

Sistema de Hotel

* Albano Alvarez

* Trabajo Práctico 5

* Programación 2 - Ingeniería en Computación

1. Descripción del contexto

Imaginemos un hotel moderno que ofrece alojamiento temporal para viajeros, ya sea por turismo o negocios. En este contexto, el hotel funciona como un sistema organizado, donde el personal de recepción gestiona reservas, ingresos y egresos de los clientes, asignación de habitaciones, cobros, y disponibilidad general. Cada cliente puede realizar una reserva, alojarse por un período determinado y luego retirarse, dejando la habitación libre para el próximo huésped.

2. Identificación de partes

Dentro del sistema, identificamos los siguientes componentes principales:

- **Hotel:** es la entidad central que reúne todos los elementos y coordina su funcionamiento.
- **Habitación:** unidad física ofrecida para el alojamiento.
- **Recepción:** punto de control operativo donde se registran reservas, check-in/check-out, cobros, etc.
- **Cliente:** persona que utiliza los servicios del hotel.
- **Reserva:** compromiso entre cliente y hotel para el uso de una habitación en determinada fecha.

3. Atributos de las partes

- **Hotel:**
 - habitaciones: listado de todas las habitaciones disponibles.
 - recepcion: referencia al área de gestión operativa.
- **Habitación:**
 - número: identifica la habitación de forma única.
 - tamaño: capacidad del cuarto (single, doble, etc)
 - estado: indica si está disponible, ocupada, en limpieza, etc.
 - precio: valor monetario por noche.

- **Recepción:**
 - reservas: lista de reservas actuales y futuras.
- **Cliente:**
 - nombre, correo electrónico, DNI: datos personales para registro.
 - reserva: reserva activa realizada por ese cliente.
- **Reserva:**
 - fechaReserva: fechas de estadia.
 - precio: precio total a pagar por la estadía.

4. Comportamientos

Hotel:

- abrir() / cerrar(): indica si el hotel está en funcionamiento o no.
- verDisponibilidad(): permite saber qué habitaciones están libres para reservar.

Habitación:

- actualizarEstado(): cambia el estado según si está ocupada, libre, o en limpieza.
- verEstado(): permite consultar si la habitación está disponible.

Recepción:

- buscarReserva(): permite localizar reservas existentes por nombre, fecha o habitación.
- reservarHabitacion(): crea una nueva reserva asociada a un cliente y una habitación.
- checkIn(): marca el ingreso de un cliente y cambia el estado de la habitación a ocupada.
- checkOut(): marca la salida de un cliente y libera la habitación.
- liberarHabitacion(): vuelve a poner una habitación como disponible.
- cobrar(): realiza el cobro del importe de la reserva.

Cliente:

- realizarPago(): acción mediante la cual paga su estadía.

Reserva:

- verPrecio(): muestra el costo total de la reserva, calculado a partir de la habitación asociada y duración.

Ejemplo de flujo y cambios en la información

1. Un cliente llama para reservar.

- La **recepción** ejecuta `reservarHabitacion()`, crea una **reserva** y asocia una **habitación**.
- El estado de la habitación cambia a "reservada".

2. El cliente llega al hotel.

- La **recepción** hace `checkIn()`, cambia el estado de la **habitación** a "ocupada".

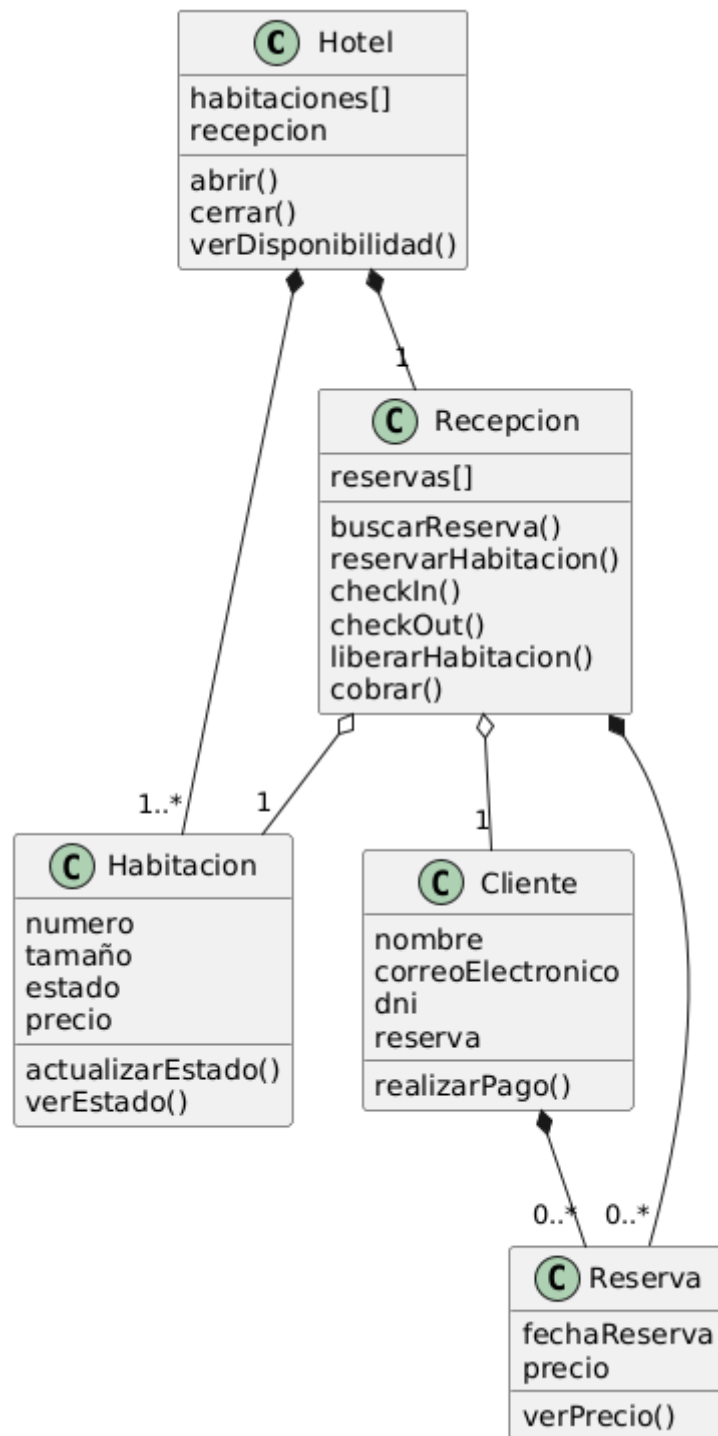
3. Durante la estadía:

- La **reserva** contiene la fecha y precio, la **recepción** puede consultar el estado o cobrar.

4. El cliente se va:

- Se hace `checkOut()`, se llama a `liberarHabitacion()`, cambiando su estado a "disponible".
- El cliente hace `realizarPago()` y la **recepción** ejecuta `cobrar()`.

ULM



@startuml

```
class Hotel {
    habitaciones[]
    recepcion
    abrir()
    cerrar()
    verDisponibilidad()
}
```

```
class Habitacion {
    numero
    tamaño
    estado
    precio
    actualizarEstado()
    verEstado()
}
```

```
class Recepcion {
    reservas[]
    buscarReserva()
    reservarHabitacion()
    checkIn()
    checkOut()
    liberarHabitacion()
    cobrar()
}
```

```
class Cliente {
    nombre
    correoElectronico
    dni
    reserva
    realizarPago()
}
```

```
class Reserva {
    fechaReserva
    precio
    verPrecio()
}
```

Hotel *-- "1..*" Habitacion
Hotel *-- "1" Recepcion

Recepcion *-- "0..*" Reserva
Recepcion o-- "1" Cliente
Recepcion o-- "1" Habitacion

Cliente *-- "0..*" Reserva

@enduml