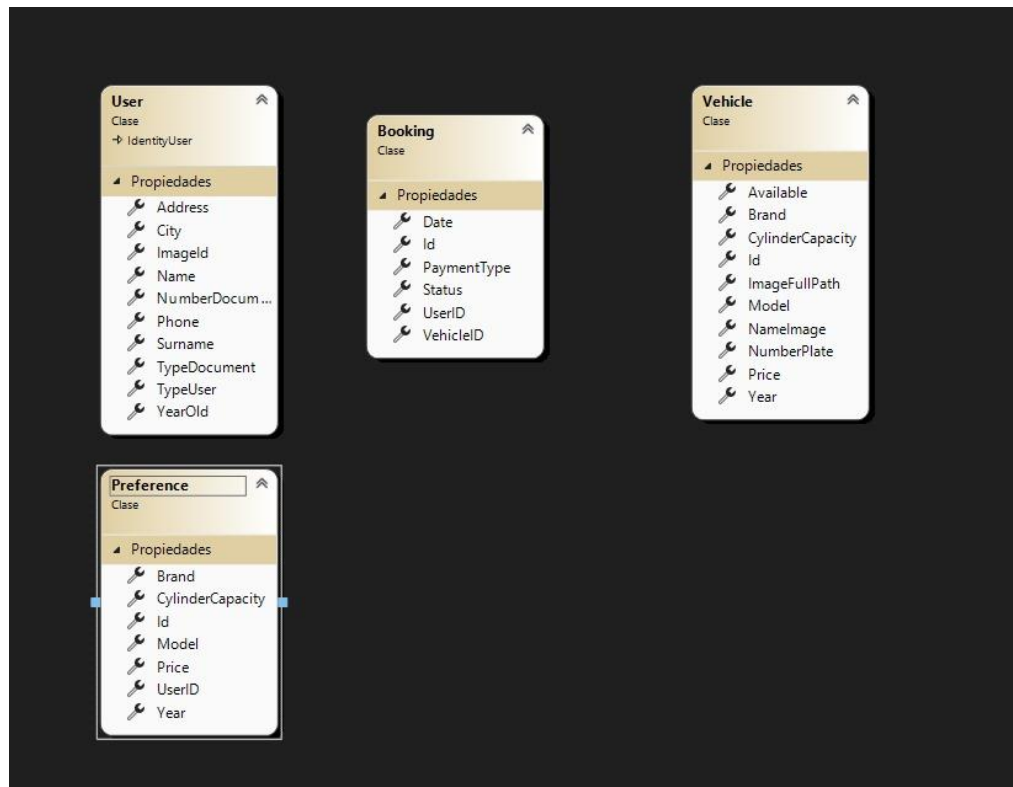


1. La empresa Miles Car Rental requiere el desarrollo de una aplicación para el manejo de la renta de sus vehículos. Inicialmente, necesitan que cuando un cliente realice una solicitud de un vehículo, la aplicación liste las diferentes opciones que ofrece la compañía y que pueda almacenar la información del cliente, sus preferencias, así como realizar reservas y finalizar la renta del vehículo. Dado que se encuentra en una fase exploratoria, se solicitan los siguientes entregables:
 - A. Diagrama de clases.



- B. Pseudocódigo que describa el funcionamiento de la aplicación.

Clase Cliente:

Atributos:

- id
- tipo documento
- numero documento
- nombres
- apellidos
- ciudad
- dirección
- edad
- número de teléfono
- correo electrónico
- contraseña

Clase Vehículo:

Atributos:

- id
- placa
- marca
- modelo
- año
- cilindraje
- imagen
- disponibilidad

Clase Reserva:

Atributos:

- cliente id
- vehículo id
- fecha
- tipo de pago
- estado

Clase Preferencia:

Atributos:

- id
- cliente id
- marca
- modelo
- año
- cilindraje
- precio

Métodos:

- agregarCliente(cliente)
- agregarVehículo(vehículo)
- agregarReserva(cliente, vehículo, fecha)
- agregarPreferencia(reserva)
- ListarPagosPorTarjeta()
- ListarPagosAlDestino()
- ListarReservasActivasyCanceladas()
- ListarPersonasMayor40DeBogotáYReservas ()

Algoritmo principal:

1. Inicializar la aplicación de renta
2. Mientras el usuario del tipo user no decida salir:
 - a. Mostrar menú de opciones:
 1. Registrarse
 2. Iniciar Sesión
 3. Hacer reserva
 4. Salir
 - b. Leer la opción ingresada por el usuario
 - c. Según la opción seleccionada:
 - Caso 1:
 1. Ingresar los datos para registrarse
 - Caso 2:
 1. Solicitar los datos para iniciar sesión
 2. Listar vehiculos
 - Caso 3:
 1. Seleccionar vehículos ver los detalle y solicitar datos de reserva
 3. Listar Reservas
 - Caso 4:
 1. Salir del programa
3. Mientras el usuario del tipo admin no decida salir:
 - a. Mostrar menú de opciones:
 1. Registrarse
 2. Iniciar Sesión
 3. Agregar Vehiculos
 4. Listar Reservas activas y canceladas
 5. Listar reservas con pago de tarjeta
 6. Listar reservas con pago en destino
 7. Listar reservas con personas mayores de 40 años, de Bogotá y que tengan reserva
 8. Salir
 - b. Leer la opción ingresada por el usuario
 - c. Según la opción seleccionada:
 - Caso 1:
 1. Ingresar los datos para registrarse
 - Caso 2:
 1. Solicitar los datos para iniciar sesión
 2. Listar vehículos
 - Caso 3:
 1. Solicitar los datos del vehículo para agregar
 3. Listar vehículos

- Caso 4:
 1. Lista Reservas activas y canceladas
- Caso 5:
 1. Lista reservas con pago de tarjeta
- Caso 6:
 1. Lista reservas con pago en destino
- Caso 7:
 1. Listar reservas con personas mayores de 40 años, de Bogotá y que tengan reserva
- Caso 8:
 1. Salir del programa

2. Proceso para la configuración de la aplicación

- a. Descargar del siguiente repositorio git el código fuente <https://github.com/juliannova-web/PruebaMilesCarRenta.git>
- b. Cambiar la configuración en los appsettings al server y contraseña del equipo donde esta el servidor de MYSQL configurado

```
"ConnectionStrings": {
  "LocalConnection": "Server=localhost;Database=MilesCarRenta; user=root; password=julian.novaC360;"
}
```

- c. Ejecutar en la consola de administrador de paquetes los comandos
 - Add-migration InitialDB
 - Update-database

Tener en cuenta que aparezca como proyecto inicial el que dice PruebaMilesCarRenta.Web

- d. La aplicación esta configurada que al ejecutar automáticamente cree unos usuarios que son:

Primero

- Nombre usuario: administrador_miles_car@yopmail.com
- Contraseña: 123456
- Tipo de usuario: Admin

Segundo

- Nombre usuario: usuario_miles_car@yopmail.com
- Contraseña: 123456
- Tipo de usuario: User

También ya existen unos vehículos ingresados en el sistema para pruebas