

Una central de reservas hoteleras de ámbito internacional nos encarga una aplicación que resuelva sus reservas en tiempo real. La central agrupa los hoteles por zonas: "La Manga del Mar Menor", "La Costa Verde", "La Asturias Prerrománica", "El camino de Santiago", etc. (por simplicidad un hotel está en una sola zona). En esta empresa, el término Hotel es una forma genérica de referirse a un establecimiento hostelero que ofrece habitaciones reservables; podría ser pensión, hostel, alberge, casa rural, hotel, etc.

Cada "hotel" (establecimiento hotelero) tendrá habitaciones que se reservarán para determinado día durante una noche. De nuevo, por simplicidad, una reserva será para una habitación y por una noche.

Los clientes de la central de reservas (tour operadores y agencias de viaje) acceden a la central de reservas a través de una aplicación web y hacen sus operaciones on-line (en nuestro caso simularemos la interfaz web con la consola), por lo tanto debemos permitir:

- Listar habitaciones disponibles para una fecha en una zona
- Hacer reservas y cancelarlas
- Listar las reservas de un establecimiento en unas fechas
- Para casos de emergencia, eliminar todas las reservas de un establecimiento en una fecha.
- Para casos de emergencia, reubicar reservas; esto es, mover todas las reservas de un hotel en una fecha a otros hoteles de la misma zona
- Cargar y grabar ficheros con la información de zonas, hoteles, habitaciones y reservas

Deberá realizarse un programa Java con arreglo a los siguientes requisitos:

Información a tratar

1. Zonas:

- a. Código. Entero. Actúa como identificador
- b. Nombre. String
- c. Descripción. String

2. Hoteles:

- a. Código. Entero. Identificador
- b. Nombre. String
- c. Tipo. Carácter Tipo. Carácter (P, pensión; S, hostel; H, hotel; R, casa rural; etc.)
- d. Estrellas. Entero 1..5

3. Habitación:

- a. Código. Entero. Identificador
- b. Planta. Entero
- c. Ocupación max. (Nº de PAX). Entero
- d. Precio noche. Euros

4. Reserva:

- a. Código. Entero. Identificador
- b. DNI de cliente. String
- c. Fecha de entrada. String en formato aaaa/mm/dd

2. Opciones de menú

- Listar zonas
 - Una línea en pantalla para cada zona que muestre todos sus datos
- Listar todos los hoteles
 - Una línea en pantalla para cada hotel que muestre todos sus datos y el nombre de la zona en la que está
- Listar habitaciones disponibles para una fecha en una zona.
 - Deberá pedir por teclado el identificador de zona y la fecha
 - Mostrará en pantalla una línea por cada habitación disponible con la siguiente información: el nombre del hotel, tipo, nº de estrellas, planta y precio
 - El listado estará ordenado por precio de menor a mayor
- Hacer una Reserva
 - Pedirá por teclado el código de la habitación, la fecha de entrada y el DNI del cliente
 - Debe verificar que la habitación está disponible, en caso contrario es un error
 - Escribirá en pantalla el código generado para la reserva
- Cancelar una reserva
 - Pedirá por teclado el código de la reserva
- Listar las reservas de un cliente
 - Pedirá por teclado el DNI del cliente
 - Cada reserva será una línea en pantalla con la siguiente información: nombre hotel, planta habitación, fecha y precio

- Listar las reservas de un hotel
 - Pedirá por teclado el código del hotel y el rango de fechas en el que se desea buscar
 - Cada reserva será una línea en pantalla con la siguiente información: nombre hotel, planta habitación, DNI del cliente, fecha y precio
- Emergencia: Eliminar todas las reservas de un Hotel un día
 - Pedirá por teclado el código del hotel y la fecha a evacuar
- Emergencia: Reubicar todas las reservas de un hotel un día
 - Pedirá por teclado el código del hotel y la fecha a reubicar
 - Cada reserva de ese día se debe asignar a otro hotel de la misma zona en una habitación del mismo tamaño, no importan las diferencias de precio
 - Se borrará la reserva del hotel de evacuado y se registrará una nueva en el hotel receptor

3. Situaciones de error a controlar:

- Se considerarán errores no recuperables los problemas genéricos de entrada/salida (disco desconectado, estropeado, etc.) y los de ficheros no existentes. El programa terminará de forma ordenada y mostrará un mensaje explicativo al usuario.
- Son situaciones que no provocarán la terminación de la ejecución las que provengan de incoherencias (errores de lógica) en los datos suministrados por el usuario. Si hay errores en los datos suministrados por el usuario el programa no debe terminar y debe informar al usuario con un mensaje en pantalla alusivo.
- Si hay inconsistencias en los datos leídos desde fichero se dejará constancia de ellos en un fichero de log pero el programa

arrancará con normalidad, habiendo cargado los datos correctos y descartado las líneas con incoherencias.

- Se deben controlar como mínimo las siguientes situaciones:
 - Búsqueda en una zona inexistente
 - Tratar de reservar una habitación inexistente
 - Cancelación de reserva no registrada
 - Listar reservas de hotel inexistente

4. Carga de datos desde fichero

- El programa al iniciarse cargará los datos desde fichero
- El fichero estará en formato texto, organizado en líneas (registros).
- Cada línea está compuesta por campo separados por el carácter Tabulador ('\t')
- Cada registro contiene información relativa a una zona, un hotel, una habitación o una reserva. Se usarán estos formatos de registro:

Registro de zona:

1<código><nombre><descripción>

Registro de hotel:

2<código><código-zona><nombre><tipo><estrellas>

Registro de habitación:

3<código><código-hotel><planta><ocupación><precio>

Registro de reserva:

4<código><código-habitación><DNI><fecha>

- Los registros no tienen porque aparecer agrupados ni en determinado orden pero sí se cumplirá que una reserva concreta aparecerá detrás de su habitación, esta, a su vez, detrás de su hotel y este de su zona

- El fichero puede contener líneas inválidas: en blanco, mal formadas o con información incoherente. El programa debe descartarlas pero debe cargar las siguientes
- No se debe informar al usuario de los errores de líneas inválidas pero deben quedar registrados en un fichero de log llamado log.txt
- Son líneas mal formadas las que :
 - Son de tipo desconocido; no empiezan por 1, 2, 3 ó 4
 - No tienen el número de campos correcto para el tipo de registro
 - Contienen números mal formados (p.e. "1d3,2" no es un número)
- Las incoherencias que pueden aparecer en el fichero son del estilo del apartado control de errores lógicos : zona, hotel o habitación inexistente.

5. Grabación de fichero de salida:

- Al finalizar el programa se grabarán de forma automática los datos contenidos en memoria en un fichero de texto plano que tendrá el mismo formato que el de entrada.

6. Diseño:

- Debe haber una única clase que actúe de interfaz con el usuario y: Será la única que podrá leer de teclado y escribir en pantalla
 - No contendrá lógica de la aplicación
- Todas las situaciones de error deben controlarse utilizando excepciones. No es necesario crear un tipo de excepción para cada posible situación errónea, se pueden agrupar en excepciones lógicas y gestiones de persistencia.

- Los métodos encargados de ordenar listas deben devolver listas (no String) y las ordenaciones no alterarán las listas originales.
- La carga y grabación de ficheros debe realizarse desde una clase específica a tal fin
- Todos los incidentes producidos en la carga deben ser informados en un fichero de log
- Los diferentes tipos de alojamiento deben modelarse usando herencia