

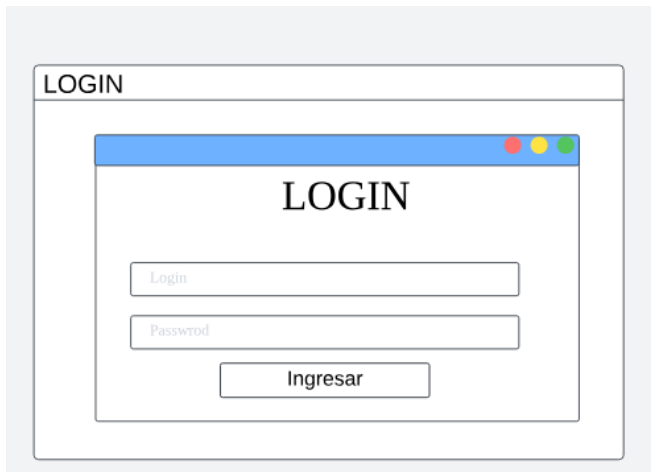
Eric Alarcon Dolyenko - 202220287
Daniel Humberto Velez - 202221703
Andres Botero Ruiz - 202223503
Manolo Hernández Rojas - 202224469

Documento Implementación y Diseño Interfaz Gráfica

Primera Iteración del Diseño:

Imágenes:

Package Login:



Package Admin General:

The image displays 10 mockups for a car management system, organized into three rows. Each mockup is a window with a blue title bar, a 'LOG OUT' button, and a set of window control buttons (red, yellow, green).

- Mockup 1 (Top Left):** Titled 'Bienvenido Admin General'. It contains two columns of buttons: 'Dar de baja un vehiculo' (with a 'Placa:' input field), 'Registrar nuevo vehiculo', 'Modificar Seguros', 'Modificar Informacion Sede', 'Revisar gráfico por sede', 'Modificar Tarifa por Categoría', 'Revisar Mantenimiento', and 'Generar Log de un Vehiculo (su historial)' (with a 'Placa:' input field).
- Mockup 2 (Top Right):** Titled 'MODIFICAR INFORMACION SEDE'. It features two dropdown menus: 'Que sede va a cambiar' and 'Que aspecto va a cambiar', followed by an 'ACEPTAR' button.
- Mockup 3 (Middle Left):** Titled 'ADMIN GENERAL'. It has a 'Registrar vehículo' section with input fields for 'Placa', 'Marca', 'Modelo', 'Color', 'Transmision', 'Combustible', 'Capacidad de pasajeros', 'Sede', and 'Categoría', and a 'Confirmar' button.
- Mockup 4 (Middle Center):** Titled 'Modificar Seguros'. It includes a 'Que seguro desea editar:' section with radio buttons for 'Multas', 'Solo Choques', and 'Todo Riesto', an 'Inserte la nueva tarifa' input field, and an 'ACEPTAR' button.
- Mockup 5 (Middle Right):** Titled 'Modificar Tarifa por Categoría'. It features a 'Que categoría desea editar:' section with radio buttons for 'Van', 'Clasico', 'Tractor', 'Lujo', 'Todoterreno', and 'SUV', an 'Inserte la nueva tarifa:' input field, and an 'ACEPTAR' button.
- Mockup 6 (Bottom Left):** Titled 'Revisar Mantenimiento'. It includes an 'El carro a revisar es:' input field, a 'Dar de baja?' section with radio buttons for 'Si' and 'No', and an 'ACEPTAR' button.
- Mockup 7 (Bottom Center):** Titled 'Revisar gráfico por sede'. It features a 'Sede a revisar:' section with radio buttons for 'RadiatorSprings', 'CarburetorCanyon', 'Emeryville', and 'MotorSpeedway', and a 'Gráfico:' section showing a calendar grid with green squares indicating events.

ADMIN LOCAL

LOG-OUT Consola Admin Local

- Crear Usuario (Cliente)
- Registrar Trabajador
- Modificar información Trabajador
- Crear Usuario (Cliente)

LOG-OUT Modificar Trabajador

Trabajador a modificar

Que aspecto desea modificar:

☐ Nombre
 ☐ Documento de Identidad

☐ Fecha de Nacimiento
 ☐ Imagen

☐ Nacionalidad
 ☐ Sede de Trabajo

☐ Rol del empleado

Nuevo Valor

ACEPTAR

Registrar Trabajador

Relleno los siguientes campos

Datos personales

Nombre completo

Fecha de nacimiento (dd/mm/YYYY)

Nacionalidad

No. Documento de identidad

Foto documento Escoge una Imagen

Sede

Tipo de trabajador

☒ Montador ☐ Inventario

Registrar Usuario(Cliente)

Rellena los siguientes campos

Datos personales

Nombre completo

Fecha de nacimiento (dd/mm/YYYY)

Nacionalidad

No. Documento de identidad

Foto documento Escoge una Imagen

Método de Pago

Tarjeta de Credito

Tipo de Tarjeta

No. Tarjeta

Fecha Vencimiento (mm/YYYY)

Fecha Vencimiento (mm/YYYY)

Licencia de Conducion

Foto Licencia Escoge una Imagen

No. licencia de Condución

País de expedición

Fecha de Vencimiento

Package Cliente:

CLIENTE

LOG OUT

Cliente Actual: Guido

Escoge una sede

Radiator Spring

Catálogo

Emeryville

Catálogo

Carburator Canyon

Catálogo

Motor Speedway

Catálogo

LOG OUT

Cliente Actual: Guido

Realizar reserva

Día recogida

Hora Recogida

Día entrega (dd/mm/yyyy)

Hora Entrega

Sede Entrega

Categoría vehículo

Confirmar

LOG OUT

Crear Usuario (Cliente)

Relleno los siguientes campos

Datos personales

Nombre completo

Fecha de nacimiento (dd/mm/yyyy)

Nacionalidad

No. Documento de identidad

Foto documento

Escoge una Imagen

Número de celular

Email

Tarjeta de Crédito

Tipo de Tarjeta

No. Tarjeta

Fecha Vencimiento (mm/yyyy)

Fecha Vencimiento (mm/yyyy)

Licencia de Condución

Foto Licencia

Escoge una Imagen

No. Licencia de Condución

Pais de expedición

Fecha de Vencimiento

LOG OUT

Cliente Actual: Guido

Formalizar alquiler

ID de reserva

Deses adicionar nuevos conductores?

Deses cambiar datos de su reserva?

Seguro

Confirmar

LOG OUT

Cliente Actual: Guido

Escoge una opción

Iniciar Reserva

Formalizar Alquiler

Entregar un Vehiculo

nota: Debe tener una reserva

LOG OUT

Modificar Datos de la Reserva

¿Que categoría desees editar?

Fecha Devolución

Hora Devolución

Sede de Entrega

Nuevo valor

ACEPTAR

LOG OUT

Entregar vehiculo

Ingresar la placa

ID de reserva

Confirmar

LOG OUT

Cliente Actual: Guido

Catálogo de: Radiador Springs

Placa	Marca	Modelo	Color	Transmisión	Combustible	Tipo
EQR485	renault	duster	negro	manual	gasolina	vans
KYC321	volkswagen	terra	rojo	manual	corriente	clasico
WOT7712	ford	trend	rosa	manual	gasolina	lujo
LAA743	kia	contemporary	rosa	manual	gasolina	todoterreno

LOG OUT

Nuevos Conductores

Nombre

Número de licencia

Pais de licencia

Fecha de Vencimiento

Foto documento

Escoge una Imagen

Confirmar

Package Empleado Inventario:

Empleado Inventario

LOG OUT

Empleado Invetario: Sally

El carro a revisar es _____

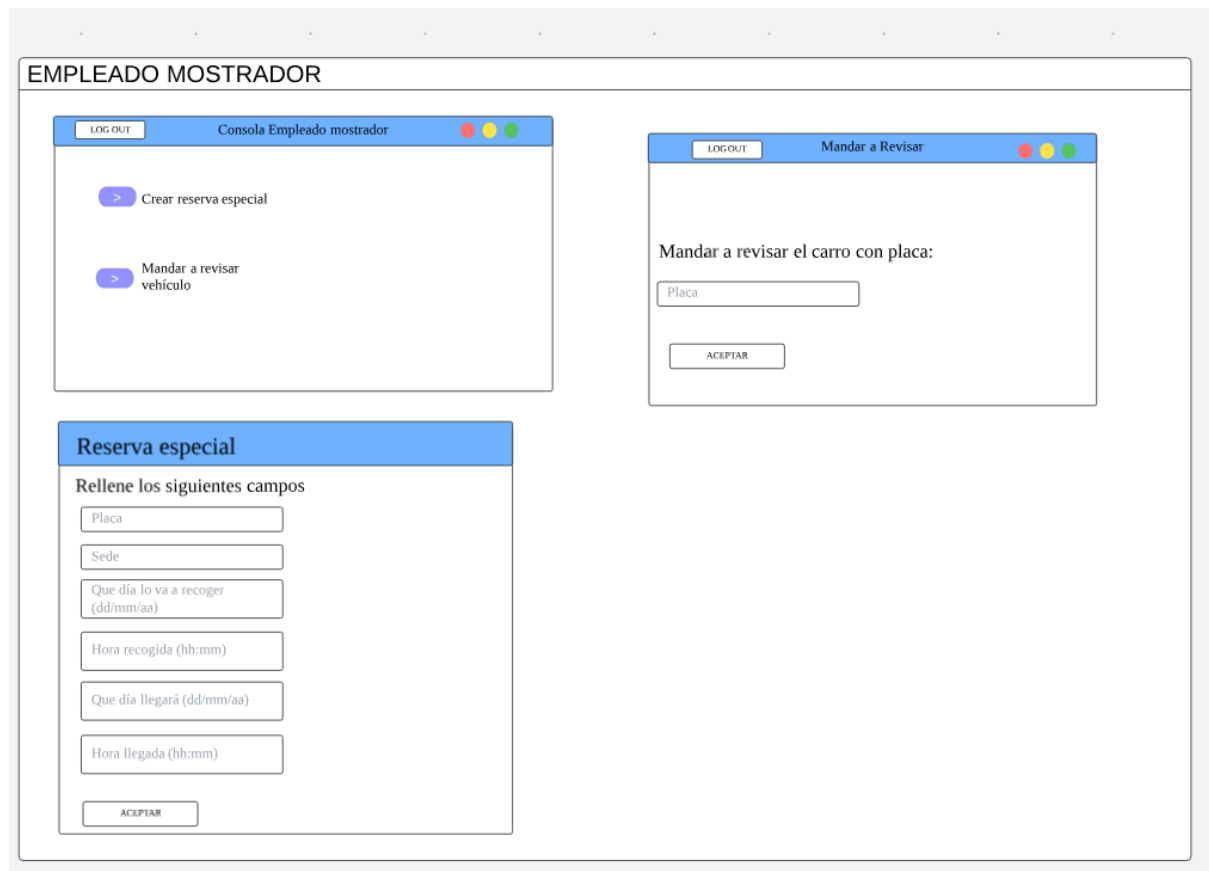
Ponerlo en manteinimiento?

Si

No

ACEPTAR

Package Empleado Mostrador:



Descripción General:

En las imágenes de la parte superior podemos encontrar nuestra primera idea para la interfaz gráfica. Donde la lógica de la implementación es separar por paquetes los diferentes tipos de usuario que tiene la aplicación. A pesar de que la lógica de la implementación no haya cambiado, ciertos aspectos estéticos sí sufrieron cambios, por facilidad en la implementación.

Cambios:

Entre los mayores cambios que tuvimos frente a estas imágenes y el concepto final fue darle una tonalidad de color diferente con un patrón de colores entre rojos y amarillos. Además agregamos los botones de “LOGOUT” en algunas páginas en la parte inferior o botones de “volver” de tal manera que el usuario siempre tenga la opción de ir a otro menú sin ingresar nuevos valores que puedan afectar el diseño.

Además, para los botones de opción múltiple que se plantearon en las imágenes de arriba, se cambiaron a los componentes de un JComboBox, ya que nos pareció adecuado y más efectivo a la hora de obtener los datos cuando un usuario seleccionaba esa opción. Cabe recalcar, que la información que se obtiene a partir de los JComboBox es información que

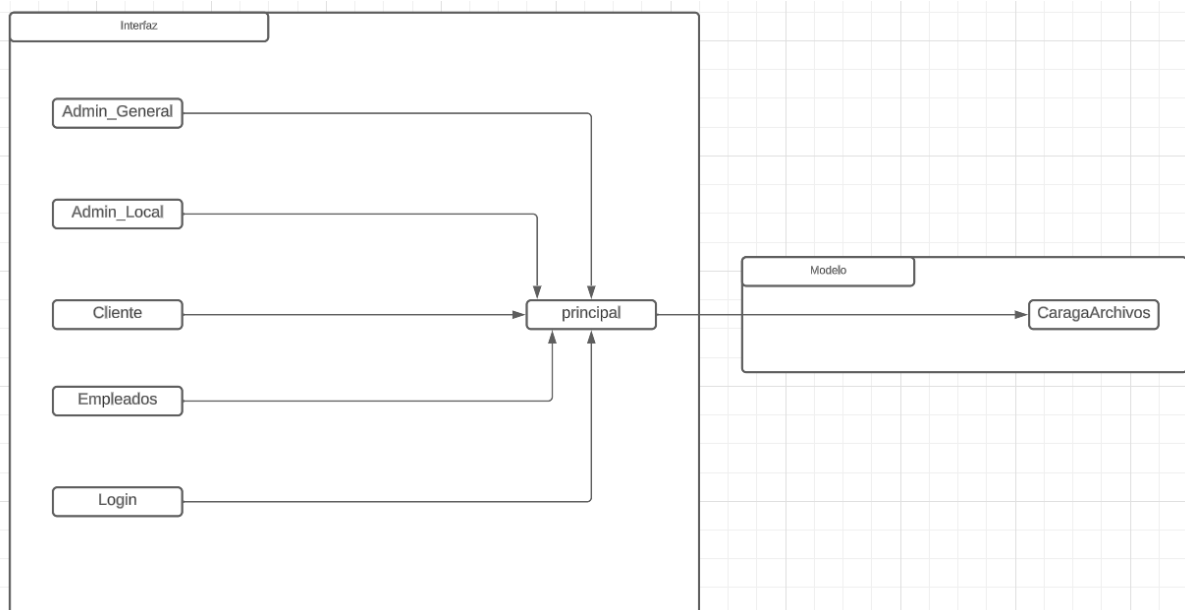
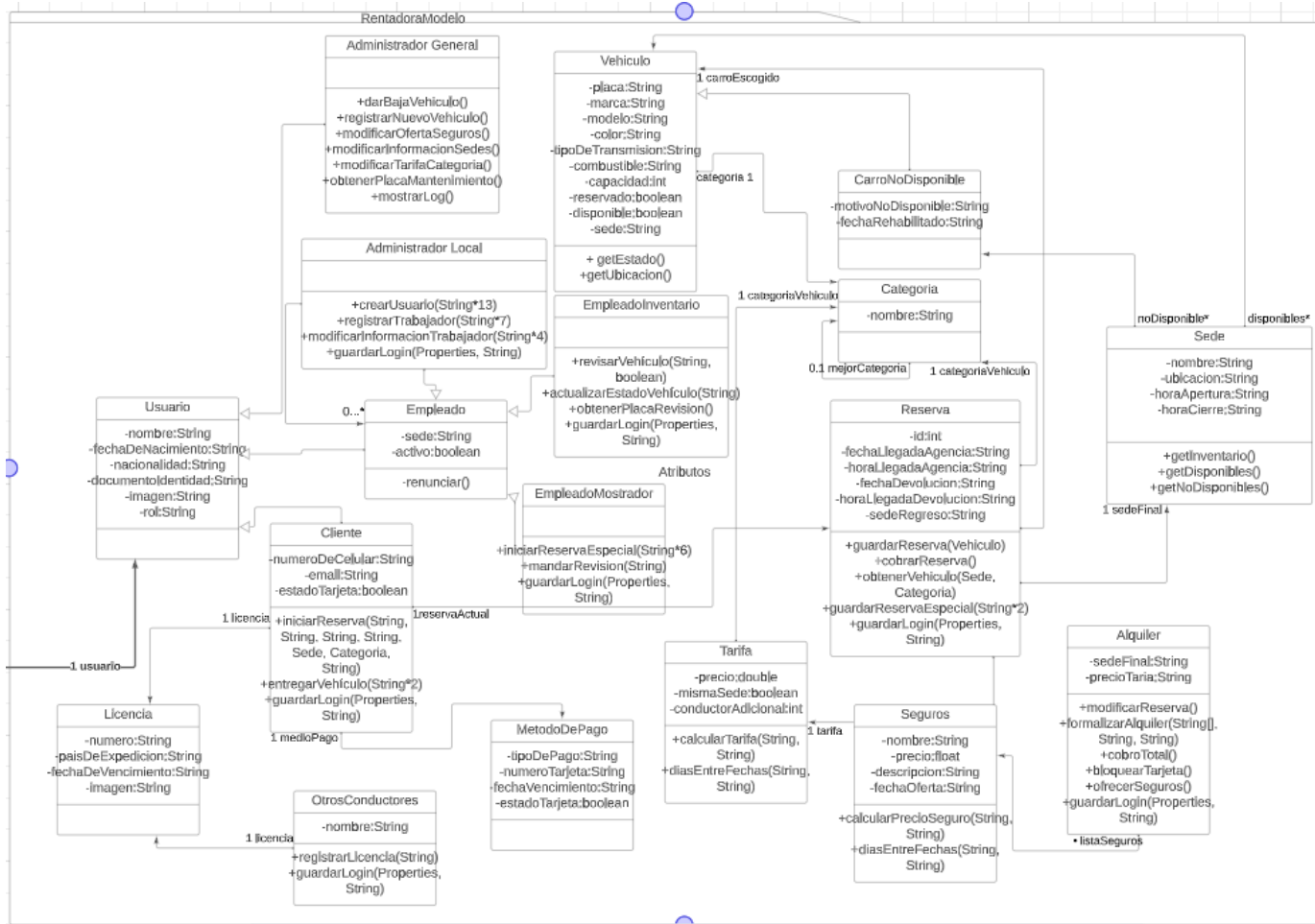
necesita ser escrita en un formato muy específico, por lo que de esta manera se evita el problema de que el usuario ingrese una sede que no existe, o una categoría mal escrita. Por último, a pesar de que en algunas imágenes está el recuadro para agrandar o modificar el tamaño de la pestaña, al final esta opción de reajustar el tamaño lo quitamos.

Adiciones:

Las adiciones que se hicieron a la interfaz fue un paquete que se llama “Principal”, el cual su propósito es tener a la clase Principal.java que es un JFrame en el cual se contienen a todos los diferentes paneles de las imágenes anteriores. Además el layout que se usará es un cardLayout, de tal manera que los diferentes requerimientos son divididos en distintas ventanas que van apareciendo ante el usuario dependiendo de con qué usuario y contraseña ingrese y las opciones que escoja. En adición, hay una clase que se llama “verify” que se encarga de revisar si la información que se obtiene del usuario es congruente y lógica para no obtener errores en la ejecución de los requerimientos.

Además, se implementaron clases para mostrar si la acción de un usuario se completó satisfactoriamente o ocurrió un error controlado. Estas clases son “Notificación” y “Error Display” respectivamente, los cuales son JDialogs. Notificación, se despliega cuando un usuario realiza una acción satisfactoriamente, por ejemplo, realizar una reserva, formalizar un alquiler etc... El error display, le indica a un usuario si no ha llenado todos los campos necesarios con información válida para realizar un requerimiento, o si no se ha obtenido la respuesta esperada de un método. No obstante, estos errores son controlados, y no afectan la funcionalidad de la aplicación.

Diagrama de clases y relaciones:



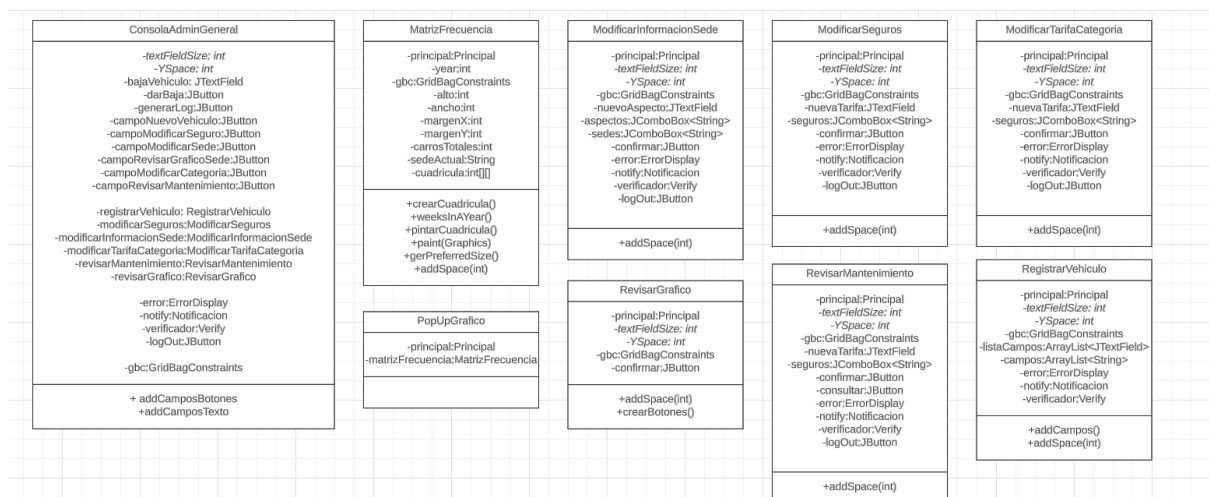
Descripción General:

El primer UML describe cómo se conectan los componentes del modelo entre ellos. Es importante mencionar que la clase “consola” fue reemplazada por los paquetes de interfaz gráfica. Sin embargo, notamos que “consola” tenía aspectos de la lógica de la implementación por lo que se implementó una nueva clase que se llama “cargar Archivos” que se ocupa de cargar los diferentes archivos de textos que tienen toda la información de las reservas, alquileres, etc...

Conexión con la Interfaz Gráfica

La clase “Principal” es la que tiene acceso a la clase de cargar Archivos, usuarios, y sedes, por lo que los demás paneles tienen que pedirle a la clase principal que obtenga la información necesaria. Esto se hizo con el propósito de que solo haya un canal para poder conectarse con el modelo y sea más sencillo de revisar cómo se comunican los paquetes de interfaz con los del modelo. Ahora se va a explicar a mayor detalle cómo cada uno de los paquetes de la interfaz (admin general, cliente, etc... están compuestos)

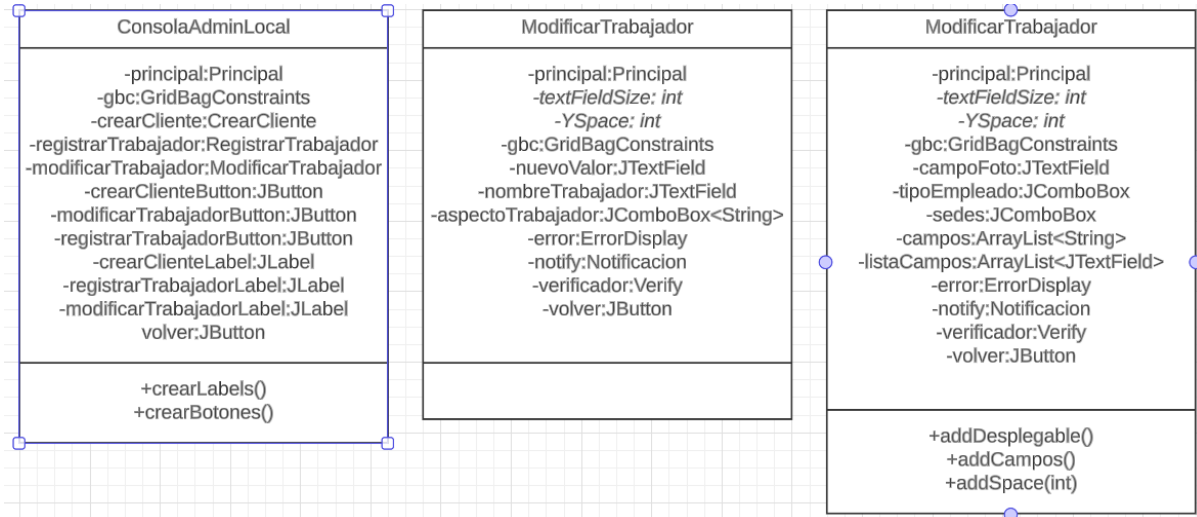
Paquete Admin General:



El paquete de administrador general cuenta con 8 requerimientos y una consola general. La consola general funciona como puente para poder abrir el panel correspondiente al requerimiento que se desea. Es vital mencionar que el requerimiento nuevo que era necesario implementar con respecto al gráfico de la cantidad de carros en una sede por un lapso de tiempo, se decidió implementar como un requerimiento propio del administrador general, ya que consideramos que esta información es relevante sólo para este tipo de usuario.

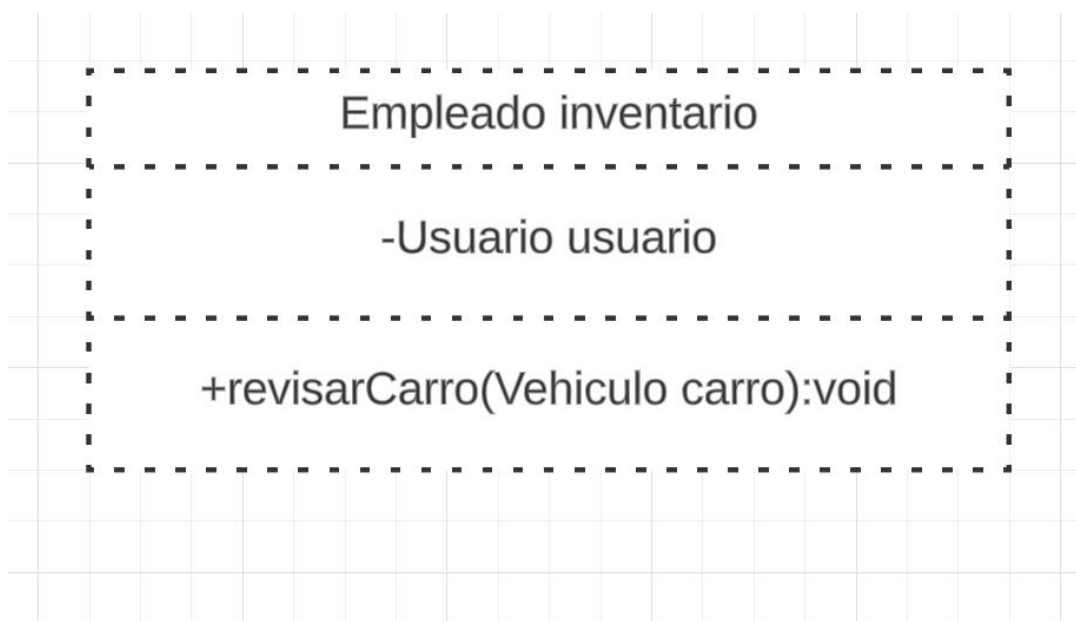
Ahora, dentro de los requerimientos individuales en el método del actionListener de cada uno, se instancia un objeto de tipo de Administrador General para acceder a sus métodos. Esta es la única conexión con el modelo de este paquete que no pasa por el permiso de principal.

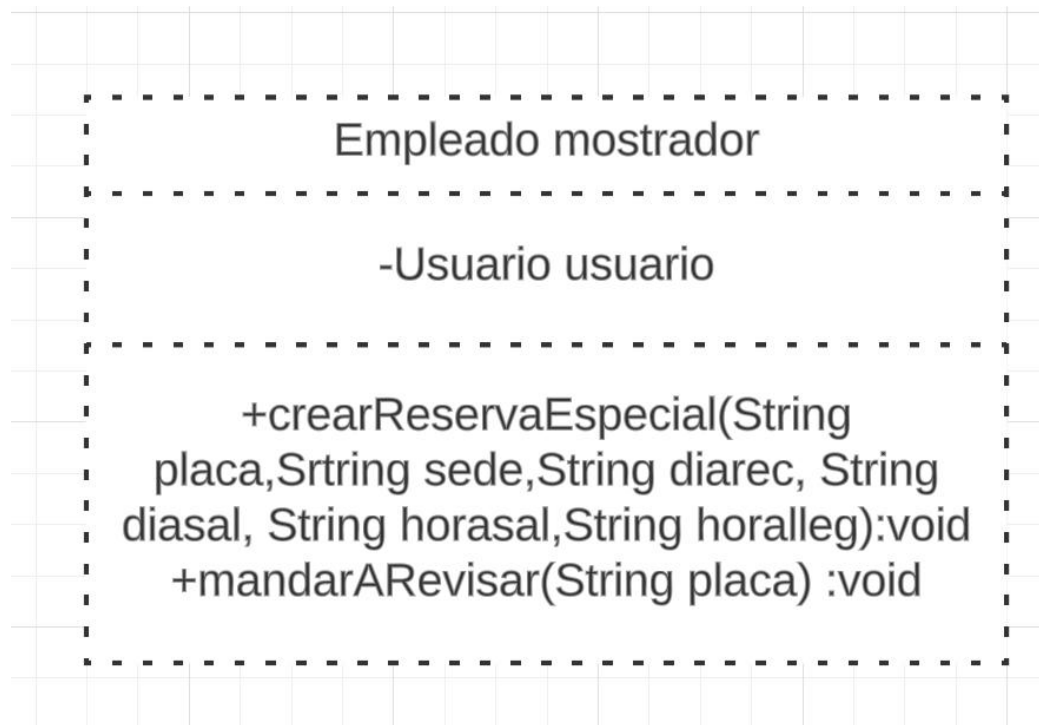
Paquete Admin Local:



Similarmente al paquete de administrador general, se instancia en los requerimientos individuales un objeto de tipo administrador local para comunicarse con el modelo y hacer la lógica correspondiente. Además es importante mencionar que para esta clase, se hace una verificación constante con la clase “verify” para restringir el poder que tiene este empleado. En otras palabras, el administrador local solo tiene poder con empleados y vehículos que están en su misma sede, de otro caso mostrará un mensaje que establece que a pesar de que la acción se pueda ejecutar, él no la puede hacer ya que está fuera de su jurisdicción.

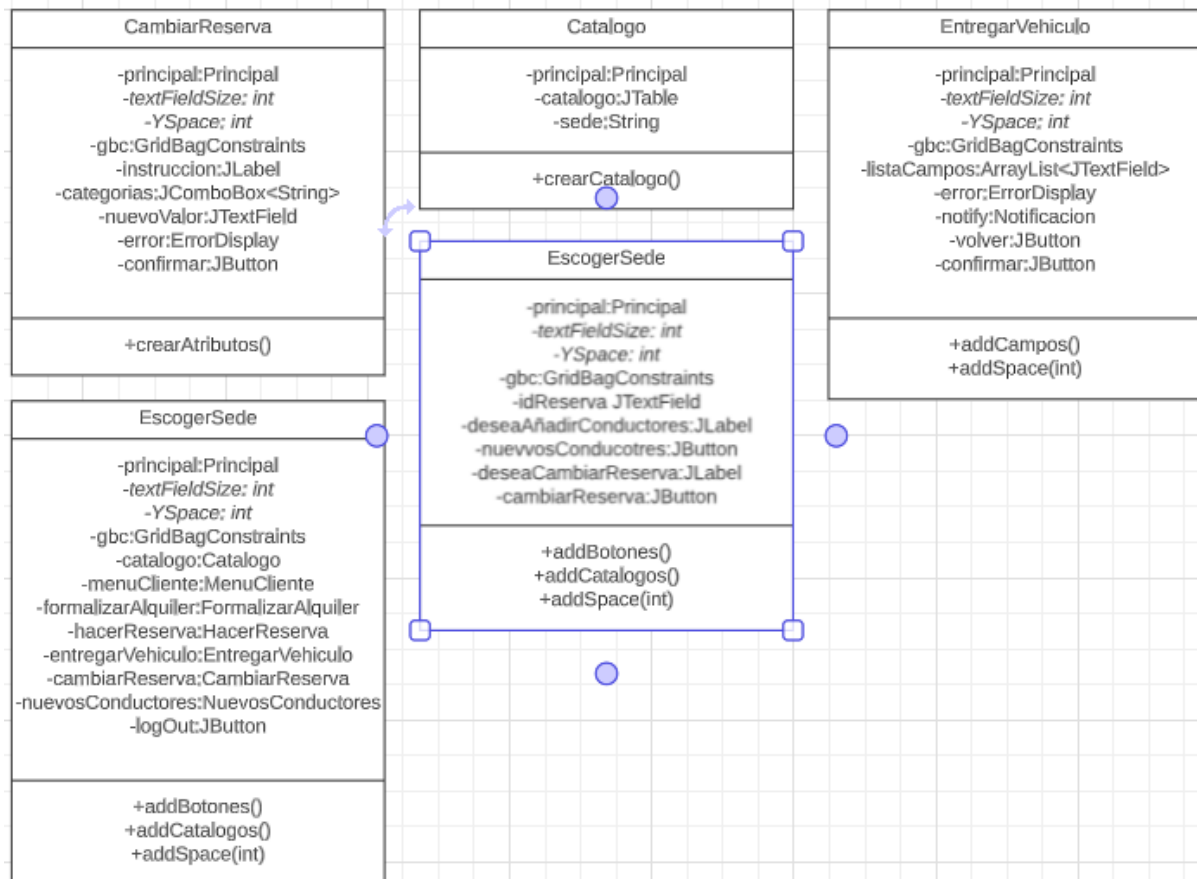
Paquete Empleado Inventario y Mostrador

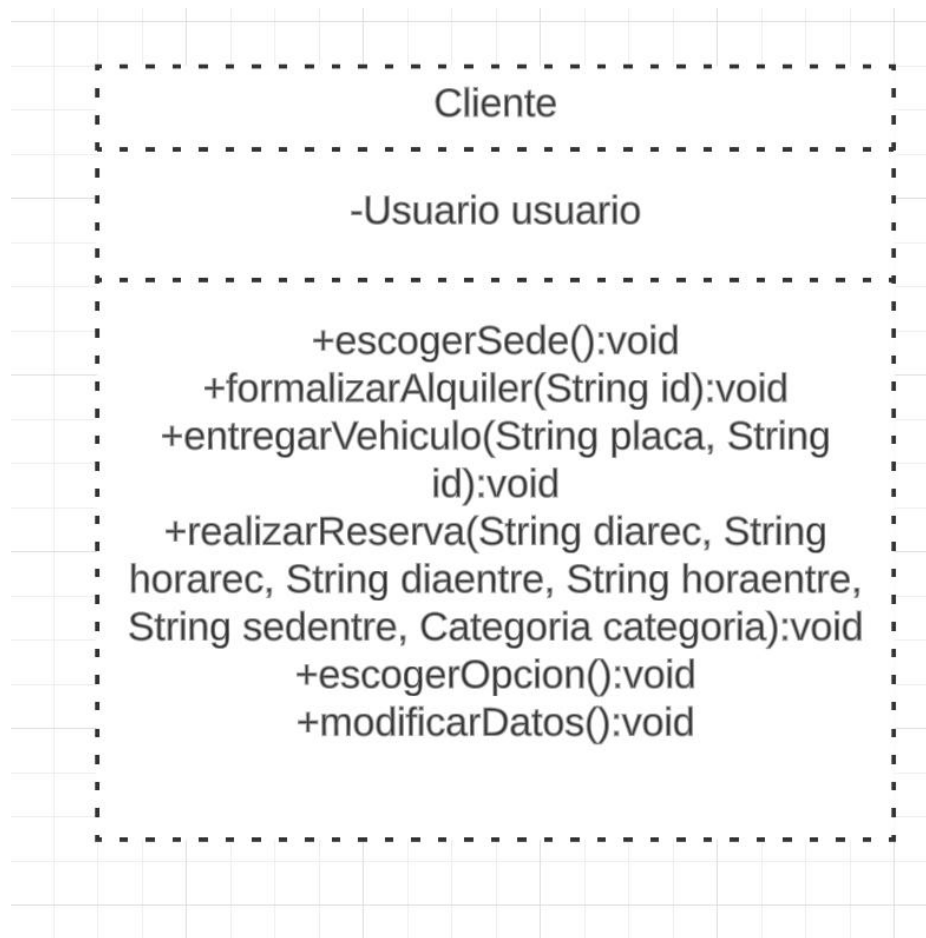




Los dos empleados también están restringidos por una sede, y a diferencia del administrador Local, la mayoría de sus acciones son pre hechas. Esto quiere decir, que no tienen el poder para escoger que carro van a mandar a revisar, o arreglar pero se les asigna uno automático dependiendo de donde están trabajando.

Paquete Cliente





Finalmente el paquete cliente tiene varios requerimientos, pero funcionan de manera similar ya que cada requerimiento instancia un objeto de tipo Cliente que se comunica con el modelo para realizar las acciones correspondientes, sin embargo, es importante mencionar que la restricción de sede también aplica en este caso puesto que si se hace una reserva en una sede, no se puede formalizar un alquiler en otra sede ya que esto no tendría sentido. Para verificar en qué sede se ubica la reserva, el paquete se comunica con la clase “principal” para poder saber en qué momento está.

Trade Offs:

Dentro de los trade offs que se tomaron, se estableció que principal sea una clase que se comunica con frecuencia con el modelo, y los demás paneles tienen que pasar por principal para pedir la información pertinente. Por lo que se vuelve una clase muy pesada lo cual puede afectar la facilidad de incluir nuevas funcionalidades, o si una clase quiere pedir un aspecto muy específico.