Documentación Proyecto Tres

1. Generalidades

a. Propósito del sistema

La idea en la que se centra esta aplicación gira entorno al sistema conocido como *Property Management System* o PMS y su implementación en el entorno de un hotel. El principal objetivo es que por medio de esta aplicación el hotel sea capaz de administrar elementos y actividades esenciales para su funcionamiento.

Entre las funcionalidades del sistema se encuentran el manejo de inventario de las habitaciones y el catálogo de servicios ofrecidos por el hotel, el registro de consumos, el manejo de los aspectos relacionados a las reservas, huéspedes y pago. Adicionalmente el sistema también cuenta con un programa separado del PMS el cual está diseñado para el uso de los huéspedes del hotel, dicho programa le permite al huésped consultar la disponibilidad de habitaciones en un rango de fechas, hacer reservas y realizar el pago inmediato.

b. Objetivos del diseño

Con esta aplicación se busca brindarle una plataforma de tipo PMS que le permita a empleados y usuarios de un hotel tener acceso a la información del mismo y realizar acciones básicas para su funcionamiento.

El sistema garantiza en primer lugar la seguridad y protección de los datos, esto implica que ciertos datos tienen acceso restringido según el rol del usuario, por ejemplo un empleado del restaurante no tiene acceso para modificar los datos de las tarifas. De igual manera no es posible acceder al sistema sin un usuario y contraseña registrado.

En segundo lugar, el sistema está disponible sin restricciones de tiempo debido a que su funcionamiento no se ve afectado por agentes externos al usuario que esté haciendo uso del mismo en el momento.

Finalmente, se buscó priorizar la fiabilidad y usabilidad de la aplicación. Los requerimientos expuestos a lo largo de las guías de trabajo de los proyectos han sido implementados de forma que el usuario solo tenga que escoger una opción en el menú de opciones e ingresar los datos necesarios para realizar una acción o hacer una consulta. Otros aspectos como la carga de datos y el almacenamiento de los mismos han sido automatizados para agilizar el sistema y facilitar el uso del mismo.

2. Especificaciones Proyecto 3

Adicional a los requerimientos detallados en el análisis, para este proyecto se implementan los siguientes requerimientos:

- Además de las características ya definidas de los cuartos ahora la descripción de un cuarto incluye:
 - o Tamaño en metros cuadrados
 - o Aire acondicionado
 - Calefacción
 - o Tamaño de la cama o cama
 - Televisión
 - o Ropa de cama y tapetes hipoalergénicos
 - Plancha
 - o Secador de cabello
 - Voltaje AC
 - o Tomas USB (pueden ser de tipo A o C)
 - Incluye desayuno.
- El hotel tiene características propias, tales como:
 - o Parqueadero pago en el hotel
 - o Parqueadero gratuito en el hotel
 - o Piscina
 - Zonas húmedas
 - o BBO
 - o Wifi gratis
 - o Recepción 24 horas
 - Admisión de mascotas
- La aplicación debe ser capaz de recibir pagos con tarjetas de crédito y por lo tanto debe tener un mecanismo que le permita integrarse con las diferentes pasarelas de pago. Para esto se recibe la información de la tarjeta y la información de un pago a hacer. Como retorno se debe informar si el pago fue exitoso o no.
- La aplicación debe tener un nuevo programa con interfaz gráfica que sea independiente al PMS y sea de uso exclusivo para huéspedes. Dicho programa cuenta con las siguientes funcionalidades:
 - o Crear usuario para un huésped.
 - o Consultar la disponibilidad de habitaciones en un rango de fechas.
 - o Reservar una habitación después de validar su disponibilidad
 - o Pagar inmediatamente una reserva de habitación con tarjeta de crédito.
- La aplicación genera reportes al administrador con las ventas del restaurante del hotel. Los reportes cubren aspectos como:
 - Ventas por producto
 - o Valor de las facturas a lo largo del tiempo

- o Relación entre el valor del consumo en el restaurante y el valor del consumo por noche en cada cuarto.
- o Reservas por mes
- La implementación del proyecto incluye pruebas de integración sobre las funcionalidades relacionadas al proceso de reserva y pruebas unitarias sobre las funcionalidades para la carga de archivos.

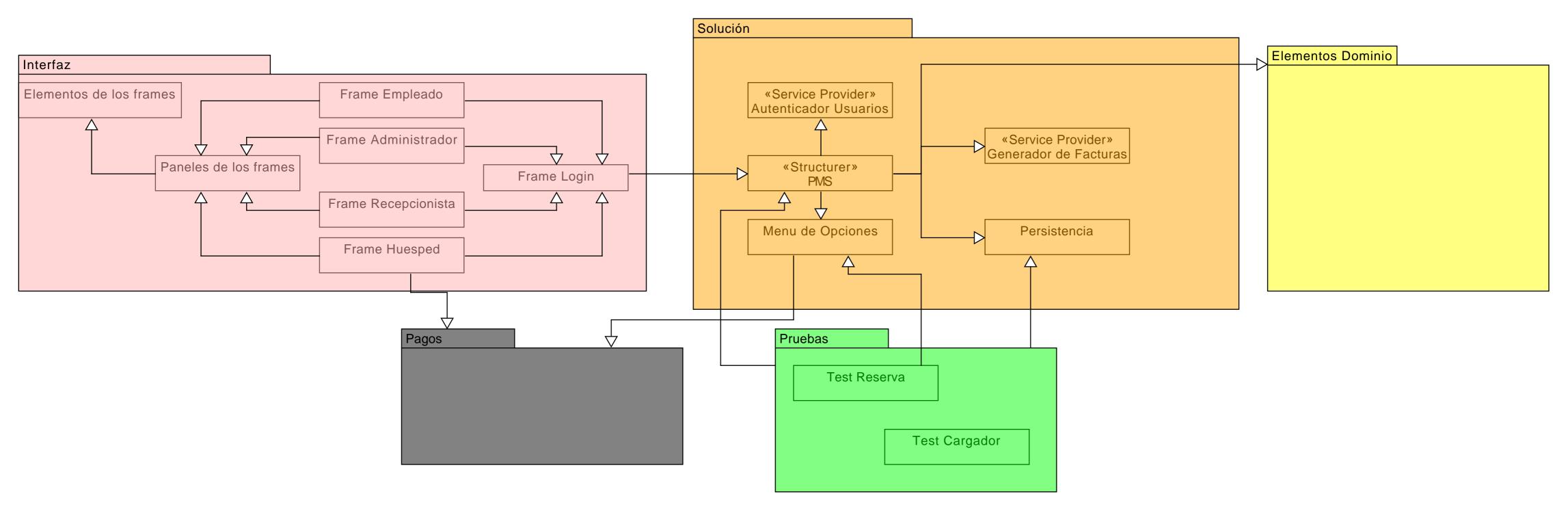
Estos requerimientos vienen acompañados a su vez de nuevas restricciones:

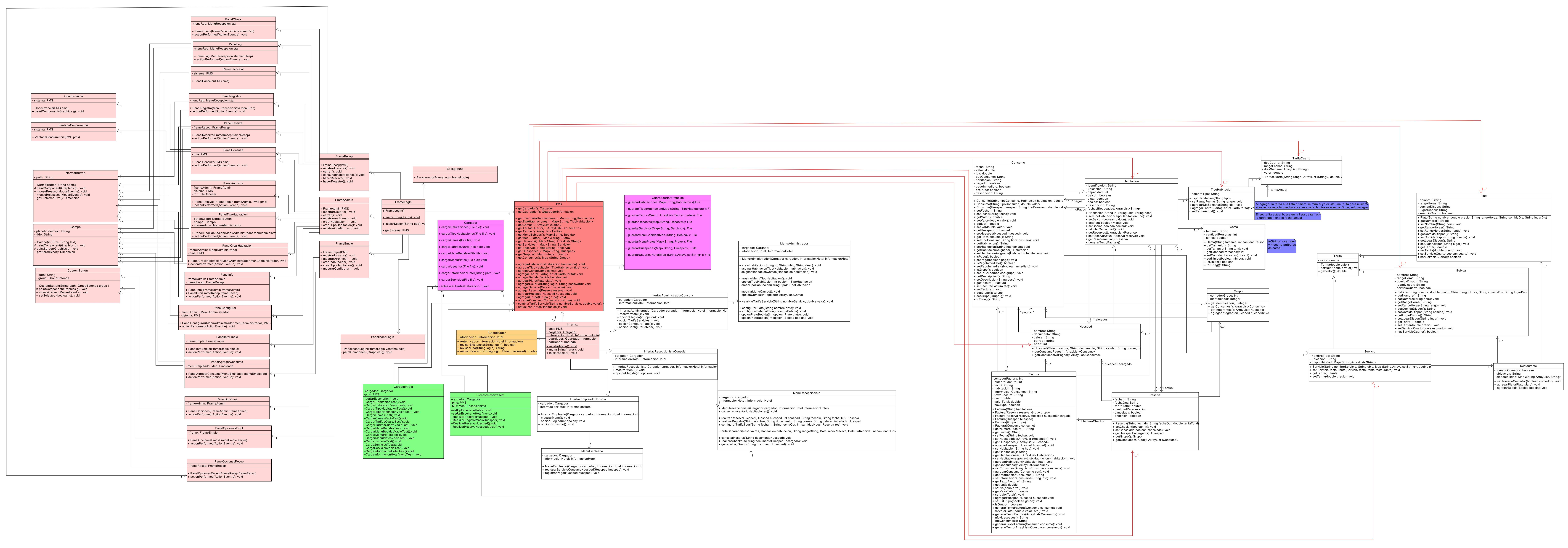
- Cuando se consulte la información de una habitación también deben aparecer las características del hotel.
- Al momento de hacer el checkout el recepcionista seleccionara una de las pasarelas de pago disponibles e ingresara la información de la tarjeta y de su dueño, seguido a esto hará el cobro de las noches y los servicios consumidos por el huésped.
- La aplicación debe soportar cambios en la implementación de las pasarelas. Para esto debe haber un archivo de texto con la configuración de las pasarelas previamente implementadas. Cada línea del archivo representa el nombre de una clase que implemente dicha pasarela de pago.
- Al hacer una reserva por medio de la interfaz de huéspedes se deben ingresar los datos de los huéspedes que se hospedan en la habitación.
- Si un huésped paga su reserva inmediatamente obtiene un descuento del 10% y al momento del checkout solo se le cobrarán los consumos que haya tenido.
- La interfaz para huésped funciona con los mismos datos del PMS.
- Los reportes al administrador incluyen gráficas generadas por el programa.
- las pruebas deben cubrir tanto los caminos correctos como los posibles caminos de excepción.
- Las pruebas no deben incluir la interfaz de la aplicación.

3. Elementos y sus estereotipos

En este documento no se van a detallar los elementos del dominio, la interfaz y la solución, puesto que estos fueron detallados en las anteriores entregas.

Pagos: Funciona como un Service Provider que se encarga de realizar los pagos por tarjeta en las diferentes pasarelas de pago. Cada pasarela de pago es representada como una clase que extiende a la clase abstracta PaymentMethod.





Frame Huésped: es un Interfacer encargado de satisfacer los requerimientos funcionales relacionados con el nuevo tipo de usuario, huésped. Estos requerimientos a su vez fueron implementados en el Menú Huéspedes.

Pruebas: Se encarga de hacer pruebas el cargador y al proceso de hacer reservas en diferentes escenarios.

4. Observaciones

Se pudo implementar las gráficas de forma que estas se puedan actualizar mientras se ejecuta la aplicación. Sin embargo, para que esto sea posible se tuvo que sobrescribir el método runnable.

Resultó problemático la implementación de los nuevos atributos de las habitaciones dado que implicaba cambiar la ya existente clase Habitación. Para solucionar este problema, se intentó usar un patrón Decorador aunque sin éxito, puesto que implementarlo implica que no se pueda ejecutar con atributos.

Debido a la gran cantidad de atributos que se debían era mucha información para una sola ventana, por lo cual cuando se iba a enviar la información se creó un JDialog para que la información que falta se ingresará en este y no fuera tan abrumante tanta información en una sola ventana.

No fue añadido al UML el panel y GUI para la aplicación de huésped. Sin embargo se intentó ilustrar su conexión y relación con el sistema en otros gráficos.

5. Comparaciones con las entregas anteriores

A la hora de realizar los cambios pedidos en esta entrega se pudo notar que agregar atributos fue una labor complicada. Es posible que de haber implementado un patrón de diseño diferente en la etapa anterior de desarrollo este problema no se hubiera presentado.

La organización que se le había dado a las anteriores entregas facilito en parte la implementación del nuevo sistema, puesto que este se pudo adaptar con relativa facilidad a las clases ya implementadas sin tener que crear clases totalmente nuevas, de manera que se reutilizó parte del código previamente elaborado.