



Enunciado de la práctica 2:

GESTIÓN DE RESERVAS DE UN RESTAURANTE

El objetivo de esta práctica es completar la simulación de funcionamiento de la gestión de reservas, mesas y pedidos, durante los tres turnos de comida de un restaurante.

Cada una de las reservas incluye los siguientes datos:

- Número de reserva.
- Nombre del cliente.
- Situación deseada: Terraza o Interior.
- Número de personas: máximo 8 personas por mesa.
- Hora de la reserva: 13:00, 14:00 o 15:00.
- Preferencia de menú: vegano, sin Gluten, completo.

Para cada pedido se tienen los siguientes datos:

- Número de pedido (el de la reserva).
- Nombre del cliente.
- Situación deseada.
- Número de personas.
- Turno de hora en el que servirá el pedido.
- Preferencia de menú.
- Número de la mesa (si el número de personas de la reserva es mayor de 4 se le asignará una de 8, teniendo en cuenta la situación indicada en la reserva).

En esta ocasión, el programa creará aleatoriamente las mesas y las almacenará en un árbol binario de búsqueda (el criterio de ordenación es el número de mesa). A continuación, mostrará el menú. Cada mesa tendrá los siguientes elementos:

- Número de la mesa: de 1 a 20.
- Capacidad: 4 u 8 personas.
- Situación: Terraza/Interior. Tendremos 10 mesas en cada ubicación.
- Lista enlazada con los pedidos que se han servido en ella. El orden de ordenación será el número de pedido.

Además, el programa generará aleatoriamente reservas que almacenará en una lista doblemente enlazada, ordenadas por hora, situación y número de comensales. Las reservas, al llegar la hora de su turno comer, se convierten en pedidos: Se toman los datos de la reserva y se les asigna mesa (buscando alguna compatible en el ABB de mesas, que no tenga ocupado ese turno), saliendo de la lista doblemente enlazada y pasando a la lista enlazada correspondiente a su mesa en el ABB. Si no es posible atender la reserva por no tener mesa compatible para ella, se modificará el turno y se mantendrá en la lista doble de reservas para el siguiente turno (si es necesario, se creará un turno a las 16:00, y si no es suficiente, otro a las 17:00, ...).

Es obligatorio usar los TAD's Lista, LDoble y ABB. Es decir, hay que implementar, como mínimo, las clases **NodoLista**, **NodoLDoble**, **Lista** y **Ldoble** (al menos con sus operaciones habituales), una clase **NodoArbol** y una clase **ArbolBB** (al menos con sus operaciones habituales).



El **programa** mostrará un **Menú con las siguientes opciones**:

1. Mostrar el contenido del árbol de mesas con recorrido Inorden, mostrando de cada mesa: número, situación y número de pedidos atendidos hasta el momento (inicialmente serán 0 pedidos). En el listado también hay que indicar la altura del árbol y el número de nodos que tiene (inicialmente 20).
2. Borrar una mesa concreta que indique el usuario.
3. Crear una nueva mesa, solicitando los datos al usuario, e insertarla en el ABB de mesas.
4. Generar aleatoriamente un número de reservas y añadirlos a la lista doble de reservas. Se pedirá al usuario el número de reservas a generar.
5. Mostrar en pantalla los datos de la lista de reservas.
6. Vaciar la lista de reservas.
7. Solicitar por teclado un tipo de menú y eliminar de la lista doble todas las reservas de ese menú.
8. Solicitar por teclado un turno y atender todas las reservas de ese turno, creando los pedidos que sea posible y cambiando de turno a los que no hayamos podido atender.
9. Solicitar por teclado una mesa y mostrar en pantalla sus datos y la lista de pedidos atendidos.
10. Solicitar por teclado una situación y mostrar en pantalla los datos de las mesas (con sus pedidos atendidos) de esa situación.
11. Salir.

NORMAS PARA LA REALIZACIÓN Y ENTREGA DE LA PRÁCTICA

Las prácticas se realizarán en grupo (el mismo **para las dos prácticas** de la asignatura).

1. La práctica se implementará en C++, utilizando CodeBlocks. Debe entregarse un fichero comprimido incluyendo **todos los ficheros fuente del proyecto C++** y el documento descrito en el punto 4. Se subirá un fichero por grupo a la plataforma **antes de la fecha indicada**. El nombre del fichero será el nombreapellido1apellido2 de alguno de los alumnos.
2. En la **defensa se verificará la autoría de la práctica entregada y será calificada con APTO/NO APTO, siendo necesario obtener APTO para ser evaluado de la práctica**.
3. La entrega de prácticas copiadas supondrá el suspenso de la asignatura en esta convocatoria para todos los alumnos implicados.
4. Además del proyecto, se subirá en un fichero *.pdf*, que tendrá los siguientes apartados:
 - a. Nombre y DNI de los alumnos del grupo.
 - b. Detalles y justificación de la implementación:
 - Especificación concreta de la interfaz de los TAD's implementados.
 - Solución adoptada: descripción de las dificultades encontradas.
 - Diseño de la relación entre las clases de los TAD implementados (Explicación de los métodos más destacados y del comportamiento del programa).
 - Bibliografía.

PRÁCTICA 2: FECHAS E INDICACIONES

La práctica 2 se entregará antes del **16 de diciembre a las 23:59 horas**. La **defensa, individual y obligatoria** para ser calificado, será durante la **clase del día 18 de diciembre**. Para su realización es necesario traer el código entregado en el portátil o en un pen drive (en caso de no traer el portátil).

EN CASO DE NO PRESENTARSE A LA DEFENSA NO SE CALIFICARÁ LA ENTREGA.