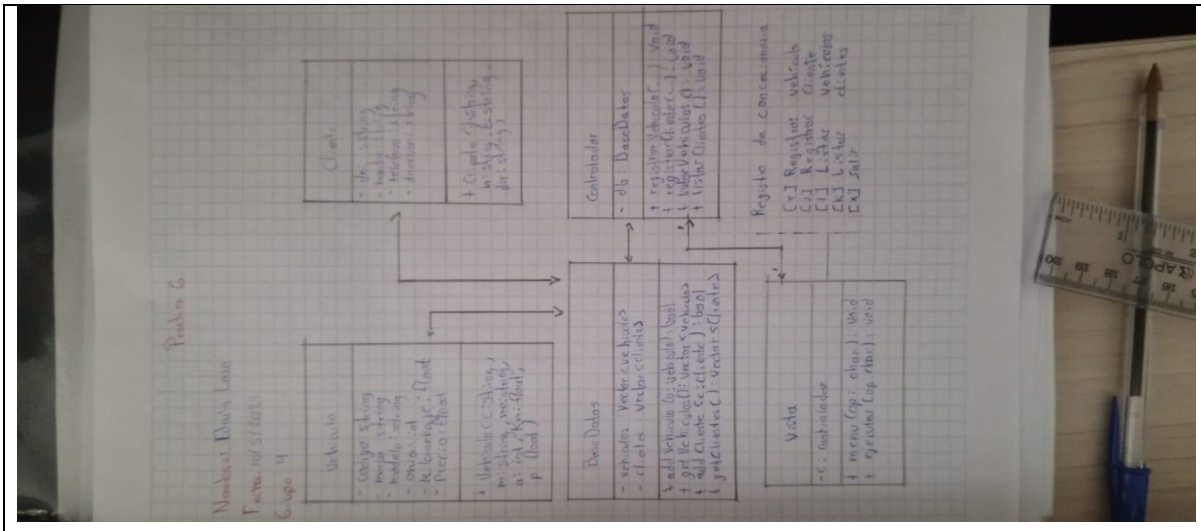
Guía de Práctica de aplicación y experimentación de los aprendizajes de la Universidad Politécnica Salesiana	
Carrera:	**CIENCIAS DE LA COMPUTACIÓN**
Nivel:	**2do**
Asignatura:	**PROGRAMACIÓN APLICADA**
Desarrollado por:	**Edwinlazo**
Grupo:	**4**
Resultados de Aprendizaje:	**Programar utilizando el paradigma Modelo – Vista – Controlador**
Indicador de logro:	**Se realizo el programa sin inconvenientes**
Práctica/Deber Número:	**5**
Horas Dedicadas:	**1 hora**

DESCRIPCIÓN DE LA PRÁCTICA:

Analice las clases creadas en la práctica e identifique cuales son los atributos y métodos, y construya el diagrama de clases con sus correspondientes clases, si es un algoritmo coloque su flujograma correspondiente.



1.1 En base al diagrama de clases generado/algoritmo, construir la aplicación

CLASE:logica

```
#include <iostream>
#include "../headers/logica.h"

void Logica::agregarVehiculo(std::string c, std::string ma, std::string mo, int a, float
km, float p) {
   Vehiculo v(c, ma, mo, a, km, p);
   vehiculos.push_back(v);
   std::cout << "Vehiculo agregado correctamente.\n";
}

void Logica::agregarCliente(std::string d, std::string n, std::string t, std::string dir) {
   Cliente c(d, n, t, dir);
   clientes.push_back(c);
   std::cout << "Cliente agregado correctamente.\n";
}

void Logica::listarVehiculos() {
   std::cout << "\n--- Lista de Vehiculos ---\n";
   for (const auto &v : vehiculos) {
      std::cout << v.codigo << " - " << v.marca << " " << v.modelo
            << " (" << v.anio << ") - " << v.kilometraje
            << " km - $" << v.precio << "\n";
   }
}

void Logica::listarClientes() {
   std::cout << "\n--- Lista de Clientes ---\n";
   for (const auto &c : clientes) {
      std::cout << c.dni << " - " << c.nombre << ", "
            << c.telefono << ", " << c.direccion << "\n";
   }
}
```

CLASE: vehhiculo

```
#ifndef VEHICULO_H
#define VEHICULO_H

#include <string>

class Vehiculo {
public:
   std::string codigo;
   std::string marca;
   std::string modelo;
   int anio;
   float kilometraje;
```

```
    float precio;

    Vehiculo(std::string c, std::string ma, std::string mo, int a, float km, float p);
};

#endif
```

CLASE: vista

```
#include <iostream>
#include <limits>
#include "../headers/vista.h"

void Vista::menu() {
    char opcion;
    do {
        std::cout << "\n=== REGISTRO DE CONCESIONARIA ===\n";
        std::cout << "[r] Registrar Vehiculo\n";
        std::cout << "[j] Registrar Cliente\n";
        std::cout << "[l] Listar Vehiculos\n";
        std::cout << "[k] Listar Clientes\n";
        std::cout << "[x] Salir\n";
        std::cout << "Seleccione una opcion: ";
        std::cin >> opcion;
        ejecutar(opcion);
    } while (opcion != 'x');
}

void Vista::ejecutar(char opcion) {
    std::string c, ma, mo, d, n, t, dir;
    int a;
    float km, p;

    switch(opcion) {
        case 'r':
            std::cout << "Codigo: "; std::cin >> c;
            std::cout << "Marca: "; std::cin >> ma;
            std::cout << "Modelo: "; std::cin >> mo;
            std::cout << "Anio: "; std::cin >> a;
            std::cout << "Kilometraje: "; std::cin >> km;
            std::cout << "Precio: "; std::cin >> p;
            logica.agregarVehiculo(c, ma, mo, a, km, p);
            break;
        case 'j':
            std::cout << "DNI: "; std::cin >> d;
            std::cout << "Nombre: ";
std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
            std::getline(std::cin, n);
            std::cout << "Telefono: "; std::getline(std::cin, t);
            std::cout << "Direccion: "; std::getline(std::cin, dir);
```

```
                logica.agregarCliente(d, n, t, dir);
                break;
            case 'l':
                logica.listarVehiculos();
                break;
            case 'k':
                logica.listarClientes();
                break;
            case 'x':
                std::cout << "Saliendo del sistema...\n";
                break;
            default:
                std::cout << "Opcion no valida.\n";
        }
}
```

1.2 Generar una clase main que cumpla con los siguientes requisitos planteados en el problema:

Método: main	Capturas de Pantalla con cada opción ejecutada

`#include "headers/vista.h"`
 	
`int main() {`
`  Vista vista;`
`  vista.menu();`
`  return 0;`
`}`