

Instrucciones. Utilizar el diagrama Entidad-Relación y Relacional del caso hoteles.

Consideraciones generales

1. El sistema tiene que ser una aplicación sencilla Java SE que utiliza Maven como gestor de dependencias, con interfaces gráficas de usuario, utilizar motor MongoDB con la librería MongoClient para el manejo de las operaciones de persistencia.
2. Para la selección de fechas en el sistema es necesario que se utilice un componente selector gráfico. Se recomienda utilizar LGoodDatePicker, pero la decisión final es del equipo diseñador del sistema.
3. Las librerías externas deben incluirse como dependencias en el archivo POM.
4. Incluir todas las validaciones correspondientes para cada interfaz gráfica como pueden ser: validación de datos de entrada (por ejemplo, formato y campos vacíos), que no se permita guardar información repetida (por ejemplo, que existan 2 usuarios repetidos), validaciones lógicas.
5. La usabilidad del sistema es de suma importancia, por lo tanto, la información debe presentarse de forma legible incluyendo formatos de fechas. El sistema debería informar siempre de los resultados de las operaciones realizadas a los usuarios con mensajes claros y concisos.

Entregables del proyecto obligatorios

1. Sistema de interfaz gráfica coherentes que incluyan información de muestra:
 2. El sistema ya deberá tener los datos del hotel, las habitaciones, categoría y tarifa.
 3. El sistema deberá hacer la reservación dependiendo si es cliente o agencia de viajes. Al final mostrar la reservación con las fechas y el costo a pagar de la(s) habitaciones reservadas.
4. Diagrama de clases de dominio.
5. Modelo entidad relación.
6. Mapeo de prototipos de interfaz gráfica con clases de dominio.
7. Proyecto Java SE de Netbeans con Maven como gestor de dependencias
8. URL del repositorio público de Github con el código fuente actualizado.
9. Documentación Javadoc en formato HTML.

NOTA: En caso de no incluir algún entregable el proyecto no se calificará.

ACLARACIÓN: no se trata de que hagan el proyecto como tradicionalmente los hacen (cada quien, en su pc, se pasan los cambios por drive o USB, luego los juntan) y suban todo al final al Git como si fuera Google Drive. Se trata de que vayan subiendo los cambios en Git gradualmente, de tal forma que la otra persona los descargue, agilice el desarrollo y mejore la mantenibilidad del software.