

Diseño

1. Contexto del Problema:

Para la interfaz es necesario tener en cuenta que van a haber distintos usuarios con distintas funcionalidades. No todos los usuarios van a tener el mismo acceso a la información que otros. Por cada usuario que haya es necesario tener su propia interfaz.

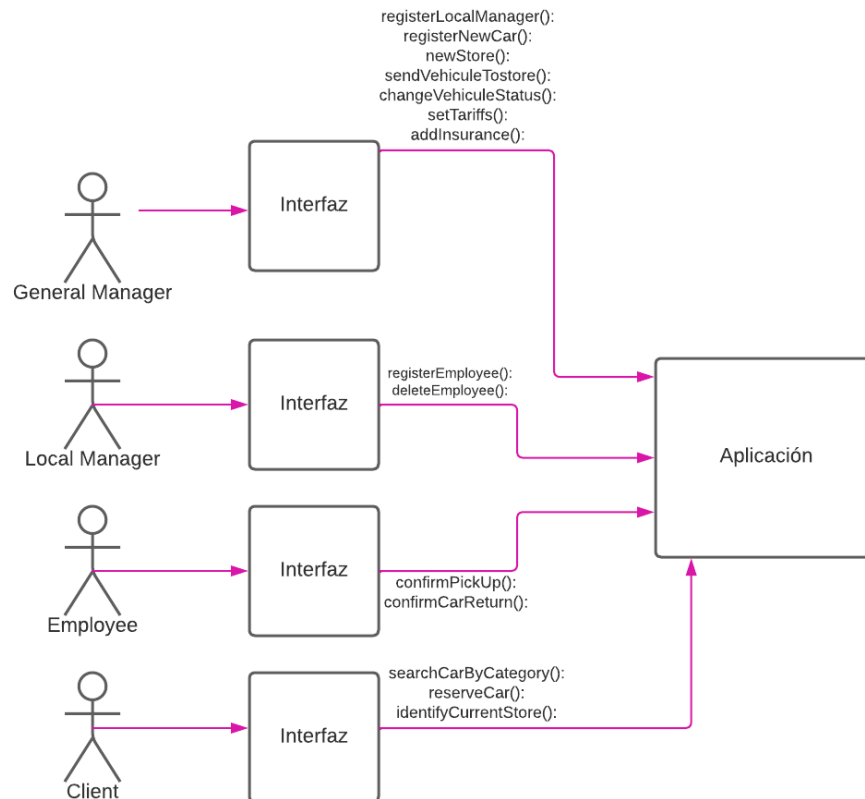


Figura 1: Definición del contexto del problema

Como se puede ver en el diagrama van a haber 4 usuarios distintos: General Manager, Local Manager, Employee y Client. La interfaz interaccionara con la aplicación de la misma manera sin importar el usuario. Y realizará los requerimientos hechos por los usuarios.

Diagrama de Alto Nivel (1^{era} iteración):

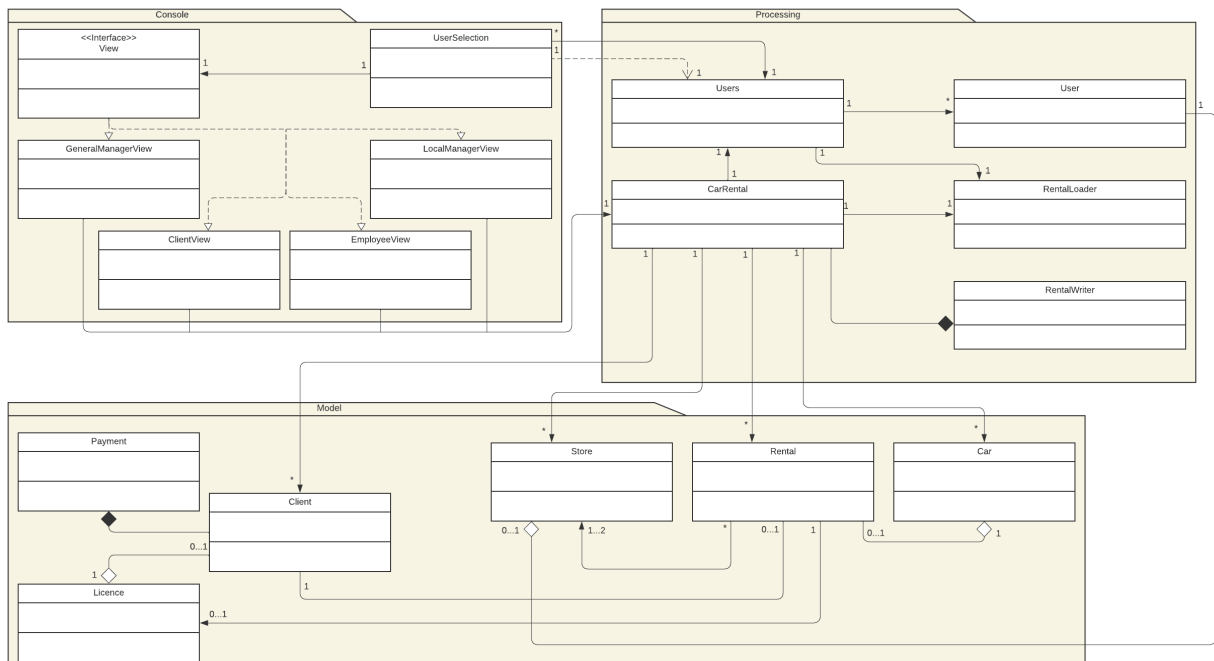
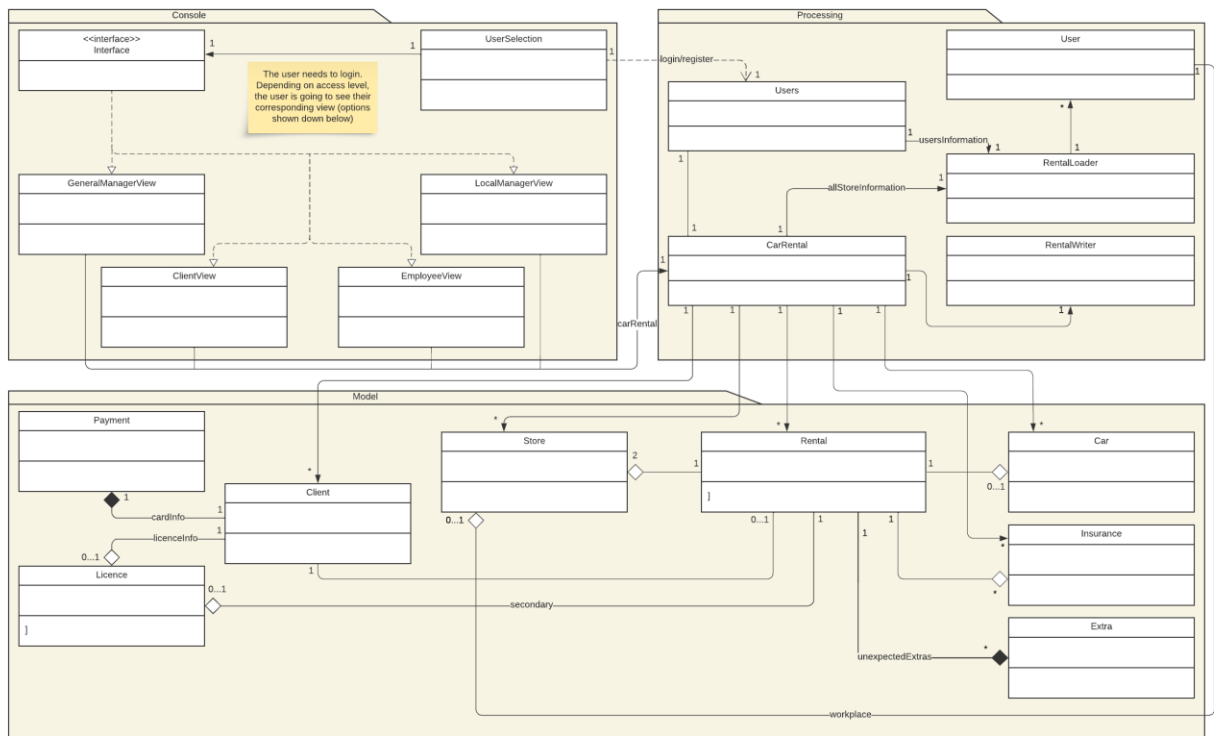


Diagrama de Alto Nivel (2^{da} iteración):



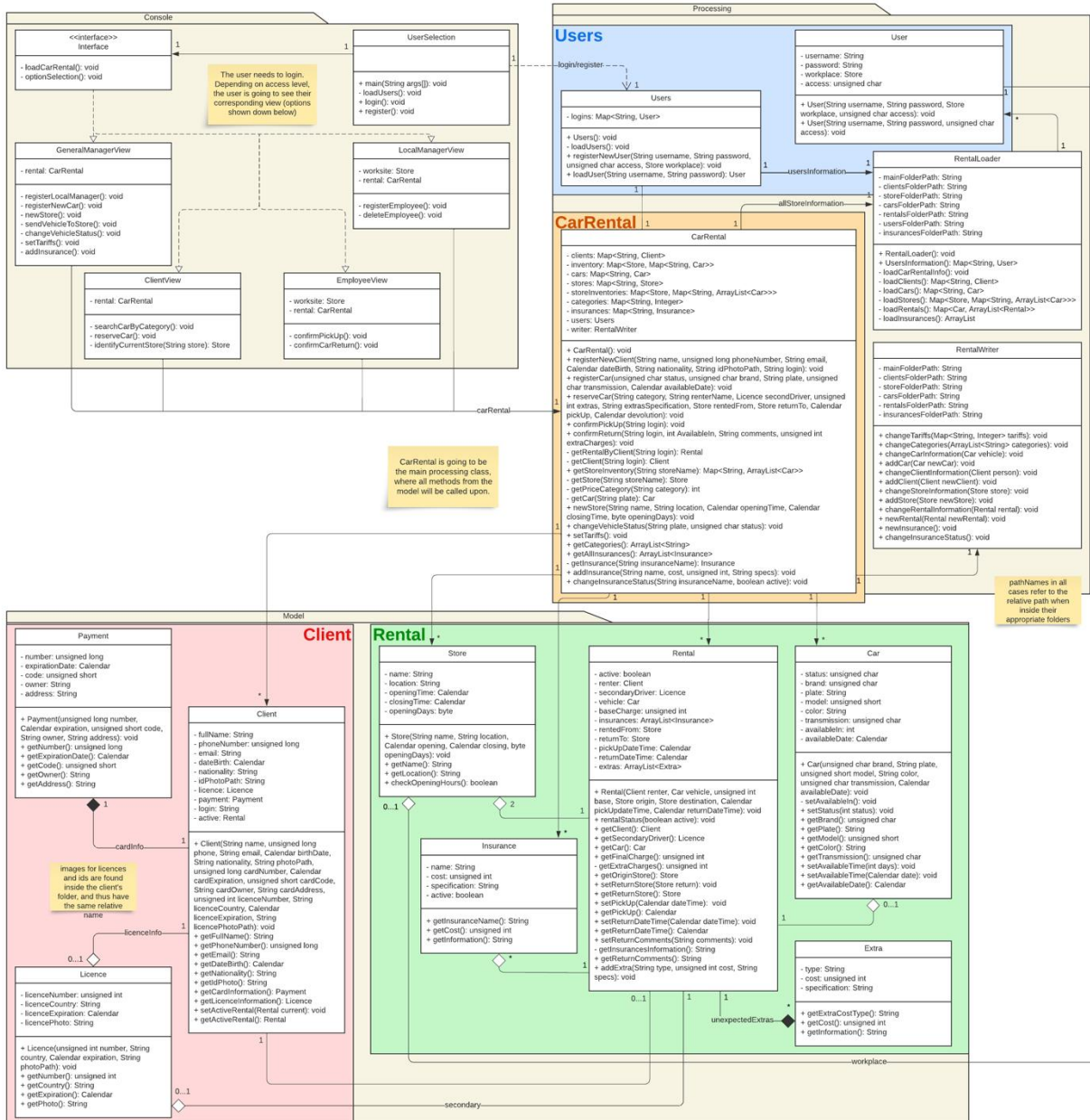
En primer lugar, desde la consola, en la interfaz se puede ver como se generan relaciones de realización con las clases asociadas a la vista de cada uno de los tipos de usuarios que usan la aplicación. Esta relación existe debido a que cada una de estas clases es la encargada de mostrar la información a la que su usuario correspondiente tiene acceso. Por ello, la selección de usuario

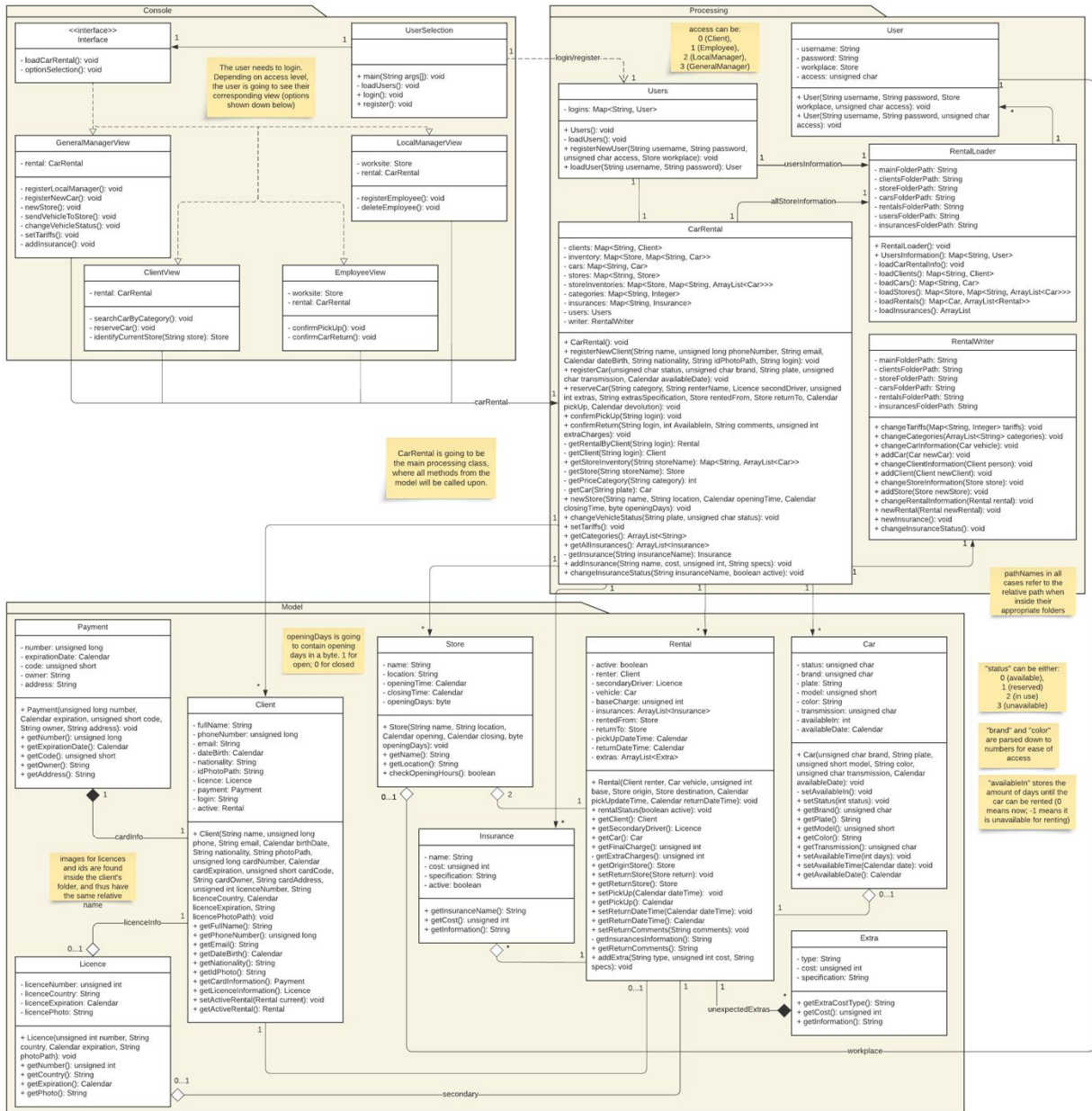
tiene una conexión directa con la interfaz, pues es quien le dice a la interfaz el tipo de usuario que ha iniciado sesión. Al seguir por la selección de usuario existe una nueva relación de realización con Users que tiene relación directa con RentalLoader, donde se guarda la información de tanto nuevos usuarios como existentes. Por ello, RentalLoader tiene relación directa con User, pues dicha clase es la responsable de definir todos los objetos User con su respectiva información.

Regresando a la vista por un momento, las 4 vistas para General Manager, Local Manager, cliente y empleado tienen relación directa con CarRental en el procesamiento. Esto porque CarRental contiene toda la información de la compañía y por ello es relevante que de ahí cada vista de usuario tenga definidos los atributos a los que puede tener acceso, pues el Local Manager solo debería poder ver los aspectos relacionados a su sede mientras el General Manager podría tener acceso a todo. Por esta misma razón CarRental tiene relación directa con RentalLoader y Users. Adicionalmente, también tiene relación con RentalWriter, pues mediante esta clase, los usuarios con la jerarquía necesaria pueden realizar ajustes sobre la compañía.

CarRental establece las conexiones con varias de las clases del modelo, dado que la compañía justamente maneja un sistema de rentas con distintas tiendas y vehículos, con clientes y seguros asociados en cada caso. Por ello, viendo la clase Rental, esta tiene relaciones de agregación con una tienda, un seguro, una licencia y un vehículo. Dicha relación de agregación existe dado que la renta de cualquier carro hecha en cualquier tienda finaliza, lo que quiere decir que, aunque esta desaparezca, los objetos Store, Insurance, Licence y Car seguirán existiendo. A su vez tiene relación de composición con los costos Extra, pues estos si dependen de una renta existente para seguir existiendo, dado que surgen a causa del propio alquiler. La clase cliente tiene una relación de agregación con la licencia, dado que esta no depende únicamente del cliente para poder existir, a causa de que pueden haber licencias secundarias para otros conductores del vehículo rentado. Sin embargo, el cliente tiene una relación de composición con la clase Payment, pues la información de pago del cliente depende justamente de él para existir. Finalmente, regresando a la clase User en el procesamiento, esta tiene relación directa con Stores en el modelo, pues los usuarios pueden estar asignados como empleados a una tienda específica.

UML Completo (con y sin divisiones generales):





Responsabilidades:

#	Responsabilidad	Componente
1	Seleccionar el usuario para acceder a la aplicación	Consola
2	Mostrar vistas por usuario	
3	Cargar usuario	Processing\Users
4	Crear nuevo usuario	Processing\CarRental
5	Registrar nuevo cliente	
6	Registrar un vehículo	
7	Confirmar recoger un vehículo	
8	Confirmar devolver un vehículo	
9	Encontrar la renta hecha por un cliente específico	
10	Encontrar un cliente	
11	Conseguir el inventario de vehículos de una tienda	
12	Encontrar una tienda por su nombre	
13	Conseguir el precio de una categoría específica	
14	Encontrar un vehículo por su placa	
15	Crear una nueva tienda	
16	Cambiar el estado de un vehículo dado su placa	
17	Definir la tarifa de un servicio	
18	Obtener todas las categorías existentes	
19	Obtener todos los seguros existentes	
20	Conseguir el seguro dado su nombre	
21	Añadir un nuevo seguro	
22	Cambiar el estado de un seguro dado su nombre	
23	Cargar archivos de: información de usuarios, clientes, vehículos, tiendas, rentas hechas y seguros.	Processing

24	Escribir o reescribir la información dentro de los archivos según cambios	
25	Al hacer una renta se asocia el cliente que la hace con su información pertinente: nombre, teléfono, correo, fecha de nacimiento, nacionalidad, licencia, log in y estado de la renta	Model\Client
26	Se asocia la renta activa con el cliente	
27	Al cliente se le agrega información completa de su licencia: número, país, fecha de vencimiento y foto	
28	La información del pago del cliente (datos de la tarjeta): número, fecha de vencimiento, código, propietario, dirección	
	Se crea la renta de un vehículo: estado, cliente, conductor secundario en el caso de haberlo, vehículo, cargo base, seguro, tienda de donde se rentó, tienda donde se devuelve, fecha para recogerlo, fecha para devolverlo y cargos extras en el caso de haberlos.	Model\Rental
	A la renta se le agrega la información completa del vehículo: estado, marca, placa, modelo, color, transmisión, disponibilidad en cuantos días, disponibilidad en fecha.	
	A la renta se le agrega la información completa del seguro: nombre, costo, especificaciones y estado.	

	Para saber el vehiculo de donde se renta y donde se devuelve se le agrega la información completa de dicha tienda: nombre, ubicación, hora de apertura, hora de cierre, días en los que abre.	
	Se definen los cargos extras que se llegan hacer: tipo, costo, especificación.	