

# **Informe Sistema de Reservas de un Hotel**

**Curso: Desarrollo de Software II**

**Docente: Cristian Camilo Cuadrado Beltrán**

**Autor: Camilo Andrés Urrutia Guzmán**

**Fecha: 20/04/2024**

## Descripción del sistema:

El Sistema de Reservas de Hotel es una aplicación web que permite a los usuarios buscar, reservar y gestionar reservas de habitaciones de hotel de manera online. El sistema cuenta con dos tipos de usuarios principales: clientes y administradores.

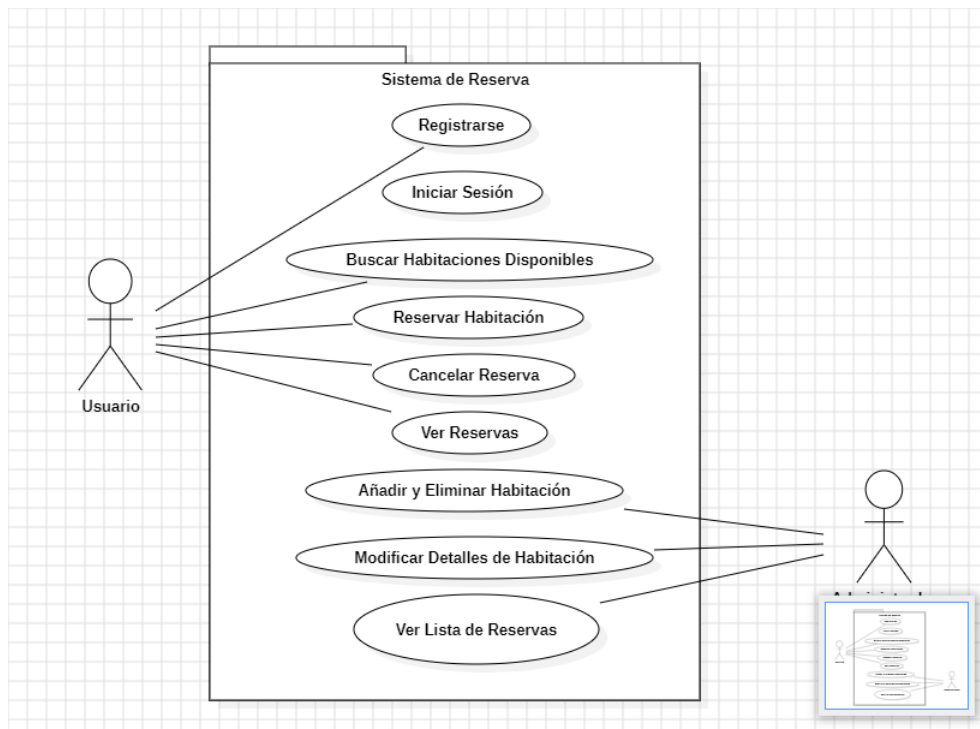
Los clientes pueden registrarse en el sistema, iniciar sesión, buscar habitaciones disponibles por rango de fechas, hacer reservas, cancelar reservas y ver el historial de sus reservas pasadas y activas.

Por otro lado, los administradores del hotel tienen la capacidad de agregar, eliminar y modificar habitaciones, así como ver la lista completa de reservas realizadas por los clientes.

## Diagramas UMLs:

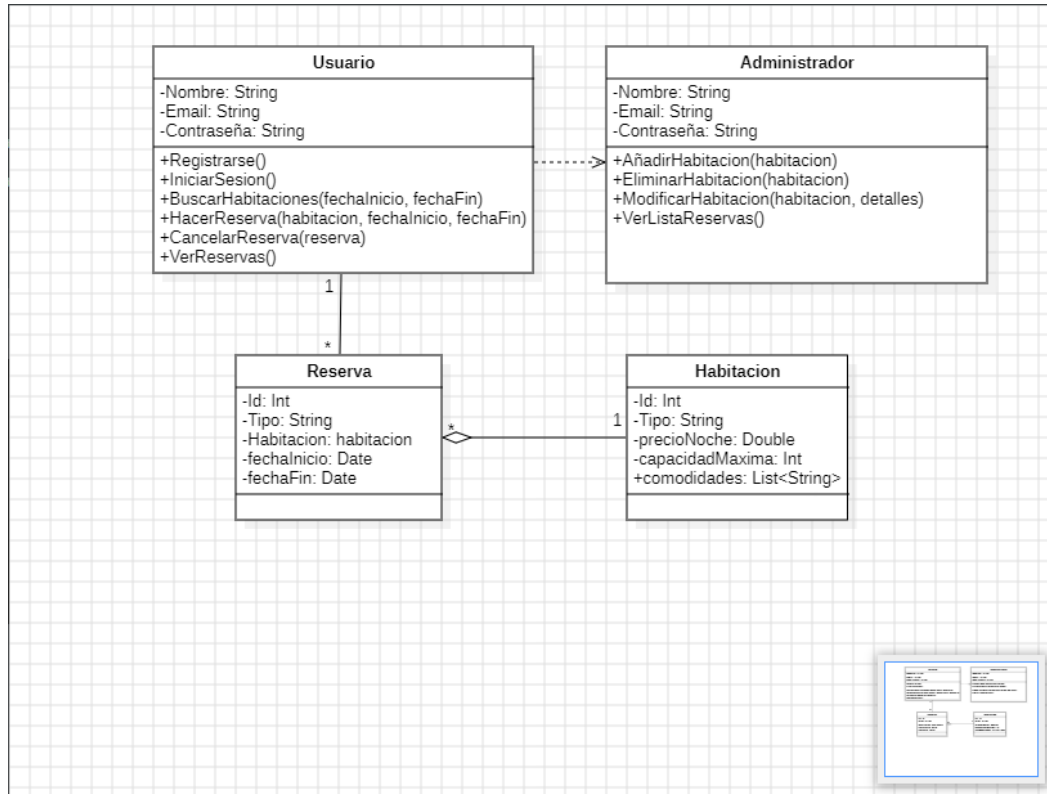
### Diagrama de Casos de Uso:

El diagrama de casos de uso muestra las interacciones principales entre los actores (clientes y administradores) y el sistema. Describe las funcionalidades clave que cada actor puede realizar.



## Diagrama de Clases:

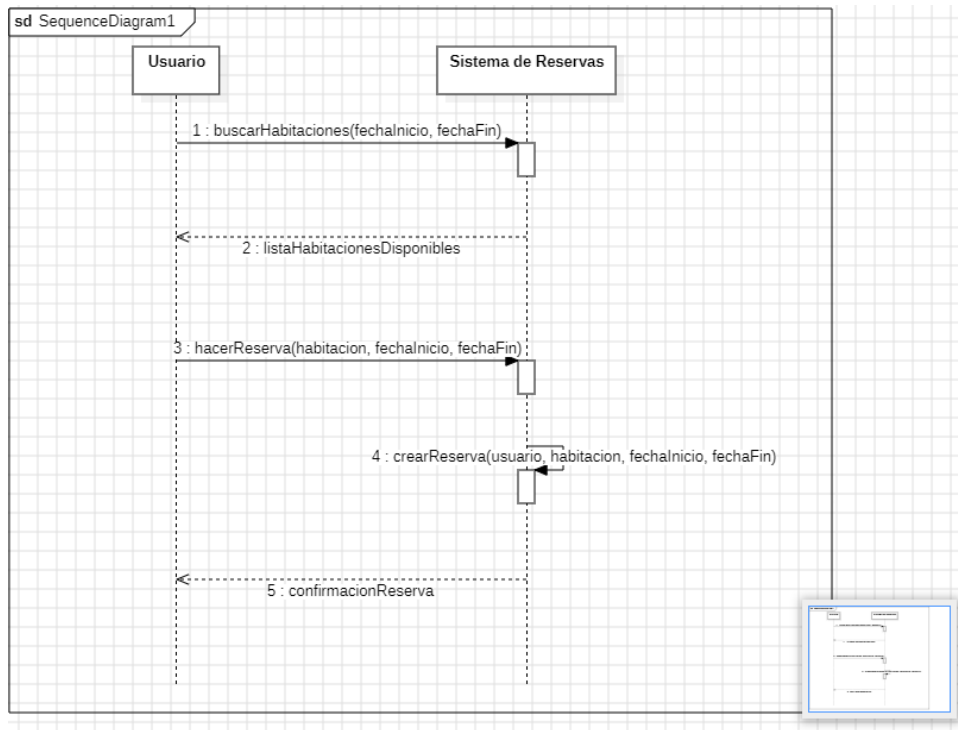
El diagrama de clases representa las entidades principales del sistema (Usuario, Administrador, Reserva, Habitación) y sus relaciones. Muestra los atributos y métodos de cada clase, así como las asociaciones y multiplicidades entre ellas.



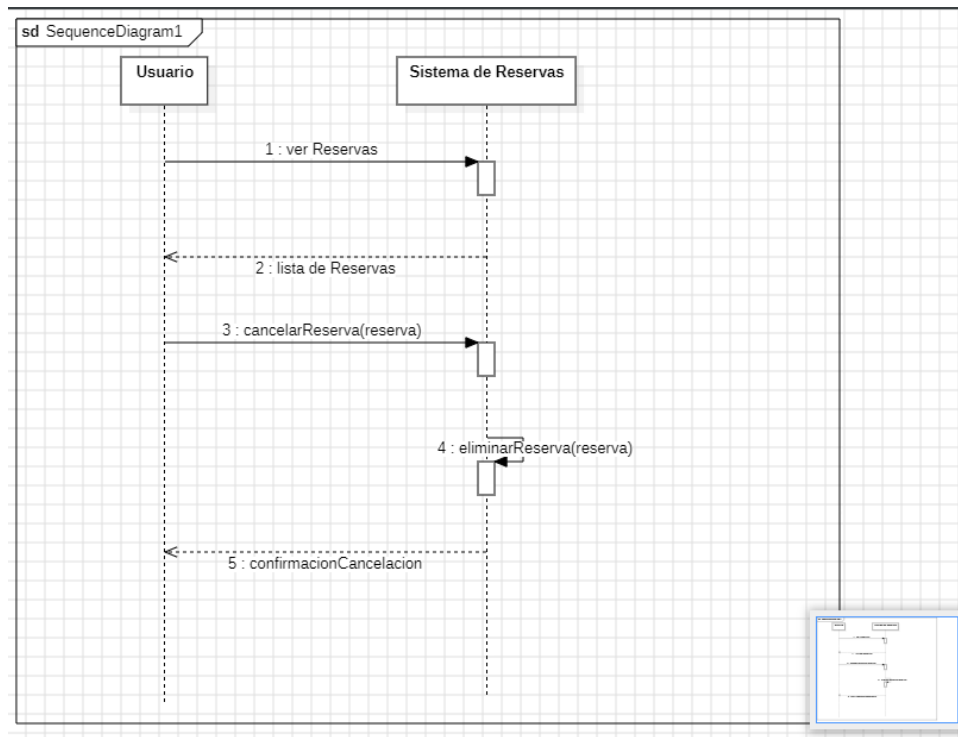
## Diagramas de Secuencia:

Los diagramas de secuencia ilustran cómo interactúan los objetos y las clases del sistema en dos escenarios específicos: hacer una reserva y cancelar una reserva. Muestran el flujo de mensajes y la secuencia de operaciones que se llevan a cabo.

### Reserva:



## Cancelar:



## **Explicación del Código:**

El sistema está implementado en Java y consta de las siguientes clases principales:

### **Usuario:**

```
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

public class Usuario {
    private String nombre;
    private String email;
    private String contraseña;
    private List<Reserva> reservas;

    // Métodos: registrarse(), iniciarSesion(), buscarHabitaciones(), hacerReserva(),
    cancelarReserva(), verReservas()
}
```

Esta clase representa a un usuario del sistema y contiene métodos para realizar acciones como registrarse, iniciar sesión, buscar habitaciones disponibles, hacer reservas, cancelar reservas y ver sus reservas.

### **Administrador:**

```
import java.util.List;

Public class Administrador {
    private String nombre;
    private String email;
    private String contraseña;

    // Métodos: añadirHabitacion(), eliminarHabitacion(), modificarHabitacion(),
    verListaReservas()
}
```

Esta clase representa a un administrador del hotel y contiene métodos para agregar, eliminar y modificar habitaciones, así como para ver la lista completa de reservas realizadas.

**Reserva:**

```
import java.util.Date;

Public class Reserva {

    private int id;

    private Usuario usuario;

    private Habitacion habitacion;

    private Date fechaInicio;

    private Date fechaFin;

    // Constructor y getters/setters

}
```

Esta clase representa una reserva de habitación y contiene información sobre el usuario que realizó la reserva, la habitación reservada, las fechas de inicio y fin, y un identificador único.

**Habitación:**

```
import java.util.List;

Public class Habitacion {

    private int id;

    private String tipo;

    private double precioNoche;

    private int capacidadMaxima;

    private List<String> comodidades;

    // Constructor y getters/setters

}
```

Esta clase representa una habitación del hotel y contiene información sobre el tipo de habitación, el precio por noche, la capacidad máxima y las comodidades disponibles.

**SistemaReservas:**

```

import java.util.ArrayList;
import java.util.Date;
import java.util.List;
public class SistemaReservas {
    private static List<Usuario> usuarios;
    private static List<Habitacion> habitaciones;
    private static List<Reserva> reservas;
    // Métodos para gestionar reservas, habitaciones y usuarios
}

```

Esta clase es la principal encargada de gestionar las reservas, las habitaciones y los usuarios. Contiene métodos para realizar operaciones como buscar habitaciones disponibles, hacer reservas, cancelar reservas, agregar habitaciones, eliminar habitaciones, modificar habitaciones y ver la lista de reservas.

### **Main:**

```

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Date;
import java.util.List;
import java.util.Scanner;
public class Main {
    private static final Scanner scanner = new Scanner(System.in);
    public static void main(String[] args) {
        // Inicialización y flujo principal del programa
    }
}

```

Esta clase contiene el método main que es el punto de entrada del programa. Aquí se inicializan los datos de prueba, se crea la interfaz de usuario basada en consola y se ejecutan las operaciones principales del sistema.

## **Instrucciones para Ejecutar el Sistema:**

1. **Abrir el proyecto en tu entorno de desarrollo preferido:** Abre el proyecto en tu entorno de desarrollo seleccionado. Esto puede ser Eclipse, IntelliJ IDEA, NetBeans u otro.

2. **Configurar el JDK (si es necesario):** Si aún no has configurado el JDK en tu entorno de desarrollo, es posible que necesites hacerlo para que el entorno pueda compilar y ejecutar el código Java correctamente.

3. **Compilar el código:** Utiliza las herramientas de compilación proporcionadas por tu entorno de desarrollo para compilar todas las clases del proyecto.

4. **Ejecutar el programa:** Después de compilar el código, puedes ejecutar el programa desde tu entorno de desarrollo. Si hay una clase principal designada, ejecútala según las opciones proporcionadas por tu entorno de desarrollo.

5. **Seguir las instrucciones en la consola:** Una vez que el programa esté en ejecución, sigue las instrucciones que aparezcan en la consola de tu entorno de desarrollo para interactuar con el programa.

6. **Verificar la salida:** Después de realizar cualquier acción en el programa, verifica la salida para asegurarte de que todo haya funcionado según lo esperado.