

## **Proyecto 1 – Entrega 3 – PMS**

### **Contexto del problema**

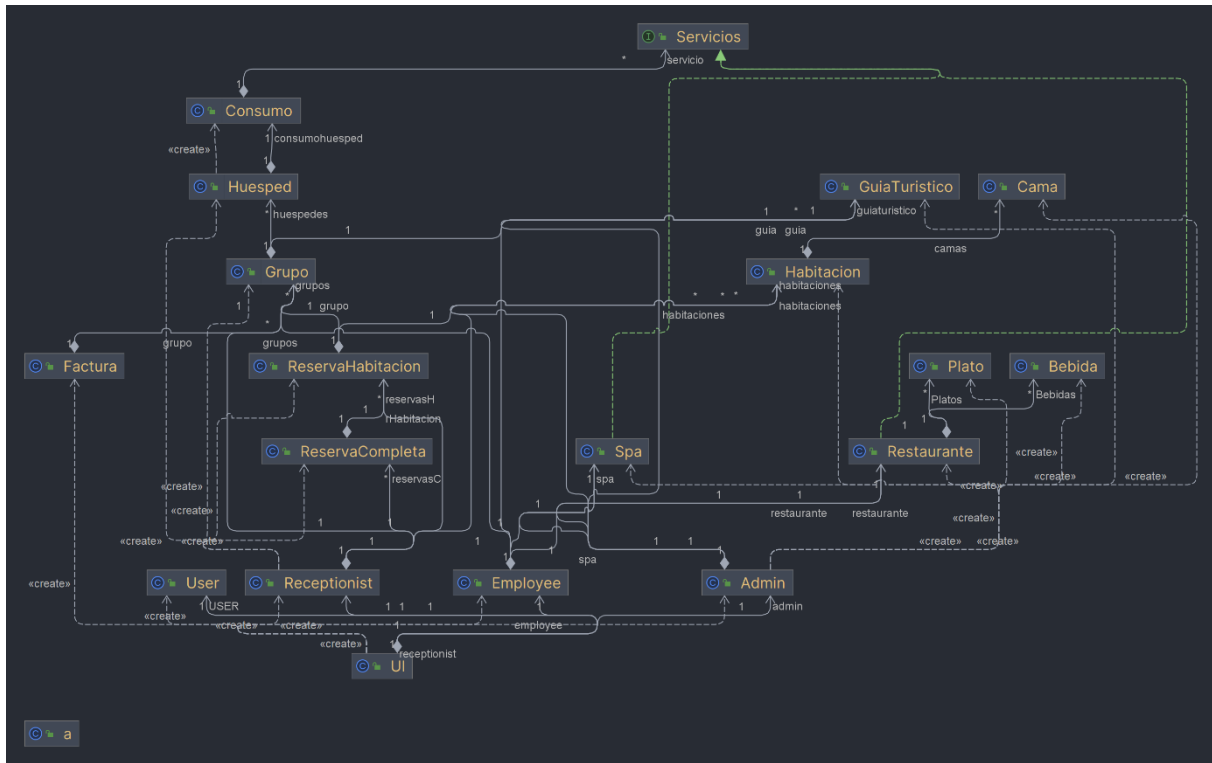
El texto describe un sistema llamado PMS (Property Management System) que se encarga de administrar varios elementos importantes de un hotel. El sistema consta de tres grandes funcionalidades: un inventario de habitaciones del hotel, un catálogo de servicios y un registro de consumo, y el manejo de todos los aspectos relacionados con las reservas, los huéspedes y los pagos.

El inventario de habitaciones consta de habitaciones tipo estándar, suite y suite doble, cada una con un identificador, ubicación e información sobre su capacidad y características, como balcón, vista y cocina integrada. Las camas tienen información sobre su tamaño y capacidad. El sistema permite al administrador crear habitaciones una por una o cargar un archivo con información de nuevas habitaciones o habitaciones existentes. Los empleados de la recepción del hotel pueden consultar el inventario y las características de las habitaciones para saber quién las está ocupando o quién las tiene reservadas en una determinada fecha.

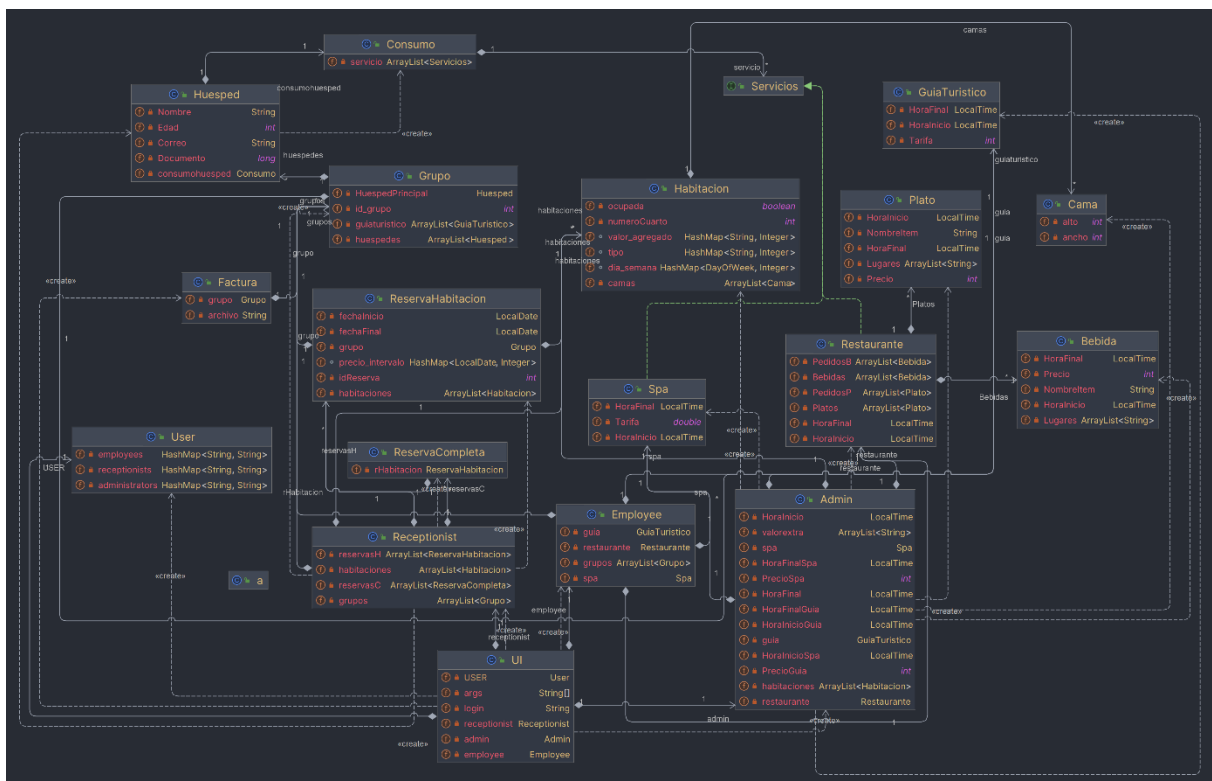
El sistema también permite al administrador cargar tarifas por tipo de habitación, válidas para ciertas fechas y días de la semana. El sistema toma la tarifa más económica en caso de que haya más de una tarifa para la misma habitación en la misma fecha. El administrador también puede cambiar las tarifas de los servicios, cargar los menús del restaurante y configurar lo necesario para cada plato. El sistema registra los servicios que han sido consumidos por un huésped, los pagos que se han hecho y genera las facturas correspondientes.

Los huéspedes del hotel pueden hacer reservas a través de un empleado de la recepción, y las habitaciones necesarias quedarán bloqueadas en las fechas de la reserva. En el momento del registro, el empleado pedirá los datos básicos del huésped y sus acompañantes para registrarlos en el sistema. Un grupo de huéspedes y sus acompañantes pueden tener una o varias habitaciones asignadas y se consideran como un solo grupo. El límite de personas por habitación depende de la capacidad de las camas. Los servicios de restaurante pueden ser tomados en el comedor o en la habitación, pero no todas las opciones del menú pueden ser llevadas a la habitación. El sistema tiene información completa sobre el menú, incluyendo nombres, precios y rangos de horas disponibles.

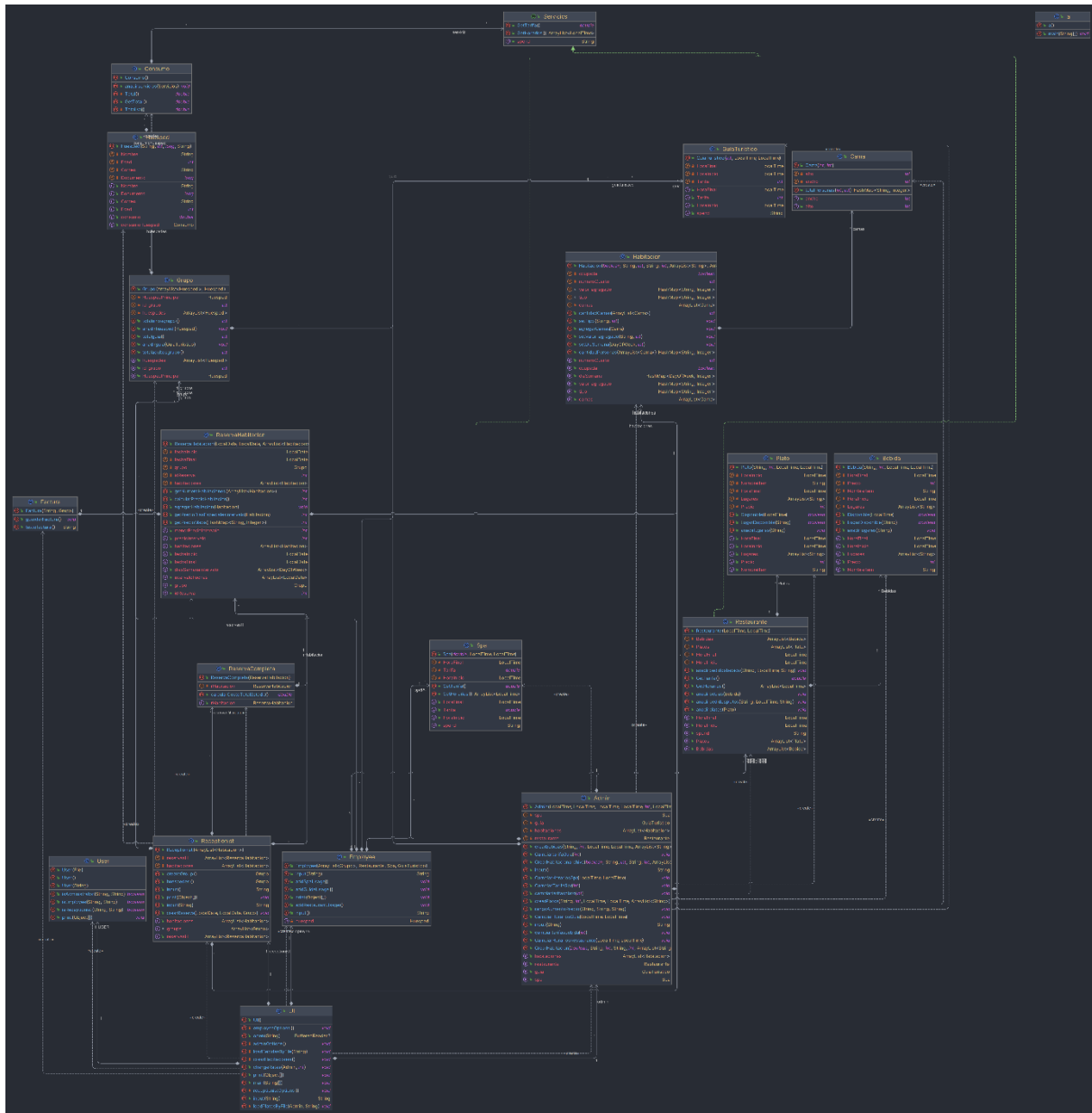
## Diagrama de diseño UML de alto nivel (inicial)



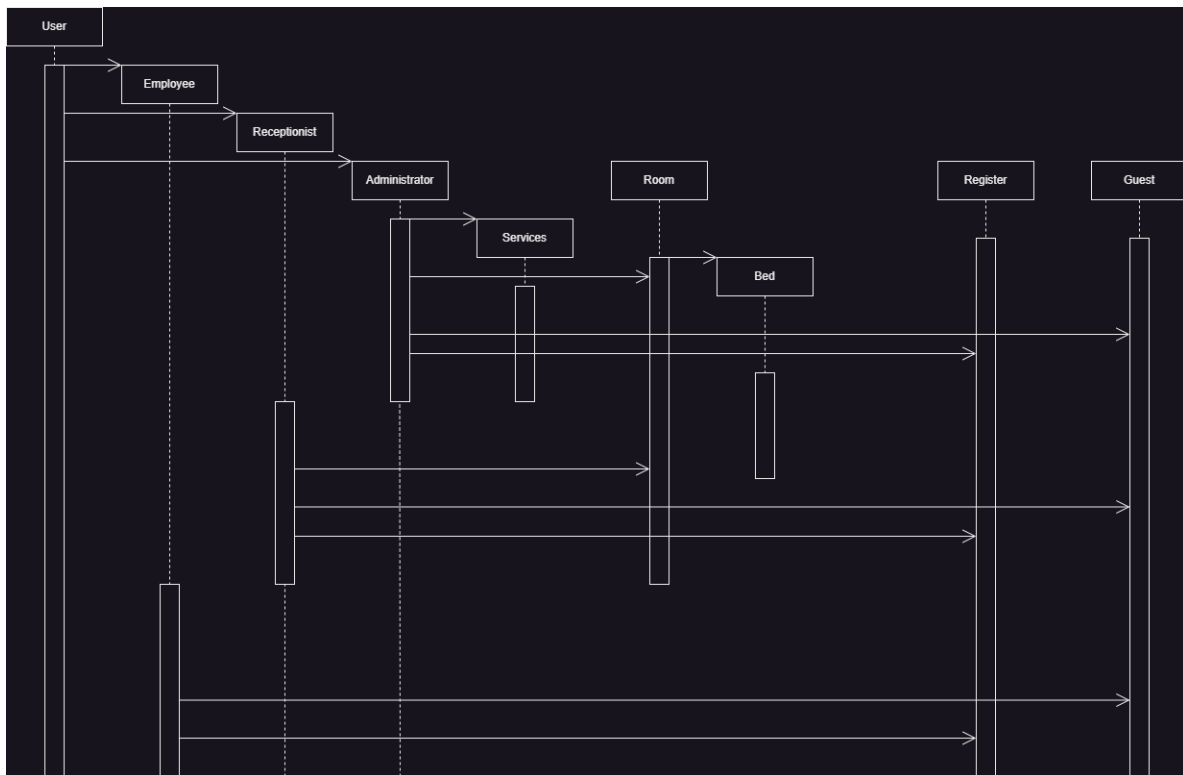
## Diagrama de diseño UML de alto nivel (medio)



### Diagrama de diseño UML de alto nivel (final)



## Diagrama de secuencia



## **Documentación de las funciones de cada Clase**

Para evitar redundancia no se dará soporte a la función del constructor

### **Bebida:**

Disponible: Indica si la bebida está disponible de acuerdo con los atributos de Hora Inicio y Hora final

### **Plato:**

Disponible: Indica si el plato está disponible de acuerdo con los atributos de Hora Inicio y Hora final

### **Restaurante:**

GetPlatos\Bebidas: Obtiene cada uno de los platos y bebidas del restaurante

Disponibilidad: Retorna un hashmap con platos y bebidas cuyo valor es un booleano que indica si están disponibles

GetHorarios: Retorna los horarios del restaurante

GetPrecios: Retorna un hashmap con cada uno de los items del menú cuyo valor es el respectivo precio.

GetPedidosP: Retorna los platos pedidos por el cliente

GetPedidosB: Retorna las bebidas pedidas por el cliente

GetTarifa: Retorna el costo total de los pedidos

### **Spa:**

GetTarifa: Retorna el precio por el uso.

GetHorarios: Retorna cuando está disponible el Spa

**(Spa y Restaurante al ser servicios individuales se les relaciono una interfaz con métodos abstractors de GetTarifa y GetHorarios)**

### **Guía turístico:**

Solo contiene getters de su respectiva tarifa y horarios de disponibilidad

### **Consumo:**

GetTotal: Retorna el consumo total por cada uno de los servicios del cliente, también se incluye el precio de la habitación con los impuestos.

### **Cama:**

TotalPersonas: Retorna el total de las personas que caben por cama de acuerdo a su tipo o tamaño (No tiene sentido que una cama que es de tipo para una persona, aunque tenga tamaño considerable quepan 2)

### **Complementario:**

GetComplemento: Retorna el complemento de la habitación

### **Habitación:**

GetPrecioTotal: Retorna el precio total de la habitación dependiendo de la fecha

GetNumeroCamas: Retorna el total de camas de la habitación

GetnumeroCuarto: Retorna el id del cuarto.

GetTipo: Retorna el tipo de cuarto.

GetComplementos: Retorna los complementos de la habitación

Cambiartarifasfechas: Cambia las tarifas de acuerdo con una fecha previamente asignada

AgregarTarifas: Agrega tarifas de la habitación a fechas no asignadas

### **Huesped**

Solo contiene getters para cada uno de sus atributos (Nombre, edad, documento, correo)

### **Grupo:**

IdGrupo: Retorna el id del grupo

GetHuespedPrincipal: Retorna el huésped responsable del grupo es decir el que tiendra a su nombre la reserva y la factura.

### **ReservaHabitaciones:**

Tiene getters del nombre del huésped principal, del intervalo de tiempo de cada habitación, del número de habitaciones usadas.

AgregarHabitaciones: Agrega habitaciones a la reserva.

### **ReservaCompleta:**

GetHorarioIntervalo: Retorna el intervalo de la reserva.

GetHabitaciones: Retorna una lista de las habitaciones usadas en la reserva.

GetPrecioTotal: Retorna el precio total de las habitaciones reservadas.

### **Factura:**

TotalServicioIndividual: Crea un costo total del consumo de los servicios individuales.

TotalHabitación: Crea un costo del valor de la habitación

TotalServicioGrupal: Crea un costo del total de servicios grupales, es este caso solo el guía turístico.

TotalConImpuetos: Crea un costo del total consumido con impuestos.

GenerarFactura: Genera la factura con los consumos y costos con el id de la reserva

### Asignación de responsabilidades

#	Responsabilidad	Componente
1	Indicar si la bebida está disponible	Bebida
2	Indicar si el plato está disponible	Plato
3	Obtener cada uno de los platos y bebidas del restaurante	Restaurante
4	Obtener los horarios del restaurante	
5	Obtener los platos y bebidas pedidos por cliente	
6	Obtener el costo total de los pedidos	
7	Saber la disponibilidad del spa	Spa
8	Calcular el precio por el uso del spa	
9	Saber la tarifa y horarios del guía turístico	Guía turístico
10	Saber el total de personas que caben en una cama	Cama
11	Saber el complemento de la habitación	Complementario
12	Saber el precio total de la habitación dependiendo de la fecha	Habitación
13	Saber el total de camas de la habitación	
14	Saber el id del cuarto	
15	Saber el tipo de cuarto	
16	Saber los complementarios de la habitación	
17	Agregar y cambiar tarifas	Huésped
18	Saber la información del huésped	
19	Saber la id del grupo	Grupo
20	Saber quién tiene a su nombre la reserva y la factura	
21	Saber el nombre del huésped principal	ReservaHabitaciones
22	Saber el intervalo de tiempo de cada habitación	
23	Saber el número de habitaciones usadas	
24	Agrega habitaciones a la reserva	
25	Saber el intervalo de reserva	ReservaCompleta
26	Retornar una lista de las habitaciones usadas en la reserva	
27	Retornar el precio total de las habitaciones reservadas	Factura
28	Calcular el costo total del consumo de los servicios individuales	
29	Calcular el costo del valor de la habitación	
30	Calcular el costo total de servicios grupales	
31	Calcular el costo del total con impuestos	
32	Generar una factura con los consumos y costos con el id de la reserva	

## **Persistencia**

Para nuestro proyecto, la descripción del paso de la información por el programa se describe a continuación. Inicialmente el administrador carga la información del hotel referente a los servicios especiales (esta parte está relacionada directamente con las clases de Restaurante, Plato, Bebida Spa y GuiaTuristico), a las habitaciones (esta parte está relacionada con las clases de Cama, habitación y ReservaHabitacion) y la reserva en sí (esta parte incluye la clase de ReservaCompleta, Factura y las mencionadas anteriormente). El recepcionista tiene acceso a la información referente a las reservas de la habitación, de donde se obtienen los datos de las habitaciones usadas y sus costos respectivos). Los empleados del hotel tienen acceso a las clases relacionadas a su área en concreto (restaurante, spa, etc.) y puede acceder al inventario de este para poder generar facturas del consumo de los huéspedes (o, en dado caso, dejarlo para cobrar al momento del check-out). Al final el administrador puede generar una factura con todos los costos hechos por el huespued (o huéspedes en caso de que se esté alojando un grupo).

## **Justificaciones**

El proyecto presenta un estilo de control delegado debido a las restricciones del proyecto. Cada usuario presenta diferentes opciones y diferentes accesos por lo tanto cada usuario (empleado, recepcionista y administrador) juega un rol de coordinador. Un estilo centralizado complicaría su implementación.

El proyecto tiene dos grandes divisiones, la primera que es la implementación de servicios y la segunda que es la implementación de alojamiento. Los servicios se encuentran divididos en individuales y grupales, donde los individuales son el restaurante, el spa los cuales debido a métodos de Tarifa y Horario son agrupados a través de una interfaz, los servicios grupales son el guía turístico. Se dividieron de dicha forma para facilitar el cálculo del gasto y la creación de la factura.

El alojamiento a pesar de que también es un servicio grupal es tratado como un elemento distinto debido a que cuenta con demasiadas opciones difíciles de agrupar con el guía turístico. Además, se crearon dos tipos de reserva que en un principio pueden parecer redundantes que es la ReservaCompleta y la ReservaHabitación, se realizaron con el fin de que al acceder al inventario se pueda acceder a las reservas de habitación por aparte.

La factura se genera a partir del consumo de los huéspedes y a partir del consumo del grupo, donde la reserva completa contiene la información del precio total a pagar por alojamiento.

Las responsabilidades se asignaron de acuerdo con los componentes (clases) presentados en los diagramas UML y a las funcionalidades de cada uno de estos. Una vez ya estaban definidos los métodos que iban a utilizar cada una de las clases, asignar las responsabilidades fue algo muy sencillo.



