

PROYECTO 2**README****Integrantes:**

○ Bernal Esquivel Mariana Morayma	320298787
○ Castillo Hernández Eduardo	420003557
○ Raya Pérez José Luis	319659058
○ Vázquez Dávila José Adolfo	423033366

SISTEMA DE RESERVACIONES DE HOTEL EN CANCÚN⇒ Problemática:

La industria hotelera se enfrenta al desafío de modernizar continuamente sus sistemas de reservación para satisfacer las expectativas de los clientes que buscan rapidez, eficiencia y personalización. En un entorno altamente competitivo, los hoteles deben ofrecer una plataforma de reservación que no solo cubra las necesidades básicas de selección y reserva de habitaciones sino que también proporcione una experiencia rica y personalizable. Los sistemas deben ser capaces de manejar diversas preferencias y requerimientos de los usuarios, como elegir tipos de habitación específicos, explorar y añadir paquetes turísticos y ofertas especiales. Además, la plataforma debe ofrecer soporte multilingüe para alcanzar una audiencia global, adaptándose a mercados que hablan inglés, español y portugués.

En el desarrollo de tal sistema, se deben considerar los siguientes puntos:

- Inicio de sesión de usuarios: Sencillo, eficaz, seguro y robusto
- Selección de habitaciones: Los usuarios deberían poder buscar habitaciones que se ajusten a sus necesidades específicas de alojamiento, tanto por la duración de su estadía como por la cantidad de acompañantes.
- Personalización de habitaciones: El sistema deberá permitir a los usuarios personalizar su experiencia, eligiendo servicios adicionales.
- Aplicación de ofertas: Detectar automáticamente las promociones aplicables para ofrecer la mejor tarifa posible.
- Paquetes de turismo: Ofrecer la opción de agregar experiencias turísticas de la zona al hacer la reserva, lo cual podría incluir tours, actividades locales y otras experiencias.
- Confirmación de reservación: Enviar automáticamente una confirmación por correo electrónico con todos los detalles de la reserva y la información necesaria.
- Visualización de ofertas y paquetes: Presentar de manera clara y atractiva las ofertas especiales y paquetes disponibles para que los usuarios puedan tomar decisiones informadas.
- Idiomas: Implementar interfaces en inglés, español y portugués para atender a una base de usuarios diversa.

Uso: `java -jar target/proyecto2-1.0-SNAPSHOT.jar`

Patrones implementados.

- **MVC**

Modelo:

- decoradores: creación de decoradores con el patrón de diseño Decorator para agregar servicios adicionales a la habitación reservada.
- interfaces: Observador y Sujeto, indispensables para el patrón de diseño Observer el cual informa a los usuarios de las ofertas disponibles.
- habitacionesYPaquetes: creación de las habitaciones y paquetes de turismo con el patrón de diseño Prototype.
- objetos: creación de los objetos Oferta y Usuario.
- reservaciones: manejo de fechas de entrada y salida de un huésped.

Vista:

- HotelFacade: La fachada de la interfaz para simplificar su uso en el main.

Controlador:

- repositorios: Carpeta con clases donde se crean las colecciones de usuarios, habitaciones, ofertas y paquetes.

- **Observer**

Para el sistema de ofertas y promociones especiales, se optó por usar Observer. Creamos clases Usuario que implementa Observador; y repositorioOferta que implementa Sujeto.

Usamos Observer para notificar al usuario suscrito, la aparición de nuevas ofertas para que este pueda visualizarlas y en dado caso que alguna sea válida para su habitación, automáticamente se aplique la mejor oferta posible al costo final de su estadía en la habitación (No aplica extras ni paquetes de turismo).

La manera de generar ofertas es automática por medio de hilos, con un descuento variado y que selecciona entre los 3 tipos de habitaciones, al generarla, se le notificará al usuario suscrito. El sistema seguirá generando las ofertas hasta que el usuario seleccione los pasos finales para la reserva de su habitación. De igual forma al continuar a la selección de los paquetes, se detendrá el generador de ofertas y eliminaremos al usuario como observador, pues al ya aceptar una oferta, el usuario no podrá volver a aplicar una segunda oferta.

- **Prototype:**

Usamos este diseño al crear las habitaciones y paquetes de turismo, clonando las respectivas instancias para crear más de cada tipo. De esta forma se esconde la complejidad de crear nuevas instancias, del cliente y además nos permite integrar mejor la parte de decorator a diferencia del patrón factory.

Creamos una clase que representa la habitación del hotel como lo haríamos típicamente en POO con el agregado de usar el método clone() de java, para luego delegar a otra clase la generación de múltiples instancias y almacenarlas en una colección que sirve como nuestra base de datos, a la cual podemos acceder posteriormente en tiempo de ejecución para modificar su estado de disponibilidad.

- **Decorator:**

Este patrón de diseño nos ofrece la posibilidad de añadir servicios adicionales a nuestro objeto habitación de manera dinámica y flexible. En nuestro caso, tenemos 3 tipos distintos de habitaciones las cuales tienen un precio fijo por día, y quedará a decisión del usuario el agregar una cena romántica, champagne, chocolates o un ramo de flores. Sea cual sea la elección el precio total de la reserva aumentará correspondientemente. Esta forma de envolver al objeto nos permite tener todo un conjunto de clases que cambian su comportamiento final sin tener que hacer cada una de las combinaciones posibles de manera separada.

- **Facade:**

Utilizamos este patrón de diseño por la sencilla razón por la que fue concebido. Proveer una interfaz unificada y simplificada de un conjunto de interfaces dentro de un subsistema.

Toda nuestra ejecución del programa se delega al objeto fachada el cual nos brinda métodos que facilitan del lado del cliente la resolución de varias acciones al mismo tiempo pero sin perder el acceso a las individuales. En nuestro caso la clase HotelFacade actúa como el nexus entre todas las demás clases del programa y define todos los métodos para seleccionar el idioma, el login de un usuario, mostrar el menú y submenús, mandar un correo de confirmación y ser el entorno de prueba a distintos tests que amenacen con la robustez del programa. Todo esto para que en la parte del main solo se haga un llamado a una instancia de esta clase junto a su método simplificado.

Los usuarios precreados para ingresar al sistema son:

Usuario	Contraseña
carlos	carlito23
maria	mary35
juan	juan97