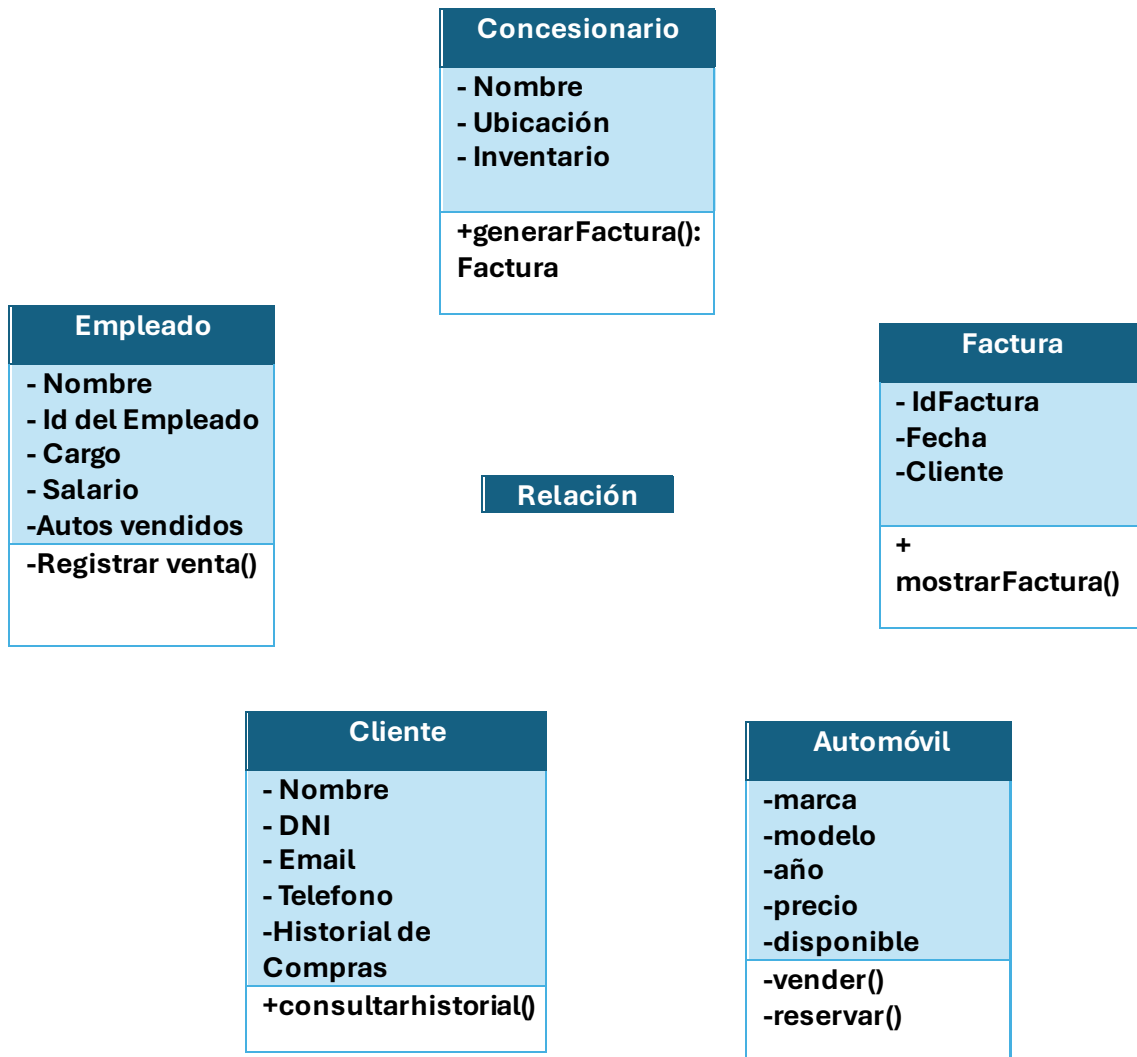


DIAGRAMA de UML



CODIGO

```
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
```

```
// Clase Automóvil
```

```
class Automovil {
    private String marca;
    private String modelo;
    private int anio;
```

```
private float precio;
```

```
private boolean disponible;
```

```
public Automovil(String marca, String modelo, int anio, float precio) {
```

```
    this.marca = marca;
```

```
    this.modelo = modelo;
```

```
    this.anio = anio;
```

```
    this.precio = precio;
```

```
    this.disponible = true;
```

```
}
```

```
public float getPrecio() {
```

```
    return precio;
```

```
}
```

```
public boolean isDisponible() {
```

```
    return disponible;
```

```
}
```

```
public void vender() {
```

```
    if (disponible) {
```

```
        disponible = false;
```

```
        System.out.println("El automóvil " + modelo + " ha sido vendido.");
```

```
    } else {
```

```
        System.out.println("El automóvil no está disponible.");
```

```
    }
```

```
}
```

```
@Override
```

```
public String toString() {
```

```
        return marca + " " + modelo + " (" + anio + ") - $" + precio + (disponible ? "[Disponible]" : "[No disponible]");
    }
}
```

// Clase Cliente

```
class Cliente {

    private String nombre;

    private String dni;

    private String email;

    private String telefono;

    private List<Automovil> historialCompras;

    public Cliente(String nombre, String dni, String email, String telefono) {

        this.nombre = nombre;

        this.dni = dni;

        this.email = email;

        this.telefono = telefono;

        this.historialCompras = new ArrayList<>();

    }

    public void comprarAutomovil(Automovil automovil) {

        if (automovil.isDisponible()) {

            historialCompras.add(automovil);

            automovil.vender();

            System.out.println("El cliente " + nombre + " ha comprado el automóvil " + automovil);

        } else {

            System.out.println("El automóvil no está disponible para la compra.");

        }

    }

}
```

```
public void consultarHistorial(){  
    System.out.println("Historial de compras de " + nombre + ":");  
    for (Automovil auto : historialCompras){  
        System.out.println("- " + auto);  
    }  
}  
}
```

// Clase Empleado

```
class Empleado {  
    private String nombre;  
    private String idEmpleado;  
    private String cargo;  
    private float salario;  
  
    public Empleado(String nombre, String idEmpleado, String cargo, float salario) {  
        this.nombre = nombre;  
        this.idEmpleado = idEmpleado;  
        this.cargo = cargo;  
        this.salario = salario;  
    }  
  
    public void registrarVenta(Automovil automovil, Cliente cliente) {  
        cliente.comprarAutomovil(automovil);  
        System.out.println("El empleado " + nombre + " registró la venta.");  
    }  
}
```

// Clase Factura

```
class Factura {  
    private String idFactura;
```

```

private Date fecha;

private Cliente cliente;

private Automovil automovil;

private float total;


public Factura(String idFactura, Cliente cliente, Automovil automovil) {

    this.idFactura = idFactura;

    this.fecha = new Date();

    this.cliente = cliente;

    this.automovil = automovil;

    this.total = automovil != null ? automovil.getPrecio() : 0;

}


public void mostrarFactura() {

    System.out.println("Factura ID: " + idFactura);

    System.out.println("Fecha: " + fecha);

    System.out.println("Cliente: " + cliente);

    System.out.println("Automóvil: " + automovil);

    System.out.println("Total: $" + total);

}

}

```

```

// Clase principal Concesionario

public class Concesionario {

    public static void main(String[] args) {

        // Crear automóviles

        Automovil auto1 = new Automovil("Toyota", "Corolla", 2022, 20000);

        Automovil auto2 = new Automovil("Honda", "Civic", 2023, 22000);


        // Crear clientes
    }

}

```

```
    Cliente cliente1 = new Cliente("Juan Perez", "12345678", "juan@example.com",  
"555-1234");
```

```
    // Crear empleados
```

```
    Empleado empleado1 = new Empleado("Maria Gomez", "E001", "Vendedora",  
1500);
```

```
    // Mostrar inventario inicial
```

```
    System.out.println("Inventario inicial:");
```

```
    System.out.println(auto1);
```

```
    System.out.println(auto2);
```

```
    // Registrar una venta
```

```
    System.out.println("\nRegistrar venta:");
```

```
    empleado1.registrarVenta(auto1, cliente1);
```

```
    // Consultar historial de compras del cliente
```

```
    System.out.println("\nConsultar historial del cliente:");
```

```
    cliente1.consultarHistorial();
```

```
    // Mostrar estado del inventario después de la venta
```

```
    System.out.println("\nInventario después de la venta:");
```

```
    System.out.println(auto1);
```

```
    System.out.println(auto2);
```

```
    // Generar una factura
```

```
    System.out.println("\nGenerar factura:");
```

```
    Factura factura1 = new Factura("F001", cliente1, auto1);
```

```
    factura1.mostrarFactura();
```

```
    }
```

```
}
```

El diagrama UML se organiza colocando las clases en un pentágono con Concesionario en la punta, Empleado en la esquina izquierda, y Factura en la esquina derecha. Cliente se ubica en la parte inferior izquierda, mientras que Automovil se coloca en la parte inferior derecha. Concesionario tiene una relación de composición con Factura, y de asociación con Empleado. Factura tiene una relación de agregación con Cliente y Automovil, mientras que Cliente está asociado a Automovil debido a que un cliente puede comprar varios vehículos. Las multiplicidades se especifican adecuadamente para representar la cantidad de instancias involucradas en cada relación.

```
PS C:\Users\MONICA\Github ESPE> ^C
PS C:\Users\MONICA\Github ESPE>
PS C:\Users\MONICA\Github ESPE> c::; cd 'c:\Users\MONICA\Github ESPE'; & 'C:\Program Files\Eclipse Adoptium\jdk-17.0.13.11-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\MONICA\AppData\Roaming\Code\User\workspaceStorage\e95c0a7cddb3f29783afaaeb04885d74\redhat.java\jdt_ws\Github ESPE_5a981b4b\bin' 'Concesionario'
Inventario inicial:
Toyota Corolla (2022) - $20000.0 [Disponible]
Honda Civic (2023) - $22000.0 [Disponible]

Registrar venta:
El automóvil Corolla ha sido vendido.
El cliente Juan Perez ha comprado el automóvil Toyota Corolla (2022) - $20000.0 [No disponible]
El empleado Maria Gomez registró la venta.

Consultar historial del cliente:
Historial de compras de Juan Perez:
- Toyota Corolla (2022) - $20000.0 [No disponible]

Inventario después de la venta:
Toyota Corolla (2022) - $20000.0 [No disponible]
Honda Civic (2023) - $22000.0 [Disponible]

Generar factura:
Factura ID: F001
Fecha: Sun Dec 08 21:31:40 ECT 2024
Cliente: Cliente@7cd84586
Automóvil: Toyota Corolla (2022) - $20000.0 [No disponible]
Total: $20000.0
PS C:\Users\MONICA\Github ESPE> |
```

GITHUB

<https://github.com/DevilDibu/ESPE>