Unidad dos

Curso: UML - Ingeniería de Software

¿Qué vimos hasta ahora?

- Ciclo de Vida de Proyectos de Software

  Análisis, Diseño, Implementación

- Modelación...

- Diagramas UML
  - Elementos estructurales
  - Elementos de comportamiento
  - Relaciones

Diagramas UML

- Use-Case diagrams
- Diagramas de Componentes
  (Software)
- Diagramas de Distribución
  (Hardware)
- Diagramas de Actividades
  (Flujograma)
- Diagramas de Colaboración
  (Mensajes entre objetos para cambiar su estado)
- Diagramas de Secuencias
  (Línea de vida - Colaboración secuencializada)
- Diagramas de Estado
  (State Machines)

Modelación

%system "xdvi modelacisn.dvi"

Práctica:
Modelos en análisis y diseño de motores eléctricos:

- Eléctrico: tensiones, corrientes, campos electromagnéticos, inductancias, resistencias.
- Mecánico: rigidez, densidad, movimiento, fuerzas pares.
- Térmico: disipación de calor, transferencia de calor
- Fluido: Flujo de aire refrigerante

¿Qué modelos deben considerarse para?

- ¿Cuánto pesa un motor?
- ¿Cuánto se calienta el motor?
- ¿Cuánta vibración crea un motor?
- ¿Cuánto tardarán en gastarse los rodamientos?

... Orientado a Objetos

Programación: Smalltalk, Simulación

Theoria generalizada:

- Análisis Orientado a Objetos
- Diseño Orientado a Objetos
- Modelación Orientado a Objetos - OMT

  Modelación cierra el abismo entre análisis y diseño.

- Programación = Implementación orientado a Objetos

Programación Orientado a Objetos

C++ ampliación de sintaxis de C

java máquina virtual portable, interpretadores

Lisp, Forth

Perl, Tcl Rapid Prototyping, Scripting

Bases de datos relacionales

Eiffel, Python, Smalltalk Lenguajes con OO inherente

Programación Orientado a Objetos

Conceptos básicos

- Identidad
- Clasificación
- Polimorfismo
- Herencia

Programación Orientado a Objetos

Ejemplo: Geometria

Jerarqía de Clases:

- punto
- linea
- trapezoide

Diagrama de Clases

Lavadora

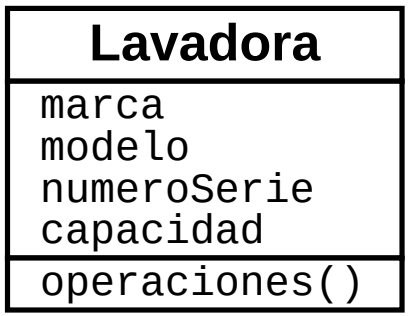
Clase

Electrodomesticos

Paquete - jerarquía de clases

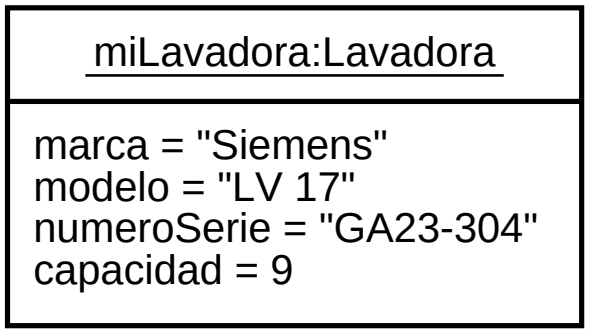
Electrodomesticos::Lavadora

Classpath - via de acceso



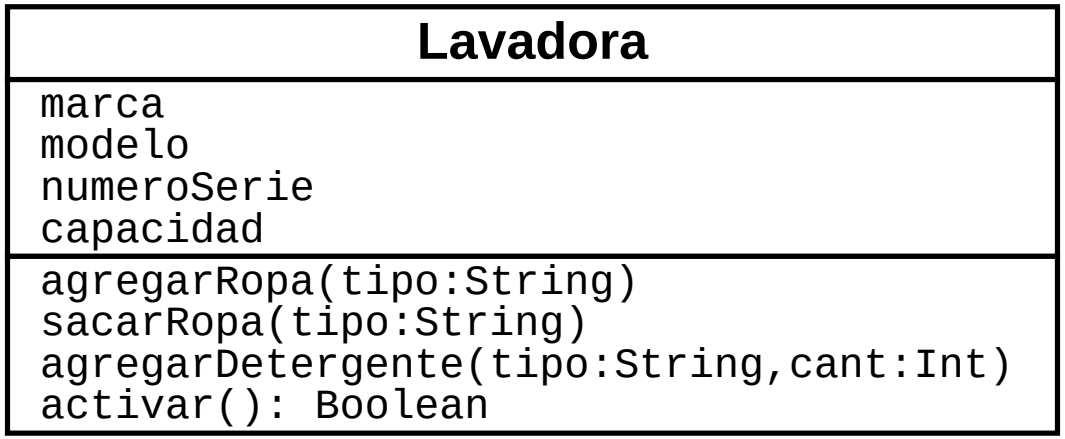
Clase con atributos y operaciones

Diagrama de Clases




```
Instancia de la Clase Lavadora
Instancias anónimas ":Lavadora"
```

Diagrama de Clases



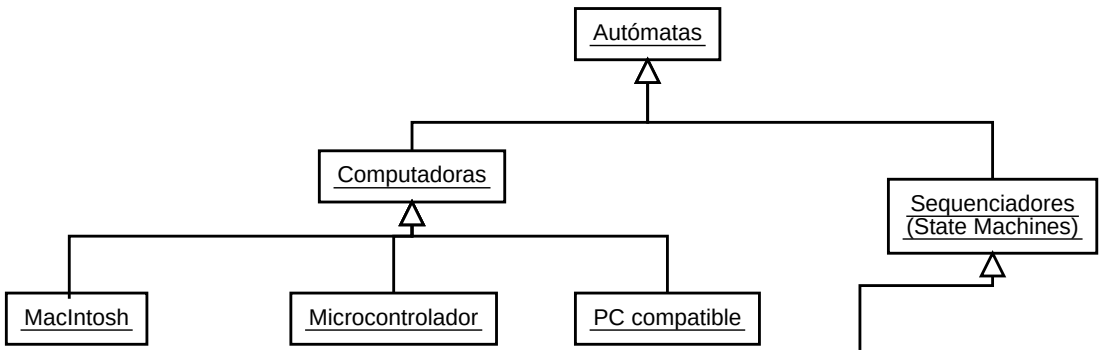
Firma (Signature) de operaciones

Relaciones entre Clases

Asociación

- uni/bi-direccional, rol
- Restricción {}
- Multiplicidad 1, *, n..m, a,b,c
- Calificación

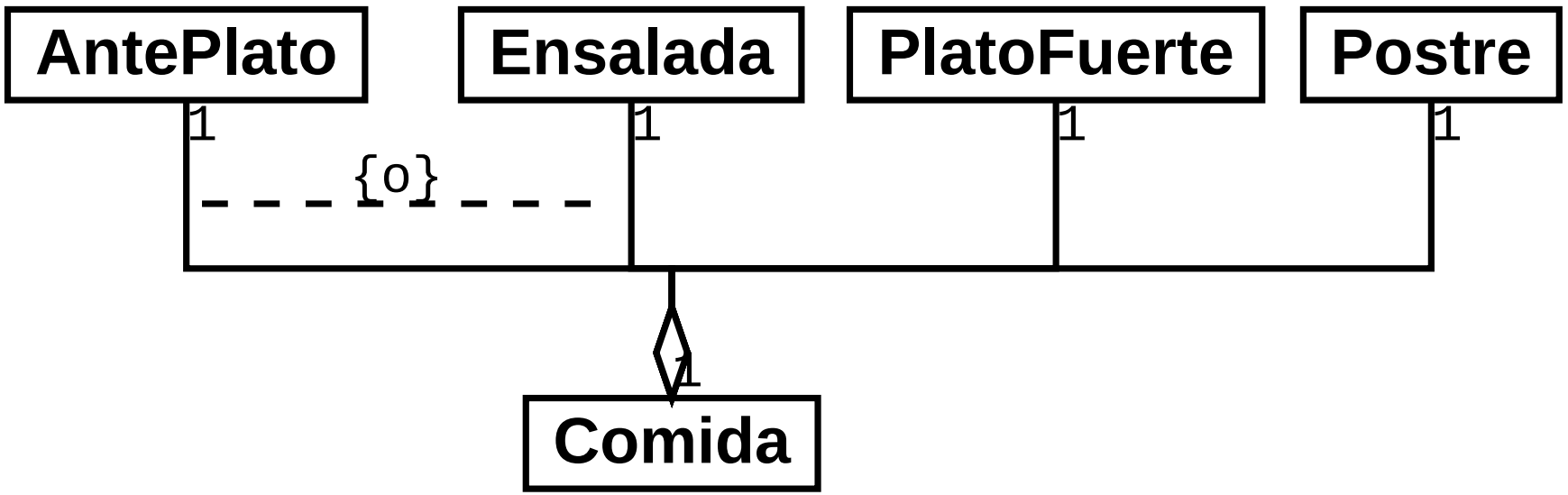
Herencia y generalización



Relaciones entre Clases

Agregación

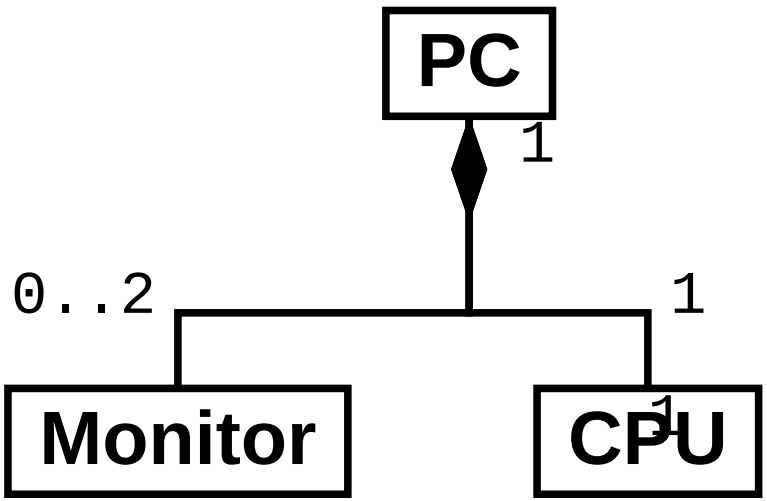
Agregación con restricción



Relaciones entre Clases

Composición

Agregación exclusiva



Práctica:

Realizar el diagrama de Clases del ejemplo "Geometrí
a"

Interfaces

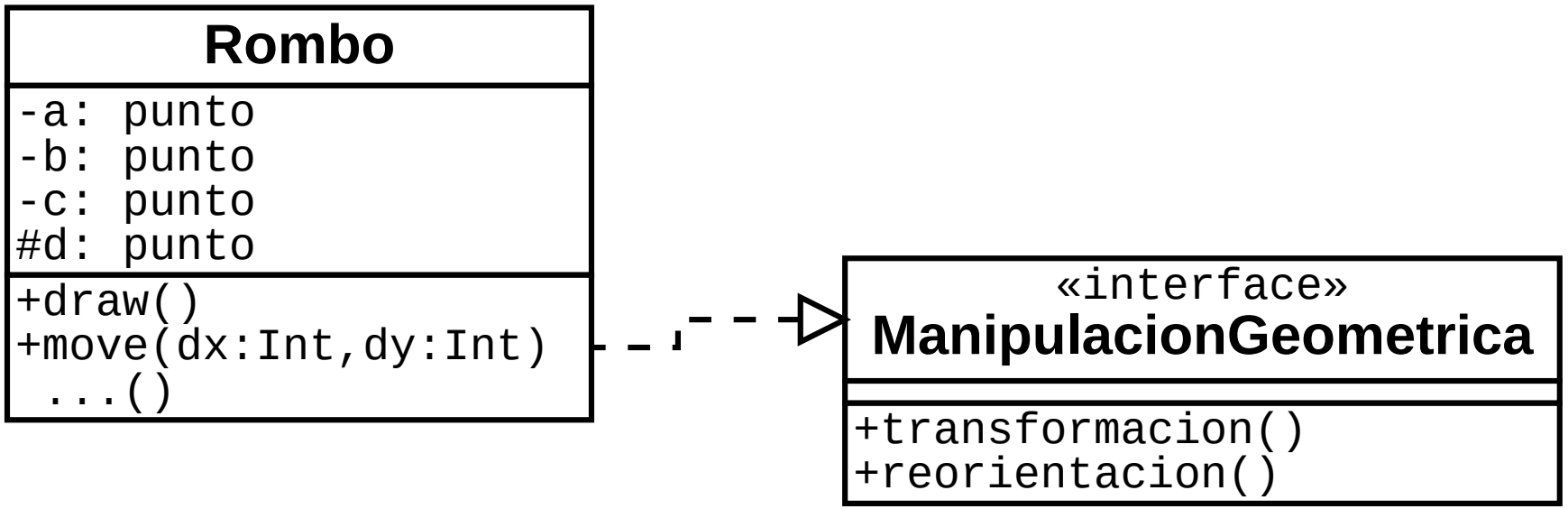
Interfaz es un conjunto (nombrado) de operaciones

que presenta una clase hacia el sistema

Clase que implementa: ofrece Interfaz

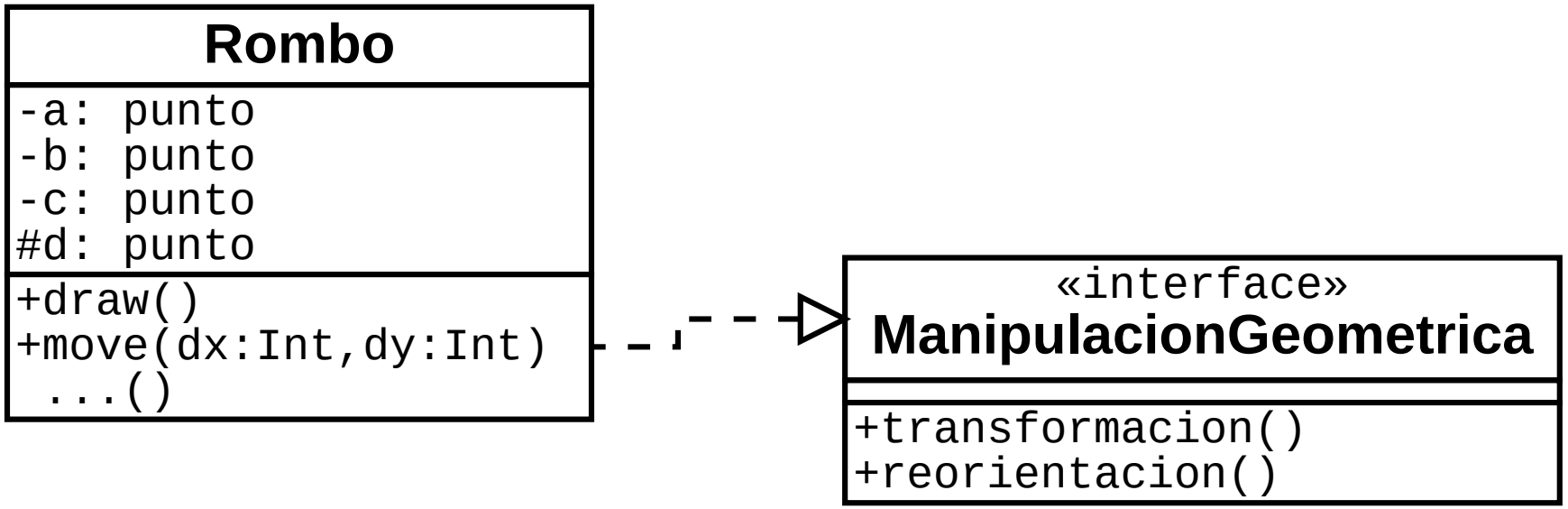
Abstracción: Un interfaz es un conjunto de operaciones.

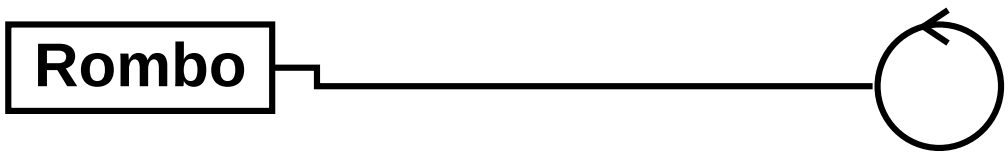
una o más clases pueden *realizar* un interfaz.



Interfaz

Notación omitida:

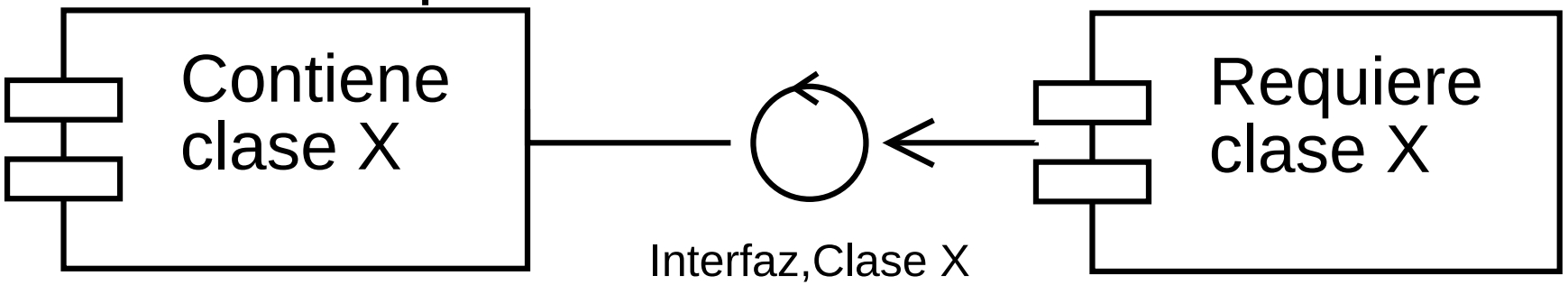




ManipulaciónGeometrica

Componentes, Clases e Interfaces

Una componente es una "biblioteca" de clases.
Hacia el sistema *realiza* el conjunto de interfaces de las clases implementadas en ella.



. . .

Encapsulación, Privacidad

Attributos/operaciones pueden ser:

- + visibles, públicos -
- - privados - solo la clase misma
- # protegidos - solo subclases

Visibilidad de attributos y operaciones