**DOCUMENTO DE IMPLEMENTACIÓN**

**CONTEXTO:**

Diagrama

Descripción generada automáticamenteEl programa PMS está orientado para el uso de los empleados, por lo que hay distintos tipos de empleados que pueden usar el programa con funciones diferentes. El usuario se relaciona directamente solo con la aplicación. Esta aplicación es la que se conecta los controladores del programa, y estos controladores buscan la información en la base de datos, que en este caso está en archivos de texto. Después de obtener la información, se la dan a la aplicación y este se la da al usuario.

**REQUERIMIENTOS FUNCIONALES:**

Como se mencionó anteriormente, cada tipo de empleado cuenta con funciones diferentes que salen de sus labores. Por eso encontramos los siguientes requerimientos.

* **Requerimientos funcionales administrador:**
* Cargar un archivo .csv con la información de las habitaciones en el inventario.
* Crear una habitación en el inventario.
* Cargar tarifa de habitación por tipo de cuarto, en un rango de fechas y en ciertos días de la semana.
* Establecer o cambiar la tarifa de un servicio del hotel.
* Cargar menús del restaurante.
* Crear un producto de restaurante.
* **Requerimientos funcionales recepcionista:**
* Crear reserva a nombre de uno o varios huéspedes.
* Generar factura para una reserva.
* Consultar inventario de habitaciones.
* **Requerimientos funcionales empleado:**
* Registrar el consumo de un servicio a nombre de uno o más huéspedes.
* **Requerimientos funcionales generales:**
* Cargar los datos de los usuarios.
* Generar archivo de log con el historial de reservas de un grupo de huéspedes.
* Avisar si no hay una tarifa definida en los próximos 365 días.

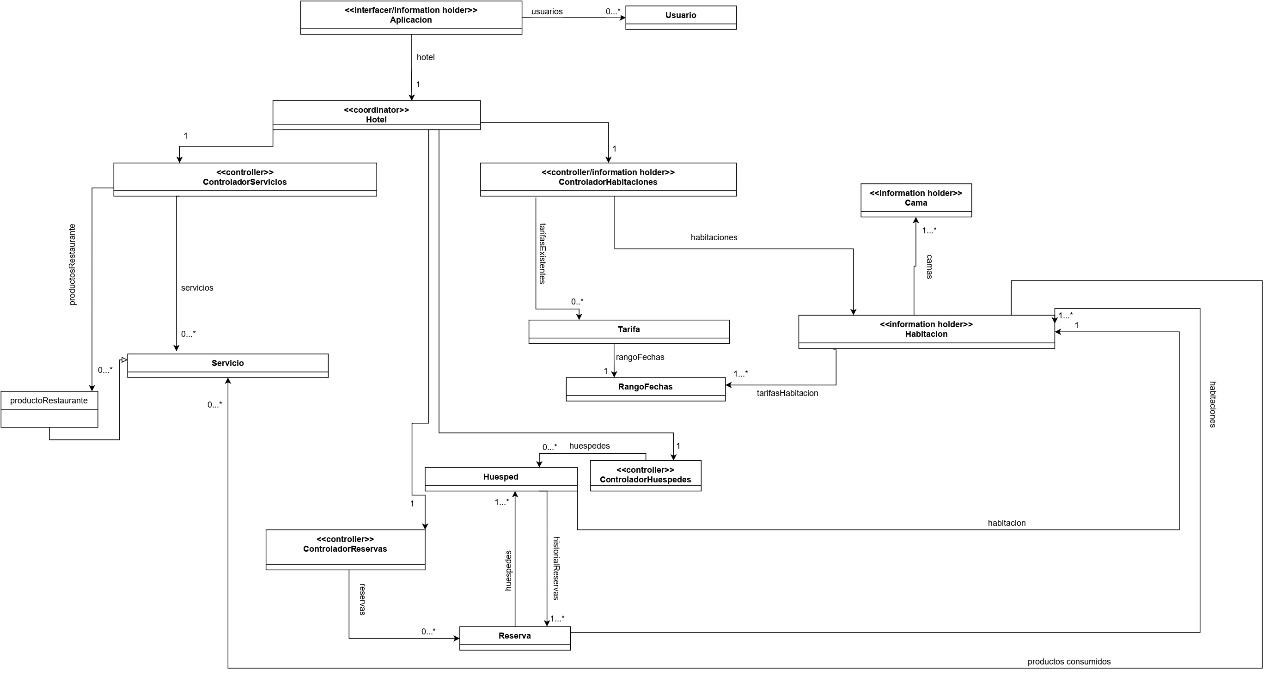
**RESPONSABILIDADES:**

A su vez, cada requerimiento tiene lo cumple un controlador específico. En la siguiente tabla se pueden ver las responsabilidades asignadas a cada controlador.

|  |  |  |
| --- | --- | --- |
| # | Responsabilidades | Componente |
| 1 | Avisar si no hay una tarifa definida en los próximos 365 días. | Controlador Habitaciones |
| 4 | Cargar tarifa de habitación por tipo de cuarto, en un rango de fechas y en ciertos días de la semana. |
| 5 | Cargar un archivo .csv con la información de las habitaciones en el inventario. |
| 6 | Consultar inventario de habitaciones. |
| 9 | Crear una habitación en el inventario. |
| 11 | Generar archivo de log con el historial de reservas de un grupo de huéspedes. | Controlador Huéspedes |
| 2 | Cargar los datos de los usuarios. | Controlador Reservas |
| 7 | Crear reserva a nombre de uno o varios huéspedes. |
| 12 | Generar factura para una reserva. |
| 3 | Cargar menús del restaurante. | Controlador Servicios |
| 8 | Crear un producto de restaurante. |
| 10 | Establecer o cambiar la tarifa de un servicio del hotel. |

**DIAGRAMAS DE CLASE:**

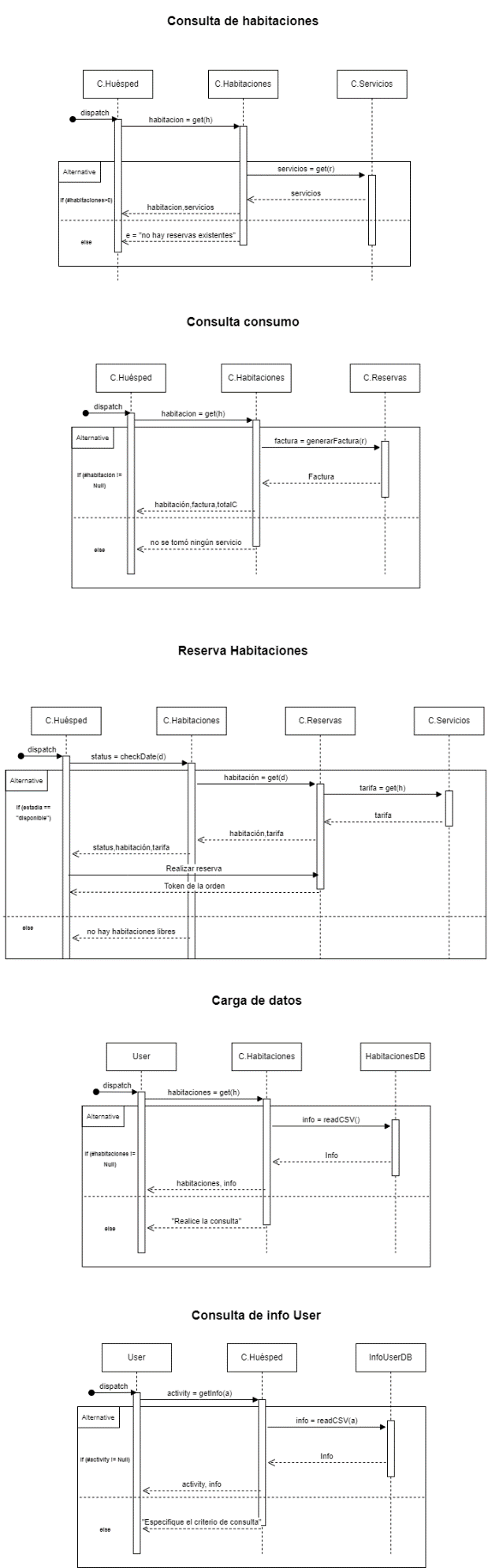
En el diagrama de clases se puede ver todo lo mencionado anteriormente, como la relación del usuario con el programa es exclusivamente a través de la aplicación, mientras que esta se conecta con el corazón del programa para obtener la información solicitada.

Además del UML detallado con todos los atributos y métodos, también está el diagrama de alto nivel.

Diagrama

Descripción generada automáticamente

**SECUENCIAS:**

****

**FUNCIONAMIENTOS:**

* Archivos .txt: En el programa se crearon varios archivos .txt para guardar la información ordenada de cada una de las clases. Para mantener la información, cada vez que se añade un nuevo objeto de una clase, se crea una nueva línea, en la que cada atributo se separa con un “;”.
* Consola: la aplicación del programa (consola), primeramente, pide a un iniciar sesión, que funciona como una aplicación convencional (usuario y contraseña); dependiendo del rol que cumple el empleado, a este se le presentaran varias funciones que puede cumplir el programa.
* Persistencia: para la persistencia, se usa la biblioteca FileWriter, que como su nombre indica, permite escribir sobre los archivos de texto anteriormente mencionados, para mantener almacenada la información.g