

Documento de Diseño

Realizado por: Andrés Santiago Neira Socha, Omar Mauricio Urrego Vásquez y Juan Esteban Aguirre Olarte

1. Aspectos de Diseño

Para este apartado hay que entender el mundo del problema en el que se nos sitúa. Este consiste en construir un sistema de hoteles “PMS” el cual se encargará de administrar los elementos más importantes presentes en la operación de un hotel, como lo pueden ser: el manejo de inventario, servicios, reservas, registros y facturaciones. Para ello el sistema estará habilitado para 3 tipos de empleados; administrativos, recepcionistas (hereda características de “Empleados”) y empleados. Así mismo, cada uno de ellos cuenta con una serie de requerimientos sobre lo que debe hacer el sistema para que cada uno de ellos cumpla sus necesidades operacionales.

Por consiguiente, tras haber analizado el contexto del problema se identificaron los siguientes requerimientos funcionales.

- | | |
|--------------------------------------|--|
| • Registro de Administrador | • Conocer el inventario de habitaciones |
| • Registro de Empleado | • Actualizar el registro de habitaciones |
| • Registro de Recepcionista | • Realizar Check-in |
| • Establecer Inventario | • Abrir un registro de consumo |
| • Crear Tarifas asociadas | • Realizar Check-out |
| • Crear Tarifas de Servicios | |
| • Modificar las tarifas de servicios | |
| • Modificar el registro de consumo | |

* *Administrador – Recepcionista – Empleado* *

De manera análoga, existen requerimientos no funcionales como lo son la seguridad de la información, los métodos de pago y la verificación de número de personas informadas al momento de la reserva.

En consecuencia, se decidieron una serie de restricciones que el sistema iba a tener con relación al contexto y la identificación de requerimientos y estos fueron:

- | | |
|-----------------------------|---------------------------|
| • Restricciones de tiempo | • Limitaciones del diseño |
| • Restricciones de hardware | • Mantenibilidad |

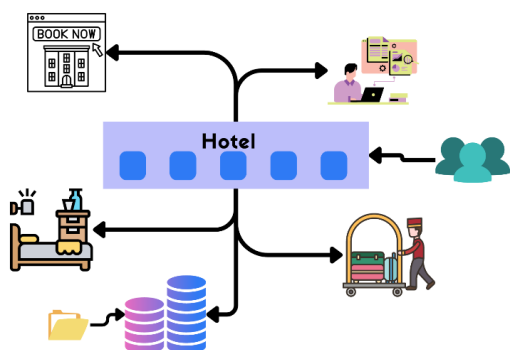
Ahora bien, a partir de esta identificación se decidió subdividir al “Sistema Hotel” en diversos componentes con el objetivo de reducir el acoplamiento entre funcionalidades. Por tanto, se pensó dividir este componente en 5 subcomponentes principales: Administrador de

Usuarios¹, Sistema de Inventarios², Sistema de Servicios³, Sistema de Consumo⁴ y Sistema de Reservas⁵.

Estas componentes están basadas en la frontera de únicamente realizar funciones fundamentales para un hotel, por lo que se descarta cualquier tipo de consideraciones subjetivas dentro de la solución. Es decir, no contemplar aspectos que puedan aumentar la complejidad del problema dado que no son del todo específicos o necesarios para todos los casos.

Por otro lado, para identificar los aspectos de diseño se usó una metodología reduccionista, en la que se trataba de descomponer los componentes utilizados para la solución, a priori, en subconjuntos relacionados lo menos posible con otros subcomponentes. Esto se debe a que el primer diagrama de dominio presentaba excesivas relaciones entre componentes, aspecto que puede llegar a ser conflictivo al momento de la implementación en Java. Como, por ejemplo, el descarte de la “clase” Catálogo_Servicios la cual no se relacionaba bien con el

requerimiento funcional de modificación de servicios; o, por ejemplo, la adición de un registro para usuarios.



En sintonía, para la descomposición iterativa de componentes se preveo las reglas de dominio identificadas en el análisis. Adicionalmente, teniendo en cuenta esto se creó un componente específico para el manejo de archivos de entrada y salida.

En resumen, para la operación del sistema se eligió un control delegado entre los 5 sistemas/Objetos-Componentes⁶ principales (asociado con su respectivo usurario del sistema) que participan en la solución final.

¹ Sistema encargado del registro de usuarios en el sistema PMS.

² Sistema encargado de la creación y modificación del inventario.

³ Sistema encargado de la creación, modificación e inicialización del catalogo de servicios asociados al hotel.

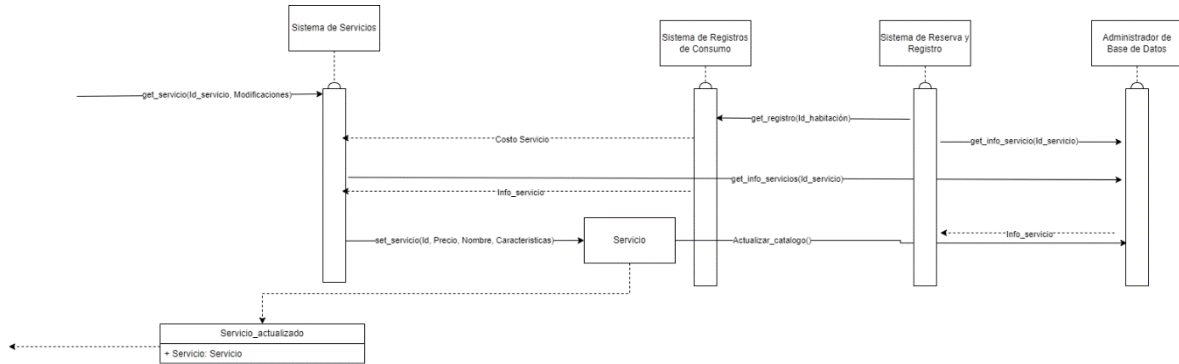
⁴ Sistema encargado de la creación y modificación de un registro de consumo asociado a una habitación por huésped.

⁵ Sistema encargado de los procesos asociados, a priori y posteriori, con el Check-in y Check-out.

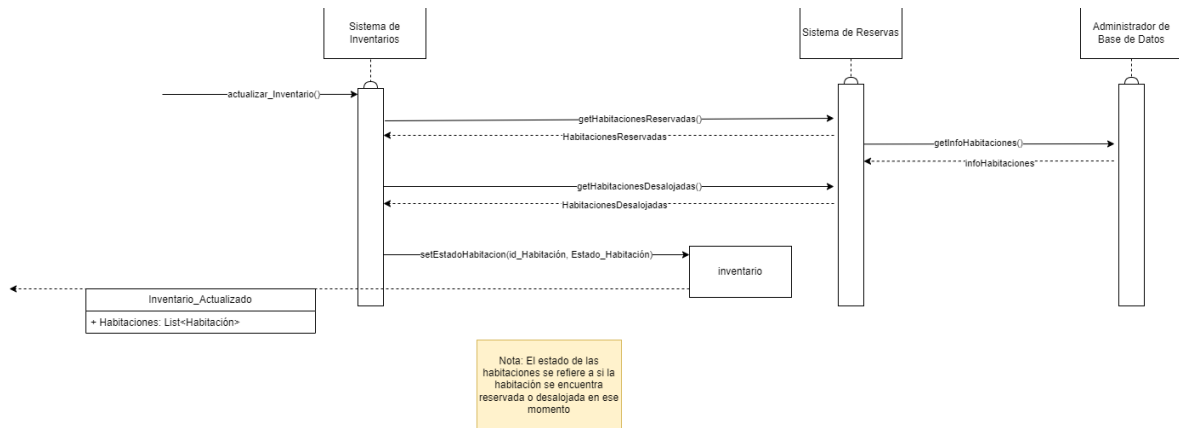
⁶ La componente administrativa tiene el estereotipo de “Coordinator” (Asignar tareas dentro del sistema). La componente de inventario tiene el estereotipo de “Information Holder” (Pues almacena la información de inventario). La componente de servicios tiene el estereotipo de “Information Holder e Interfacer” (Almacena y modifica información). La componente de Consumo tiene el estereotipo “Service Provider”. La componente de Reserva tiene el estereotipo de “Interfacer, Service Provider y Structurer” (Transforma el inventario, provee la reserva junto con su registro de consumo y mantiene una relación entre huésped-habitación-registro de consumo.)

3. Diagrama de Secuencia

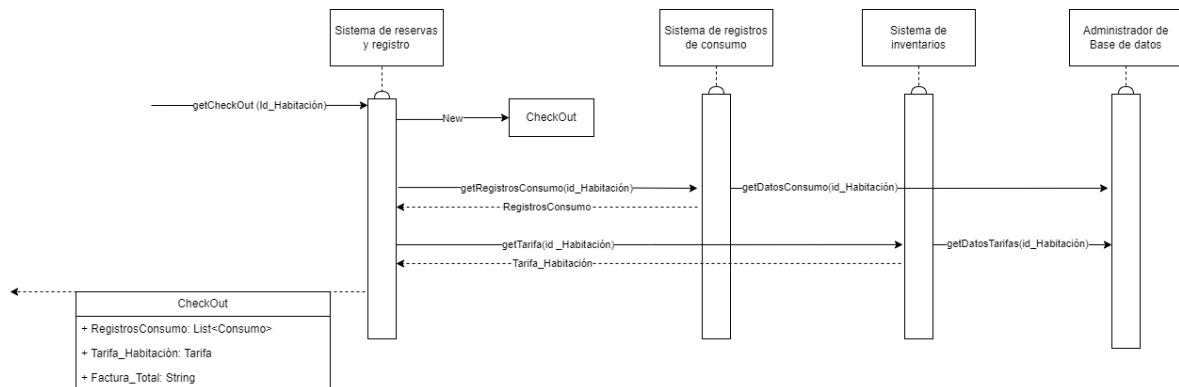
“Modificar Servicio”:



“Actualizar Inventario”:



“Check-out”:



4. Diseño de Persistencia del sistema

Llegados a este punto se decidió usar una persistencia de datos por serialización y archivos txt, esto se debe a que entendiendo la persistencia como una acción de preservación de toda información concerniente a un objeto ya sean características actuales, pasadas, eliminadas... etc. Es decir, la capacidad que tiene un objeto para seguir existiendo tras haber realizado ciertos cambios en el. Por lo cual, analizando el contexto del problema el uso de persistencia, para los archivos, resulta practico, pues un hotel es un sistema complejo en el que se presentan una serie de subsistemas cuyas relaciones no están definidas explícitamente; asimismo, este sistema se adapta y evoluciona en el tiempo a medida que es afectado por las partes y el contexto mismo.

Por consiguiente, para manejar y crear los archivos de inventario y servicios, se decidió usar serialización con archivos txt. Es decir, que los archivos de entrada (inventario y servicios) serán leídos por el sistema y codificados en objeto, luego serán guardados en una carpeta exclusiva para estos; así pues, la complejidad asociada a manejo y edición de archivos de texto será reducida, además se podrán detectar los cambios en el estado de objetos.

En sintonía, se realizará un proceso de deserialización, en los archivos de salida, para decodificar estos objetos en archivos txt (nótese que aunque se sigan usando archivos txt la complejidad para usar estos no es la misma), por si se quiere tener un registro “tangibile de los inventarios”; cabe resaltar que tanto los archivos de entrada como salida serán guardados en otra carpeta (Entrada / Salida) y serán nombrados con la fecha en que fueron procesados en el sistema.

Por su parte, para el manejo y creación de facturas asociadas a una habitación y huésped se decidió usar archivos txt. Por tanto, estos serán sobrescritos o escritos al momento de generar la factura en el Check-out. De la misma manera, estos serán guardados en una carpeta de facturas en la que cada factura estará denominada por el nombre y el rango de días de estancia del huésped.