa misma habitación. Si es este el caso, se ampliará la reserva actual hasta el máximo día posible (en que será ocupada la habitación por otra persona) y se abrirá una reserva nueva en otra habitación (si es posible).

- Incrementar en uno el día actual. El programa deberá comprobar, en las habitaciones ocupadas, si hay algún huésped que abandone el hotel hoy y, en su caso, sacará la reserva de la lista de reservas y creará un fichero "facturaxxx.txt" con los datos de la factura que incluirá los siguientes campos (donde xxx será el número de reserva):
  - Nombre del huésped
  - Fecha actual
  - Número de habitación
  - Número de reserva
  - Fecha de entrada
  - Número de días
  - Precio de la habitación por día
  - Precio total

Con el fin de no complicar el programa sin necesidad, se supondrá que el año está formado por 12 meses de 30 días cada mes.

Así mismo, después de las operaciones de inserción de una nueva reserva o modificación de reserva, deberá reorganizarse la lista de reservas agrupando primero por número de habitación y luego por fecha de entrada a la habitación. Eso quiere decir, por ejemplo, que una nueva reserva número 7 sobre la habitación 1 con una fecha de entrada anterior al 09/06/04 quedará la primera en la lista de reservas. En este sentido para poder asignar correctamente el número de reserva a la siguiente entrada habrá que guardar en una variable el número de la última reserva.

### 3. REQUISITOS

**Recordar que para poder acceder a la sesión de prácticas correspondiente es necesario entregar un pequeño esquema de lo que se va a realizar en la práctica.**

En esta práctica, habrá que entregar una documentación de programa completa para el programa principal, explicando el programa principal y las clases usadas. Si no se presenta la documentación ANTES de acceder al laboratorio, no se permitirá la entrada a la sesión de prácticas.

### 4. ENTREGA

La entrega de esta segunda práctica debe incluir **el código** con el programa principal (`##hotel.cpp`), y la implementación de las clases (`##listah.cpp`, `##listah.h`, `listar.cpp`, `listar.h`)

La entrega puede realizarse por e-mail a la dirección del profesor encargado del grupo o en disquette al iniciar la siguiente sesión de prácticas, pero siempre ANTES de empezar la siguiente sesión de prácticas.

Cualquier práctica entregada fuera de plazo no será admitida para su corrección.

## ENTREGA DE PROGRAMAS:

**Al comenzar la sesión 7 de prácticas (10, 11, 12, 20 y 21 de mayo de 2004)**