

Diseño y Programación Orientada a Objetos

Proyecto 1

Entrega 2 - Diseño

Documento de análisis

María Viviana Lara Peña 202122798

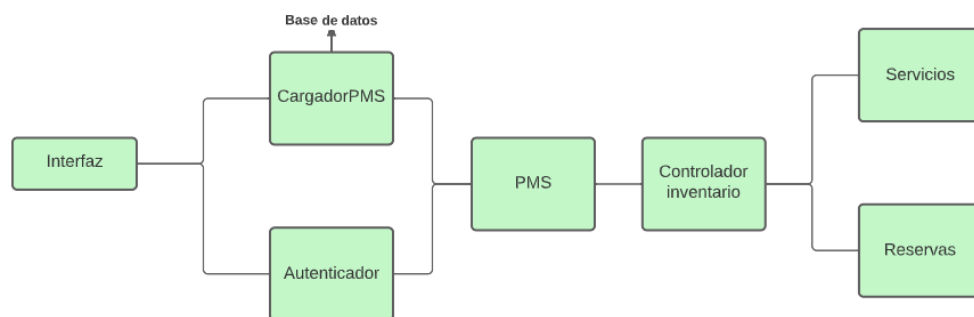
Giuliana Volpi Muñoz 202123986

David Tobón Molina 202123804

Opciones extra:

- Responsibility-Driven Design
- Definir estilos de control
- Descripción de candidatos (estereotipos)
- Agrupación/Organización de elementos
- Tabla de responsabilidades
- Diseño de colaboraciones

Para comprender el diseño realizado se planteó un diagrama de funciones que resume la forma en la que nos aproximamos y describimos la situación de estudio.



Descripción Diagrama 1:

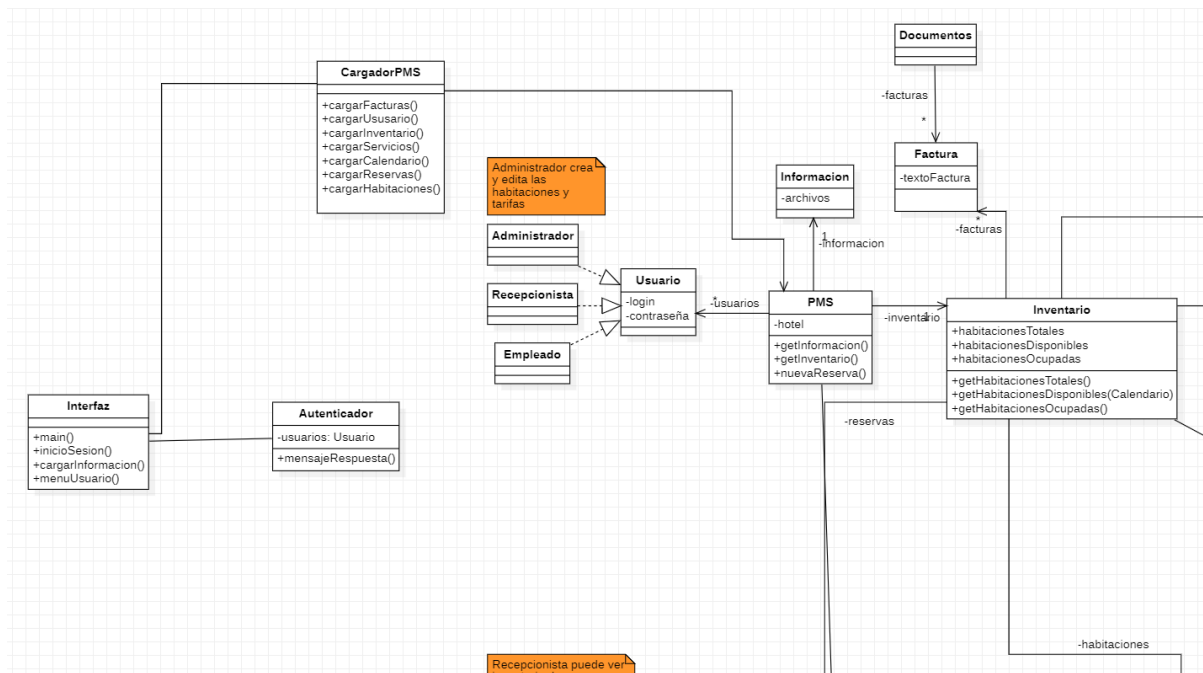
La situación de estudio fue descompuesta en los siguientes elementos básicos a través de los cuales se manejará la información y se resolverán las transacciones deseadas entre los protagonistas y su contexto.

La aplicación comienza con una interfaz donde el usuario se puede registrar e ingresar al programa. Se identifica el tipo de usuario a través de autenticador y se cargan los datos.

Sigue la parte central de la aplicación que toma el nombre del modelo de central de reservas y control del hotel, en esta parte está toda la información del hotel, los datos actuales y los pasados, todas las características y funcionalidades del hotel, este núcleo está directamente conectado con el controlador

de inventarios que es donde se registra la disponibilidad de las habitaciones y a partir de dicho inventario se puede realizar una reserva que es uno de los requisitos principales de la aplicación.

Al haber realizado una reserva se puede acceder a las funcionalidad y servicios del hotel, cubriendo así todas las necesidades de la aplicación PMS.



Descripción Diagrama 2 Inicio de Aplicación:

Interfaz y Autenticador:

La aplicación comienza en la interfaz principal con el método main. Además de correr el programa, este es el espacio en el cual el usuario se registra, inicia sesión y carga los archivos asociados al programa. Por último, se le muestra al usuario el menú y al seleccionar una opción podrá seguir interactuando con la aplicación.

Se decidió darle estas cuatro responsabilidades a la interfaz principal ya que para que la aplicación pueda desarrollarse es necesario que corran las cuatro y en la primera instancia del programa.

El autenticador es una barrera de protección del programa ya que al tentativo usuario que ingrese al programa e intente iniciar sesión, el autenticador se encargará de validarlo y de identificar el tipo de usuario que es, después de esto podrá seguir con las otras responsabilidades de la interfaz.

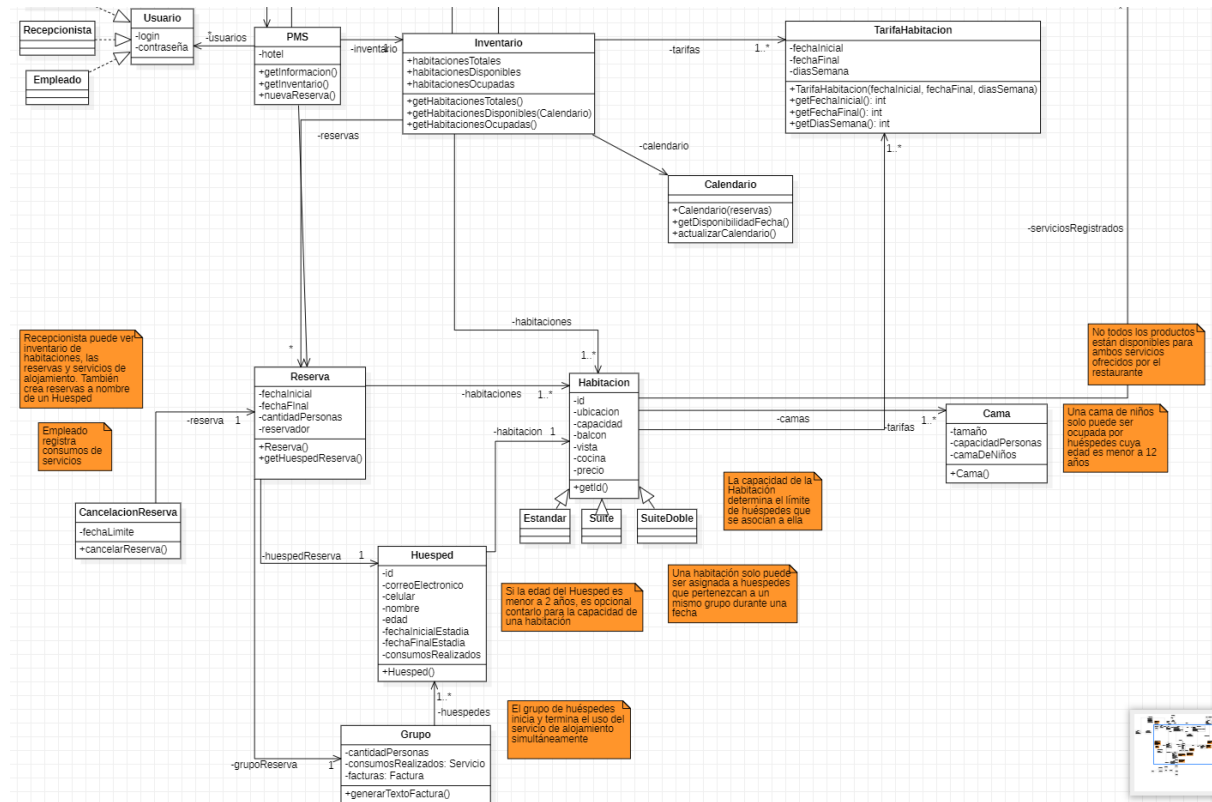
Cargador PMS y PMS:

Es la clase que se encarga de cargar toda la información necesaria para el funcionamiento de la aplicación, dicha información está dividida por temas, las facturas pasadas, el inventario hasta la última actualización realizada, los usuarios existentes, los servicios, el calendario y las habitaciones.

Toda esta información cargada estará a disposición del núcleo de la aplicación, es decir de la clase llamada "PMS", dicha clase representa la idea del hotel, su integridad, todas sus características y datos. Consideramos la clase "PMS" como núcleo de la aplicación ya que a través de ella se realiza la funcionalidad principal, realizar una reserva.

Usuario:

Se decidió crear una clase “Usuario” ya que el rol de quien esté utilizando la aplicación es relevante para el funcionamiento de la misma. Esta es una superclase para Administrador, Recepcionista y Empleado quienes heredan los atributos y métodos de “Usuario” que por el momento se reducen a un login y una contraseña, pero esto se podría profundizar y realizar de forma más detallada, por lo tanto, dejamos los tipos en usuarios en clases separadas.



Descripción Diagrama 3 Sistema de Reservas e Inventario:

Como estipulado en el enunciado, un solo huésped debe realizar la reserva, éste, accede a la información del hotel, es decir a los tipos de habitaciones a través del núcleo de la aplicación, el PMS que al iniciar la aplicación cargó los archivos de los datos de las habitaciones y sus características y de la disponibilidad de ellas.

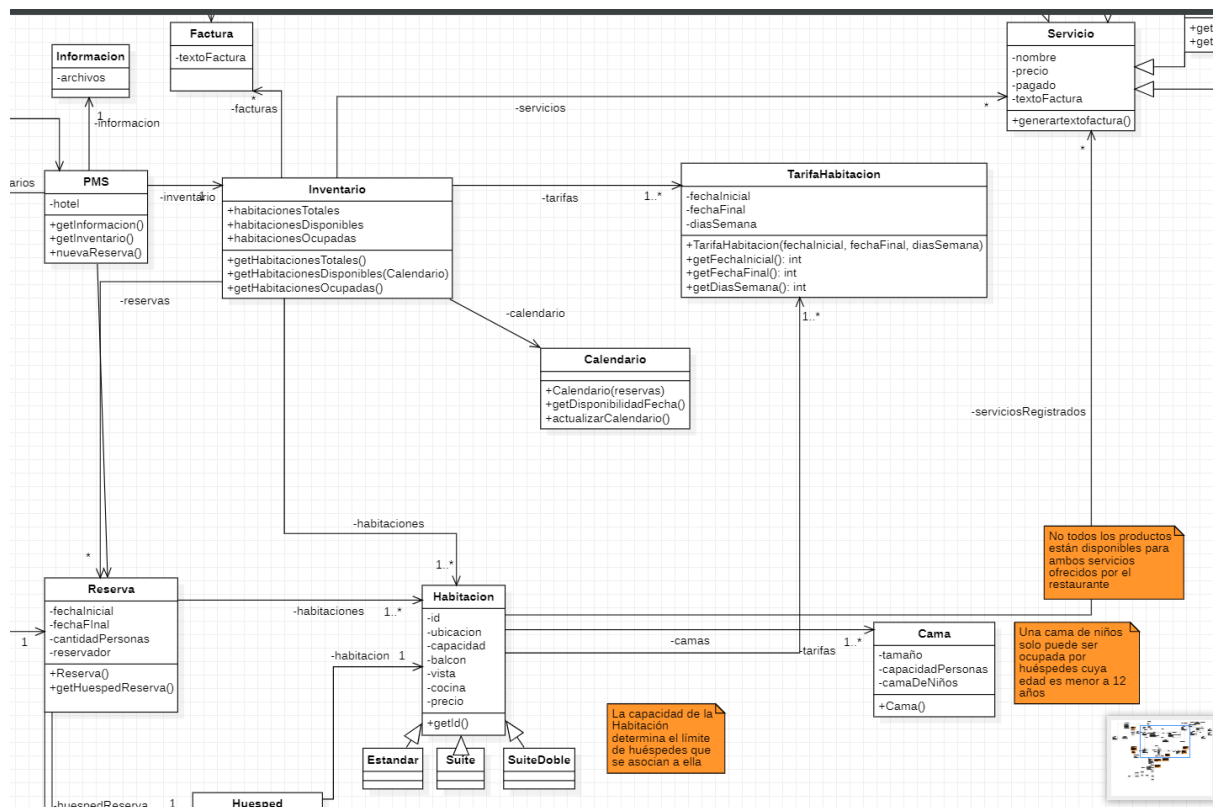
Inventario y Calendario:

Para mostrarle al huésped la disponibilidad de las habitaciones decidimos crear una clase inventario de donde se pueda obtener la información de las habitaciones existentes y cuáles de ellas están disponibles y cuales ocupadas. Para esto, el inventario se actualiza con una clase llama calendario que a su vez está conectada con la clase de reservas, cuando se procesa una reserva el calendario se actualiza y cambia la disponibilidad de las habitaciones de acuerdo a la reserva realizada y esta es la base de datos que se utiliza para saber si las habitaciones que un nuevo usuario desea están disponibles o no. La actualización del calendario ocurre en el método constructor de dicha clase. Otra funcionalidad importante de la clase “Calendario” es garantizar persistencia en el programa ya que en dicha clase quedan registradas las disponibilidades de las habitaciones durante todo el año.

Reserva:

Si los días que el huésped desea alojarse en el hotel resultaron disponibles se puede realizar la reserva, para esto se requiere el rango de fechas, la cantidad de personas y el método constructor se encargará de realizar la reserva. A partir de una reserva existente el usuario tendrá acceso a todas las demás funcionalidades y servicios del hotel.

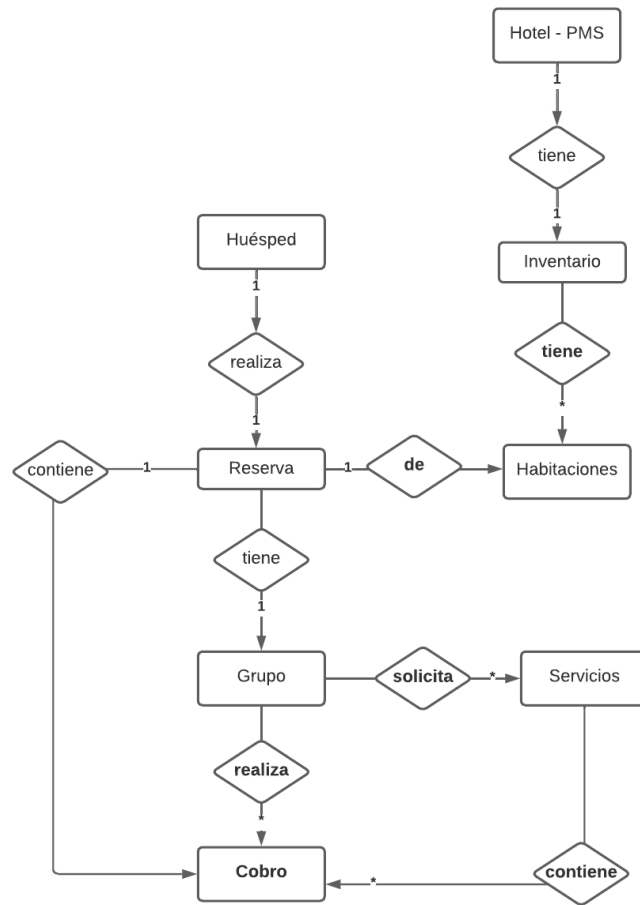
Siguiendo las restricciones establecidas, se creó una clase llamada “Cancelación Reserva” donde se podrá cancelar la reserva si se está dentro de los límites de tiempo establecidos.



Descripción Diagrama 4 Habitaciones y Tarifas:

Habitación es una interfaz de la cual heredan tres clases, cada una correspondiente a cada tipo de habitación: estándar, suite o suite doble. La información de las habitaciones se podrá acceder por otras clases a través de su id. Cuando un usuario consulte el inventario, las reservas contendrán la información de las fechas en que cierta habitación este en uso. Por su parte, una habitación tendrá como atributo una lista de tarifas, las cuales se aplicarán durante ciertos días entre dos fechas, y determinarán el valor del atributo *precio* de la habitación.

Para garantizar la persistencia del programa se creó la clase “Factura” donde se guardan las facturas que se realicen.



Modelo entidad - relación