TRABAJO PRÁCTICO:

SOFTWARE ADMINISTRATIVO “HOTEL”

Integrantes:

* Huerta Santiago
* Luzardi Franco
* Querejeta Matias

Asignatura: Laboratorio lll

Año: 2021

**Desarrollo:**

Para comenzar con nuestro proyecto decidimos plantear primero un modelo de hotel para la realización concreta de este software, es decir, nos planteamos los distintos tipos de habitaciones y servicios extra que el hotel ofrece, de esta forma procedimos a realizar un código acorde a la idea.

Durante la realización del proyecto trabajamos con GitHub y Discord, de esta forma nos manteníamos actualizados e informados de los cambios presentes en nuestro proyecto, además de que manteníamos una comunicación activa entre los distintos integrantes

Para la realización del diagrama UML utilizamos el plugin ObjectAid el cual nos facilito el diseño del mismo. Gracias a esto pudimos visualizar nuestro proyecto y corregir errores

**Descripción del diagrama:**

Como primera instancia tenemos al Controlador. Este tiene como atributo un Hotel ya que se va a encargar de tareas que no son propias de dicha clase, como puede ser el guardado de información mediante archivos.

Dejando las clases utilitarias nos encontramos con la clase Usuario y Hotel. Esta primera es una clase abstracta, de la cual heredan Admin y Recepcionista. Su única función es contener atributos propios de un Usuario. Dentro de la clase Hotel vamos a encontrar varias colecciones del tipo TreeMap (elementos se encuentran ordenados por orden ascendente de Clave) Donde se van a guardar: Habitaciones Recepcionistas, Clientes y Reservas. También tiene un atributo del tipo double totalIngresos que guarda información de las ganancias. La clase Cliente es una clase básica que guarda información de los clientes. Como agregado tiene un ArrayList<String> historial y es modificado por el Recepcionista. Una de las clases mas importantes es la clase Reserva. Al igual que el Cliente es generada por un Recepcionista. Esta se encarga de especificar fechas de ingresos y salidas de los clientes (check-in, check-out) y los asigna a una Habitación. Además tiene un ArrayList<Productos> consumos, que registra los consumos hechos por un cliente. Siendo Producto una Enumeración.

La clase Habitación además de contener atributos trabaja también con una Enumeración llamada TipoHabitacion.

El encargado de crear Habitaciones y dar de alta Recepcionistas es el Admin.

La clase IOGenericoUtil, como su nombre lo indica, aplica de forma genérica el funcionamiento de Archivos. La cual nos permite trabajar de forma más eficiente y nos ahorra las sobrecarga de métodos según su tipo.

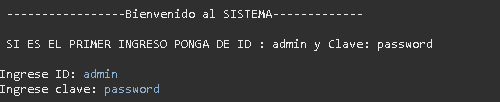
Otra clase Utilitaria es FechaHoraUtil, esta trabaja con LocalDateTime y DateTimeFormatter. En esta se declara las horas específicas de check-in y check-out. También nos permite formatear el tiempo según nos parezca.

Agregamos una clase Menú que desplegará unas opciones por pantalla para el Usuario, donde al principio le pedirá que ingrese su nombre de usuario y contraseña para acceder al sistema. Para esto, nos ayudamos con una clase llamada Password y restringe los datos de dicha contraseña. Verifica que sea alfanumérica y limita su longitud.

Con todas estas clases damos por entendido que son suficientes para la realización de este trabajo.

**Manual del Usuario:**

1- Como primera instancia se deberá ingresar al sistema, para ello tiene que escribir su usuario y su contraseña. Si es la primera vez que ingresa debera acceder con el usuario admin predeterminado. Usuario: *admin* Password: *password*.



*2-* Una vez adentro, para su mayor seguridad es necesario que cambie sus datos de ingreso. Para ello vaya a *4. Administrar Información Propia->2. Modificar Informacion.*

*3-* Dentro de los menús si usted quiere volver a las opciones anteriores solamente debe presionar la opción (cero).

*4-* Menú administrador:

·

*Administrar Recepcionista:*

o *Dar alta Recepcionista:* Se procede a la creación de un Recepcionista. Pedirá un dni, una contraseña(alfanumérica, mayor a 8 dígitos y menor a 20) y un Nombre.

o *Habilitar/Deshabilitar Recepcionista:* se listarán todos los Recepcionistas existentes en el sistema y deberá ingresar el dni del cual se va a modificar.

o *Eliminar Recepcionista:* se listarán todos los Recepcionistas existentes en el sistema y deberá ingresar el dni del cual se espera eliminar.

o *Ver Listado de Recepcionistas:* se listarán todos los Recepcionistas existentes en el sistema. Se mostrara dni, nombre y su estado (habilitado o no habilitado).

· *Administrar habitaciones:*

o *Creación de habitación:* Se procede a la creación de una habitación. Se pedirá un numero de habitación, especificar de que tipo es (matrimonial, familiar o individual) y el valor de la misma. Este valor tiene que ser mayor a $200.

o *Modificar tipo de habitación:* se listarán todas las habitaciones existentes en el sistema y deberá ingresar el numero de la cual se va a modificar.

o *Modificar precio de la habitación:* se listarán todas las habitaciones existentes en el sistema y deberá ingresar el numero de la cual se va a modificar. El precio ingresado deberá ser mayor a $200.

o *Eliminar habitación:* se pedirá el número de habitación a eliminar. Si la encuentra se procederá a pedir una confirmación.

o *Ver todas las habitaciones:* se listarán todas las habitaciones existentes en el sistema.

· *Administrar clientes:*

o *Ver registros de clientes:* Se listarán todos los clientes dentro del sistema.

o *Eliminar cliente del sistema:* se listarán todos los clientes existentes en el sistema y deberá ingresar el número de dni a eliminar.

· *Administrar información propia:*

o *Ver información propia:* se mostrará nuestro usuario y nuestro nombre.

o *Modificar información:* se pedirá que ingrese un nuevo usuario, una nueva contraseña y un nombre. Se procede a una confirmación de datos.

· *Cerrar sesión:* para cerrar sesión o cambiar de usuario solo tiene que presionar la opción (cero).

*5-* **Menú Recepcionista**:

· *Gestión de reservas:*

o *Buscar Cliente:* ingrese numero de dni del cliente ya creado previamente en el sistema.

Una vez asociado al cliente tiene que seleccionar el tipo de habitación que requiera. Deberá ingresar el numero de la misma para asociarla y las fechas de ingreso y salida de la misma en formato dd/MM/yyyy.

o *Check- in:* se listarán las reservas cuya fecha de ingreso coincide con la de hoy. Seleccione la que desea realizar el check-in.

o *Check-out:* se listarán las reservas validas para realizar esta acción. Deberá seleccionar una y se procederá a realizar la factura de la misma.

o *Ver reservas:* se listarán todas las reservas existentes en el sistema.

o *Cancelar reserva:* introduzca el id de la reserva que desea cancelar.

o *Agregar consumo en una habitación:* deberá asociar una habitación y se procederá a listar los productos disponibles. Seleccione el producto que desea agregar a la cuenta.

· *Gestión de clientes:*

o *Ver lista de clientes:* se listarán todos los clientes existentes en el sistema

o *Modificar datos de un cliente:*  debera ingresar el dni del cliente y debera seleccionar que dato desea modificar.

· *Gestión de habitaciones:*

o *Ver habitaciones libres:* se listarán todas las habitaciones libres dentro del sistema.

o *Ver habitaciones ocupadas:*  se listarán todas las habitaciones ocupadas dentro del sistema.

· *Modificar información propia:*

o *Ver información propia:* se mostrará nuestro usuario y nuestro nombre.

o *Modificar información:* se pedirá que ingrese un nuevo usuario, una nueva contraseña y un nombre. Se procede a una confirmación de datos.

· *Cerrar sesión:* para cerrar sesión o cambiar de usuario solo tiene que presionar la opción (cero).

**Cuestiones a Mejorar:**

* + Durante el proyecto consideramos la implementación de una interfaz gráfica, comenzamos la ejecución de esta idea pero no pudimos llevarla a cabo. Sería bueno en un futuro poder terminar esta idea para poder generar un software más amigable con el usuario.

Además a través de esta interfaz lograríamos la exclusión de prints y scan que se encuentran en muchas clases.

* + Realizar una mejor organización de las funciones para que el uso del programa sea mas simple todavía.

**Conclusiones:**

Fue un trabajo interesante el cual nos llevó a desarrollar mayormente nuestras capacidades además todos nos vimos en la situación en la que tuvimos que trabajar con compañeros que no conocíamos, por lo cual fue una buena experiencia respecto al trabajo en equipo.

Durante el trabajo tuvimos varios imprevistos con distintas clases, pero el más grande fue el hecho de que intentamos implementar una interfaz gráfica antes de terminar el código, por lo cual esto nos dificulto y atrasó mucho, finalmente decidimos hacer un menú simple y dejar el gráfico para una futura implementación.

Uno de los problemas que tuvimos fue la superposición de reservas, es decir, que no coincidan reservas en un mismo periodo en determinada habitación, creemos que era el principal desafío del trabajo e investigando en distintas fuentes lo pudimos resolver.

**Links Extras:**

**-GitHub:** https://github.com/iammatuke/HOTEL-LAB3

**Bibliografia**:

* [https://stackoverflow.com](https://stackoverflow.com/)
* [https://lawebdelprogramador.com](https://lawebdelprogramador.com/)
* [https://javahispano.org](https://javahispano.org/)
* Enum -<https://jarroba.com/enum-enumerados-en-java-con-ejemplos/#:~:text=Un%20enumerado%20(o%20Enum)%20es,la%20implementaci%C3%B3n%20de%20la%20clase>.

- <https://javadesdecero.es/avanzado/enumerados-enum-ejemplos/>

* Tutorial para UML:

- <https://www.youtube.com/watch?v=4H7Aft7kLD4>

* Singleton:

-https://www.arquitecturajava.com/ejemplo-de-java-singleton-patrones-classloaders/