**­­­­Descripción de la solución:**

Para la solución planteada para el PMS se plantean las siguientes clases:

* PropertyManagementSystem
* Usuario (Abstracta)
* Huésped
* Titular
* Reserva
* Estadía
* Factura
* Disponibilidad
* Cama
* Habitación
* Pago
* Consumible<Interface>
* Servicio<Abstract>
* Producto
* Spa
* Restaurante
* ProductoRestaurante
* Alojamiento
* FileManager

**Estereotipos**

|  |  |
| --- | --- |
| **Nombre de Clase** | **Estereotipo** |
| * PropertyManagementSystem | **Controller** |
| * Usuario | **Information holder** |
| * Huésped | **Information holder** |
| * Titular | **Information holder** |
| * Habitación | **Service Provider** |
| * Reserva | **Interfacer** |
| * Estadía | **Structure** |
| * Factura | **Interfacer** |
| * Disponibilidad | **Information holder** |
| * Cama | **Information holder** |
| * Pago | **Information holder** |
| * Consumible<Interface> | **<Interface>** |
| * Servicio<Abstract> | **Information holder** |
| * Producto | **Information holder** |
| * Spa | **Service Provider** |
| * ProductoRestaurante | **Service Provider** |
| * Alojamiento | **Service Provider** |
| * Restaurante | **Service Provider** |
| * FileManager | **Service Provider** |

**Información de clases**

**PropertyManagementSystem**

Es quién se encarga de hacer las actividades propias del sistema. Tales como cargar los precios, validar los usuarios (recepcionista, administrador), registrar el huésped, cargar las habitaciones, registrar las estadías previas, crear una reserva e iniciar una estadía

NOTA: La información se carga a través de la clase FileManager, esto quiere decir que las habitaciones, las estadías registradas, los productos y los precios estarán almacenados en diferentes archivos.

**Usuario**

Se compone de un login(user) y un password. El usuario solo puede ser un empleado del hotel. Cada usuario tiene un rol designado, el cual va a ser usado para validar las funcionalidades dentro del sistema. Este rol es un enum.

**Huésped**

El huésped va a ser la clase padre que se usa como un information holder para almacenar toda la información que se requiere de los huéspedes (edad, nombre etc.)

NOTA: Si es un huésped, pero no es titular, se es tratado como un acompañante por lo que no se pasa ningún recargo por los consumibles facturados, ya que estos van directamente al titular.

**Titular (Hereda de Huésped)**

Es hijo de huésped, contiene más información sobre el huésped (email, teléfono), debido a que sobre este se hace la facturación de los consumibles y la estadía de la habitación

**NOTA Grupo:**

Se considera un grupo si la lista de huéspedes de la clase estadía es mayor a uno.

**Reserva**

Tiene un identificador, la información necesaria para la realizar la reserva, como es el caso de las habitaciones, las personas que se van a quedar y las fechas de entrada/salida del hotel.

NOTA: La reserva no es lo mismo que una estadía, la reserva se contiene en una estadía, ya que esta estadía empieza el día en el que el/los huésped(es) empiecen a hospedarse en el hotel

**Estadía**

La estadía inicia en el momento en el que los huéspedes llegan al hotel, sobre la estadía se va a hacer el recargo sobre las habitaciones usadas, además de almacenar las demás facturas de consumibles generadas durante el tiempo de alojamiento.

**Factura**

La factura es generada por cada consumible, ya que cada consumible sabe como generar la factura, la factura tiene información del valor total del consumible y de su respectivo pago.

**Pago**

Tiene la información del precio y del día en el que registra el pago

NOTA: El pago existe porque puede pasar que el consumible se pague inmediatamente o se deje a la factura final (si no se paga de inmediato el pago es nulo), así que como la estadía contiene la información de todas las facturas solo tiene que verificar aquellas que no tienen pago registrado.

**Disponibilidad**

Es contenida dentro de la habitación contiene el precio, el estado de disponibilidad (reservada, ocupada, mantenimiento), la fecha y la reserva asociada (esta puede ser nula).

NOTA: la habitación tiene la información de las 365 disponibilidades del año, para casos en la que el precio de la habitación cambia dependiendo del día del año, o en que la habitación ya se encuentra reservada para cierto tiempo.

**Cama**

Contiene la información de capacidad de la cama, es decir cuantos niños y cuantos adultos caben dentro de la cama.

**Habitación**

Contiene el numero de habitación, la capacidad, la disponibilidad, el tipo de habitación y las características de esta (balcón, cocina, vista). Además, se encarga de generar el precio para la reserva.

**Consumible<Interface>**

Señala el contrato funcional de facturar() ya que cada consumible tendrá que generar la factura que se almacenará en estadía.

**Producto**

Contiene la información del id, el nombre del producto y el precio. Esta clase solo es un information hoder.

**ProductoRestaurante (Hereda de producto)**

Contiene lo mismo que el producto, solo que, con ciertas restricciones ligadas al restaurante, como es el caso de que si se lleva a la habitación o el rango de disponibilidad horaria.

**Restaurante**

Contiene la información de los productosRestaurante, así como el rango horario de cada producto.

**Spa**

Contiene uno o más servicios o productos usados/adquiridos dentro de este.

**Alojamiento**

Contiene la lista de habitaciones usadas en el alojamiento, es el encargado de generar la factura de las habitaciones usadas en función a los días de hospedaje

**FileManager <static>**

Es el que se encarga de cargar, y almacenar los archivos que se requieren para el property management system.