

Universidad de las Fuerzas Armadas-ESPE

Segundo nivel

Tarea:

Creación de Objetos y UML

Alumno

[Alejandra Renata Salazar](#)

POO

Docente

Ing. LUIS ENRIQUE JARAMILLO MONTAÑO

27 de noviembre de 2024

Descripción de la Actividad:

Diseñe 5 objetos diferentes con su correspondiente diagrama UML, asegurándose de mostrar las relaciones entre ellos.

Problemática:

En un taller mecánico, se lleva a cabo el proceso de reparación de vehículos. El flujo incluye:

1. El cliente registra su vehículo en el sistema.
2. Un mecánico asignado evalúa el vehículo y genera una orden de reparación.
3. Los repuestos requeridos son registrados y asociados a la orden.
4. Una vez completada la reparación, el cliente realiza el pago (efectivo o tarjeta).
5. Se actualiza el estado del vehículo en el sistema.

Se modela este flujo utilizando diagramas de clases UML y definiendo las clases, atributos y métodos correspondientes, junto con las relaciones entre ellas.

Análisis del Problema y Diseño de Clases

A partir de la descripción del proceso, se identifican las siguientes clases:

1. **Cliente:** Representa al propietario del vehículo.
2. **Mecánico:** Representa al empleado encargado de las reparaciones.
3. **Orden Reparación:** Representa el registro del proceso de reparación de un vehículo.
4. **Repuesto:** Representa los elementos necesarios para la reparación.
5. **Pago:** Representa la forma de realizar el pago (efectivo o tarjeta).

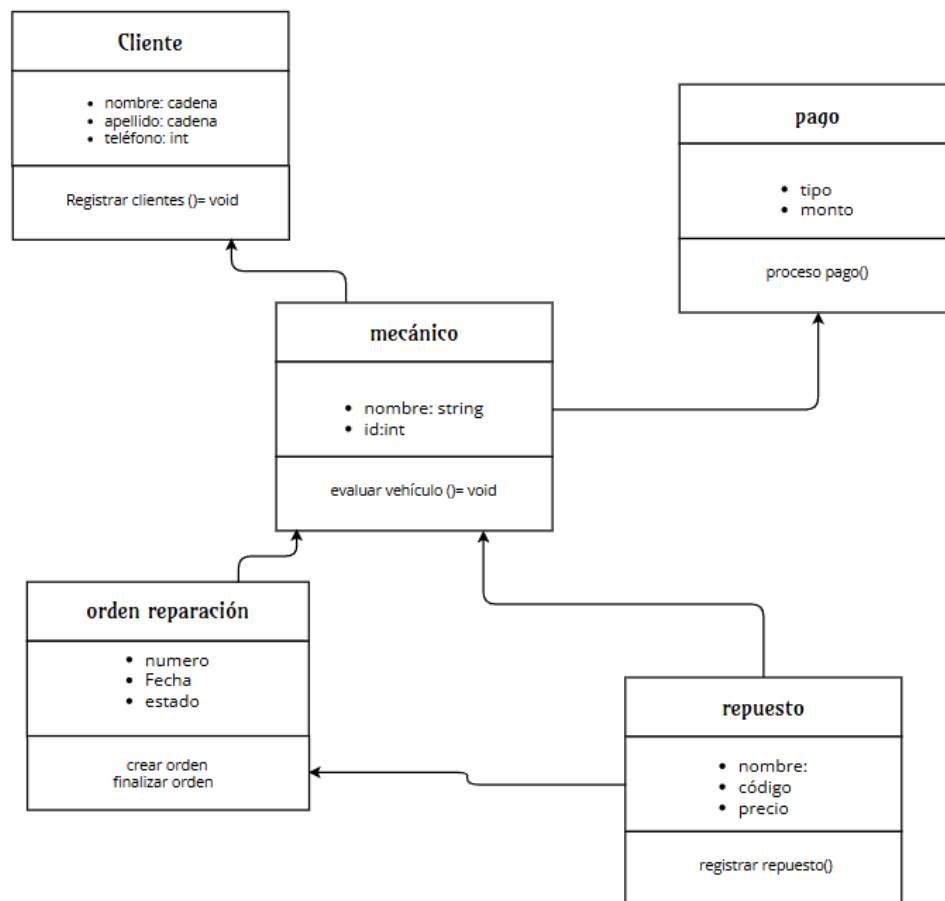
Atributos y Métodos de Cada Clase:

Clase	Atributos	Métodos
<ul style="list-style-type: none"> • Cliente • Mecánico • Orden Reparación • Repuesto • Pago 	<ul style="list-style-type: none"> • nombre, apellido, telefono • id, nombre • número, fecha, estado • nombre, código, precio • tipo, monto 	<ul style="list-style-type: none"> • registrar Cliente () • evaluar Vehículo () • crear Orden (), finalizar Orden () • registrar Repuesto () • procesar Pago ()

Relaciones entre Clases

- **Cliente y Orden Reparación (Asociación):** Un cliente puede generar varias órdenes, pero una orden pertenece a un cliente.
- **Mecánico y Orden Reparación (Asociación):** Un mecánico puede gestionar múltiples órdenes.
- **Orden Reparación y Repuesto (Composición):** Una orden está compuesta por uno o más repuestos.
- **Orden Reparación y Pago (Agregación):** Una orden puede tener un pago asociado.

UML



Código

Clase Cliente

```
1  class Cliente {
2      public String nombre;
3      public String apellido;
4      public int telefono;
5
6      public Cliente (String nombre, int apellido, String telefono) {
7          this.nombre = nombre;
8          this.apellido = apellido;
9          this.telefono = telefono;
10     }
11
12     public void registrarCliente() {
13         System.out.println ("Cliente registrado: " + nombre + " " + apellido);
14     }
15 }
```

Clase Mecánico

```
16  class Mecanico {
17      public int id;
18      public String nombre;
19
20      public Mecanico(int id, String nombre) {
21          this.id = id;
22          this.nombre = nombre;
23      }
24
25      public void evaluarVehiculo() {
26          System.out.println("Vehículo evaluado por: " + nombre);
27      }
28  }
```

Clase Orden Reparación

```
30  class OrdenReparacion {
31      public int numero;
32      public String fecha;
33      public String estado;
34
35      public OrdenReparacion(int numero, String fecha, String estado) {
36          this.numero = numero;
37          this.fecha = fecha;
38          this.estado = estado;
39      }
40
41      public void crearOrden() {
42          System.out.println("Orden de reparación creada: " + numero);
43      }
44
45      public void finalizarOrden() {
46          this.estado = "Finalizada";
47          System.out.println("Orden de reparación finalizada: " + numero);
48      }
49  }
```

Clase Repuesto

```
51 class Repuesto {
52     public String nombre;
53     public int codigo;
54     public double precio;
55
56     public Repuesto(String nombre, int codigo, double precio) {
57         this.nombre = nombre;
58         this.codigo = codigo;
59         this.precio = precio;
60     }
61
62     public void registrarRepuesto() {
63         System.out.println("Repuesto registrado: " + nombre);
64     }
65 }
```

Clase Pago

```
67 class Pago {
68     public String tipo;
69     public double monto;
70
71     public Pago(String tipo, double monto) {
72         this.tipo = tipo;
73         this.monto = monto;
74     }
75
76     public void procesarPago() {
77         System.out.println("Pago procesado de tipo: " + tipo + " por monto: " + monto);
78     }
79 }
```

- inicio de programa

```
81 public class Main {
82     public static void main (String [] args){
83         Cliente cliente=new Cliente ("Alejita","Salazar",983107929);
84         Mecanico mecanico=new Mecanico (172763,"Andrew");
85         OrdenReparacion orden=new OrdenReparacion (1,"Dos de Febrero","Reparado");
86         Repuesto repuesto=new Repuesto ("bujia",809,25);
87         Pago pago= new Pago ("Transferencia",40);
88
89         cliente.registrarCliente ();
90         mecanico.evaluarVehiculo ();
91         orden.crearOrden ();
92         orden.finalizarOrden ();
93         repuesto.registrarRepuesto ();
94         pago.procesarPago();
95     }
96 }
97 }
```

- Ejecución

```
✓ ↗ 📄 ⚙️ 🖨️ inpt
Cliente registrado: Alejita Salazar
Vehículo evaluado por: Andrew
Orden de reparación creada: 1
Orden de reparación finalizada: 1
Repuesto registrado: bujia
Pago procesado de tipo: Transferencia por monto: 40.0

...Program finished with exit code 0
Press ENTER to exit console.
```

Explicación del código: Sistema de Biblioteca

1. Clases como Entidades:

- **Cliente:** Representa al propietario del vehículo.
- **Mecánico:** Gestiona las reparaciones.
-
- **Repuesto:** Contiene información sobre los repuestos utilizados.
- **Pago:** Registra las transacciones de pago.

2. Relaciones:

- Las clases están conectadas por relaciones de asociación, composición y agregación según el flujo descrito.

3. Propósito: Este sistema permite gestionar un taller de vehículos de forma eficiente, ampliándose según las necesidades.