



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA



Universidad de las Fuerzas Armadas

## “INFORME”

**Integrantes:**

1. Brayan Stehp Mendoza Márquez

**Curso:**

1323

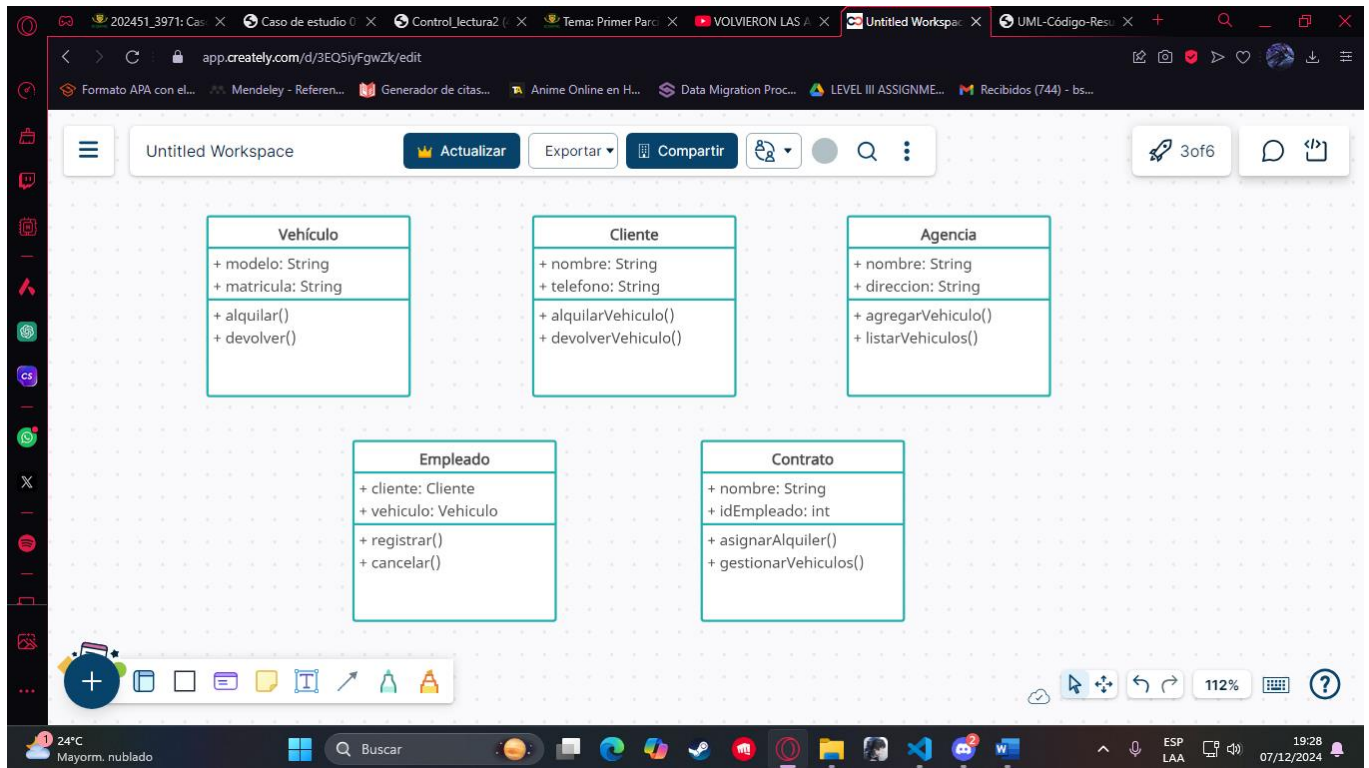
**Asignatura De Programación orientada a objetos**

**Docente:**

LUIS ENRIQUE JARAMILLO MONTAÑO

05 de diciembre de 2024

## 1.1 Creación del UML



Para crear los IMLS utilizamos la aplicación en línea “creately.com” que nos permite jugar con las figuras requeridas para hacer los objetos con sus respectivos métodos y atributos me pareció genial la app por que fue fácil de entender

## 1.2 Relaciones

Para realizar relaciones entre objetos he aprendido por diversas maneras, pero la que más me gusto fue la forma en la que me encelaron en el instituto en que estaba estudiando ya que hay me hicieron ver que casi cualquier cosa puede tener una relación y con eso aprendí bastante rápido un ejemplo rápido aparte del que estoy haciendo en este deber seria: **Persona ↔ YouTube** como seria esta relación seria así; **una persona ve videos en YouTube y YouTube es una aplicación de videos que permite ver videos a las personas.**

También vi un video en YouTube que ayuda a tender mejor como se deben de hacer las relaciones con objetos

<https://www.youtube.com/watch?v=XKk5k9OrAUg>

### 1.3 Código

En este código, trabajamos con las clases y relaciones basadas en el diagrama UML, usando ArrayList de las colecciones en Java. Esto nos permite manejar de forma dinámica los elementos como vehículos, clientes y contratos, haciendo que el programa sea más flexible y fácil de ampliar. Además, aplicamos los conceptos que vimos en clase sobre colecciones para organizar mejor los datos y resolver problemas prácticos. También utilizamos ideas del libro de Ingeniería de Software, lo que nos ayudó a diseñar el código de una manera más ordenada y orientada a objeto

```
deber.java > deber
1  import java.util.ArrayList;
2  import java.util.List;
3
4  // Clase Vehiculo
5  class Vehiculo {
6      public String modelo;
7      public String matricula;
8
9      public Vehiculo(String modelo, String matricula) {
10         this.modelo = modelo;
11         this.matricula = matricula;
12     }
13
14     public void alquilar() {
15         System.out.println("El vehículo con matrícula " + matricula + " ha sido alquilado.");
16     }
17
18     public void devolver() {
19         System.out.println("El vehículo con matrícula " + matricula + " ha sido devuelto.");
20     }
21 }
```

```
28  public Cliente(String nombre, String telefono) {
29      this.nombre = nombre;
30      this.telefono = telefono;
31  }
32
33  public void alquilarVehiculo() {
34      System.out.println("El cliente " + nombre + " ha alquilado un vehículo.");
35  }
36
37  public void devolverVehiculo() {
38      System.out.println("El cliente " + nombre + " ha devuelto un vehículo.");
39  }
40  }
41
42  // Clase Agencia
43  class Agencia {
44      public String nombre;
45      public String direccion;
46      private List<Vehiculo> vehiculos;
47
48      public Agencia(String nombre, String direccion) {
49          this.nombre = nombre;
50          this.direccion = direccion;
51          this.vehiculos = new ArrayList<>();
52      }
53  }
```

```

    public void agregarVehiculo(Vehiculo vehiculo) {
        vehiculos.add(vehiculo);
        System.out.println("Vehículo " + vehiculo.modelo + " agregado a la agencia.");
    }

    public void listarVehiculos() {
        System.out.println("Vehículos disponibles en la agencia " + nombre + ":");
        for (Vehiculo vehiculo : vehiculos) {
            System.out.println("- Modelo: " + vehiculo.modelo + ", Matrícula: " + vehiculo.matricula);
        }
    }
}

// Clase Empleado
class Empleado {
    public Cliente cliente;
    public Vehiculo vehiculo;

    public Empleado(Cliente cliente, Vehiculo vehiculo) {
        this.cliente = cliente;
        this.vehiculo = vehiculo;
    }

    public void registrar() {
        System.out.println("El cliente " + cliente.nombre + " ha registrado el alquiler del vehículo " + vehiculo.modelo + ".");
    }

    public void cancelar() {
        System.out.println("El cliente " + cliente.nombre + " ha cancelado el alquiler del vehículo " + vehiculo.modelo + ".");
    }
}

```

```

// Clase principal
public class deber {
    Run | Debug

    public static void main(String[] args) {
        // Crear objetos
        Vehiculo vehiculo1 = new Vehiculo(modelo:"Toyota Corolla", matricula:"ABC-123");
        Vehiculo vehiculo2 = new Vehiculo(modelo:"Honda Civic", matricula:"XYZ-456");
        Cliente cliente1 = new Cliente(nombre:"María López", telefono:"555-1234");
        Agencia agencia = new Agencia(nombre:"Agencia Central", direccion:"Calle Principal 123");

        // Operaciones
        agencia.agregarVehiculo(vehiculo1);
        agencia.agregarVehiculo(vehiculo2);

        agencia.listarVehiculos();

        cliente1.alquilarVehiculo(); // Alquilar vehiculo
        cliente1.devolverVehiculo(); // Devolver vehiculo

        Empleado empleado1 = new Empleado(cliente1, vehiculo1);
        empleado1.registrar(); // Registrar alquiler
        empleado1.cancelar(); // Cancelar alquiler
    }
}

```

```

// Clase Vehiculo
class Vehiculo {
    String modelo;
    String matricula;

    public Vehiculo(String modelo, String matricula) {
        this.modelo = modelo;
        this.matricula = matricula;
    }
}

```

```

// Clase Cliente
class Cliente {
    String nombre;
    String telefono;

    public Cliente(String nombre, String telefono) {
        this.nombre = nombre;
        this.telefono = telefono;
    }
}

```

```

// Clase Agencia
class Agencia {
    String nombre;
    String direccion;

    public Agencia(String nombre, String direccion) {
        this.nombre = nombre;
        this.direccion = direccion;
    }
}

```

```

PS C:\Users\RYZEN\Github> & 'C:\Program Files\Eclipse Adoptium\jdk-21.0.5.11-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\RYZEN\AppData\Roaming\Code\User\workspace
eStorage\d3bfb36972b0a5123e3df3d6db7639e\redhat.java\jdt_ws\Github_3b5e9415\bin' 'deber'
Vehículo Toyota Corolla agregado a la agencia.
Vehículo Honda Civic agregado a la agencia.
Vehículos disponibles en la agencia Agencia Central:
- Modelo: Toyota Corolla, Matrícula: ABC-123
- Modelo: Honda Civic, Matrícula: XYZ-456
El cliente María López ha alquilado un vehículo.
El cliente María López ha devuelto un vehículo.
El cliente María López ha registrado el alquiler del vehículo Toyota Corolla.
El cliente María López ha cancelado el alquiler del vehículo Toyota Corolla.

```

Python  
Python  
Python  
Run: deber