



SESIÓN 03:

Relación de Asociación

Relación de Agregación y Composición



Una relación es un vínculo entre dos objetos, este vínculo, se presenta por que, ambos objetos, se necesitan mutuamente para lograr la realización de una actividad o la realización de unos servicios. Estas relaciones nacen por la misma dinámica social o por que la naturaleza así lo estableció.

Que nunca se me olvide:

LAS RELACIONES DEPENDEN DEL CONTEXTO
"Dominio del Problema"



Definición de relaciones

Pregrado

Ingeniería de
Sistemas





Tipos de relaciones entre clases

Pregrado

Ingeniería de
Sistemas

- Composición
- Agregación
- Asociación
- Dependencia – Uso
- Herencia



- Se constituye entre el *todo* y la *parte*.
- La clase *todo* controla la existencia de las clases *parte*. Mientras existe el *todo*= existe la *parte*.
- Al crear un objeto del *todo* se crea un objeto de la *parte*
- Objeto A *tiene un/posee un/contiene un* objeto de la clase B
- Los objetos suelen estar compuestos de conjuntos de objetos más pequeños; un coche es un conjunto de motor y carrocería, un motor es un conjunto de piezas, y así sucesivamente.
- Si se elimina el *todo* se eliminan las *partes*.
- Contiene un atributo, que puede ser una colección y además de ello la clase que contiene la colección debe tener un método que agregue los elementos a la colección.

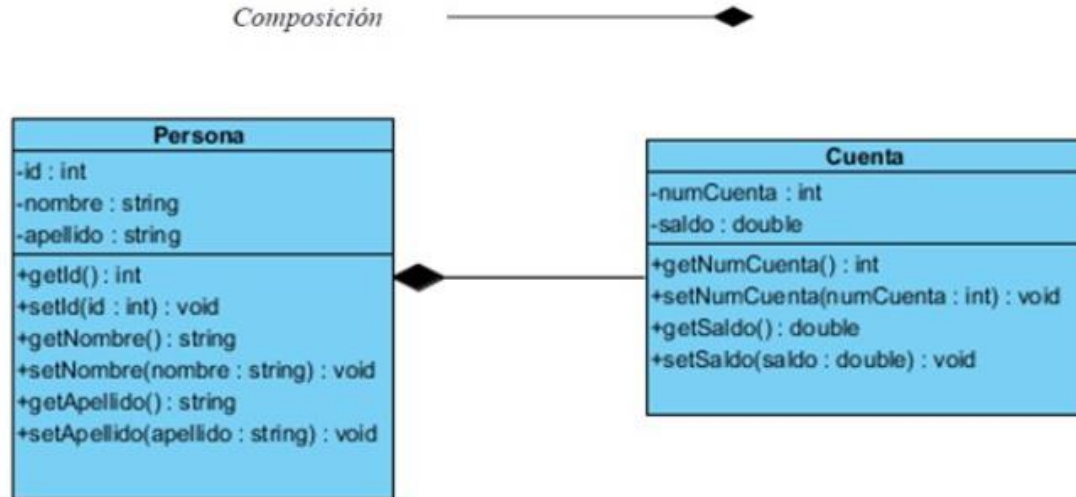


Composición - Notación

Pregrado

Ingeniería de
Sistemas

- o La notación para representar las composiciones, es una línea con un rombo lleno en el extremo apuntando a la clase **todo**





Composición - Implementación

Pregrado

Ingeniería de
Sistemas

```
Public class Persona {  
  
    private int id;  
    private String nombre;  
    private String apellido;  
    public Cuenta cuenta = new Cuenta();  
  
    public int getId() {  
        return id;  
    }  
    public void setId(int id) {  
        this.id=id;  
    }  
    public string getNombre() {  
        return nombre;  
    }  
    public void setNombre(String nombre) {  
        this.nombre=nombre;  
    }  
    public string getApellido() {  
        return apellido;  
    }  
    public void setApellido(String apellido) {  
        this.apellido=apellido;  
    }  
    public int getCuenta() {  
        return cuenta;  
    }  
    public void setCuenta(Cuenta cuenta) {  
        this.cuenta=cuenta;  
    }  
}
```

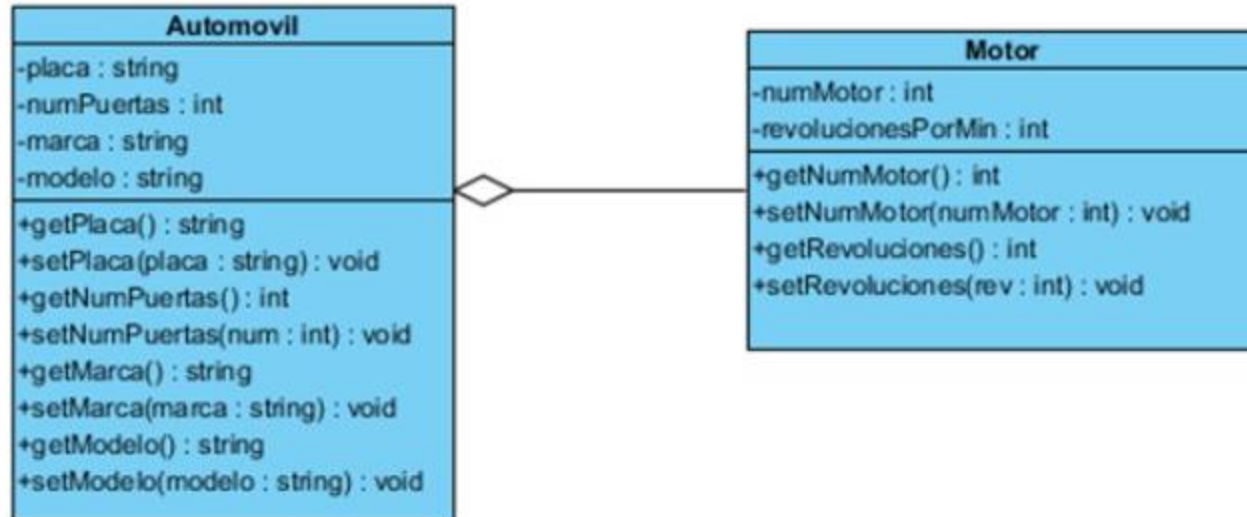
```
public class Cuenta {  
  
    private int numCuenta;  
    private double saldo;  
  
    public int getNumCuenta() {  
        return NumCuenta;  
    }  
  
    public void setNumCuenta(int numCuenta) {  
        this.numCuenta=numCuenta;  
    }  
  
    public double getSaldo() {  
        return saldo;  
    }  
  
    public void setSaldo(double saldo) {  
        this.saldo=saldo;  
    }  
}
```



- Existe una relación de agregación si en el enunciado del problema hay expresiones de la forma "tiene_un", "es_parte_de"...
- Se constituye entre el *todo* y la *parte*.
- La existencia del *todo* no condiciona la existencia de la *parte*.
- Si se elimina el *todo* NO se eliminan las *partes*.
- Contiene un atributo, que puede ser una colección, es decir un array, vector, etc, y además de ello la clase que contiene la colección debe tener un método que agregue los elementos a la colección.



- La notación para representar las agregaciones, es una línea con un rombo vacío en el extremo apuntando a la clase **todo**





Agregación - Implementación

Pregrado

Ingeniería de
Sistemas

```
public class Automovil {  
    private String placa;  
    private int numPuertas;  
    private String marca;  
    private String modelo;  
    public Motor motor;  
  
    public String getPlaca() {  
    }  
  
    public void setPlaca(String placa) {  
    }  
  
    public int getNumPuertas() {  
    }  
  
    public void setNumPuertas(int num) {  
    }  
  
    public String getMarca() {  
    }  
  
    public void setMarca(String marca) {  
    }  
  
    public String getModelo() {  
    }  
  
    public void setModelo(String modelo) {  
    }  
}
```

```
public class Motor {  
    private int numMotor;  
    private int revolucionesPorMin;  
  
    public int getNumMotor() {  
    }  
  
    public void setNumMotor(int numMotor) {  
    }  
  
    public int getRevoluciones() {  
    }  
  
    public void setRevoluciones(int rev) {  
    }  
}
```



- Desarrollar una aplicación que permita registrar las ventas del día de una tienda de productos electrónicos.
- La aplicación debe permitir tener un resumen de las ventas del día.
- También debe permitir tener el listado de todas las ventas.
- Se debe aplicar la programación en capas.



Universidad **César Vallejo**

Licenciada por Sunedu
para que puedas salir adelante