Documento de Diseño – “Actualizado”

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

***Nota:*** *Cualquier cambio respecto al documento de diseño original será indicado con “Cambio” o con un inciso seguido de “New”.*

**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 usuarios: 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
* Registro de Empleado
* Registro de Recepcionista
* Establecer Inventario
* Crear Tarifas asociadas
* Crear Tarifas de Servicios[[1]](#footnote-2)
* Modificar las tarifas de servicios[[2]](#footnote-3)
* Modificar el registro de consumo
* Conocer el inventario de habitaciones
* Actualizar el registro de habitaciones
* Realizar Check-in
* Abrir un registro de consumo
* Realizar Check-out

*\* Administrador – Recepcionista – Empleado \**

De manera análoga, existen requerimientos no funcionales como lo son la *(“Cambio”)* seguridad de la información, disponibilidad del sistema, mantenimiento de datos y acceso a la información.

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
* Restricciones de hardware
* Restricciones lógicas sobre las operaciones del sistema[[3]](#footnote-4)
* Limitaciones del diseño
* 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 *(“Cambio”)* 7/8[[4]](#footnote-5) subcomponentes principales: Administrador de Usuarios[[5]](#footnote-6), Administrador BD[[6]](#footnote-7), Sistema de Inventarios[[7]](#footnote-8), Sistema de Servicios[[8]](#footnote-9), Sistema de Consumo[[9]](#footnote-10), Sistema de Reservas[[10]](#footnote-11), Hotel[[11]](#footnote-12) y/o Vista[[12]](#footnote-13).

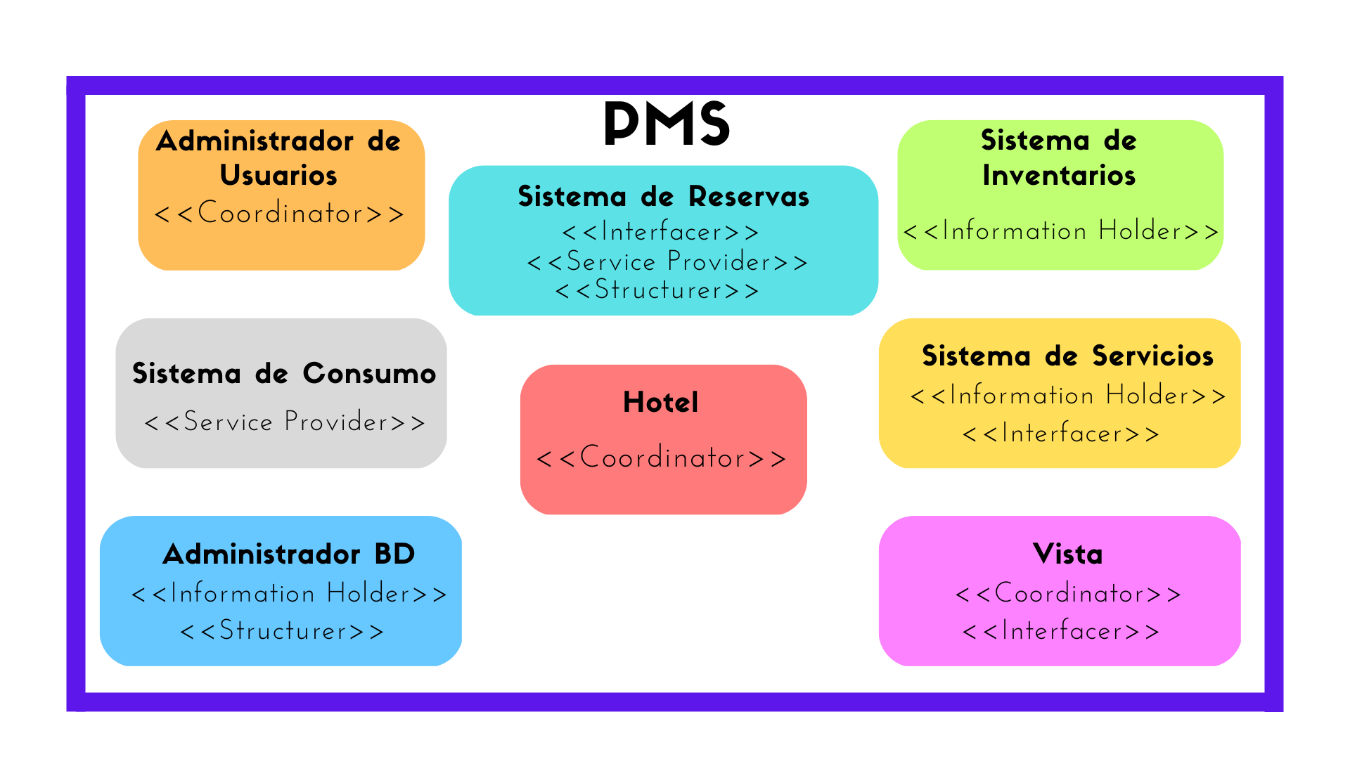
Estas componentes están basadas en la frontera de únicamente realizar funciones fundamentales para un hotel, por lo que se descarta cualquier tipo de consideraciones subjetivas dentro de la solución. Es decir, no contemplar aspectos que puedan aumentar la complejidad del problema dado que no son del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

Diagrama

Descripción generada automáticamentePor 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, entre muchas más.

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 7/8 sistemas/Objetos-Componentes principales (asociado con su respectivo usurario del sistema) que participan en la solución final.



*(“New”) Figura Ilustrativa “Estereotipos de las Componentes de Diseño”*

* *(“Cambio”)* La componente administrativa de usuarios tiene el estereotipo de “**Coordinator**”, ya que asigna tareas dentro del sistema para cada usuario registrado en el sistema.
* *(“Cambio”)* La componente de inventario tiene el estereotipo de “**Information Holder”,** pues esta almacena la información del inventario de habitaciones.
* *(“Cambio”)* La componente de servicios tiene el estereotipo de “**Information Holder e Interfacer**”, debido a que almacena y puede modificar información de ella misma para ser utilizada por otro componente.
* *(“Cambio”)* La componente de Consumo tiene el estereotipo “**Service Provider**”, porqué solo ofrecer una serie de opciones (“servicio complejo”) para un mismo tipo de servicio que es consumo.
* *(“Cambio”)* La componente de Reserva tiene el estereotipo de “**Interfacer, Service Provider y Structurer**”, esto se debe a que transforma el inventario (disponibilidad), provee la reserva junto con su registro de consumo y mantiene una relación entre huésped-habitación-registro de consumo.
* *(“New”)* La componente de Base de Datos tiene el estereotipo de “**Information Holder y Structurer**”, ya que mantiene la información del hotel, lo que implica mantener una relación entre cada parte del sistema; así pues, gracias a ello se puede conocer la información de estas relaciones.
* *(“New”)* La componente de Hotel tiene el estereotipo de “**Coordinator**”, puesto que se encarga de delegar tareas a cada componente.
* *(“New”)* La componente de Vista tiene el estereotipo de “**Coordinator y Interfacer**”, debido a que por un lado delega tareas únicamente a “Hotel”, y por el otro lado con esta información la transforma para poder entregarla al usuario.

**2. Diagramas de Clases**

Los siguientes diagramas en formato PNG y mayor resolución, al igual que las tablas de iteraciones (responsabilidades y componentes asociados) en un archivo de Excel, se agregaron al repositorio de *github*:

Diagrama

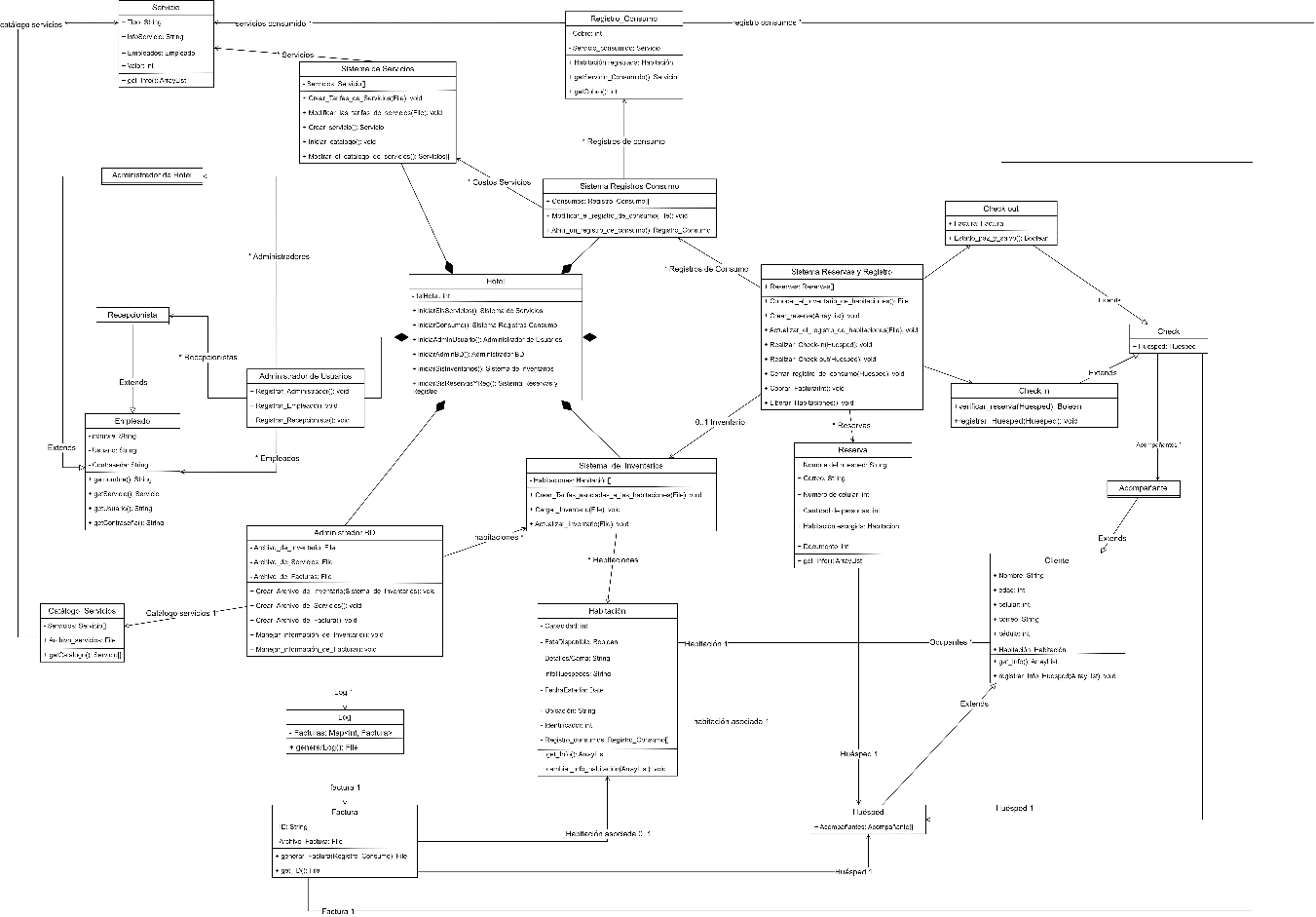
Descripción generada automáticamente

*Diagrama de Alto nivel iteración 2\**

Diagrama

Descripción generada automáticamente

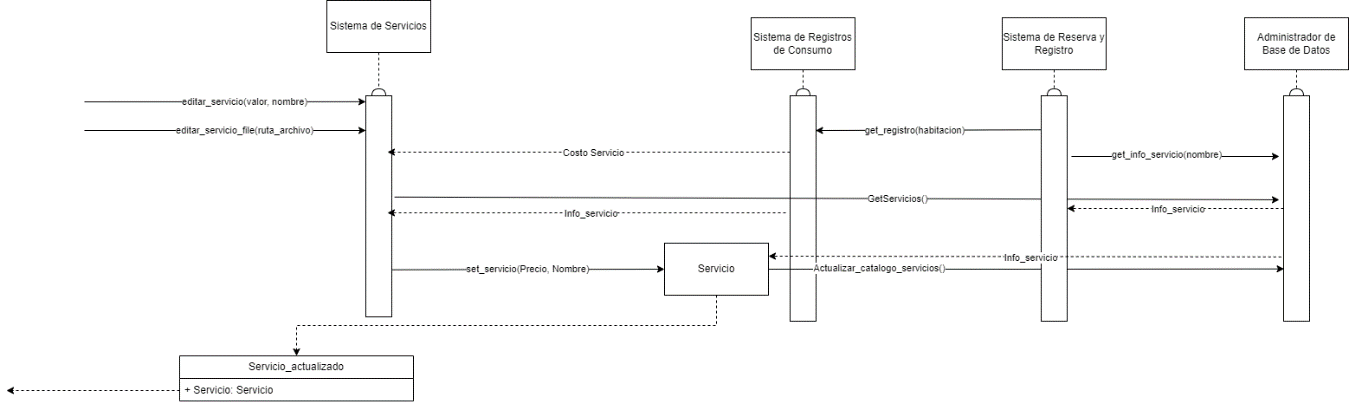
*Diagrama de Alto nivel iteración 3\**

**

*Diagrama detallado iteración 3\**

**3. Diagrama de Secuencia**

*(“Cambio”)* “Modificar Servicio”:



“Actualizar Inventario”:

Imagen que contiene Tabla

Descripción generada automáticamente

“Check-out”:

Diagrama

Descripción generada automáticamente

**4. Diseño de Persistencia del sistema**

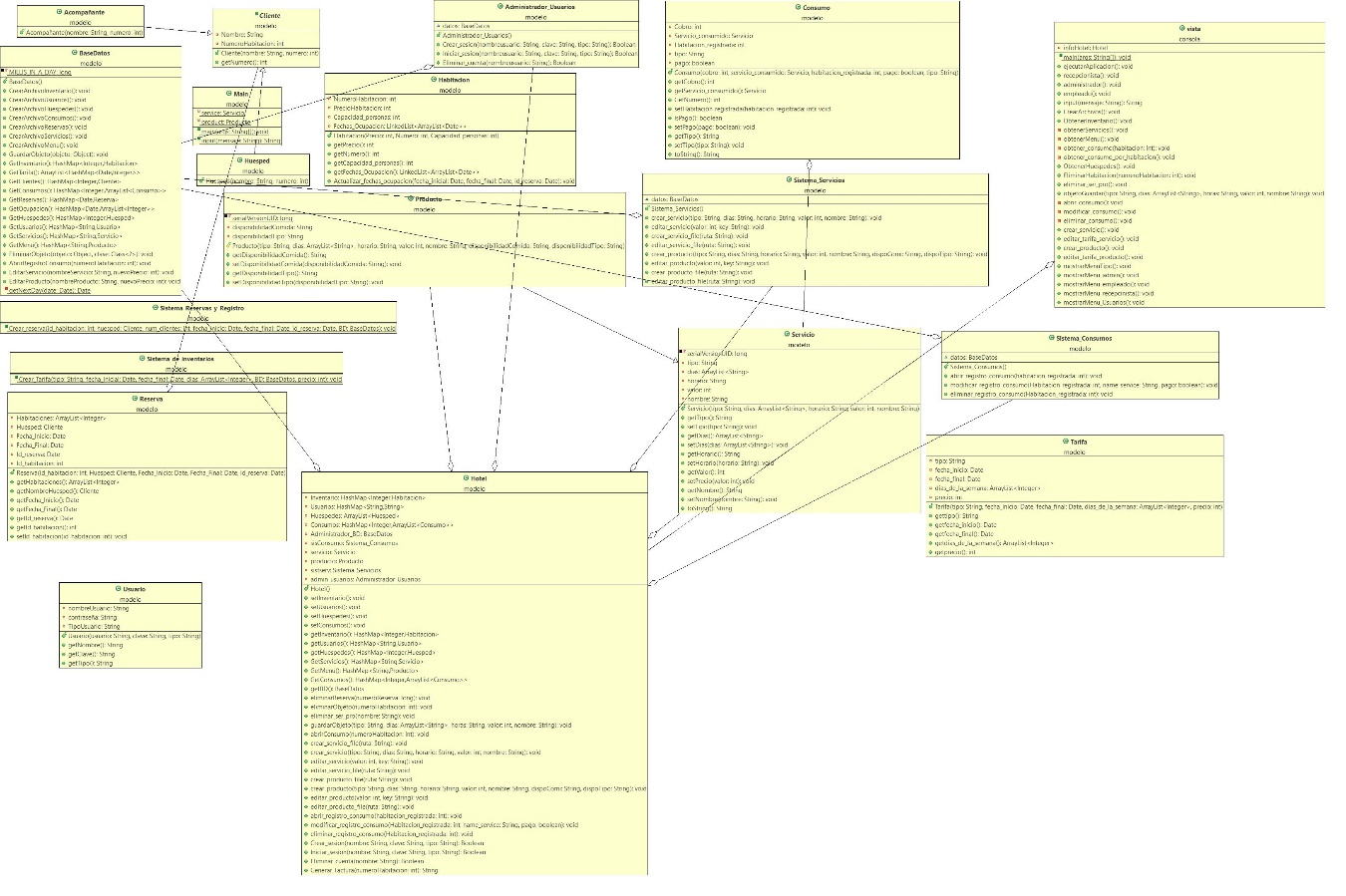
Llegados a este punto se decidió usar una persistencia de datos por serialización y archivos txt, esto se debe a que entendiendo la persistencia como una acción de preservación de toda información concerniente a un objeto ya sean características acutales, pasadas, eliminadas… etc. Es decir, la capacidad 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 descodificar 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 “tangible de los inventarios”; cabe resaltar que tanto los archivos de entrada como salida serán siempre los mismo, solo que la diferencia entre versiones depende de las modificaciones hechas por el usuario.

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.

**5. Diagrama De Clases – Versión Final-**



**6. Opciones del sistema**

Antes de comenzar, cabe aclarar que no se tiene en cuenta las fechas para el consumo de los servicios pues puede ser contraproducente no consumir un servicio por el simple hecho de que la hora actual del sistema no coincida con la hora del servicio.

El sistema PMS consta de diversas opciones para cada tipo de usuario soportado; Administrador, Recepcionista y Empleado; a fin de ello, cuando se ejecute la aplicación esta preguntará el tipo de usuario, para después preguntarle su usuario y contraseña (ver figura 1).

Texto

Descripción generada automáticamente

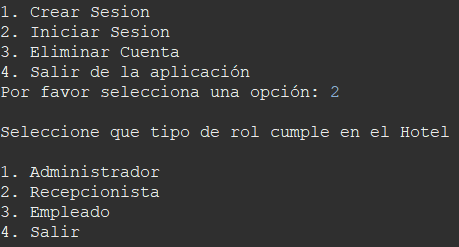
*\*Figura 1 “Opciones iniciales del sistema” \**

*Texto

Descripción generada automáticamente*

*\*Figura 1.1 “Creación de una Sesión” \**

A partir de este punto cada usuario tendrá sus respectivas opciones dependiendo del tipo de usuario elegido (ver figura 3).



*\*Figura 2 “Tipos de usuarios en el sistema” \**

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteTexto

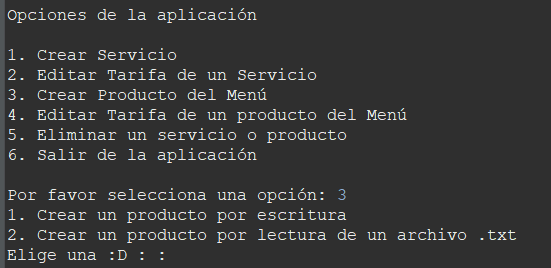
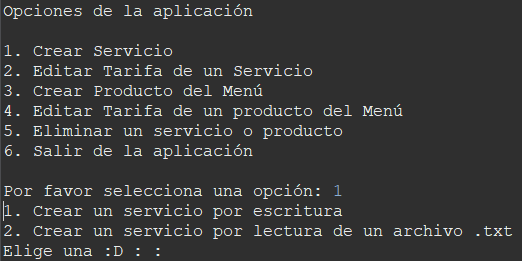
Descripción generada automáticamente

*\*Figura 3 “Opciones del sistema para cada tipo de usuario” \**

Llegados a este punto se hará una descripción y distinción de ciertas opciones especiales por usuario:

Administrador:

Las opciones del administrador tanto para servicios como para productos son similares (más allá de que producto extienda servicio) solo cambia en que producto requiere 2 parámetros extra si se quiere crear un producto ingresando valores por máquina (ver figura 5).



*\*Figura 4 “Opciones para la creación de servicios y productos dentro del sistema” \**



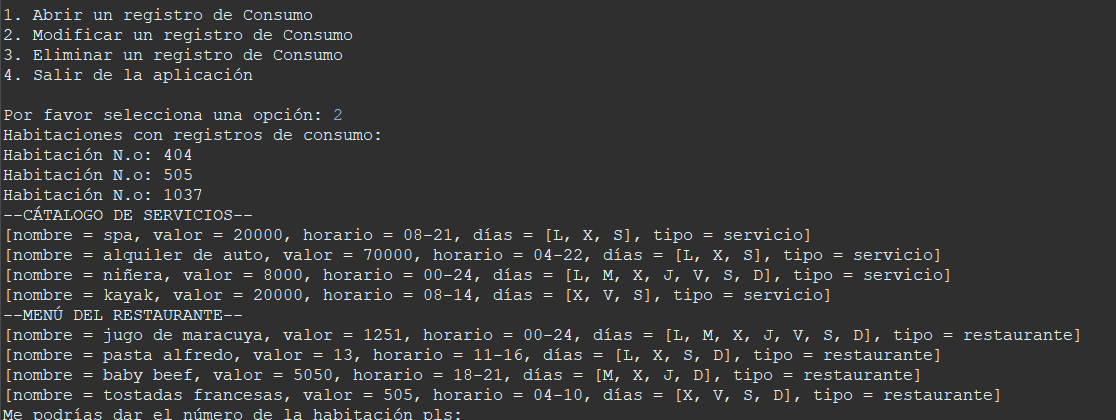


*\*Figura 5 “Diferencia entre crear un servicio y crear un producto” \**

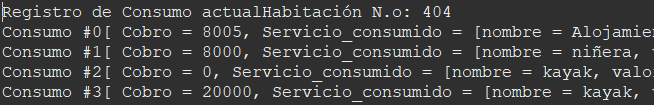
Recepcionista:

La opción de abrir un registro de consumo requiere el numero de la habitación, cabe aclarar que el primer consumo va a ser la reserva de alojamiento.

La opción de modificar un consumo (también aplicable para el usuario “Empleado”) va a requerir el nombre de la habitación, el nombre del servicio/producto y un valor booleano (true/false) que indiqué si está pago o no. Para este apartado se mostrará las habitaciones con consumos abiertos, así como el catálogo de servicios y menú de productos (ver imagen 6).



*\*Figura 6 “Vista de la opción modificar un registro de consumo” \**

**

*\*Figura 7 “Registro de consumo por habitación (Cobro, Servicio, Pagado)” \**

**7. Instrucciones de uso**

A continuación, encontrarás una serie de opciones a ejecutar en cierto orden para que entiendas el funcionamiento del sistema, tras haber visto esto junto con las opciones del sistema podrás jugar libremente con el sistema PMS.

1. Ejecuta la aplicación e inmediatamente crea un usuario y contraseña para cada tipo de usuario (ver figura 1 y figura 1.1).
2. Tras haber realizado esto, elige 1 (la opción de administrador)
3. Seguidamente, deberás crear y/o cargar los servicios y productos del menú (ver figura 3) para ello elige la opción 1 (Crear Servicio) y luego la 3 (Crear Producto del Menú) al ejecutar cada una verás la posibilidad de crear el servicio/producto a través de 2 formas (ver figura 4).
4. Llegados a este punto elige la que más te guste, pero te recomendamos la opción 2, pues es más práctica, seguidamente se te preguntará el nombre del archivo. Para ello solo digita el texto que está en “()” al final del input. Y según si hay datos o no, en el sistema, se te avisará si se creo el servicio/producto o si esté ya existe.
   1. Para editar tarifas ocurre algo similar, así que siéntete libre de elegir estas opciones si gustas.
5. Te recomendamos que elimines un servicio/producto y luego cargues el respectivo archivo nuevamente, para cerciorarte el que sistema cumple este requerimiento.
6. Ahora elige la opción 6 (Salir de la aplicación) para volver al menú principal.
7. Seguidamente, elige la opción 2 (Iniciar Sesión) para después elegir el usuario 2 (Recepcionista).
8. Ahora abre un registro de consumo, eligiendo la opción 1 e indicando el número de la habitación.
9. Luego elige la opción 2 (modificar un registro de consumo) y empieza a jugar.
   1. Seguidamente te recomendamos que cambies de usuario al administrador, cambies una tarifa, vuelvas al recepcionista o vayas al empleado (inicies sesión) y modifiques el consumo de una habitación y rectifiques que se cambió el precio en el consumo teniendo en cuenta la estructura del consumo presentada en la figura 7.
10. Ahora deja volar tu imaginación y recuerda hacer check-out.

1. *(“New”)* Este requerimiento se puede solventar vía archivo “.txt” o simplemente escribiendo las características del servicio nuevo. [↑](#footnote-ref-2)
2. *(“New”)* Este requerimiento se puede solventar vía archivo “.txt” o simplemente escribiendo las características del servicio nuevo. [↑](#footnote-ref-3)
3. *(“New”)* Uso erróneo del sistema por equivocación o decisión, causadas por el uso de operaciones que el sistema no soporta en su implementación. [↑](#footnote-ref-4)
4. *(“New”)* La vista no está involucrada en el diseño del sistema, pero es fundamental en la implementación y el funcionamiento del PMS. [↑](#footnote-ref-5)
5. Sistema encargado del registro de usuarios en el sistema PMS. [↑](#footnote-ref-6)
6. *(“New”)* Sistema encargado del manejo de archivos para el funcionamiento del PMS. [↑](#footnote-ref-7)
7. Sistema encargado de la creación y modificación del inventario. [↑](#footnote-ref-8)
8. Sistema encargado de la creación, modificación e inicialización del catálogo de servicios asociados al hotel. [↑](#footnote-ref-9)
9. Sistema encargado de la creación y modificación de un registro de consumo asociado a una habitación por huésped. [↑](#footnote-ref-10)
10. Sistema encargado de los procesos asociados, a priori y posteriori, con el Check-in y Check-out. [↑](#footnote-ref-11)
11. *(“New”)* Sistema encargado de “almacenar” todas las opciones para todos los usuarios. [↑](#footnote-ref-12)
12. *(“New”)* “Sistema/Interfaz” encargada de interactuar con el usuario. [↑](#footnote-ref-13)