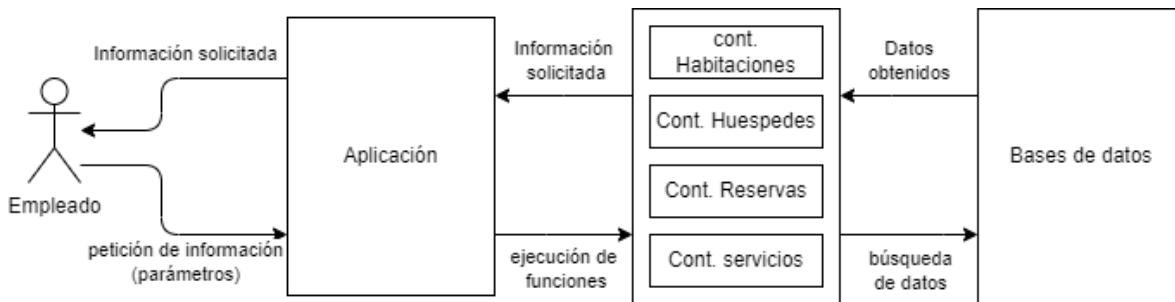


DOCUMENTO DE IMPLEMENTACIÓN

CONTEXTO:

El 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:

Requerimientos funcionales administrador:

- Cargar un archivo 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.
- Cancelar una reserva.

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:

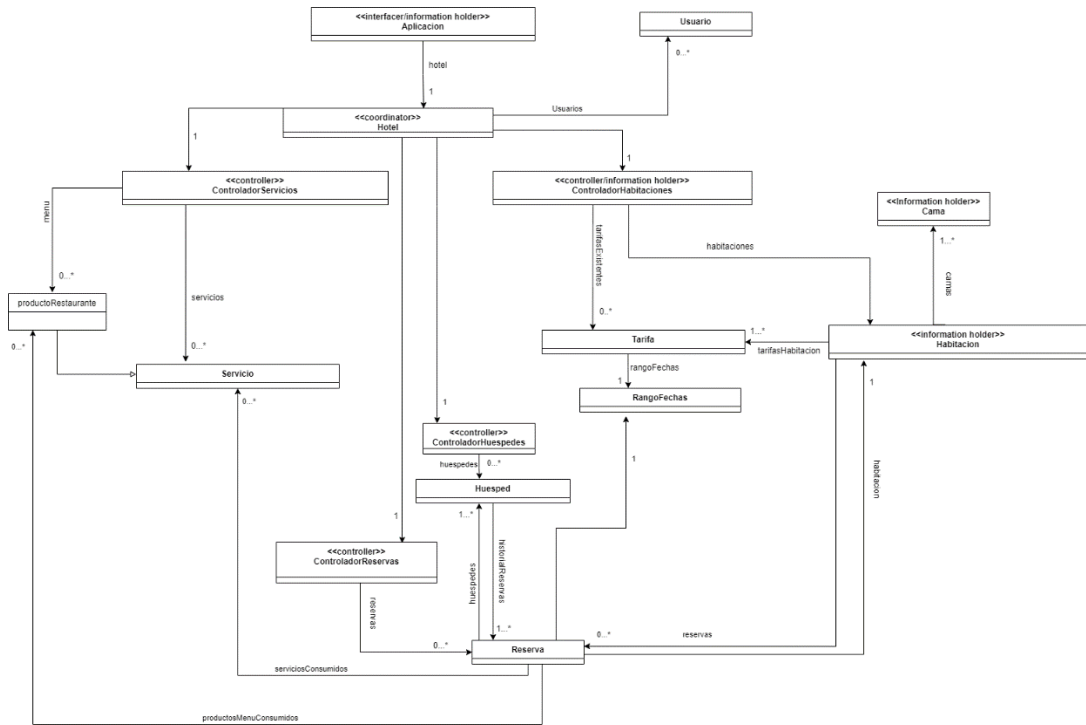
A su vez, cada requerimiento tiene lo cumple un controlador específico. En la siguiente tabla se pueden ver las responsabilidades asignadas a cada controlador.

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

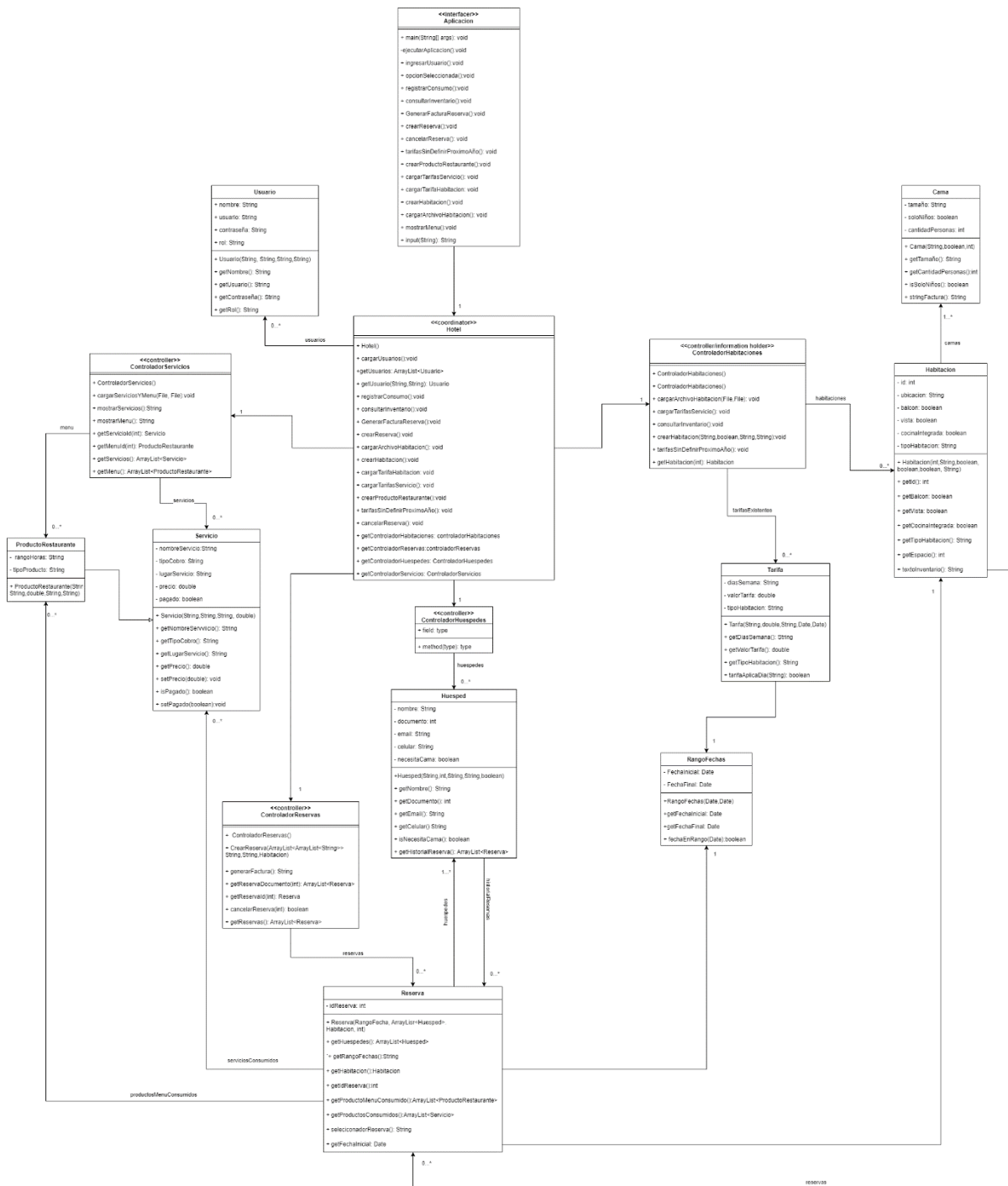
DIAGRAMAS DE CLASE:

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.

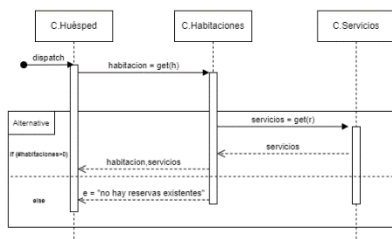
Además del UML detallado con todos los atributos y métodos, también está el diagrama de alto nivel.



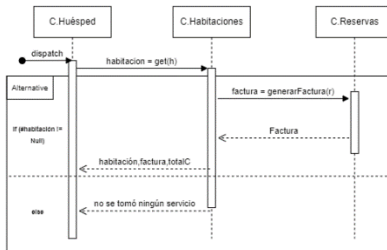
SECUENCIAS:



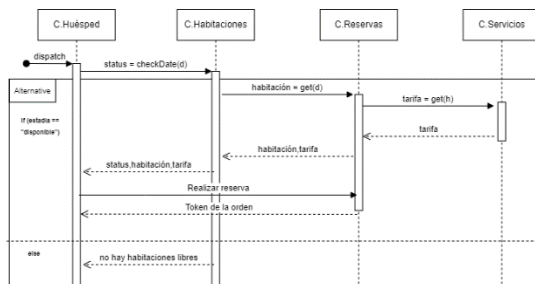
Consulta de habitaciones



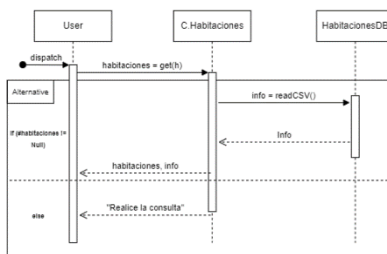
Consulta consumo



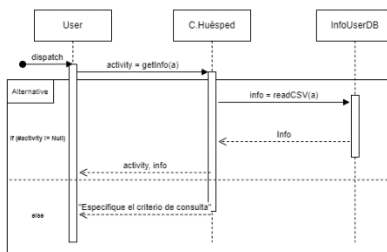
Reserva Habitaciones



Carga de datos



Consulta de info User



Cambios con respecto a la propuesta de diseño:

UML propuesta de diseño:

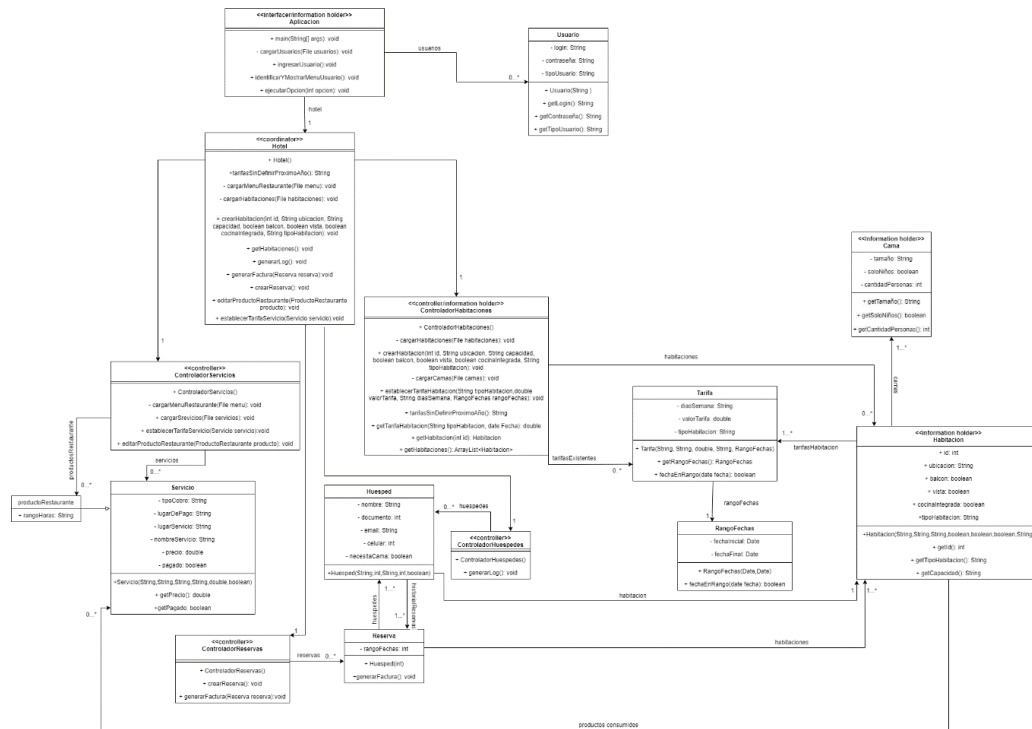
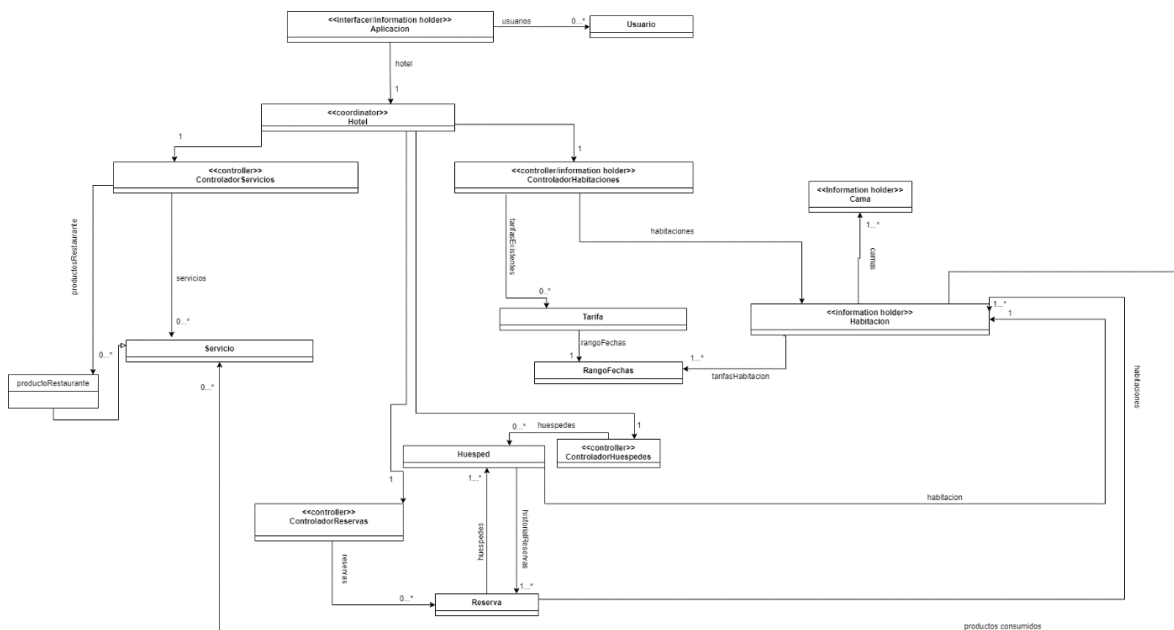


Diagrama de alto nivel propuesta de diseño:



No hubo muchos cambios respecto a la propuesta de diseño que se hizo originalmente, la mayoría de estos fueron los métodos de algunas funciones, que no se habían contemplado

en el diseño, pero esto es debido a que son métodos auxiliares que simplifican el código y son usados por los métodos principales.

El cambio más grande con respecto a la relación entre clases es el uso de una clase RangoFechas dentro de la clase Reserva, esto se hizo con el fin de poder acceder de mejor forma las fechas tanto finales como iniciales de una reserva.

Otro cambio con respecto a la relación entre clases es que ahora los huéspedes de una reserva solo pueden alojarse en una misma habitación, por lo que pasa de tener una relación de uno o varios a solo una. Por lo tanto, el huésped tampoco necesita tener una relación directa con su habitación. Esta decisión se tomó con el fin de facilitar el cobro de servicios a una habitación, ya que se puede gestionar directamente con su reserva. Esto también significa que la lista de servicios consumidos ahora pertenece a la clase Reserva y no a la clase Habitación.

También ahora es el hotel el que coordina la carga de los usuarios, por lo que es la clase Hotel que contiene una lista de usuario, y no la aplicación, como se había propuesto antes. Esto se hizo con el fin de poder mantener la mayor independencia posible del cumplimiento de los requerimientos de la clase Aplicación, apuntando a las entregas futuras que ya no usarán esta clase.

Descripción de los archivos de datos:

Habitaciones:

```
id;ubicacion;balcon;vista;cocinaIntegrada;tipoHabitacion
1;102;true;false;true;estandar
2;103;false;true;false;suite
3;104;true;1;true;suite doble
4;105;false;0;true;estandar
5;106;true;1;false;suite
6;107;false;true;1;suite doble
7;108;true;false;0;estandar
8;109;false;0;true;suite
9;110;true;1;true;suite doble
10;111;false;true;false;estandar
11;112;true;false;true;suite
12;113;false;0;false;estandar
13;114;true;1;true;suite doble
14;115;false;true;false;suite
15;116;true;false;true;estandar
16;117;false;0;false;suite
17;118;true;1;true;suite doble
18;119;false;true;false;estandar
19;120;true;false;true;suite
20;121;false;0;false;estandar
```

Como se puede apreciar en la figura 1, el archivo con la información de las habitaciones contiene seis columnas, las cuales representan lo siguiente:

Id: El identificador único que cada habitación.

Ubicación: El número de la habitación respecto a la distribución del hotel.

Balcón: Un booleano que determina si la habitación tiene balcón o no.

Vista: Un booleano que determina si la habitación tiene vista hacia el exterior o no.

Cocina Integrada: Un booleano que determina si la habitación posee una cocina integrada o no.

Tipo Habitación: Determina el tipo de habitación, los tipos posibles son “estándar”, “suite” y “suite doble”.

Cama:

```
id;tamaño;cantidadPersonas;soloNiños
4;doble;2;false
10;king;2;false
2;individual;1;false
16;doble;3;false
8;king;2;false
5;doble;2;false
3;individual;1;false
18;king;3;false
15;doble;3;false
11;queen;2;false
13;queen;2;false
19;king;3;false
20;king;3;false
12;queen;2;false
1;individual;1;true
17;doble;3;false
9;king;2;false
7;doble;2;false
6;doble;2;false
14;queen;2;false
1;queen;2;false
5;queen;2;false
```

Como se puede apreciar, el archivo tiene cuatro columnas que representan lo siguiente:

Id: El identificador único de la habitación a la que pertenecen.

Tamaño: Representa el tamaño de la cama, los tamaños posibles son “individual”, “doble”, “queen” y “King”.

Cantidad Personas: Representa el número de personas máximo que la cama puede alojar.

Solo Niños: Booleano que determina si una cama es solo apta para niños o no, las únicas camas que pueden ser aptas para niños son las individuales.

Servicios:

```
nombreServicio;tipoCobro;lugarServicio;precio
Tour por la ciudad;persona;Hotel;15000
Masaje relajante;persona;Spa;50000
Traslado al aeropuerto;grupo;Hotel;100000
Acceso al gimnasio;persona;Gimnasio;10000
Alquiler de bicicletas;grupo;Hotel;25000
Clases de yoga;persona;Gimnasio;15000
Tratamiento facial;persona;Spa;70000
Sesión de sauna;persona;Spa;20000
Sesión de jacuzzi;grupo;Hotel;50000
Paseo en bote;grupo;Playa;75000
Tour de compras;persona;Hotel;20000
Clases de natación;persona;Piscina;25000
Excursión en kayak;grupo;Playa;60000
Sesión de manicura/pedicura;persona;Spa;40000
Sesión de maquillaje;persona;Spa;60000
```

Como se puede apreciar, el archivo de los servicios tiene 4 columnas, las cuales representan lo siguiente:

Nombre Servicio: Representa el nombre del servicio que presta el hotel.

Tipo cobro: Representa si el cobro es por persona o por grupo de personas.

Lugar Servicio: Representa el lugar en el que se presta el servicio, que puede o no ser dentro de las instalaciones del hotel.

Precio: Representa el valor en pesos que tiene el servicio, en caso de ser grupal el precio representa el valor que debe de pagar cada persona que usó el servicio.

Productos del Restaurante:

```
nombreServicio;tipoCobro;lugarServicio;tipoProducto;rangoHoras;precio
Desayuno buffet;persona;Restaurante;comida;6:30-10:00;45000
Almuerzo buffet;persona;Restaurante;comida;12:00-15:00;50000
Cena buffet;persona;Restaurante;comida;19:00-22:00;55000
Café americano;persona;Restaurante;bebida;6:00-23:00;5000
Café con leche;persona;Restaurante;bebida;6:00-23:00;7000
Refresco en lata;persona;Restaurante;bebida;6:00-23:00;6000
Jugo natural;persona;Restaurante;bebida;6:00-23:00;10000
Hamburguesa con papas;persona;Restaurante;comida;11:00-22:00;20000
Pizza mediana;persona;Restaurante;comida;11:00-22:00;25000
Sándwich mixto;persona;Restaurante;comida;11:00-22:00;15000
Cerveza nacional;persona;Restaurante;bebida;18:00-23:00;8000
Cerveza importada;persona;Restaurante;bebida;18:00-23:00;12000
Coctel del día;persona;Restaurante;bebida;18:00-23:00;15000
Botella de vino;persona;Restaurante;bebida;18:00-23:00;70000
```

Como se puede apreciar, el archivo con el menú del restaurante tiene seis columnas, las cuales representan:

Nombre Servicio: El nombre del plato o bebida ofrecido por el hotel.

Tipo Cobro: Tipo de cobro, puede ser por persona o grupal.

Lugar servicio: Lugar en el que se presta el servicio.

Tipo de Producto: El tipo de producto, puede ser “comida” o “bebida”.

Rango Horas: El rango de horas del día en el que está disponible el producto del restaurante, está en formato “HH:mm-HH:mm”.

Precio: Precio en pesos colombianos del producto del restaurante.

Usuarios:

Este archivo se carga automáticamente al ejecutar la aplicación.

```

nombre;usuario;contraseña;rol
Mario Alexander;m.a13;1234;Administrador
Ana Martinez;ana_martinez;123456;Recepcionista
Luis García;luis_garcia;abc123;Empleado
Carla Jiménez;carla_jimenez;qwert;Recepcionista
Pedro Sánchez;pedro_sanchez;contraseña;Empleado
María Rodríguez;maría_rodriguez;securepwd;Recepcionista
Javier López;javier_lopez;password;Empleado
Lucía Pérez;lucia_perez;letmein;Recepcionista
Diego González;diego_gonzalez;secret;Empleado
Sofía García;sofia_garcia;iloveyou;Recepcionista
Andrés Torres;andres_torres;admin;Empleado

```

Como se puede apreciar, el archivo tiene cuatro columnas que representan lo siguiente:

Nombre: El nombre real del empleado del hotel.

Usuario: Un username con el que el empleado inicia sesión en el programa.

Contraseña: La contraseña asociada al usuario.

Rol: El rol del usuario, esto determina las opciones disponibles dentro del programa. Los roles posibles son “administrador”, “recepcionista” y “empleado”.

Huéspedes:

```

nombre;documento;email;celular;necesitaCama
Nicolas;1092850219;n.ruizp2;318849;true
Harold;1234;h.coca;312;true
nicolas;1098;n.ruizp2;315;true
hicolas;109;n.ruizp2;23;true
harol;12;ao;o;true

```

Como se puede apreciar, la información de los huéspedes se guarda en el archivo Huespedes.txt, la información de los huéspedes se escribe cuando se crea una nueva reserva, y los huéspedes que no hayan existido antes se guardarán en este documento. Las columnas son las siguientes:

Nombre: Nombre del huésped.

Documento: Documento del huésped.

Email: Correo electrónico del huésped.

Celular: Numero de celular del huésped.

Necesita cama: Booleano que especifica si el huésped requiere de una cama.

Reservas:

```
id;idHabitacion;fechaInicial;fechaFinal;Huespedes;cancelado;servicios;menu
0;10;2023-04-23;2023-04-24;harold:1092:a:a:true,false;-11:true;-9:false
```

Como se puede apreciar, la información de las reservas creadas se guarda en el archivo “Reservas.txt”. Las columnas son las siguientes:

Id: Identificador de la reserva.

Id Habitación: identificador de la habitación de la reserva.

Fecha inicial: Fecha en la que empieza la reserva.

Fecha final: Fecha en la que termina la reserva.

Huespedes: La información de los huéspedes de la reserva, separados por guiones (“-“), y su información de huésped (nombre, documento, celular, etc.) separada por dos puntos (“:”).

Cancelado: Un booleano que determina si la reserva a sido cancelada o no.

Servicios: Lista de los servicios consumidos por los huéspedes de la reserva, consiste de la id del servicio, junto a un booleano que especifica si ha sido pagado o no (ej. 9: true).

Menú: Lista de los productos del menú por los huéspedes de la reserva, consiste de la id del producto, junto a un booleano que especifica si ha sido pagado o no (ej. 9:true).

```
diasSemana;valorTarifa;tipoHabitacion;fechaInicial;fechaFinal
LMX;4000.0;estandar;2023-04-01;2023-04-02
SD;60000.0;suite;2023-04-07;2023-04-09
LMXJVSD;1.0;estandar;2023-04-01;2023-04-05
LMXJVSD;900000.0;estandar;2023-04-23;2023-04-24
```

Las tarifas que se carguen se guardarán en el archivo Tarifas.txt. Las columnas son las siguientes:

Días semana: Los días que la tarifa aplica, está formada por los días en los que aplica, siendo X el miércoles. Por ejemplo, una tarifa que aplique los lunes, martes y miércoles tendrá un valor diasSemana de “LMX”.

Valor Tarifa: El valor en pesos colombianos que tiene la tarifa.

TipoHabitación: El tipo de habitación para el que aplica la tarifa.

Fecha inicial: fecha en formato “yyyy-mm-dd” en la que empieza a ser valida la tarifa.

Fecha inicial: fecha en formato “yyyy-mm-dd” en la que termina de ser valida la tarifa.

Uso de la consola:

Ingreso de un usuario a la aplicación: Al iniciar la consola, el usuario podrá ingresar sesión, poniendo su usuario y su contraseña.

```
Bienvenido señor/a: Mario Alexander.  
Cargo: Administrador.  
1. Crear habitacion en el inventario.  
2. Cargar tarifa para un tipo de habitación  
3. Establecer o cambiar tarifa para un servicio.  
4. Crear producto de restaurante.  
5. Consultar tarifas sin crear en los proximos 365 días.  
0. Cerrar.  
Ingrese una opción: █
```

Al iniciar sesión, se le mostrará un menú de opciones, dependiendo de su cargo dentro del sistema. Para elegir una opción, el usuario debe de ingresar el número que está al inicio del comando deseado. Si se ingresa un usuario o contraseña inválidos, se volverán a pedir tanto el usuario como la contraseña hasta que ambos sean válidos.

Cambio de usuario: Si se desea cambiar de un usuario a otro sin necesidad de detener el programa, el usuario debe de usar el comando “0. Cerrar” que es universal en el menú de todos los tipos de usuario. Este comando le pedirá al usuario de nuevo su usuario y contraseña para iniciar sesión con otro usuario.

Carga de datos: La carga de todos los archivos de datos en la memoria se hace automáticamente al iniciar la aplicación, esto se hace para que no tenga que ser necesario iniciar sesión como administrador y cargarlos manualmente, de esta forma los recepcionistas y empleados pueden hacer uso de la aplicación directamente.

Creación de habitaciones: Los usuarios de tipo Administrador pueden crear habitaciones, haciendo uso de la opción 1. Las habitaciones y camas creadas se guardarán también en los archivos .txt correspondientes.

Ingreso de tarifas: Para ingresar una tarifa por medio de un usuario administrador se hace uso de la opción 2. Después el sistema pedirá el tipo de habitación, el rango de fechas y los días que será válida esa tarifa. En el tipo de habitación se debe de ingresar “estandar”, “suite” o “suite doble”. Para el rango de fechas se deben de ingresar dos fechas en formato “yyyy-mm-dd”. Y los días validos se deben de ingresar solo poniendo las letras que

representan cada día, por ejemplo, si una tarifa es válida los lunes, martes y miércoles, el administrador deberá de digitar “LMX”.

Creación de reserva: Los usuarios de tipo recepcionista pueden crear reservas en el sistema, este les pedirá el número de huéspedes y la información de cada uno, además de las fechas en las que ocurrirá la reserva y la id de la habitación en la que se alojarán. El sistema no permitirá crear la reserva y pedirá una nueva id en dos casos: En el caso en el que haya una reserva no cancelada que ocurra en el mismo tiempo en la misma habitación o que la habitación no pueda dar abasto al número de huéspedes que requieren de cama.

Cargar un consumo: Los usuarios de tipo Empleado pueden cargar un consumo a una reserva mediante el uso de la única opción que tienen, la número 1. El empleado puede cargar el consumo de dos formas. La primera es poniendo el id de la reserva, en este caso se muestran todas las reservas, con un id, la fecha en la que ocurren y el nombre y el documento del primer huésped de la reserva, el empleado solo deberá poner el id para especificar la reserva. La segunda forma es poniendo el documento de una persona de la reserva, el sistema le mostrará al empleado todas las reservas que tienen a ese huésped y el empleado solo deberá seleccionarla. Acto seguido se le pedirá al empleado que especifique si el cargo es por un servicio o un plato del restaurante, luego de la decisión se mostrará todo el catalogo para cada uno y el empleado solo deberá poner el id correspondiente.

Persistencia de los datos: La persistencia de los datos se da mediante archivos de notas .txt, los cuales se cargan en la aplicación y también se modifican acorde se haga uso de funciones del sistema. Por ejemplo, ante la creación de un objeto (habitación, tarifa, reserva, etc.) se creará una línea al final del archivo con la nueva creación, de forma que cuando se carguen los datos la siguiente vez estos persistan. En el caso de la modificación de un objeto (cancelación de reserva, cambio de tarifa de un servicio, etc.) se modificará la línea correspondiente en el archivo correspondiente para que este cambio persista.