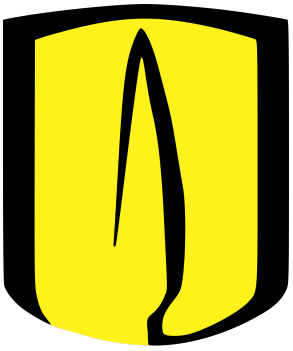
Proyecto 3 DPOO



Universidad de los Andes

2023-1

Hector Duarte Calderon - 202210781
Juan David Duarte - 202215070
David Rey -202211410

1. **Cosas que salieron bien y mal**

Durante el desarrollo de los proyectos, tuvimos dificultades para estimar el trabajo necesario debido a nuestro desconocimiento inicial de la tecnología. Por ejemplo, tuvimos que aprender a manejar la persistencia de datos sin usar bases de datos, lo que resultó ser un desafío. Además, tuvimos que considerar la compatibilidad de la aplicación tanto para PC como para móviles, lo que añadió una capa adicional de complejidad.

En el análisis del dominio, nos encontramos con problemas al tratar de entender y modelar correctamente el sistema del hotel. Por ejemplo, tuvimos que considerar aspectos como la gestión de reservas, la administración de pagos, la gestión de personal, la generación de informes y análisis, y la comunicación interna. Además, tuvimos que considerar las necesidades de diferentes actores, como los huéspedes y el administrador del hotel.

El diseño en un entorno incierto fue otro desafío que enfrentamos. Por ejemplo, tuvimos que considerar cómo manejar situaciones como las cancelaciones y cambios de reserva, el servicio de habitaciones, el acceso a Wi-Fi, y la recepción 24/7. Además, tuvimos que diseñar el sistema para manejar diferentes tipos de habitaciones y servicios, y para permitir la comunicación entre diferentes roles dentro del hotel.

A pesar de estos desafíos, también tuvimos varios éxitos. Por ejemplo, logramos desarrollar una aplicación funcional que satisfacía los requerimientos del proyecto. Además, aprendimos mucho sobre el análisis del dominio, la estimación del trabajo y el diseño en un entorno incierto, lo que nos ayudará en futuros proyectos como lo es el uso de herramientas de diseño con los es swift o javafx, también patrones de diseño, etc.

2. Decisiones tomadas

La decisión de cargar las habitaciones en listas dentro del inventario y tener un mapa con fechas para los cambios de precio parece haber sido exitosa, ya que permitió una gestión eficiente de las reservas de habitaciones y los precios. Del mismo modo se decidió cargar los usuarios de manera automática cuando se inicia la aplicación, estos usuarios se meten en un hash para facilitar los inicios de sesión.

Decidimos crear una nueva clase llamada usuario para utilizar lo de inicio de sesión, con atributos de usuario, contraseña, documento, y reservas, que es un arrayList que tiene todas las reservas creadas por este usuario.

3. Problemas durante el desarrollo

Durante el desarrollo de nuestros proyectos, uno de los desafíos más significativos que enfrentamos fue la creación de la interfaz gráfica de usuario (GUI) de la aplicación. La complejidad de este proceso fue mayor de lo que inicialmente anticipamos. Nos encontramos con problemas para garantizar una transición fluida y sin errores entre las diferentes páginas de la aplicación. Cada página tenía su propia funcionalidad y requerimientos únicos, y asegurarnos de que todas estas partes funcionan en armonía resultó ser una tarea complicada. A pesar de los desafíos, trabajamos juntos como equipo para resolver estos problemas, aprendiendo mucho en el proceso sobre la importancia de un diseño de interfaz de usuario bien planificado y ejecutado.

4. Roles de la aplicación

Administrador

En la interfaz de administrador están abiertas cuatro ventanas con las funciones que este puede ejecutar, las cuales son buscar una reserva, la cual se busca en una lista de reservas que tenemos en el inventario, y si esta se encuentra notifica. Otra ventana de crear reserva, la cual pide los datos necesarios para crear una reserva y una vez esta se crea se mete en la misma lista antes mencionada en el inventario, y las últimas dos ventanas, una que añade plato que funciona de manera similar a la de crear reserva, y la de cobrar servicio, que permite cobrar un servicio a un grupo de personas o solamente a un huésped.

Empleado

En la interfaz de empleado tiene una única ventana, que es la misma ventana de cobrar servicio que aparece en la interfaz del administrador, y funciona de la misma manera, permitiendo cobrar a un huésped o a un grupo de huéspedes.

Recepcionista

La interfaz de recepcionista tiene solamente las funciones referentes a las reservas, es decir, la ventana de crear reservas y la de buscar reservas, ambas también presentes en la ventana de administrador, y funcionan de la misma manera.

Usuario

En la interfaz de usuario el inicio de sesión de usuario reconoce si el usuario existe dentro de los datos y si el usuario existe reconoce que la contraseña sea correcta o no, si el usuario y contraseña son correctos aparece un mensaje que indica que el inicio fue correcto. En caso de que alguno sea incorrecto el sistema no dejará que se inicie la sesión

Los usuarios con una cuenta creada son guardados en un csv que se encuentra en la carpeta de datos con el nombre “usuarios.csv”. Este archivo cuenta con tres columnas, la primera para el nombre de usuario, la segunda para la contraseña del usuario y en la tercera se guarda el documento de identidad del usuario. Cuando se le da a crear cuenta, se guardan los datos en el archivo csv y al mismo tiempo en un hash que contiene estos datos para hacer el inicio de sesión rápido.

Los usuarios pueden crear sus reservas y consultar sus reservas.

5. Datos que usa la aplicación

Persistencia de datos

La persistencia de los datos se tiene en la clase inventario que almacena todos los datos necesarios para el funcionamiento del hotel.

Habitaciones: Las habitaciones están contenidas en tres listas, una lista para cada tipo de habitación (estándar, suite y doble suite)

Platos: Los platos del restaurante estarán contenidos en una lista.

Servicios: Todos los servicios fuera de las habitaciones y los platos estarán contenidos en otra lista, estos servicios pueden ser (ejemplo: spa, tours, etc…).

Usuarios: Los datos necesarios para que los usuarios inicien sesión.

Entrada de datos

Para que el PMS funcione, el administrador debe subir un archivo de tipo CSV, que contenga la información de las habitaciones, como: el tipo de habitación, la capacidad de personas, lista de camas y con indicador si es de solo niño, costo de la habitación. Más adelante se pueden reconocer más información necesaria para cada habitación.

Datos de habitaciones

- **id:** identificador de cada habitación.
- **tipo:** tipo de habitación que puede ser: estándar, suite y doble suite.
- **Costo:** El valor de dicha habitación, más adelante será de tipo doble.
- **Camas:** Información con las camas estructurada de la forma “2-2-1n”, donde se tiene la cantidad de cada cama, y si el número está acompañado de una “n”, entonces dicha cama es solo para niños; en este caso “2-2-1n”, hay dos camas de dos personas y una capa para un niño.
- **Ubicación:** piso en la cual se ubica la habitación.
- **Balcón:** true o false dependiendo si la habitación tiene o no balcón.
- **Cocina:** true o false dependiendo si la habitación tiene o no cocina.

Datos de platos

- **id:** identificador de cada plato.
- **nombre:** nombre del plato.
- **Costo:** El valor de dicho plato, más adelante será de tipo doble.
- **habitación:** true o false dependiendo si el plato puede o no llevarse a la habitación.

Datos de otro servicio

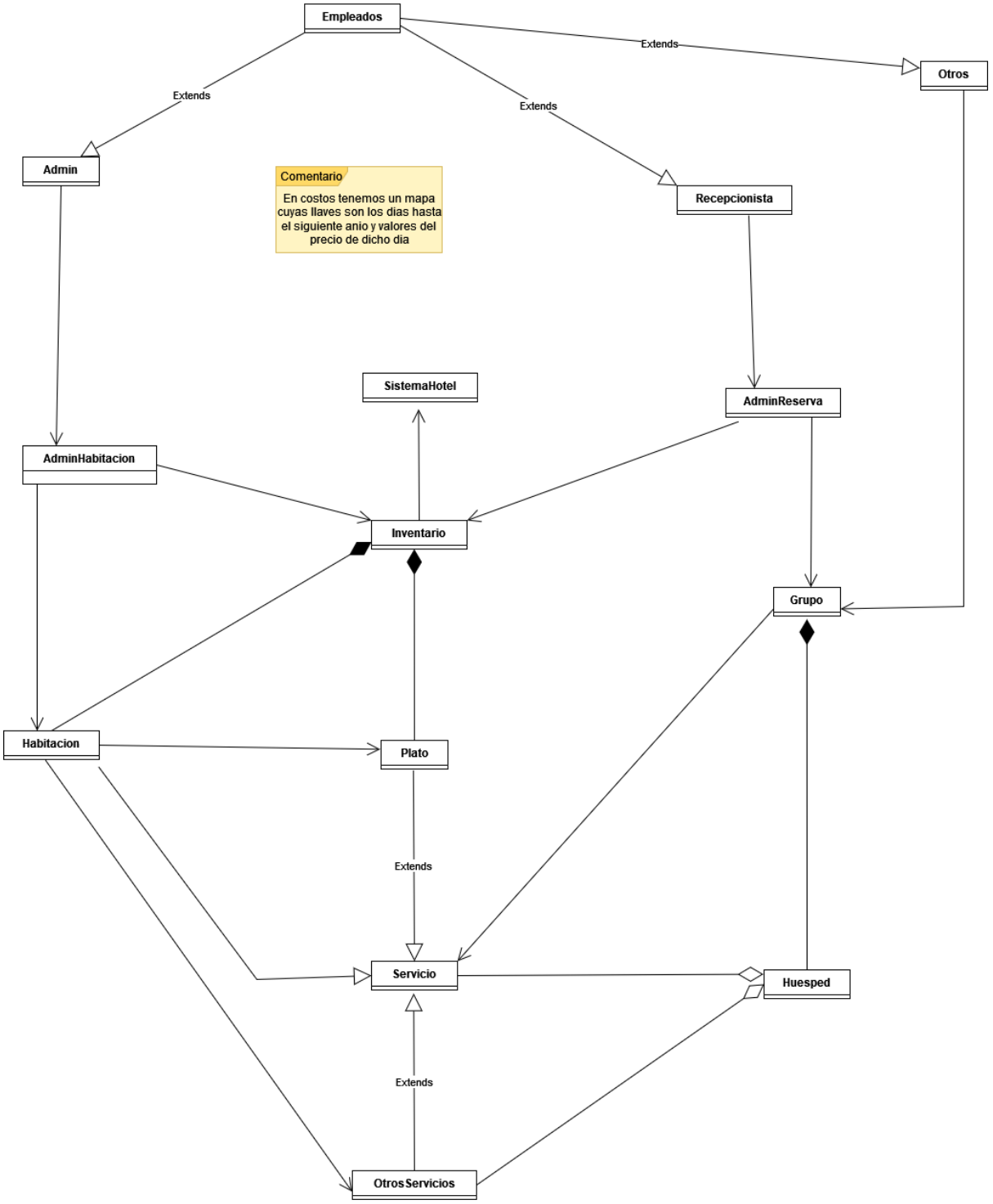
- **nombre:** nombre del servicio.
- **Costo:** El valor de dicho servicio, más adelante será de tipo double.

Datos de usuarios

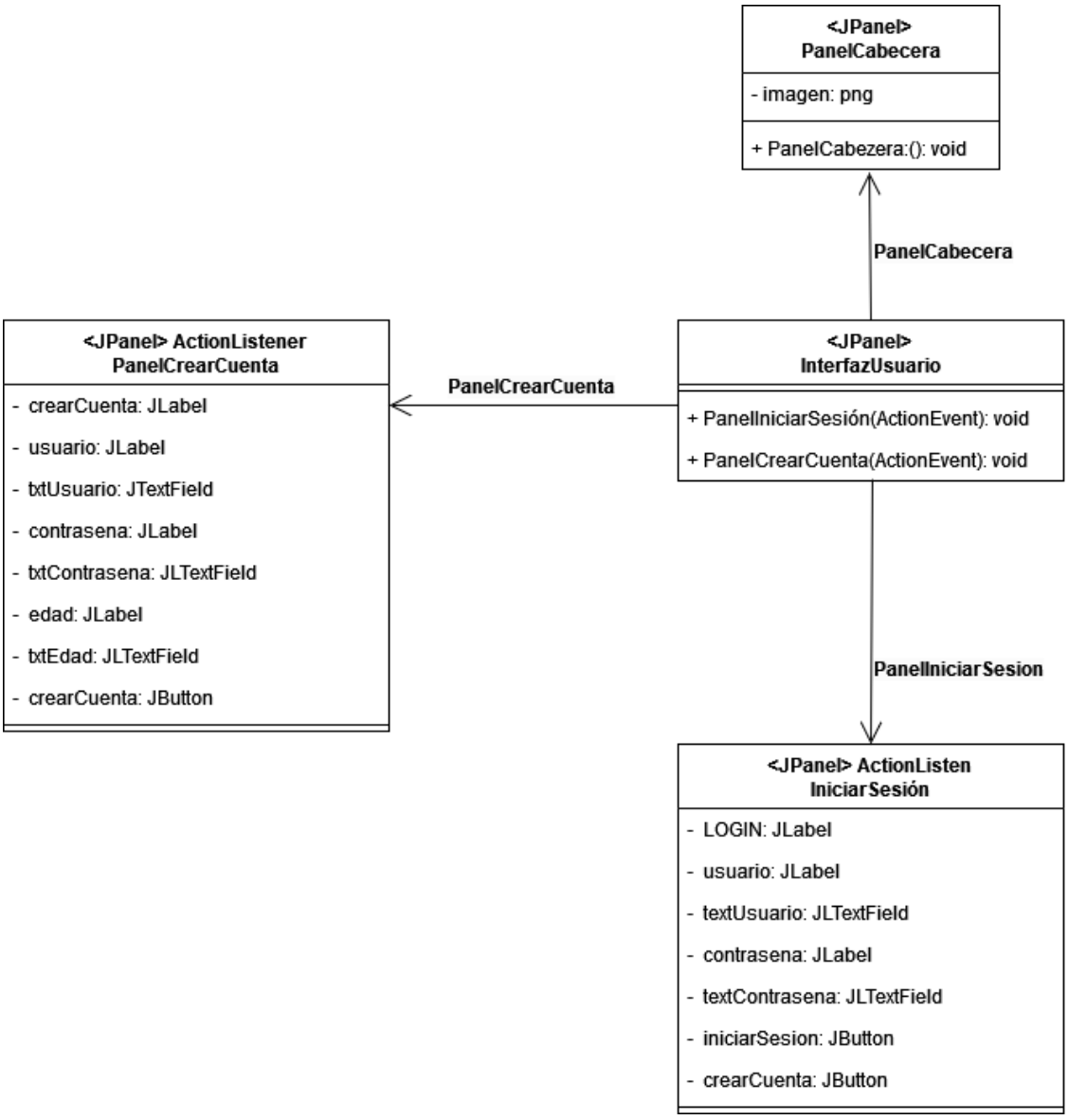
- **usuario:** nombre de usuario.
- **contrasenia:** contraseña del usuario.
- **documento:** documento del usuario.

6. UML´s

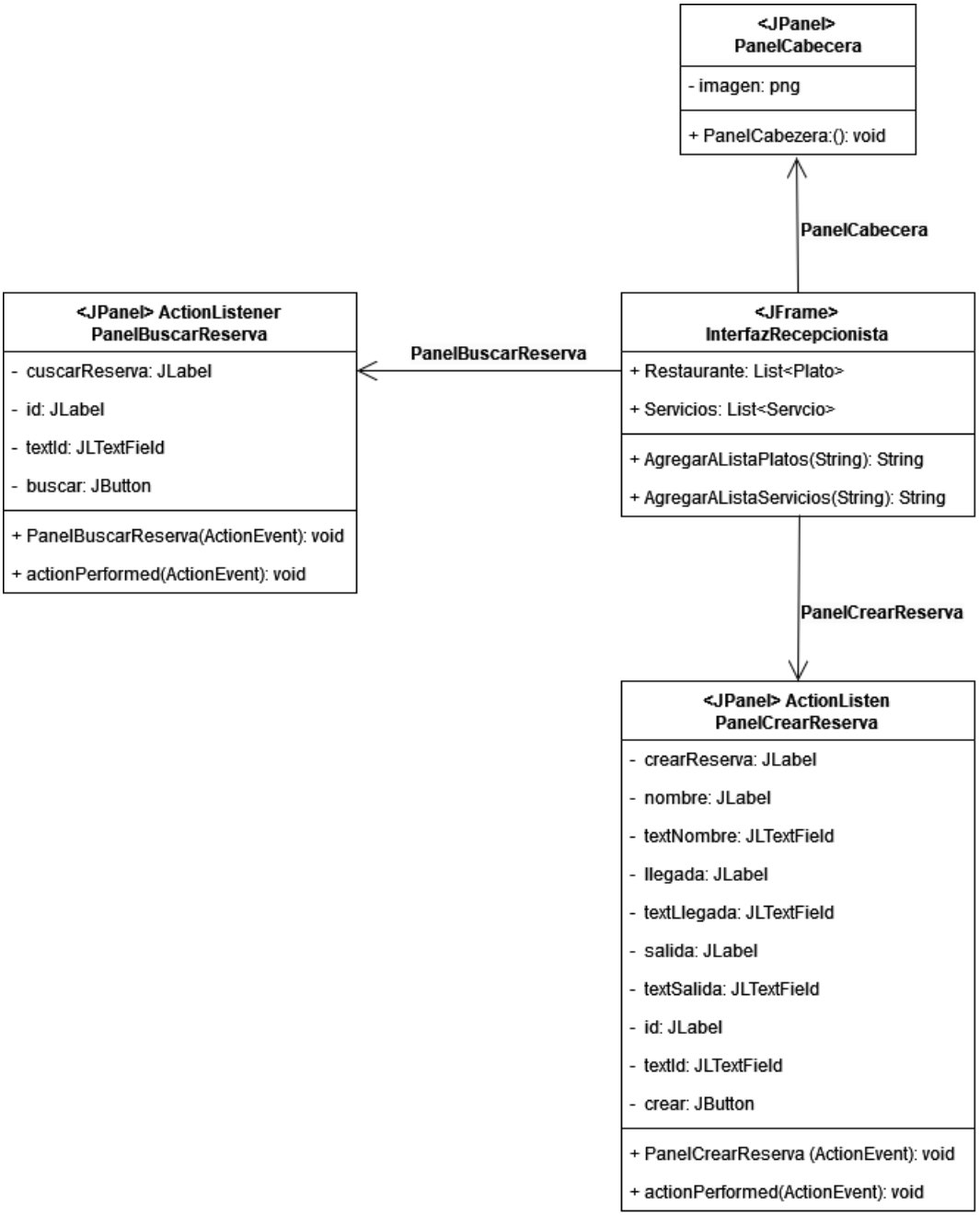
SistemaHotel



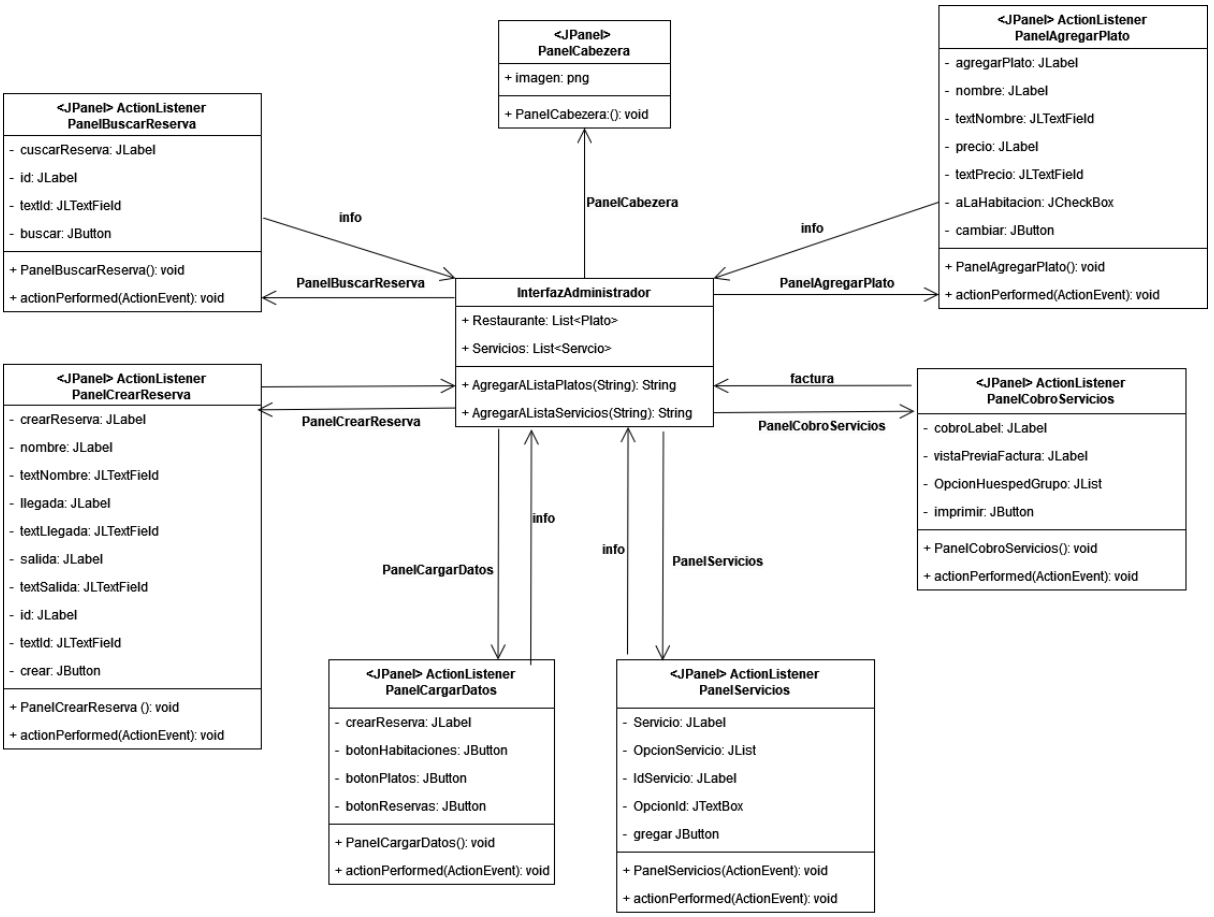
InterfazUsuario



InterfazRecepcionista



InterfazAdministrador



InterfazEmpleado