

---

### **Descripción**

La famosa cadena de hoteles City Express tiene pensado construir un sistema sencillo que tendría por propósito, un control de ocupación de las habitaciones para sus hoteles. El sistema contempla la administración del ingreso económico, control de ocupaciones donde se genere el alquiler de habitaciones, entre otros. La aplicación ya terminada puede ser utilizada en cualquiera de los hoteles de tan prestigiosa cadena. Por eso al iniciar dicha aplicación, se debería ingresar por teclado, el número de pisos del hotel y cantidad de habitaciones por piso con esto se construye la estructura inicial básica para trabajar en ella.

Si bien es cierto, estos hoteles son muy parecidos en su interior, ellos varían en el número de pisos y el número de habitaciones por piso. Se le ha contratado para que implemente este gran proyecto.

---

### **Objetivos**

El objetivo de este trabajo es aplicar y desarrollar los conceptos teórico/prácticos, principios y técnicas estudiadas en clase (paradigma estructurado). Es importante que al desarrollar el proyecto se escriba código correctamente estructurado, debidamente encapsulado y fácilmente reutilizable.

---

### **Consideraciones de implementación**

El código del proyecto debe estar estructurado adecuadamente, respetando los principios vistos en las lecciones sincrónicas:

- Las implementaciones deben hacerse en dos archivos (archivos .h) y código fuente (archivo .c) por aparte. O bien en uno solo archivo.c (con la declaración de las cabeceras de procedimientos y funciones).
- Las estructuras que representen las entidades, no deberán contener código de entrada/salida (como salida a la consola usando print, por ejemplo). Sin embargo, se pueden mostrar mensajes de comprobación al efectuar las pruebas del programa, para corroborar que se están ejecutando las funciones de manera correcta.
- Se programará en el compilador de C de Apache Netbeans 12.4 (última versión). O en el ambiente de desarrollo seleccionado por su profesor, además solo se podrán crear, estructuras de datos dinámicos (es decir con uso de punteros).

---

### **Detalles de problema**

Básicamente, a usted se le solicita la creación y desarrollo de cuatro estructuras, que son:

- a) Una estructura Contenedora donde interiormente tenga una matriz [nxm] de punteros, la idea es que la matriz simule un edificio de habitaciones, donde **n** es el número de pisos y **m**

es el número de habitaciones por piso, además donde cada celda de la matriz pueda tener un puntero que “pueda ver” o apuntar a una estructura de tipo de habitación.

- b) La estructura Habitación tendrá las siguientes variables miembro: Un identificador (string) de la habitación, un estado (char) de ocupación ('O' → ocupada, 'M' → mantenimiento, 'L' → libre), además un puntero que lo relaciona con la estructura cliente y finalmente otro atributo tipo puntero que se asigna a una variable de tipo Información.
- c) La estructura Información deberá contener solo cuatro atributos que son: el número de adultos, número de niños, número de días a disfrutar, y finalmente si la estancia es todo incluido o no.
- d) La estructura Cliente tendrá la siguiente información: el nombre del cliente, su número de identificación, la forma en la que este realiza el pago (efectivo, tarjeta de débito o tarjeta de crédito), su número de teléfono así como su número de cuenta.
- e) Toda habitación deberá ser ocupada por un número máximo de 5 personas, pero el número máximo dependerá del número de camas que tenga dicha habitación. La asignación de la cantidad de camas por cuarto deberá ser aleatoria (2-5). Si una habitación está ocupada, no puede tener menos de una persona adulta.
- f) Si la habitación tiene solo una persona, la misma tendrá un valor de \$150.00 por noche, en caso de dos personas tendrá un valor de \$250.00 por noche, en caso de tres personas tendrá un valor de \$350.00 por noche y más de tres el valor de la habitación será de \$400.00 la noche.
- g) Como en todo hotel, la entrada se realiza a la 1:00 pm y la salida es a las 11:00 am del otro día. Si la habitación es ocupada por la noche, al valor de la habitación se le hará un descuento del 25% en caso de ser una persona sola, un 20% si es el caso de dos personas, un 15% si son tres personas, en caso de ser más de tres que desean la habitación en la noche, entonces se les hace un descuento de un 10% solamente. Todo lo anterior, si “No Es Todo Incluido”. En caso de que el hotel dé un servicio de “Todo Incluido”, cada persona mayor de edad pagará por día un valor de \$300.00 y los niños \$200.00, independientemente de la hora que lleguen.
- h) En los hoteles, generalmente, las habitaciones no son iguales, dado que presentan condiciones diferentes de espacio y comodidad, por ejemplo algunas tienen ventilador, mientras que otras tienen aire acondicionado; esto hace que algunas habitaciones cumplan con pluses a la hora de ser tomadas. Entonces, es importante entender que todas las habitaciones deberían tener una clasificación (primera clase, segunda clase, tercera clase), entendiendo que esto las diferencia. Esta clasificación se debería asignar en forma aleatoria (solo para este proyecto). “PC” = Primera Clase, “SC” = Segunda Clase, “TC” = Tercera Clase. Obviamente las de primera clase, están mejor equipadas.



- i) Es recomendable que, antes de asignar una variable tipo Habitación a alguna posición en la contenedora, primero se debería haber ingresado el tipo de Habitación, una variable que le pertenece a la variable tipo Información y de una vez la variable cliente. Esta variable Información deberá ser recibida por parámetro (tipo puntero), por uno de los métodos de la variable Habitación, lo mismo ocurre con la variable cliente.
- j) Con base en todo lo anterior, se recomienda tener una estructura Contenedora que contará con una serie de funciones y procedimientos, donde algunos de ellos deberán recibir por parámetro, una variable habitación y su posición [fila][columna] para así ubicarla dentro de la matriz. Además deberá tener funciones y/o procedimientos que permitan que ayuden a tomar decisiones futuras, generando los siguientes reportes.

>>>> REPORTES <<<< que generan los métodos de la clase Contenedora	
➔ MENÚ ➔	
1. Cuántas habitaciones están libres	(5pts)
2. Cuántas están en mantenimiento	(5pts)
3. Cuántas habitaciones ocupadas	(5pts)
4. Cuántas hay desocupadas por cantidad de camas.	(5pts)
5. Realizar la ocupación de la habitación. (Ingreso)	(5pts)
6. Pagar la habitación utilizando el No. de cédula.	(5pts)
7. Liberar la habitación utilizando el No. de cédula.	(5pts)
8. Cuántas personas adultas hay el día de hoy.	(5pts)
9. Cuántos niños existen al día de hoy en el hotel.	(5pts)
10. Saber cuánto dinero recaudó hoy el hotel. (Sub-Menú)	
a. Con los clientes, "TODO INCLUIDO"	(5pts)
b. Con los clientes, "Sin TODO INCLUIDO"	(5pts)
c. Con todos los clientes que hay en el hotel.	(5pts)
Total (60 pts)	

- k) Cuando el cliente llega a la ventanilla, el sistema tendrá la opción de consultar las habitaciones que están disponibles por el número de camas (opción 4), la idea es que el cliente indique cuántas personas son las que van a ocupar la habitación y que el sistema le informe cuántas y cuáles habitaciones hay de primera clase, segunda clase y tercera clase, así el cliente podrá seleccionar (con dicha información separada por categoría), cuál habitación desea ocupar.
- l) Una vez que el cliente ha decidido cuál habitación desea, el sistema tendrá la opción de realizar la ocupación de la habitación (opción 5). El sistema debe solicitar al cliente dentro de esta opción, sus datos personales para ser incluidos dentro de la clase cliente, además se debe capturar el número de adultos, el número de niños, la cantidad de días que desea utilizar la habitación y si desea adquirir el servicio de todo incluido o no; con toda la información se incluye los datos en los campos correspondientes dentro de la clase Contenedora en la posición correspondiente de la matriz.
- m) Luego de haberse ingresado la anterior información, el sistema debe tener una opción de Pagar la Habitación (opción 6), de este modo el cliente indica el número de habitación que

estaba ocupando y con esta información el sistema se encargará de informar al usuario el monto a ser cancelado. En el pago final, hay que verificar, si la habitación es de primera clase, se requiere que al pago final, se le cobra un 30% más, si la habitación es de segunda clase se le cobra un 20% más y finalmente si la habitación es de tercera, no hay cobro adicional.

- n) Cuando la habitación ha sido cancelada, el sistema debe tener una opción para colocar la habitación en estado de “Libre” y con esto se pueda iniciar el ciclo de ocupación de esta misma habitación por un nuevo cliente (opción 7).

### ***Evaluación***

---

- 1- La tarea se realizará de manera individual ó grupal (max 2 estudiantes) y será entregado en dos semanas apartir del momento de la entrega del profesor.
- 2- Una vez terminada la tarea cada estudiante lo entregará por medio del aula virtual (sí es en parejas una entrega por pareja), en el espacio designado para ello, con su nombre e identificación, todo dentro de un archivo comprimido (.zip o .rar), titulado Proy#1Hotel-Ced#1-Ced#2.
- 3- El espacio en el aula virtual estará abierto para la entrega hasta el último día de la semana indicada.
- 4- No se recibirán tareas de otro modo que no sea por el aula virtual, por ejemplo, correo electrónico, etc, salvo por indicación manifiesta del profesor.
- 5- Queda a criterio del profesor utilizar un medio de entrega diferente cuando lo considere conveniente. No se admitirá la entrega tardía de la tarea. En este caso, se calificará con nota 0 (cero).
- 6- La tarea deberá ser un desarrollo propio del estudiante, no se aceptarán plagios parciales ni totales. En caso de comprobarse un plagio, se remitirá el documento a las autoridades respectivas para que se hagan cargo del debido proceso.
- 7- La tarea debe ser presentada y enviada al aula virtual y debe tener para ser tomada en cuenta una funcionalidad mínima requerida del 40%. Si no tiene funcionalidad mínima, la tarea no será calificada y la nota será de cero (0).

### ***Rubrica básica sugerida:***

---

1- Uso de variables, funciones, procedimientos, archivos h y c (20pts)
2- Uso adecuado de la modularidad (división de tareas), manejo de bibliotecas, llamados entre módulos (30pts)
3- Uso de Estructuras (25pts)
4- Uso de Archivos (25pts)