Proyecto 2 Entrega 2

Ana María Sánchez Mejía –202013587

David Leonardo Manrique Lesmes –201913129

María Paula Nizo Vega – 202213902

Tabla de contenido

[Contexto del Problema 1](#_Toc377787225)

[Diagrama de clases de diseño 2](#_Toc1897664852)

[Métodos 2](#_Toc430404269)

[Instrucciones 3](#_Toc720079067)

[Decisiones de Diseño 5](#_Toc26784407)

## Contexto del Problema

Se realizará una aplicación que permita al usuario poder llevar a cabo de forma sencilla una reserva de un automóvil en la ciudad, asimismo debe posibilitar que los administradores o empleados de la empresa de alquiler manejen la información y tengan acceso a las modificaciones que dependerán de sus responsabilidades. A continuación, se presentan las principales características del contexto y sus actuales funciones agregadas:

Lista funciones:

Por consola:

1. PanelFondoGeneral
2. PanelOpcionesAdminGeneral
3. PanelPrincipal
4. PanelRegistroNuevoUsuario
5. RegistrarCategoria
6. RegistrarEliminarSede
7. RegistrarEliminarSeguro
8. Ventana

Por modelo:

1. Administrador
2. Categoria
3. ClienteEspecial
4. ClienteNormal
5. Conductor
6. Empleado
7. EmpresaVehiculos
8. Inventario
9. Reserva
10. Sede
11. Seguro
12. Tarjeta
13. TrasladoReserva
14. Usuario
15. Vehiculo

## Diagrama de clases de diseño

Los diagramas que se llevaron a cabo en el sistema de interfaz en el proyecto se verá como un archivo adjunto.

## Métodos de algunas de las clases

Por consola:

* PanelFondoGeneral:
  + PanelFondoGeneral(ventanaPrincipal: Ventana, panelCentro: JPanel)
* PanelOpcionesAdminGeneral:
  + PanelOpcionesAdminGeneral(empresa: EmpresaVehiculos) throws IOException
* PanelPrincipal:
  + PanelPrincipal(ventanaPrincipal: Ventana)
  + setUsuario(user: String): void
  + setContrasena(contrasena: String): void
  + actionPerformed(pEvento: ActionEvent): void
* PanelRegistroNuevoUsuario:
  + PanelRegistroNuevoUsuario(empresa: EmpresaVehiculos)
  + setLabel(palabra: String): JLabel
  + Métodos de acceso y modificación para varios atributos.
* RegistrarCategoria:
  + RegistrarCategoria(empresa: EmpresaVehiculos)
  + setNombreCategoria(text: String): void
  + setLabel(palabra: String): JLabel
  + Métodos de acceso y modificación para varios atributos.
* RegistrarEliminarSede:
  + RegistrarEliminarSede(empresa: EmpresaVehiculos)
  + Métodos de acceso y modificación para varios atributos.
  + setLabel(palabra: String): JLabel
  + actionPerformed(e: ActionEvent): void
* RegistrarEliminarSeguro:
  + RegistrarEliminarSeguro(empresa: EmpresaVehiculos)
  + Métodos de acceso y modificación para varios atributos.
  + setLabel(palabra: String): JLabel
  + actionPerformed(e: ActionEvent): void
* Ventana:
  + panelPrincipal: PanelPrincipal
  + height: int
  + width: int
  + userAdmin: Usuario
  + empresa: EmpresaVehiculos
  + Ventana (h: int, w: int)
  + getHeight (): int
  + getWidth (): int
  + getEmpresa (): EmpresaVehiculos
  + main (args: String[]): void

## Instrucciones

La aplicación cuenta con una consola donde el administrador y el usuario podrán interactuar con la aplicación, de esta forma al iniciar el programa se observarán las siguientes opciones:

Opciones disponible tipo usuario



El programa inicia en la ventana LoginScreen que es la ventana de inicio de sesión en la cual están los campos de texto donde se pueden poner la cuenta y contraseña, en caso de que el usuario no tenga cuenta puede registrarse.

Usuarios que se pueden usar para el desarrollo del programa:

Admón. Sede: msrondon password: 5678

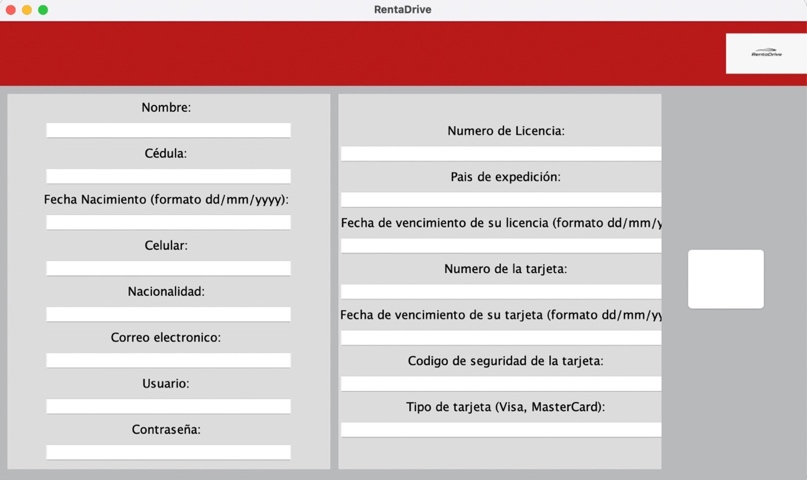
Cliente: dlesmes 1234

Empleado recepción: savillamil 1245

Admón. General: dlmanrique Password: 1234

Limpiador que usaba es: angarcia 0012

Opción crear una cuenta

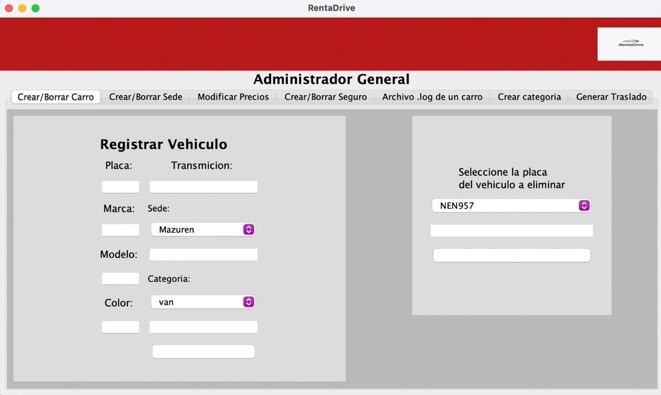


Esta es la información que se tiene en cuenta para que los usuarios que desean realizar una reserva con la empresa.

Luego se decidió mostrar los requerimientos que tiene cada uno de los usuarios se mostraran a continuación hay algunas de las instrucciones que ya cuentan con un formato, es necesario que se siga con este formato para que la aplicación pueda realizar todos los requerimientos que tiene cada uno y que cumpla con su funcionalidad.

Por último, si se desea salir de la sesión se debe oprimir el botón rojo y así ya se puede salir de la aplicación

Opción Admón. general



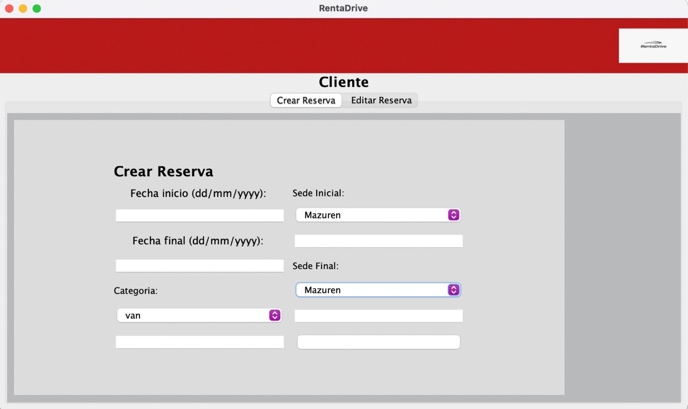
Opción Admón. sede



Opción Recepción



Opción cliente



Empleado Limpiador



## 

## Decisiones de Diseño

* Se tomó la decisión de que la clase Aplicación sea un <> para las opciones del menú de los usuarios. Esta clase tiene la función de ejecutar los procesos de autenticación, inicio de sesión, registro en la plataforma y un menú de opciones para la categoría del usuario registrado.
* Se tomó la decisión de que la clase Sede almacene un inventario de los carros que tiene cada una de las sedes.
* Se tomó la decisión de definir la clase Aplicación como un <> ya que, a través de la interacción por consola de los menús, se delega a las otras clases que están dentro del paquete de la lógica que realicen sus funciones.
* Se tiene en cuenta que para cerrar la sesión de la aplicación se deben cerrar por medio de la de las ventanas después de hacer los arreglos que cada usuario desee y al cerrarlo se guardaran los cambios.
* Para poder acceder a la gráfica de ocupación el único que la puede visualizar va a ser el administrador general