



UML- Diagrama de Clases

Víctor Custodio, 2014

Víctor Custodio, 2014

¿Qué es?

- Es la representación de las clases, interfaces y colaboraciones que serán utilizadas dentro del sistema y las relaciones que existen entre ellas.

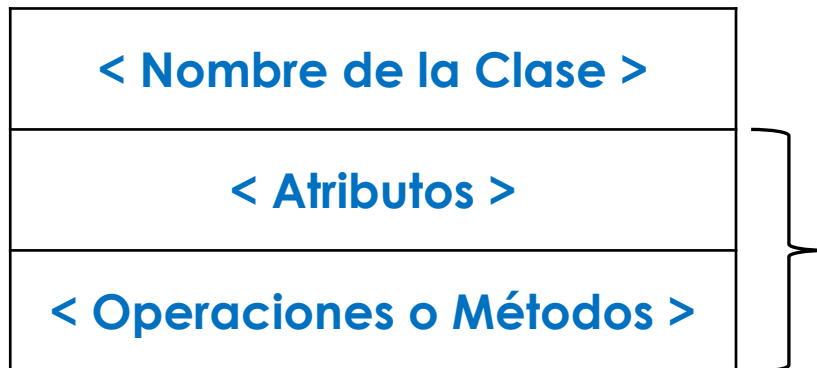
- Se utiliza para modelar la vista de diseño estática de un sistema.
- También se utilizan para construir sistemas ejecutables, aplicando ingeniería directa e inversa.

Sus elementos:

- **Clases:** Atributos, Métodos Y Visibilidad.
- **Relaciones:** Herencia, Composición, Agregación, Asociación y Uso.

Clase:

- En UML, una clase es representada por un rectángulo que posee tres divisiones:



Pueden ser:

- Public
- Private
- Protected

Atributos:

- Propiedades relevantes de un clase
- Representan su estructura
- Pueden ser simples o compuestos
 - pedido
 - hora
 - fecha

Métodos:

- Comportamiento asociado a una Clase
 - cobrar
 - tomar nota
 - servir

PedidoComida
-correlativo -fecha -hora
+tomarNota(Mesa) +cocinar() +servir() +cobrar()

Víctor Custodio, 2014

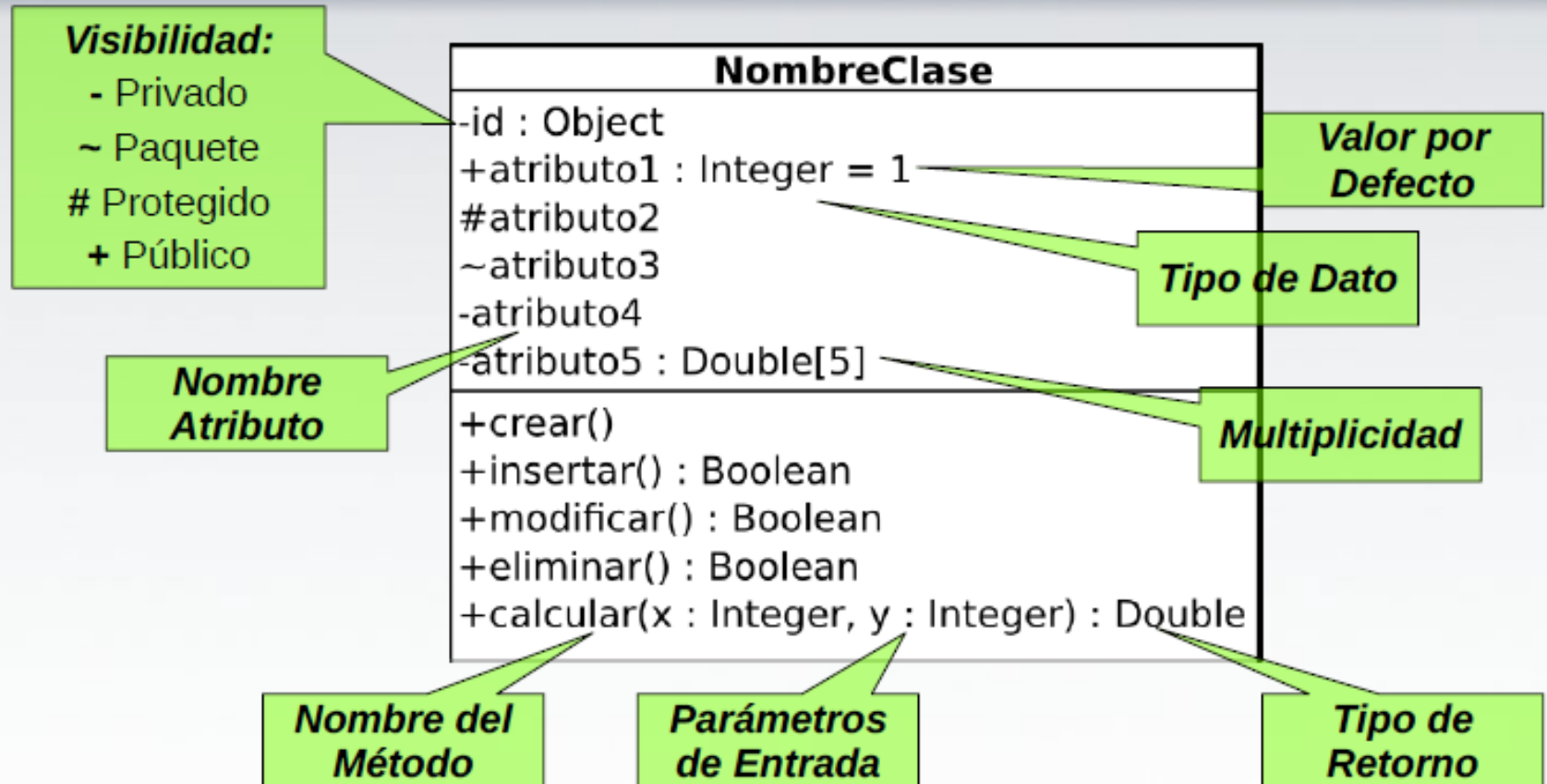
Ejemplo en UML

ComplexNumber
-r : double -i : double
+ComplexNumber(r : double, i : double) +norm() : double

Ejemplo en código java:

```
public class ComplexNumber {  
  
    private double r;  
    private double i;  
  
    public ComplexNumber(double r, double i) {  
        this.r = r;  
        this.i = i;  
    }  
  
    public double norm() {  
        return Math.sqrt(r * r + i * i);  
    }  
}
```

Notación:



Sintaxis:

Para los Atributos:

[visibilidad] [/] nombre [:tipo] [multiplicidad] [=valor por omisión] [{**propiedad**}]

Para los Métodos:

[visibilidad] nombre [(lista de parámetros)] [{**propiedad**}]

Donde un parámetro es:

[dirección (in/out/inout)] nombre: tipo [multiplicidad] [=valor por omisión]

Las **propiedades pueden una o mas de las siguientes:**

readOnly, isQuery, Concurrent, Guarded, Sequential, etcétera

... o cualquier otra predefinida ...

Relaciones:

- Interrelación de dos o más clases (cada uno con características y objetivos diferentes).

Tipos de relaciones:

- **Asociación:** Ocurre cuando las clases se conectan entre si de forma conceptual.

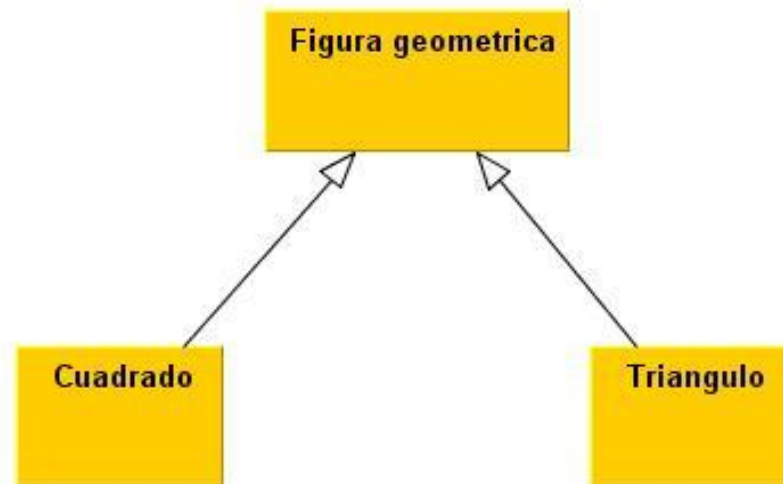


Asociaciones: En java Indican que un objeto tiene un atributo de otro objeto, también pueden indicar cardinalidades.



Herencia

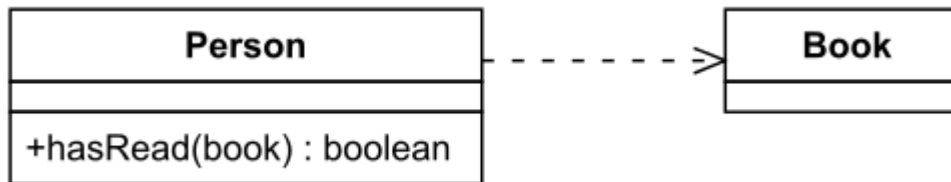
- Indica que una subclase hereda los métodos y atributos especificados por una Súper Clase, por ende la Subclase además de poseer sus propios métodos y atributos, poseerá las características y atributos visibles de la Súper Clase (public y protected)



Victor Custodio, 2014

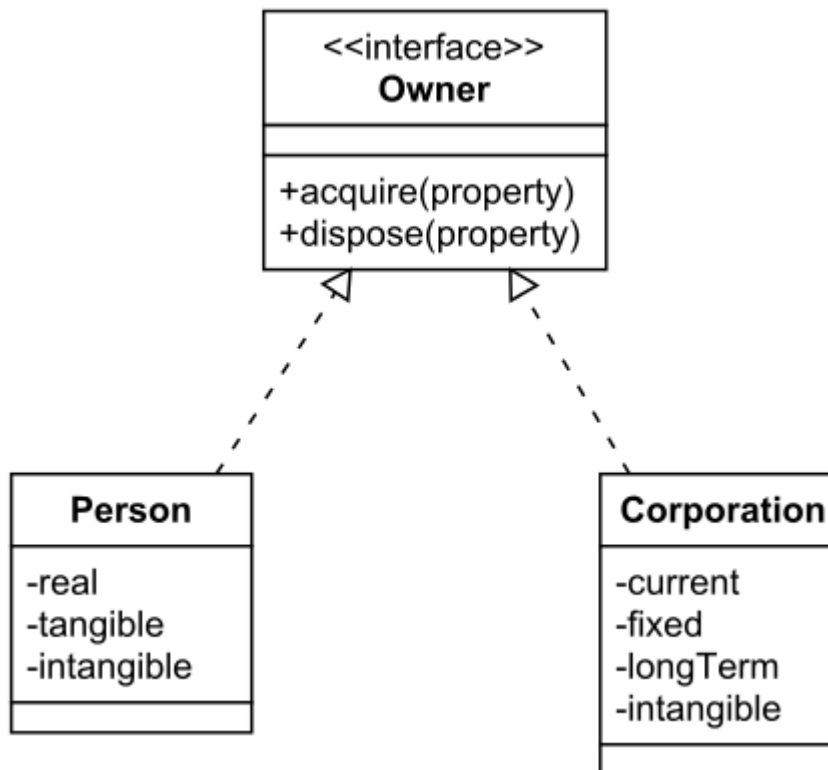
Dependencia:

- Sucede cuando una clase utiliza a otra, a esto se llama dependencia



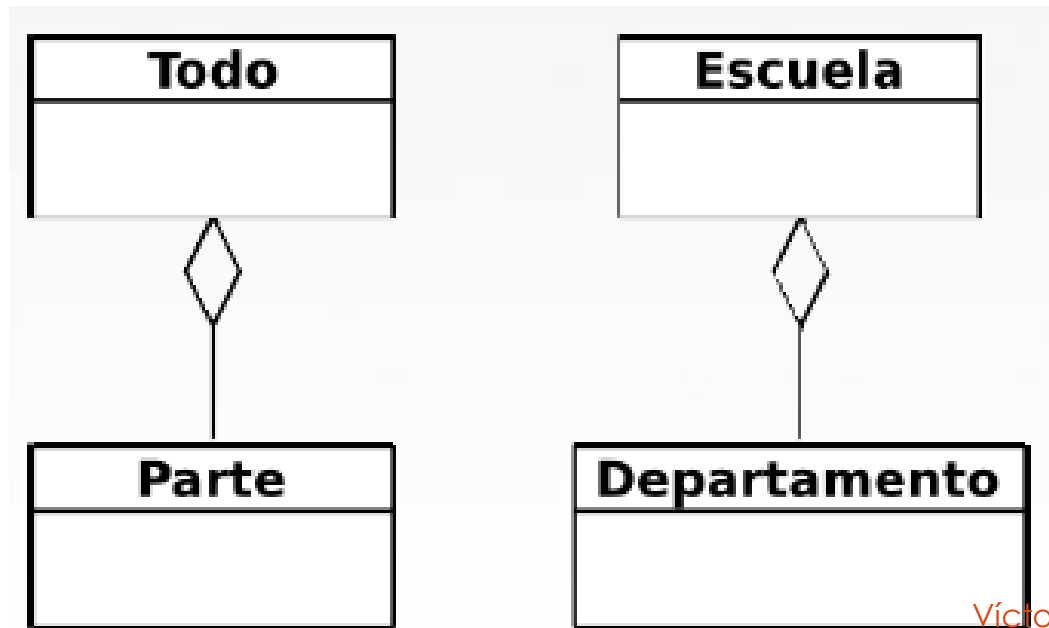
Interfaces:

- Representan relaciones de implementación de interfaces



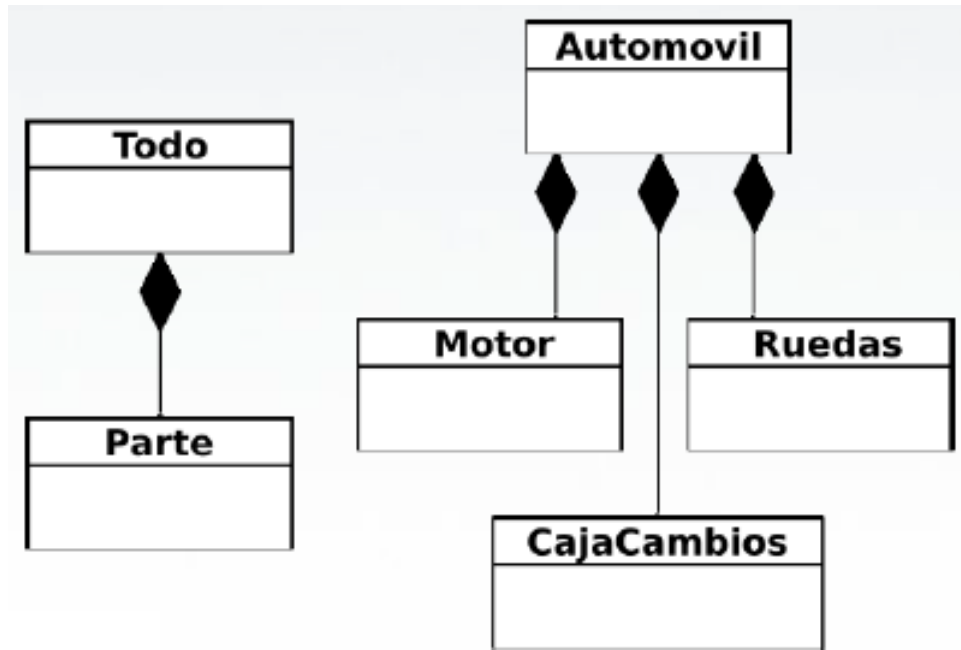
Agregación (por referencia)

- Es una relación en la que una de las clases representa un todo y la otra representa parte de ese todo

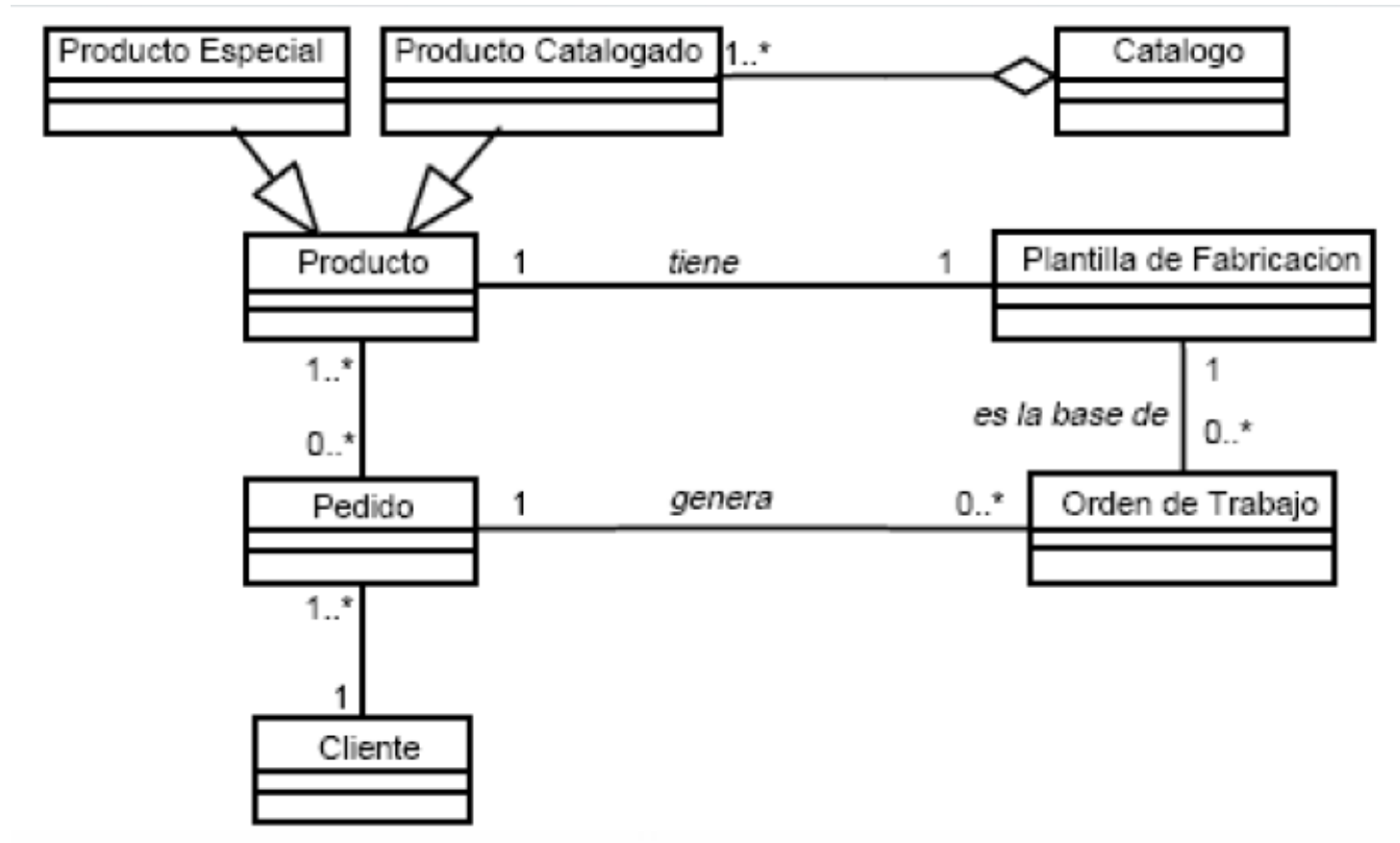


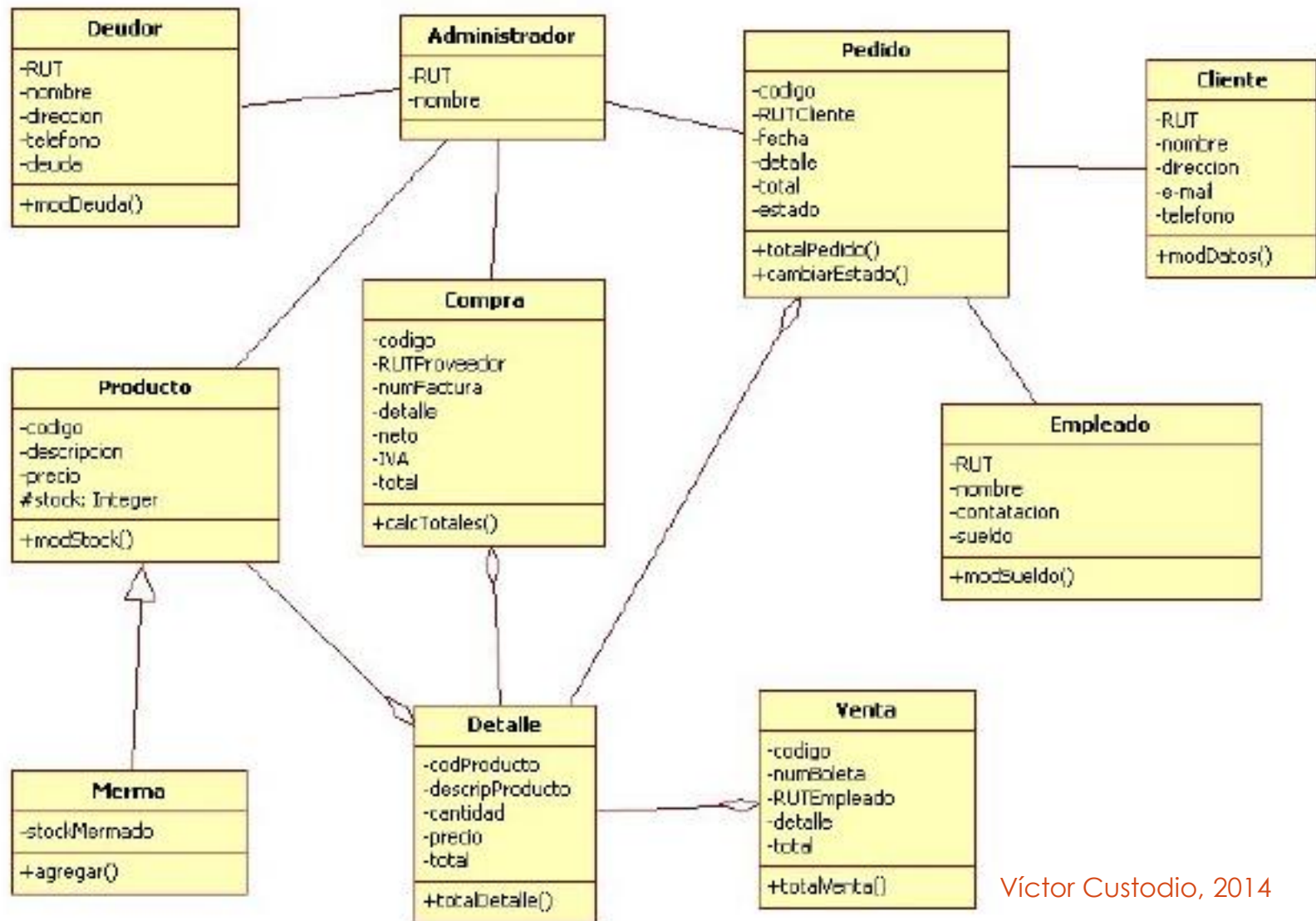
Composición (por valor)

- Es una forma más fuerte de la agregación, en la que el parte no puede existir sin el todo en nuestro sistema.



Ejemplo:





Víctor Custodio, 2014

Ejercicio

- Crea el diagrama de clases de un programa en java que gestione en garben:
 - Alumnos, profesores, cursos, aulas, horarios y libros.

Bibliografía

- "El lenguaje unificado de modelado"
Grady Booch, James Rumbaugh, Ivar Jacobson
Editorial Addison Wesley
- <http://www.youtube.com/user/piojosnos>