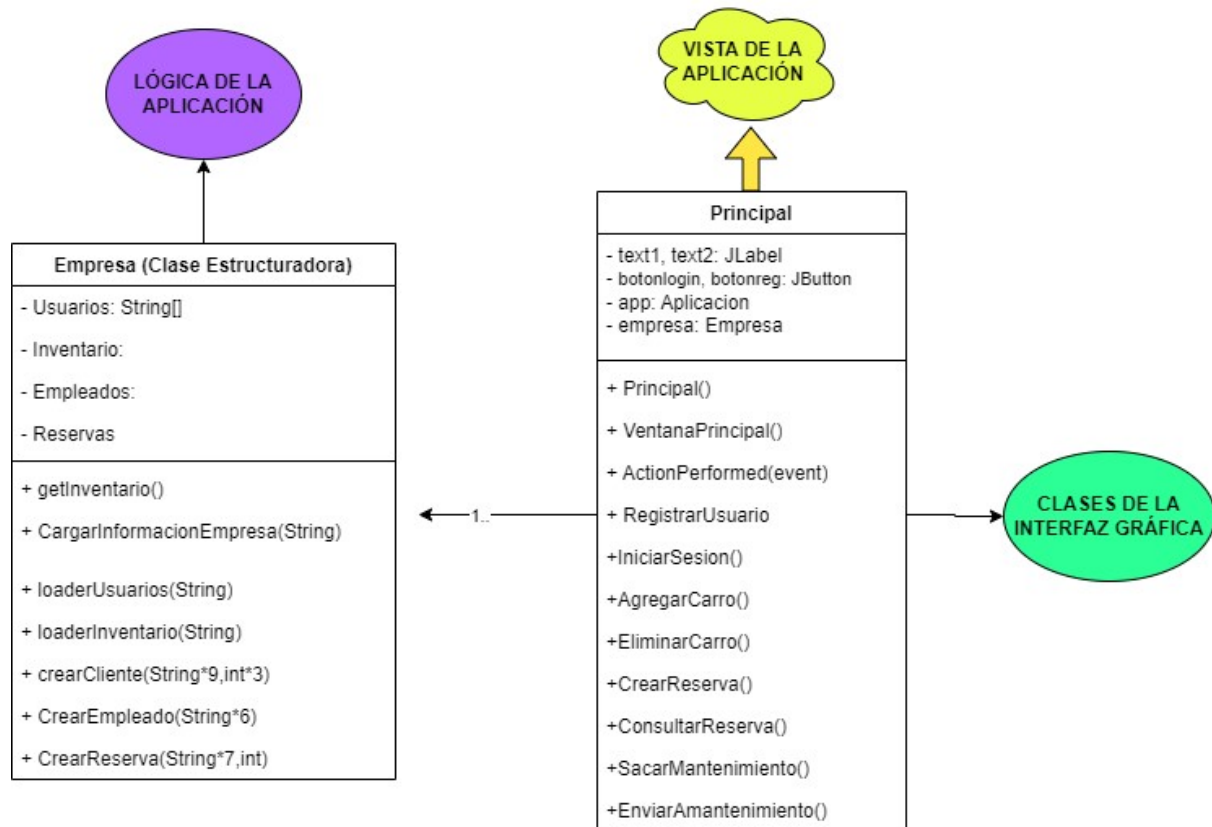


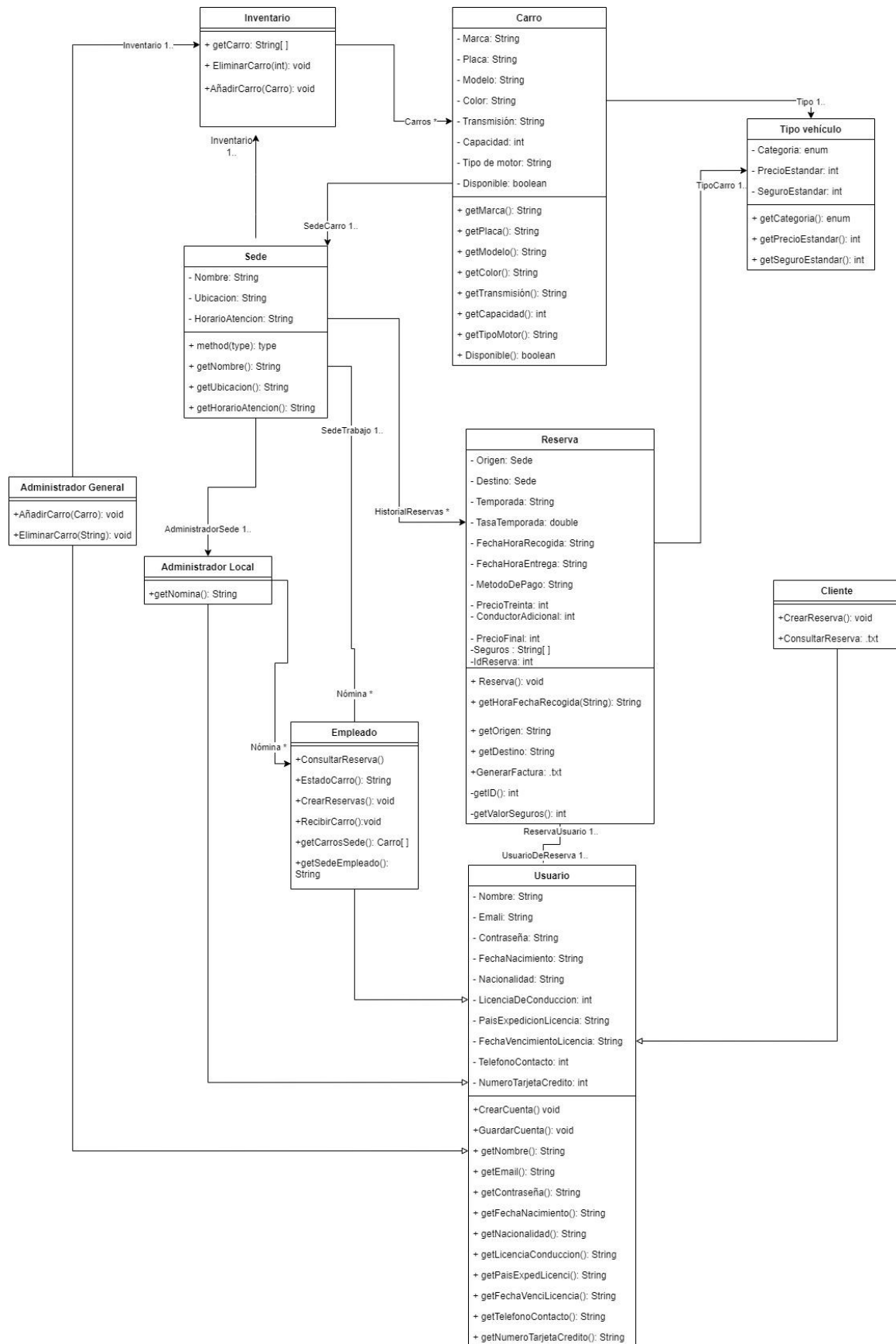
Grupo 6
Documento de Diseño
Proyecto 2 - Entrega 2

1. Planeación del acoplamiento.



En este diagrama planeamos la estructuración de la interfaz por medio de una clase "Principal". Esta clase tiene los componentes interactivos necesarios para conectarse con la interfaz pero también es el eje entre la clase empresa y la interfaz. La clase empresa fue la clase estructuradora de la lógica dónde se aglomeran todos los métodos fundamentales para resolver los requerimientos funcionales de la aplicación. Adicionalmente, desde la clase Principal planeamos correr el programa con el fin de que sea el portal para mostrar consola con elementos gráficos de la aplicación.

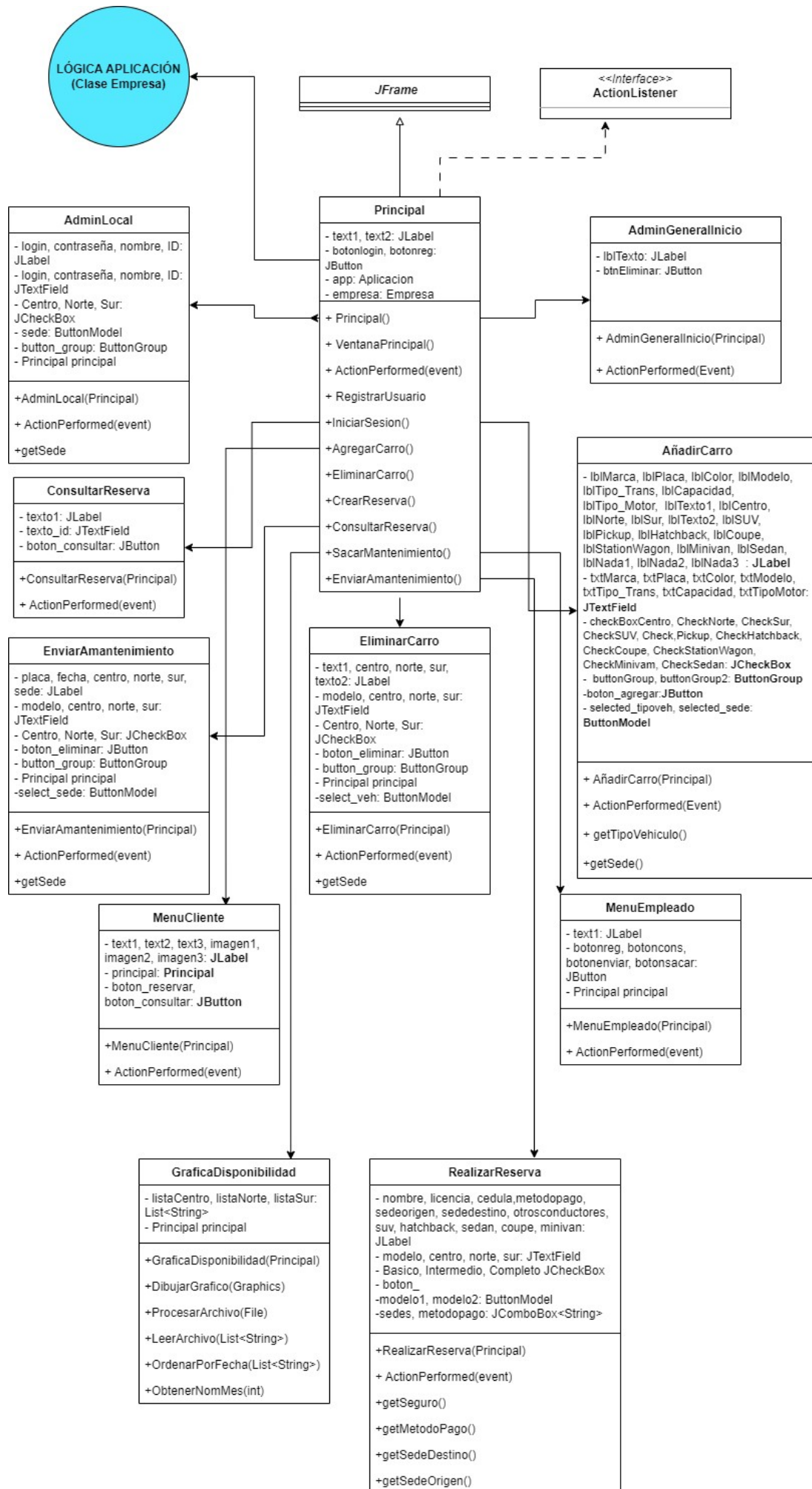
2. Diagrama de clases de la lógica del programa.



Este diagrama relaciona las clases relevantes entre los objetos que van a transmitir y gestionar la información de los requerimientos. Generalmente la clase que se relaciona con el problema le envía una solicitud a la consola, para que el usuario ingrese unos datos en la interfaz gráfica. Estos datos se envían a una clase que en la mayoría de las veces creará una instancia del objeto y lo guardará. Al tener esta información, la clase estructuradora se encarga de resolver cualquier lógica adicional necesaria del requerimiento y genera la funcionalidad para guardar la información en un archivo .txt.

3. Diagrama de clases de la interfaz gráfica.

En este diagrama hicimos la planeación sobre cómo se estructuraría la interfaz gráfica del programa. Para establecer una conexión entre la lógica del programa y la interfaz creamos la clase Principal para invocar los métodos de la clase estructuradora, Empresa. Una vez con la lógica traída dentro de una sola clase podemos crear todas las demás clases que contendrán el código para cada una de las ventanas de la plataforma. Todas estas se conectan con la clase principal por medio de una instancia en cada una de ellas y tendrán los métodos necesarios para que el usuario interactúe. Particularmente, se establecen las implementaciones de la interfaz *ActionListener* para que cada uno de los elementos que requieran de respuesta por parte del programa trabajen. También, las clases heredan la lógica de la clase y librerías de *JFrame* donde se utilizan los componentes como paneles, iconos, botones, cuadros de texto y modelos para configurar la consola visualmente y hacer que toda la aplicación funcione.



4. Responsabilidades

Responsabilidad	Rol
Registrar la información básica de un usuario para poder almacenarla en la base de usuarios.	Creación de usuarios
Reservar toda la información básica de los usuarios así como utilizarla para el posterior log-in de cada persona	Almacenador de usuarios
Crear una reserva de vehículo en la aplicación	Generador reserva
Ajustar el precio a pagar en la reserva conforme los seguros adquiridos y las tarifas por temporadas de la empresa	Caracterizador precio reserva
Permitir al empleado saber en que estado esta un vehículo y conocer su disponibilidad para el efecto de una reserva	Rastreador estado de vehículos
Mantener la persistencia de la aplicación, donde toda la información registrada quede procesada en facturas.	Archivador facturas
Permitir al administrador conocer la nómina de cada sede y tratar a los empleados como un usuario de la aplicación	Gestión de empleados
Permitir que el administrador general gestione el inventario de la empresa	Gestión de inventario
Contener toda la información acerca de los vehículos de la empresa.	Almacenador de inventario
Fuente de información de todos los vehículos existentes en la empresa junto con sus características respectivas	Base de datos