

Enunciado

Se quiere construir un programa que permita administrar las habitaciones de un hotel. El sistema tiene la información de las habitaciones, los huéspedes y las reservaciones. De una habitación el sistema conoce el tipo (simple, doble, triple), el número de la habitación y el precio por noche. Si una habitación está ocupada, se conoce la información del huésped(es) que tomo/tomaron la habitación y los consumos (nombre de producto o servicio y valor) que han sido cargados a la habitación. Si por el contrario, una habitación está reservada, solamente se conoce la información de la persona que hizo la reserva, la fecha de inicio de la estadía y el número de días/noches reservados. En el momento de check-in para una habitación (con o sin reserva), el sistema debe registrar el nombre, la edad, el número de documento de identificación, la dirección de residencia, el teléfono de cada huésped en la habitación y el número de noches de estadía. Una persona puede cancelar una reservación en cualquier momento a partir de su nombre y fecha de inicio. Cuando una persona hace check-out, debe hacer el pago de sus consumos y de su estadía. La aplicación debe mantener un registro de los recaudos en caja y del valor total de los consumos cargados a habitaciones cuyo pago está pendiente.

La primera vez que el usuario inicie la aplicación, se debe poder seleccionar un archivo de texto con la información del hotel (nombre, dirección, teléfono, país, ciudad, tipo de habitaciones, precio por tipo de habitación y la información de las habitaciones) para así poderla inicializar. La aplicación debe ser persistente de tal manera que guarde la información de los huéspedes, reservaciones y los recaudos del hotel.

El problema

El objetivo de este ejercicio es analizar, diseñar e implementar un sistema de información para el manejo de un hotel, utilizando unas estructuras de datos que tengan en cuenta las restricciones que se definen más adelante. Asegúrese de leer completamente el enunciado antes de resolver el ejercicio.

Requerimientos funcionales

El programa debe soportar los siguientes requerimientos funcionales:

R1. Cargar la información del hotel desde un archivo de texto. (Debe tener al menos 200 habitaciones).

R2. Realizar la reserva de una habitación.

R3. Eliminar la reserva de una habitación.

R4. Hacer check-in a una habitación dada su información. (Con o Sin Reserva)

R5. Hacer el check-out de una habitación.

R6. Ver la información de las habitaciones del hotel. (Información de la reserva, información del huésped, Información de la habitación)

R7. Generar un reporte con la información detallada de las habitaciones ocupadas, las habitaciones reservadas, el valor total recaudado.

R8. Buscar reserva por nombre de la persona (o parte de él).

R9. Buscar habitaciones por disponibilidad para una fecha.

R10. Buscar una habitación por nombre del huésped.

R11. Registrar un consumo en una fecha a una habitación.

R12. Mostrar el top 10 de las habitaciones mas utilizadas.

Restricciones de diseño

1. El diseño se debe hacer **MINIMIZANDO el tiempo de ejecución de todas las operaciones y el espacio que ocupan las estructuras de datos, garantizando que el programa pueda evolucionar.**
2. En particular, se deben justificar las decisiones de diseño tomadas, se debe explicar claramente en un **documento de diseño** por que se escogieron las estructuras de datos utilizadas, teniendo en cuenta los requerimientos descritos anteriormente.
3. El sistema debe ser **persistente**. Recuerde que el objetivo del ejercicio es resolver los requerimientos con las estructuras en memoria principal, por este motivo no se acepta el uso de bases de datos. Puede utilizar archivos de texto o serialización para persistir el modelo del mundo.
4. La interfaz gráfica debe ser implementada utilizando componentes de awt/swing.
5. El diseño debe estar desacoplado con interfaces.
6. Debe implementar las estructuras de datos genéricas en un paquete aparte.
- 7. No se puede utilizar Java Collections ni Cupi2 Collections.**

Entregables

Laboratorio 1:

Se debe entregar una primera versión de la interfaz gráfica de la aplicación, desarrollada con Window Builder, además se debe traer para el laboratorio el diseño completo **impreso legible**, de la aplicación y de las estructuras de datos a utilizar, realizado en Enterprise Architect para recibir retroalimentación.

El mejor ejercicio de cada sección tendrá un bono especial sobre la nota del nivel.