

# Documento de diseño de Property Management System

**Natalia Ortega, Paulina Arrazola, Natalia Espitia**

## 1. Contexto del problema

El siguiente diseño corresponde a un sistema llamado "Property Management System" o PMS, para administrar elementos importantes de los hoteles, como las habitaciones, los servicios y las reservas.

Este sistema tiene cuatro agentes principales: el administrador, los recepcionistas, los empleados, y los huéspedes. Por parte del administrador, éste tiene la potestad de crear y modificar las habitaciones, establecer las tarifas por tipo de cuarto, así como el rango de fechas para el cual aplicará esta tarifa y los días de la semana que le corresponden. Por otro lado, este agente puede modificar las tarifas de los servicios que ofrece el hotel y crear los menús del restaurante con base en platos que él introduce en el sistema. En cambio, los empleados que trabajan en la recepción son los únicos que tienen el permiso en el sistema para crear una reserva a nombre de un huésped y cancelarla. Además, estos son los encargados de ingresar al sistema los datos básicos del huésped y de sus acompañantes. Continuando con los agentes relevantes, los empleados son los encargados de reportar los consumos de los huéspedes y acceder a la factura correspondiente cuando sea necesario. Finalmente están los huéspedes, los cuales no tienen responsabilidades ya que no pueden acceder directamente al sistema, estos siempre van a actualizar su información en el PMS a través de alguno de los anteriores agentes.

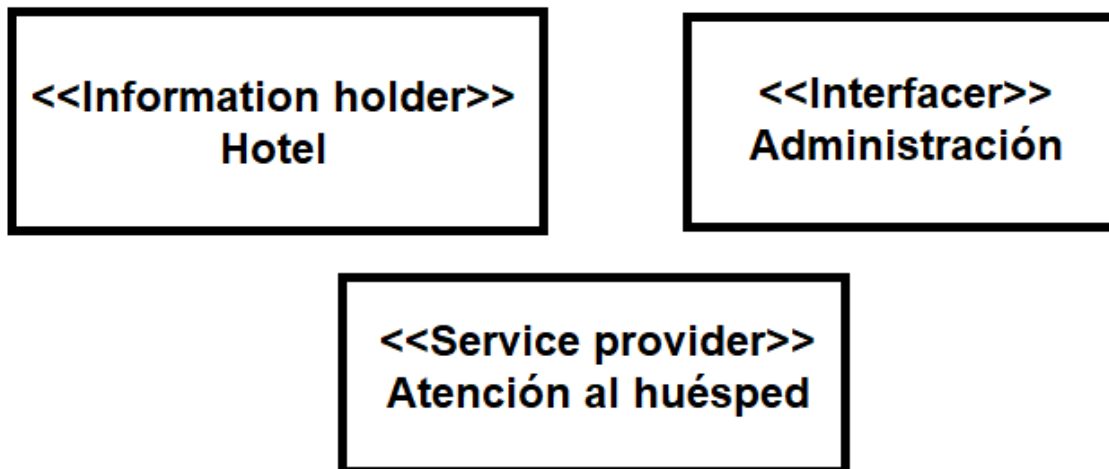
## 2. Nivel 1

### 2.1 Componentes candidatos y estereotipos

- ❖ Respecto al manejo de la información se necesita tener la información del hotel para que este luego pueda ser consultado o notificado. Dicho componente mantiene y entrega información, lo cual hace referencia al estereotipo de *Information holder* y este se denominará "Hotel".

- ❖ Es necesario recalcar las responsabilidades de que ofrecen un servicio al cliente como hacer una reserva, registrar la entrada, generar una factura entre otros, estos están enfocados en hacer un trabajo para el huésped, lo cual hace referencia al estereotipo de *Service provider* y este se denominará “Atención al huésped”.
- ❖ En orden de tener la información en el componente hotel es necesario que exista un componente cree o actualice esta información, esto significa que sería el encargado de transformar la información, por lo cual el estereotipo de este componente sería *Interfacer* y se denominará “Administración”.

La figura 1 expuesta a continuación presenta los cuatro componentes candidatos identificados con su respectivo estereotipo.



## 2.2 Responsabilidades

De forma implícita, para considerar los componentes candidatos y los estereotipos en los que encajan, se tuvieron en cuenta las responsabilidades que estos ocuparían. La siguiente tabla presenta en detalle cada una de las responsabilidades y el componente asociado que debe asumirlas.

#	Responsabilidad	Componente
1	Crear habitaciones	Administración
2	Actualizar habitaciones	
3	Crear tarifas de habitaciones	
4	Actualizar tarifas de habitaciones	
5	Configurar menú	
6	Notificar fecha de tarifa	Hotel
7	Registrar pagos	
8	Guardar información de menú	
9	Guardar información de habitaciones	
10	Guardar información de tarifas	
11	Registrar servicios	Atención al huésped
12	Generar cobro	
13	Hacer reserva	
14	Registrar entrada	
15	Registrar salida	
16	Generar facturas	
17	Cancelar reserva	
18	Registrar consumo	

## 2.3 Colaboraciones

- Realizar reserva:

Conduce a que el recepcionista realice una reserva en el sistema a nombre de un huésped. Con este fin,

- ❖ El recepcionista le provee al sistema el nombre del huésped, documento, correo electrónico, número de celular y la cantidad total de personas que se esperan.
- ❖ El sistema verifica que las fechas deseadas por tipo de habitación estén disponibles.
- ❖ Si las habitaciones están disponibles, el sistema bloquea las habitaciones necesarias quedarán bloqueadas en las fechas de la reserva.
- ❖ El sistema confirma la reserva exitosa de las habitaciones al recepcionista.

- Cancelar reserva:

Conduce a eliminar una reserva hecha previamente por el recepcionista en el sistema a nombre de un huésped. Con este fin,

- ❖ El recepcionista le debe comunicar al sistema las fechas de la reserva y el nombre del huésped que la hizo.
  - ❖ Validar que no se encuentre dentro de las 48 horas antes de la reserva.
  - ❖ En caso de que se cumpla lo anterior, el sistema debe liberar la/s habitaciones en la fecha que el huésped tiene la reserva.
  - ❖ El sistema le debe mandar una confirmación de proceso exitoso al recepcionista.
- Realizar consumos y generar las facturas correspondientes:
    - ❖ Un empleado debe informarle al sistema cuando hay un consumo por parte del huésped.
    - ❖ El sistema registra el consumo que realizó el huésped.
    - ❖ Cuando un grupo de huéspedes termine su estadía, un recepcionista podrá generar las facturas para todos los consumos realizados durante su estadía.
- Avisar de fecha disponible:
    - ❖ El sistema le debe avisar al administrador si hay alguna fecha dentro de los próximos 365 días en la que no exista una tarifa asignada para un cierto tipo de habitación.

## 3. Nivel 2

### 3.1 Hotel

Las responsabilidades asociadas a este componente son las siguientes:

- ❖ Notificar fecha de la tarifa
- ❖ Registrar pagos
- ❖ Guardar información del menú
- ❖ Guardar información de las habitaciones
- ❖ Guardar información de las tarifas
- ❖ Registrar servicios

Considerando que las responsabilidades listadas anteriormente, y que esta principalmente almacena información aportada por otras clases es factible pensarla como una clase directamente, la cual sería la clase sistema, esta tendría implícitamente la información de los pagos, habitaciones, tarifas, servicios del hotel, menú. Estas

interacciones para obtener la información las hace tanto con el componente “Administración” como con el componente “Atención al huésped”.

### **3.2 Administración**

Las responsabilidades asociadas a este componente son las siguientes:

- ❖ Crear habitaciones
- ❖ Actualizar habitaciones
- ❖ Crear tarifas de habitaciones
- ❖ Actualizar tarifas de habitaciones
- ❖ Configurar menú

Considerando las responsabilidades listadas anteriormente, podemos evidenciar que hay un gran componente de creación y actualización por lo cual se agrupara en varias clases, teniendo como principal la clase “Administrador” ya que este es el que se encarga directamente de estas responsabilidades, pero también se toma en consideración las clases que se crean, las cuales serían “Menú”, y por ende “Plato” ya que hace parte de este, “Habitación” y “Tarifas”, estas serían las clases que conformarían el componente “Administración”.

### **3.3 Atención al huésped**

Las responsabilidades asociadas a este componente son las siguientes:

- ❖ Generar cobro
- ❖ Hacer reserva
- ❖ Registrar entrada
- ❖ Registrar salida
- ❖ Cancelar reservas
- ❖ Generar facturas
- ❖ Registrar consumos

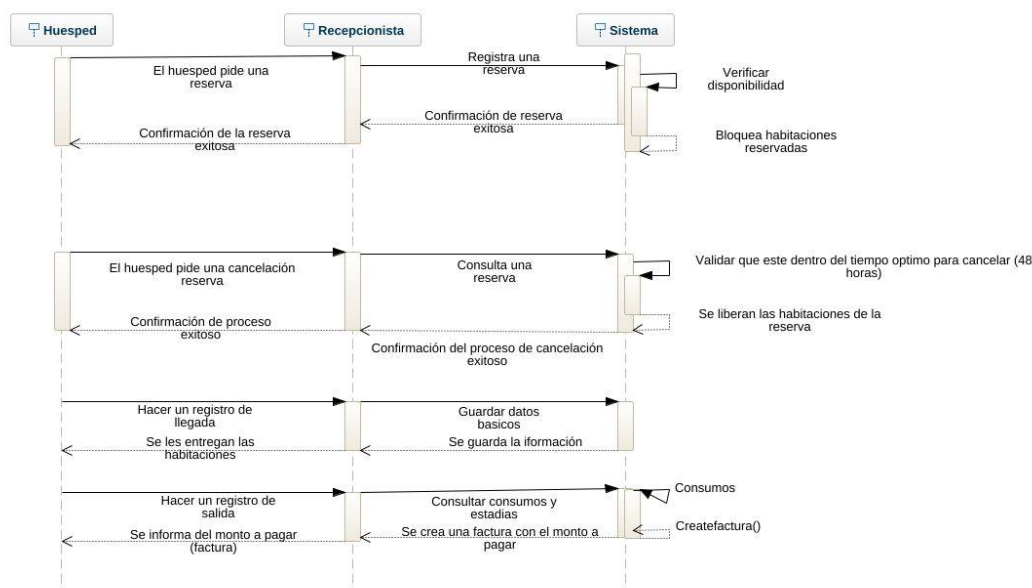
Considerando las responsabilidades listadas anteriormente, debemos tener en cuenta que estos trabajos serán ejecutados por más de una clase, las acciones relacionadas con reserva, entrada, salida y generación de factura están directamente relacionadas con el recepcionista, la responsabilidad asociada al consumo está directamente relacionada con el empleado, ya sea de spa, turismo o demás servicios que ofrezca el Hotel, y el cobro si está directamente relacionado con la habitación, todos estas responsabilidades son trabajos para el huésped. Teniendo en cuenta lo anterior y todas las clases que se relacionan para poder cumplir estos trabajos tenemos que las clases involucradas en el componente de “Atención al huésped” están: “Empleado”, “Recepcionista”, “Habitación”, “Servicios”, “Consumo”, “Factura”, “Reserva”, “Huésped”, “Grupo”.

## 4. Persistencia

Con el fin de asegurar la persistencia del sistema, se decide utilizar archivos de texto para almacenar la información de las reservas, platos, bebidas, servicios y habitaciones del hotel. Cuando alguno de los usuarios (administrador, recepcionista, empleado) decide añadir nueva información dentro de sus facultades como usuario, el sistema se encarga de guardar esta información en los archivos de texto anteriormente mencionados para acceder a ella en cualquier momento. Por otro lado, cuando se debe actualizar información que ya se encuentra dentro de los archivos de texto el sistema busca dónde se encuentra esa información dentro del archivo y la sobrescribe por los nuevos valores insertados por el usuario. En general, los archivos de texto manejados contienen los atributos de una clase separados por “;” para poder diferenciarlos.

## 5. Nivel 3

### 4.1 Diagrama de secuencia



En la imagen anterior se muestra la relación de secuencia de diferentes interacciones entre el huésped y el recepcionista que, de igual manera, lleva a una interacción entre el sistema y el recepcionista en donde este primero es el principal encargado de dar una respuesta hacia el requerimiento que se le pida desde el recepcionista. Esta interacción también se describe en el primer apartado 2.3 Colaboraciones.



En la imagen anterior se muestra la relación de secuencia una de las principales interacciones entre el sistema y el administrador que lleva a una interacción en donde este administrador se encarga de la función principal que en este caso es actualizar la tarifa de una habitación. Esta interacción también se describe en el tercer apartado 2.3 Colaboraciones.

## 6. Nivel 4

### 5.1 Diseño final

#### Actualizaciones:

- ❖ Se creo la clase bebida que anteriormente era un atributo de la clase de Menú o restaurante. Este cambio es necesario, ya que la clase plato tiene el precio del mismo, el tipo y demás características que tiene un plato. De igual manera, se completó este tipo de características en la clase de plato.
- ❖ Se creo la clase cama que se tiene las características de una cama. Gracias a esta clase fue necesario implementar una enumeración en el que describiera el tipo de huésped que se pudiera registrar en una cama.
- ❖ Se crearon varias enumeraciones nuevas como las disponibilidades que eran necesario para poder hacer una distinción mejor lo necesario en el código de java.

