

INFORME DE SISTEMA DE RESERVAS

Ingeniería de Sistemas

Michelle Sossa Nova

Cristian Cuadrado

Universidad del Sinu - Seccional Cartagena

CONTENIDO

Introducción	3
Requerimientos	4
• Funcionales	4
• No funcionales	4
UML	5
• Diagrama de clases	5
• Diagrama de casos de uso	5
• Diagrama de secuencia	6
Implementación	7
• Clases	7
• Explicación del Código de las Clases	8
• Explicación del Programa	9
Pruebas	10
Instrucciones:	10

INTRODUCCIÓN

El objetivo de este proyecto es diseñar e implementar un sistema básico usando el lenguaje de programación Java para reservas de un hotel que permita a los usuarios reservar habitaciones de forma online. El sistema debe gestionar diferentes tipos de habitaciones, controlar la disponibilidad y permitir a los usuarios hacer y cancelar reservas.

REQUERIMIENTOS

- **Funcionales**

Los usuarios deben poder:

- Registrarse e iniciar sesión en el sistema. Buscar habitaciones disponibles por fechas.
- Reservar una habitación.
- Cancelar sus propias reservas.
- Ver detalles de sus reservas pasadas y activas.

Administradores del Hotel:

- Añadir o eliminar habitaciones.
- Cambiar detalles de las habitaciones (por ejemplo, precio por noche, número de camas).
- Ver listas de todas las reservas.

Habitaciones:

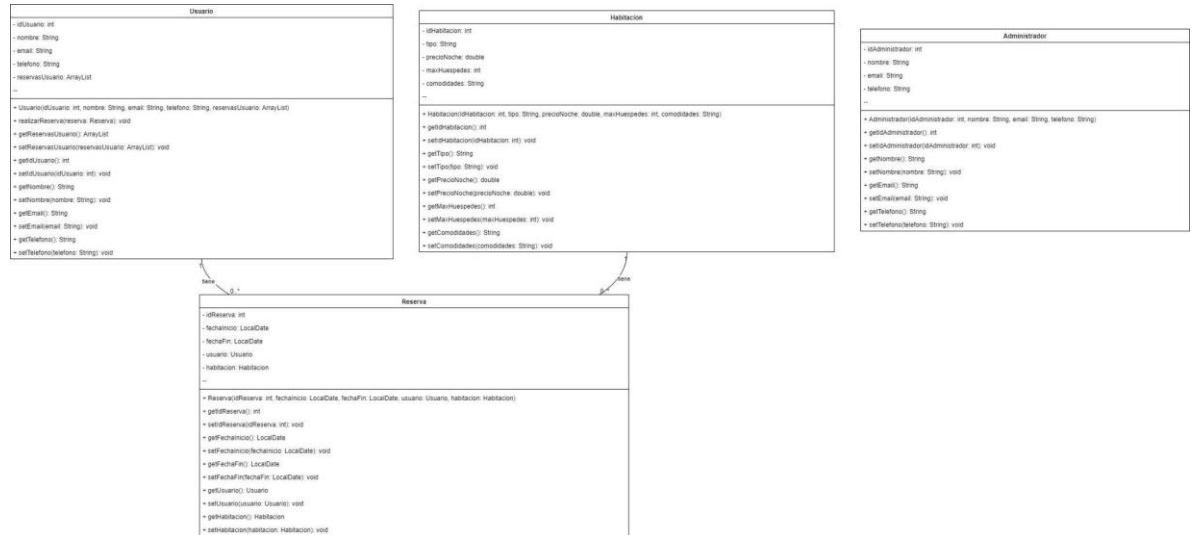
- Deben clasificarse en diferentes tipos (por ejemplo, individual, doble, suite).
- Cada habitación tiene un precio por noche, número máximo de huéspedes, y comodidades.

- **No funcionales**

- Buen aprovechamiento y funcionalidad sencilla en el uso del sistema por medio de la línea de consola.

UML

● DIAGRAMA DE CLASES



Administrador: Esta clase no está relacionada con ninguna otra clase.

Usuario: Esta clase está relacionada con la clase Reserva en una relación uno a muchos. Esto significa que cada usuario puede hacer múltiples reservas, pero cada reserva está asociada con un solo usuario.

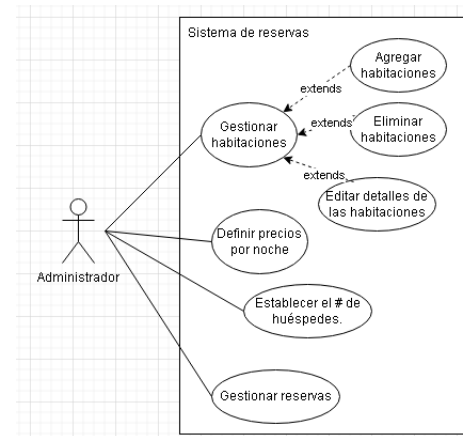
Habitación: Esta clase está relacionada con la clase Reserva en una relación uno a muchos. Esto significa que cada habitación se puede reservar varias veces, pero cada reserva está asociada con una sola habitación.

Reserva: Esta clase está relacionada con la clase Usuario y la clase Habitación en una relación de uno a uno. Esto significa que cada reserva está asociada con un solo usuario y una sola habitación.

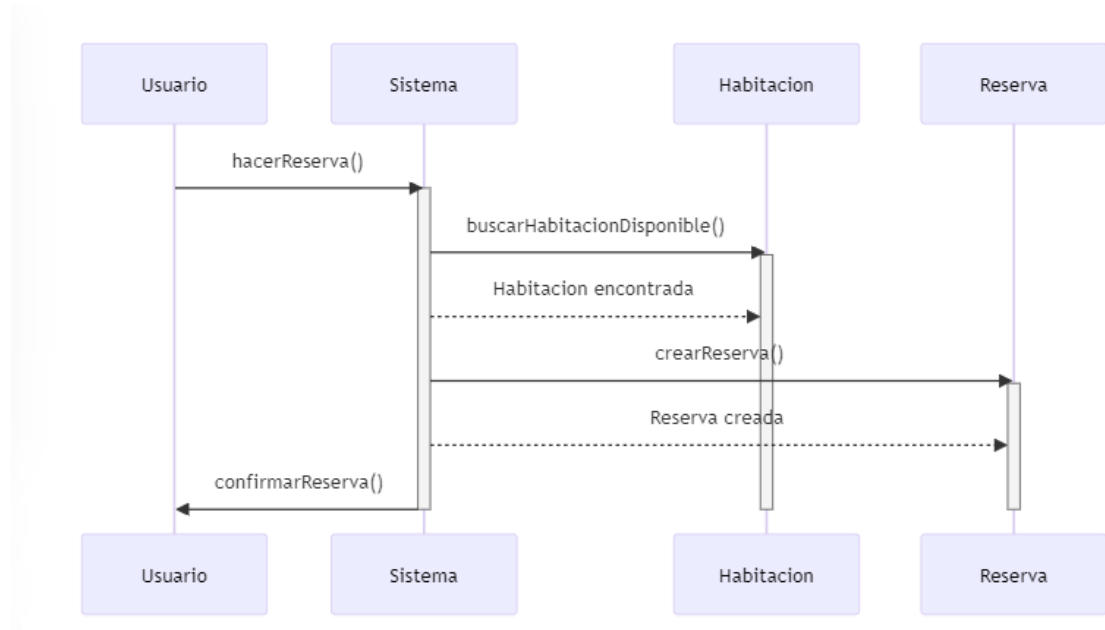
● DIAGRAMA DE CASOS DE USO

Administradores:

Agregar habitaciones.
 Eliminar habitaciones.
 Cambiar los detalles de la habitación.
 Definir el precio por noche.
 Establecer el número máximo de huéspedes.
 Ver las reservas
 Ver habitaciones disponibles
 Hacer reserva
 Cancelar reserva
 Crear usuario
 Cerrar sesión



● DIAGRAMA DE SECUENCIA

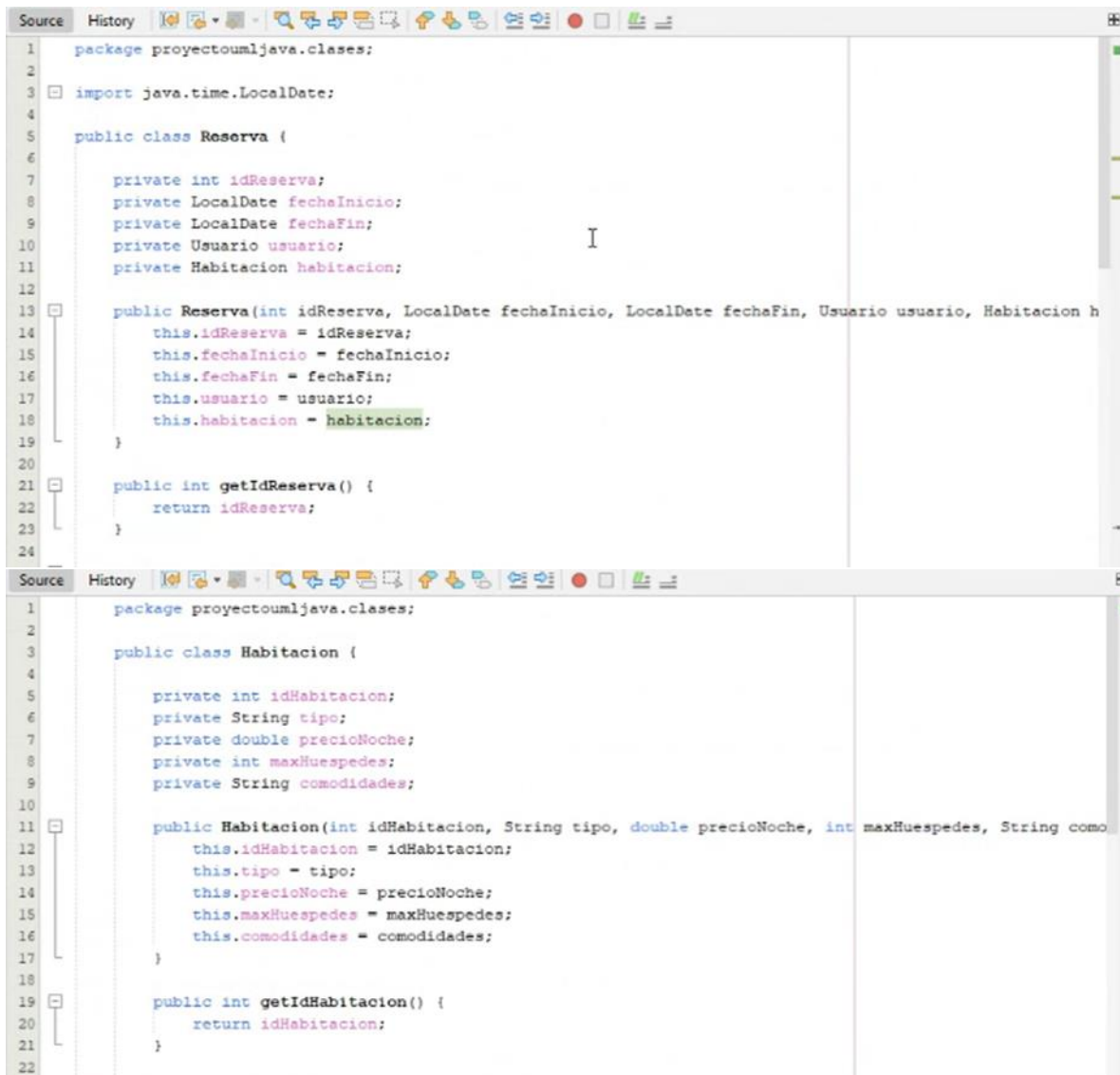


IMPLEMENTACIÓN

Detalles sobre la implementación del sistema en Java:

- CLASES

```
1 package proyectoumljava.clases;
2
3 public class Administrador {
4
5     private int idAdministrador;
6     private String nombre;
7     private String email;
8     private String telefono;
9     private String contraseña;
10
11     public Administrador(int idAdministrador, String nombre, String email, String telefono, String contraseña) {
12         this.idAdministrador = idAdministrador;
13         this.nombre = nombre;
14         this.email = email;
15         this.telefono = telefono;
16         this.contraseña = contraseña;
17     }
18
19 package proyectoumljava.clases;
20
21 import java.util.ArrayList;
22
23 public class Usuario {
24
25     private int idUsuario;
26     private String nombre;
27     private String email;
28     private String telefono;
29     private String contraseña;
30     private ArrayList<Reserva> reservasUsuario;
31
32     public Usuario(int idUsuario, String nombre, String email, String contraseña, String telefono, ArrayList<Reserva> reservasUsuario) {
33         this.idUsuario = idUsuario;
34         this.nombre = nombre;
35         this.email = email;
36         this.telefono = telefono;
37         this.reservasUsuario = reservasUsuario;
38         this.contraseña = contraseña;
39     }
40
41     public void realizarReserva(Reserva reserva) {
42         this.reservasUsuario.add(reserva);
43     }
44 }
```



The image displays two screenshots of a code editor, likely IntelliJ IDEA, showing Java code for two classes: 'Reserva' and 'Habitacion'.

The top screenshot shows the 'Reserva' class. It is located in the package 'proyectoumljava.clases'. It imports 'java.time.LocalDate'. The class has five private attributes: 'idReserva' (int), 'fechaInicio' (LocalDate), 'fechaFin' (LocalDate), 'usuario' (Usuario), and 'habitacion' (Habitacion). It has a constructor 'Reserva' that takes these five parameters and initializes the instance variables. It also has a getter method 'getIdReserva()' that returns the 'idReserva'.

The bottom screenshot shows the 'Habitacion' class. It is also in the package 'proyectoumljava.clases'. It has five private attributes: 'idHabitacion' (int), 'tipo' (String), 'precioNoche' (double), 'maxHuespedes' (int), and 'comodidades' (String). It has a constructor 'Habitacion' that takes these five parameters and initializes the instance variables. It also has a getter method 'getIdHabitacion()' that returns the 'idHabitacion'.

● EXPLICACIÓN DEL CÓDIGO DE LAS CLASES

Usuario: Esta clase representa a un usuario del sistema de gestión de hotel. Contiene atributos como `idUsuario`, `nombre`, `email`, `telefono` y `reservasUsuario`. Los métodos incluyen la capacidad de realizar reservas, obtener y establecer reservas, y acceder a la información del usuario.

Reserva: La clase `Reserva` modela una reserva realizada por un usuario para una habitación en un determinado período de tiempo. Tiene atributos como `idReserva`, `fechaInicio`, `fechaFin`, `usuario` y `habitacion`. Los métodos permiten acceder y establecer la información de la reserva.

Habitacion: Representa una habitación en el hotel, con atributos como `idHabitacion`, `tipo`, `precioNoche`, `maxHuespedes` y `comodidades`. Los métodos proporcionan acceso a esta información y permiten modificarla según sea necesario.

Administrador: Esta clase modela a un administrador del sistema de gestión del hotel, con atributos como idAdministrador, nombre, email y telefono. Los métodos permiten acceder y establecer la información del administrador.

● **EXPLICACIÓN DEL PROGRAMA**

El programa es un sistema de gestión de hotel que permite a los usuarios realizar reservas de habitaciones, y a los administradores realizar tareas de administración, como añadir o eliminar habitaciones, modificar detalles de habitaciones y ver todas las reservas.

El flujo del programa comienza con un menú de inicio de sesión donde los usuarios pueden iniciar sesión o registrarse como nuevos usuarios. Después del inicio de sesión, los usuarios pueden ver habitaciones disponibles, hacer reservas, ver sus propias reservas y cancelarlas si es necesario.

Los administradores, por otro lado, pueden iniciar sesión en una interfaz similar y acceder a opciones adicionales para administrar el sistema, como añadir, modificar o eliminar habitaciones, ver todas las reservas del hotel y crear usuarios nuevos si es necesario.

El programa está diseñado con una estructura de clases que encapsulan la funcionalidad relacionada y permiten una fácil extensión y mantenimiento del código. Se han utilizado listas para almacenar y gestionar las reservas y las habitaciones, proporcionando una estructura de datos eficiente para manejar múltiples operaciones.

El diseño del programa sigue los principios de la programación orientada a objetos (POO), como la encapsulación, la abstracción, la herencia y el polimorfismo, lo que facilita la comprensión y la modificación del código. Además, se han implementado controles de flujo para garantizar una interacción fluida y segura con el sistema.

En resumen, el programa ofrece una solución completa para la gestión de un hotel, brindando a los usuarios la capacidad de realizar reservas de habitaciones de manera sencilla y a los administradores las herramientas necesarias para administrar eficientemente el sistema.

PRUEBAS

INSTRUCCIONES:

```
Bienvenido, Juan!  
1. Ver habitaciones disponibles  
2. Hacer reserva  
3. Ver mis reservas  
4. Cancelar reserva  
5. Cerrar sesión  
Ingrese su opción: 2
```

```
Hacer Reserva:  
Ingrese el ID de la habitación que desea reservar:  
1  
Ingrese la fecha de inicio de la reserva (YYYY-MM-DD):  
2000-02-02  
Ingrese la fecha de fin de la reserva (YYYY-MM-DD):  
2000-02-04  
;Reserva realizada con éxito!
```

```
Bienvenido, Juan!  
1. Ver habitaciones disponibles  
2. Hacer reserva  
3. Ver mis reservas  
4. Cancelar reserva  
5. Cerrar sesión
```

```
Bienvenido, Admin!  
1. Ver habitaciones disponibles  
2. Hacer reserva  
3. Ver mis reservas  
4. Cancelar reserva  
5. Crear usuario  
6. Ver todas las reservas  
7. Añadir habitación  
8. Eliminar habitación  
9. Modificar detalles de habitación  
10. Ver todas las habitaciones  
11. Cerrar sesión  
Ingrese su opción: 1
```

```
Habitaciones Disponibles:  
Habitacion{idHabitacion=1, tipo=Individual, precioNoche=50.0, maxHuespedes=1, comodidades=Wifi, TV  
Habitacion{idHabitacion=2, tipo=Doble, precioNoche=80.0, maxHuespedes=2, comodidades=Wifi, TV, Bañ  
Habitacion{idHabitacion=3, tipo=Suite, precioNoche=150.0, maxHuespedes=2, comodidades=Wifi, TV, Ba
```