

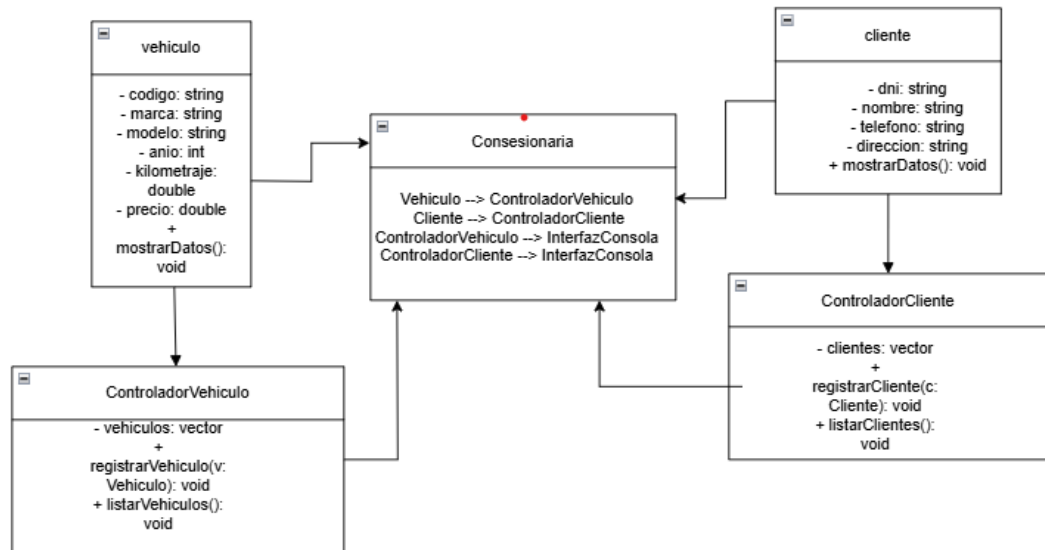
Guía de Práctica de aplicación y experimentación de los aprendizajes de la Universidad Politécnica Salesiana

Carrera:	CIENCIAS DE LA COMPUTACIÓN
Nivel:	2er
Asignatura:	PROGRAMACIÓN ORIENTADA A OBJETOS
Desarrollado por:	JORDY ROBLERO
Grupo:	1
Resultados de Aprendizaje:	
Indicador de logro:	
Práctica/Deber Número:	
Horas Dedicadas:	

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.

//Insertar el diagrama de clases / o flujograma



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

CLASE: cliente.cpp

```
#include "Cliente.h"
#include <iostream>

Cliente::Cliente(string d, string n, string t, string dir)
    : dni(d), nombre(n), telefono(t), direccion(dir) {}

string Cliente::getDNI() const {
    return dni;
}

void Cliente::mostrarDatos() const {
    cout << "DNI: " << dni << ", Nombre: " << nombre << ", Teléfono: " << telefono
        << ", Dirección: " << direccion << endl;
}
```

CLASE: cliente.h

```
//Insertar el código de la clase
#ifndef CLIENTE_H
#define CLIENTE_H

#include <string>
using namespace std;
class Cliente {
private:
    string dni, nombre, telefono, direccion;

public:
    Cliente(string d, string n, string t, string dir);
    string getDNI() const;
    void mostrarDatos() const;
};

#endif
```

CLASE: controladorCliente.cpp

```
//Insertar el código de la clase
#include "ControladorCliente.h"

void ControladorCliente::registrarCliente(const Cliente& c) {
    clientes.push_back(c);
}

void ControladorCliente::listarClientes() const {
    for (const auto& c : clientes) {
        c.mostrarDatos();
    }
}
```

CLASE: controladorCliente.h
<pre>//Insertar el código de la clase #ifndef CONTROLADOR_CLIENTE_H #define CONTROLADOR_CLIENTE_H #include "Cliente.h" #include <vector> class ControladorCliente { private: std::vector<Cliente> clientes; public: void registrarCliente(const Cliente& c); void listarClientes() const; }; #endif</pre>
CLASE: controladorVehiculo.cpp
<pre>//Insertar el código de la clase #include "ControladorVehiculo.h" void ControladorVehiculo::registrarVehiculo(const Vehiculo& v) { vehiculos.push_back(v); } void ControladorVehiculo::listarVehiculos() const { for (const auto& v : vehiculos) { v.mostrarDatos(); } }</pre>
CLASE: controladorVehiculo.h
<pre>//Insertar el código de la clase #ifndef CONTROLADOR_VEHICULO_H #define CONTROLADOR_VEHICULO_H #include "Vehiculo.h" #include <vector> class ControladorVehiculo { private: std::vector<Vehiculo> vehiculos; public: void registrarVehiculo(const Vehiculo& v); void listarVehiculos() const; }; #endif</pre>
CLASE: vehiculo.cpp
<pre>//Insertar el código de la clase #include "Vehiculo.h" #include <iostream></pre>

```
Vehiculo::Vehiculo(string cod, string mar, string mod, int an, double km, double pr)
: codigo(cod), marca(mar), modelo(mod), anio(an), kilometraje(km), precio(pr) {}
```

```
string Vehiculo::getCodigo() const {
    return codigo;
}
```

```
void Vehiculo::mostrarDatos() const {
    cout << "Código: " << codigo << ", Marca: " << marca << ", Modelo: " << modelo
        << ", Año: " << anio << ", Km: " << kilometraje << ", Precio: $" << precio <<
endl;
}
```

CLASE: register.h

//Insertar el código de la clase

```
#ifndef VEHICULO_H
#define VEHICULO_H
```

```
#include <string>
using namespace std;
class Vehiculo {
private:
    string codigo, marca, modelo;
    int anio;
    double kilometraje, precio;
```

```
public:
    Vehiculo(string cod, string mar, string mod, int an, double km, double pr);
    string getCodigo() const;
    void mostrarDatos() const;
};
```

#endif

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

```
//Inserte el código de la clase ejecutable
#include <iostream>
#include <limits>
#include "Vehiculo.h"
#include "Cliente.h"
#include "ControladorVehiculo.h"
#include "ControladorCliente.h"
using namespace std;
void mostrarMenu() {
    cout << "\n--- Menu Principal ---\n"
        << "1. Registrar Vehiculo\n"
        << "2. Registrar Cliente\n"
        << "3. Consultar Datos\n"
        << "4. Salir\n"
        << "Seleccione una opcion: ";
}
void limpiarEntrada() {
    cin.clear();

    cin.ignore(numeric_limits<streamsize>::max(),
'\n');
}
int main() {
    ControladorVehiculo cv;
    ControladorCliente cc;
    int opcion;
    do {
        mostrarMenu();
        cin >> opcion;
        limpiarEntrada();
        if (opcion == 1) {
            string codigo, marca, modelo;
            int anio;
            double km, precio;
            cout << "Codigo: "; getline(cin, codigo);
            cout << "Marca: "; getline(cin, marca);
            cout << "Modelo: "; getline(cin,
modelo);
            cout << "Ano: "; cin >> anio;
            cout << "Kilometraje: "; cin >> km;
            cout << "Precio: "; cin >> precio;
            limpiarEntrada();
            Vehiculo v(codigo, marca, modelo,
anio, km, precio);
            cv.registrarVehiculo(v);
            cout << "Vehiculo registrado
 exitosamen te.\n";
        } else if (opcion == 2) {
```

```
--- Menu Principal ---
```

1. Registrar Vehiculo
2. Registrar Cliente
3. Consultar Datos
4. Salir

```
Seleccione una opcion: 1 •
```

```
Codigo: 111
```

```
Marca: Toyota
```

```
Modelo: Corolla
```

```
Ano: 2022
```

```
Kilometraje: 25000
```

```
Precio: 30.000
```

```
Vehiculo registrado exitosamen te.
```

```
--- Menu Principal ---
```

1. Registrar Vehiculo
2. Registrar Cliente
3. Consultar Datos
4. Salir

```
Seleccione una opcion: 2
```

```
DNI: 1755786942
```

```
Nombre: Jordy
```

```
Telefono: 0991050457
```

```
Direccion: la Ecua
```

```
Cliente registrado exitosamen te.
```

```
--- Menu Principal ---
```

1. Registrar Vehiculo
2. Registrar Cliente
3. Consultar Datos
4. Salir

```
Seleccione una opcion: 3
```

```
--- Vehiculos Registrados ---
```

```
Codigo: 111, Marca: Toyota, Modelo: Corolla, Año: 2022, Km: 25000, Precio: $30
```

```
--- Clientes Registrados ---
```

```
DNI: 1755786942, Nombre: Jordy, Telefono: 0991050457, Direccion: la Ecua
```

```
--- Menu Principal ---
```

1. Registrar Vehiculo
2. Registrar Cliente
3. Consultar Datos •
4. Salir

```
Seleccione una opcion: 4
```

```
Process exited with code: 0.Saliendo del sistema...
```

```

        string dni, nombre, telefono, direccion;
        cout << "DNI: "; getline(cin, dni);
        cout << "Nombre: "; getline(cin,
nombre);
        cout << "Telefono: "; getline(cin,
telefono);
        cout << "Direccion: "; getline(cin,
direccion);
        Cliente c(dni, nombre, telefono,
direccion);
        cc.registrarCliente(c);
        cout << "Cliente registrado
exitosamente.\n";
    } else if (opcion == 3) {
        cout << "\n--- Vehiculos Registrados ---
\n";
        cv.listarVehiculos();
        cout << "\n--- Clientes Registrados ---
\n";
        cc.listarClientes();
    }
} while (opcion != 4);
cout << "Saliendo del sistema...\n";
return 0;
}

```

Problemas detectados durante el desarrollo de la práctica/deber