

# UML

## ➤ ¿Qué es?

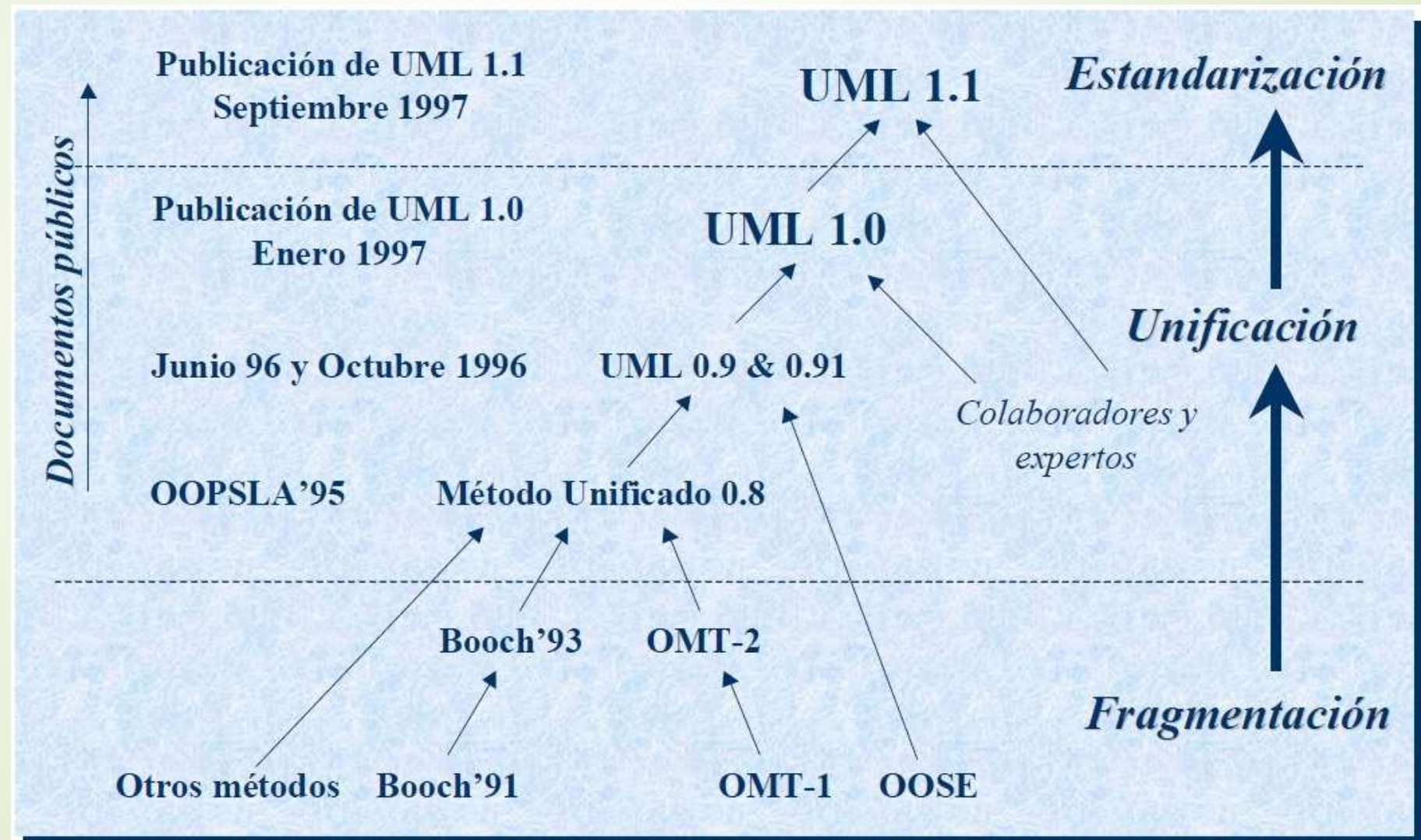
- **UML** (Unified Modeling Language) es un Lenguaje Unificado para la construcción de Modelos.
- Permite **Modelar, Construir y Documentar** todos los elementos que componen un sistema de software.

## ➤ Se origina en el año 94' concebido en su mayor parte por el aporte de 3 prestigiosos autores de otros métodos.:

- **Método de Bocch**, por Bocch.
- **OMT** (Técnica de Modelado de Objetos), por Rumbaugh.
- **OOSE** (Ingeniería de Software OO), por Jacobson.

# UML

## ➤ Evolución...



# UML

## ► ¿Por qué Modelar?

- Un modelo es una simplificación de una realidad compleja.
- Objetivos del modelado:
  1. **Visualizar** como es ó como se desea que sea un sistema.
  2. **Especificar la estructura** o comportamiento de un sistema.
  3. Proporcionar plantillas para que nos guíen en la construcción de un sistema.
  4. **Documentar** las decisiones que se han tomado a lo largo del desarrollo del sistema.
- Conclusión:

Construimos modelos de sistemas complejos por que no podemos comprenderlos, en su totalidad, a simple vista.

# UML

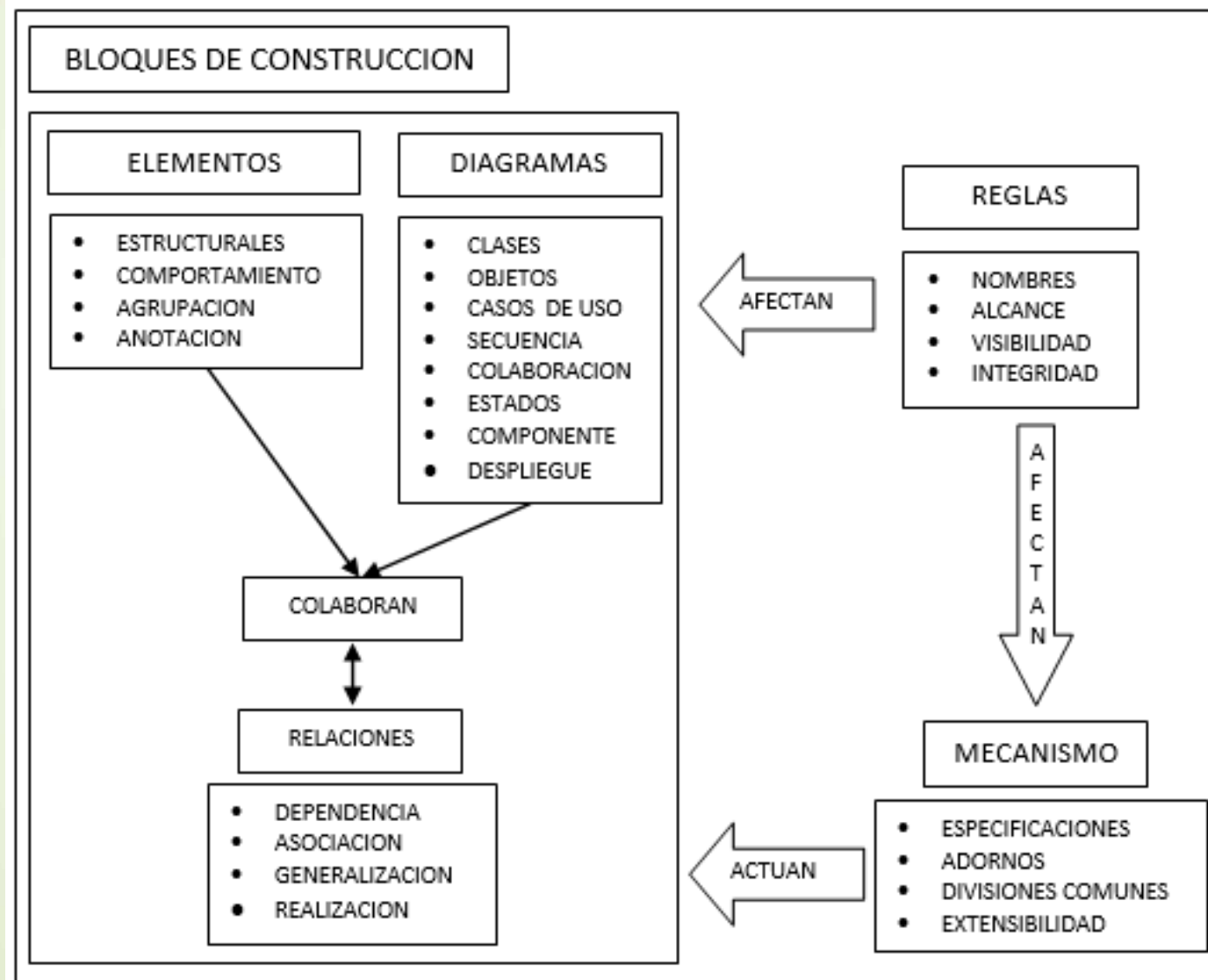
## ► Principios básicos del modelado.

- 1.** La elección de qué modelos crear tiene una profunda influencia sobre cómo se enfrenta un problema y cómo se da forma a la solución.
- 2.** Todo modelo puede ser expresado a diferentes niveles de precisión.
- 3.** Los mejores modelos están ligados a la realidad. Los modelos, cuando se apartan de la realidad, en el afán de simplificarla, no deben esconder los detalles importantes.
- 4.** Un único modelo no es suficiente. Cualquier sistema no trivial se entiende mejor a través de un pequeño conjunto de modelos casi independientes.



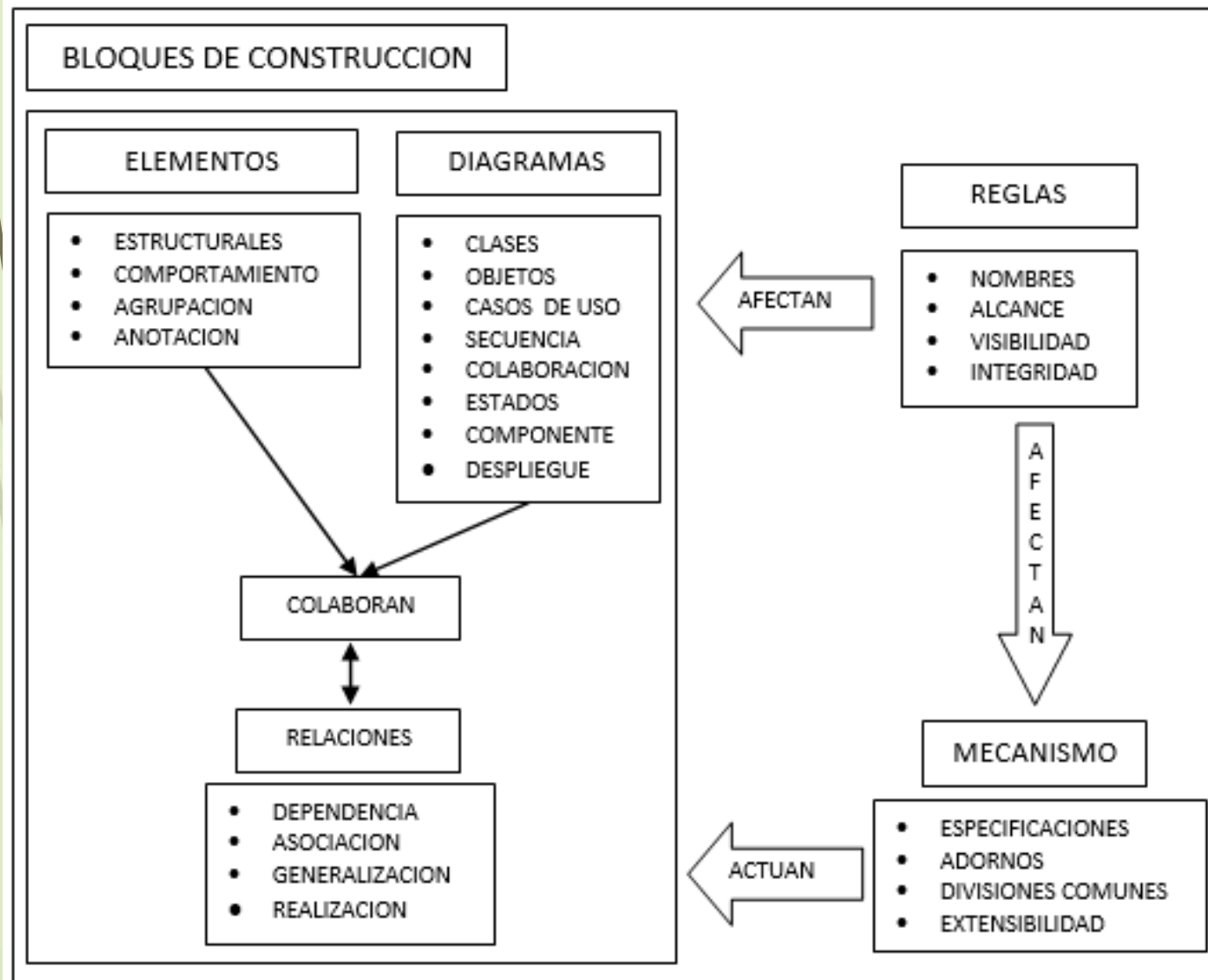
# UML

## Elementos básicos de UML.



# UML

## ► Bloque de Construcción **ELEMENTOS**



### ► Elementos Estructurales:

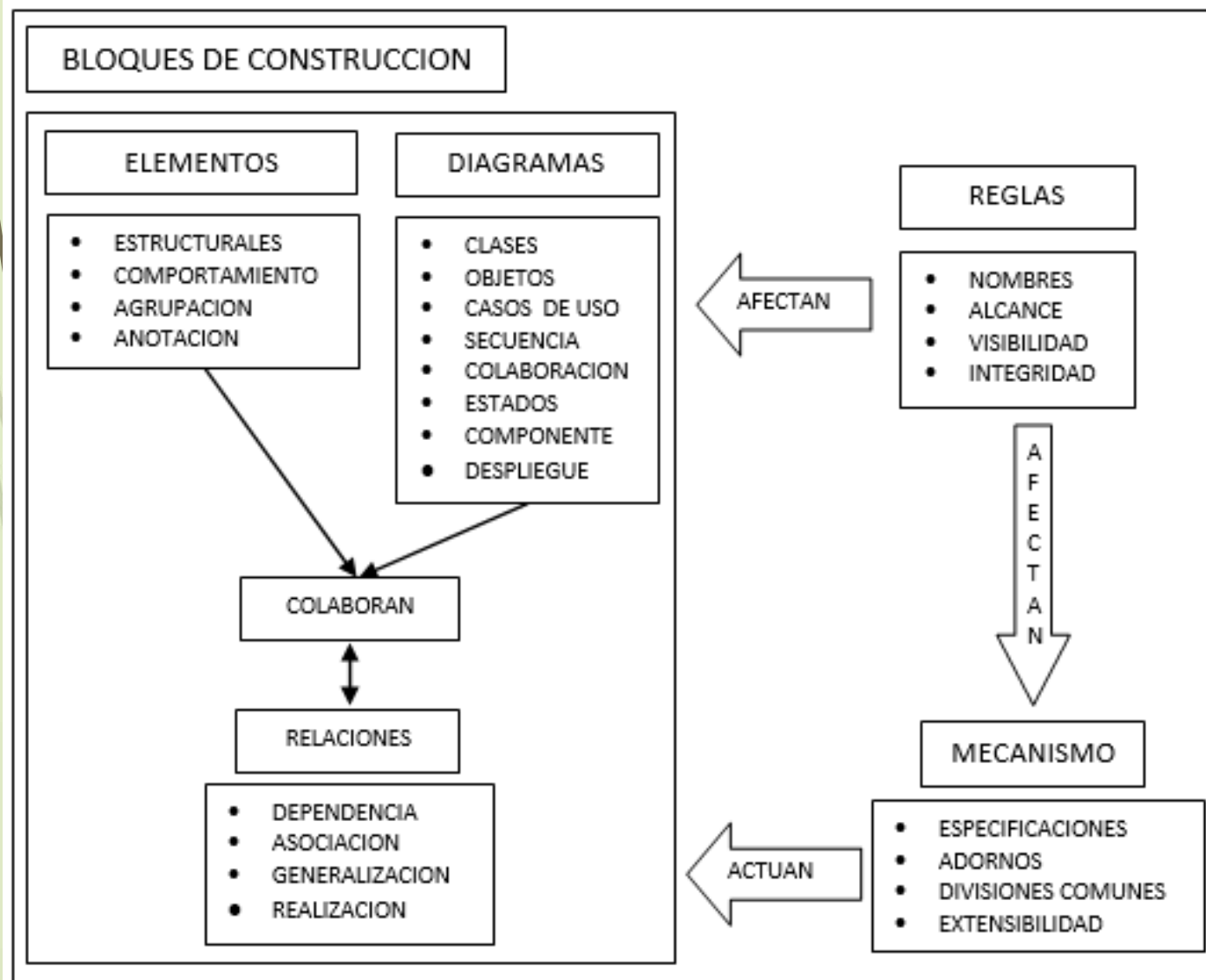
- Objetos
- Clases
- Casos de Uso
- Interfaces
- Componentes
- Nodos

### ► Elementos de Comportamiento:

- Interacciones
- Maquinas de estado

# UML

## ► Bloque de Construcción **ELEMENTOS**



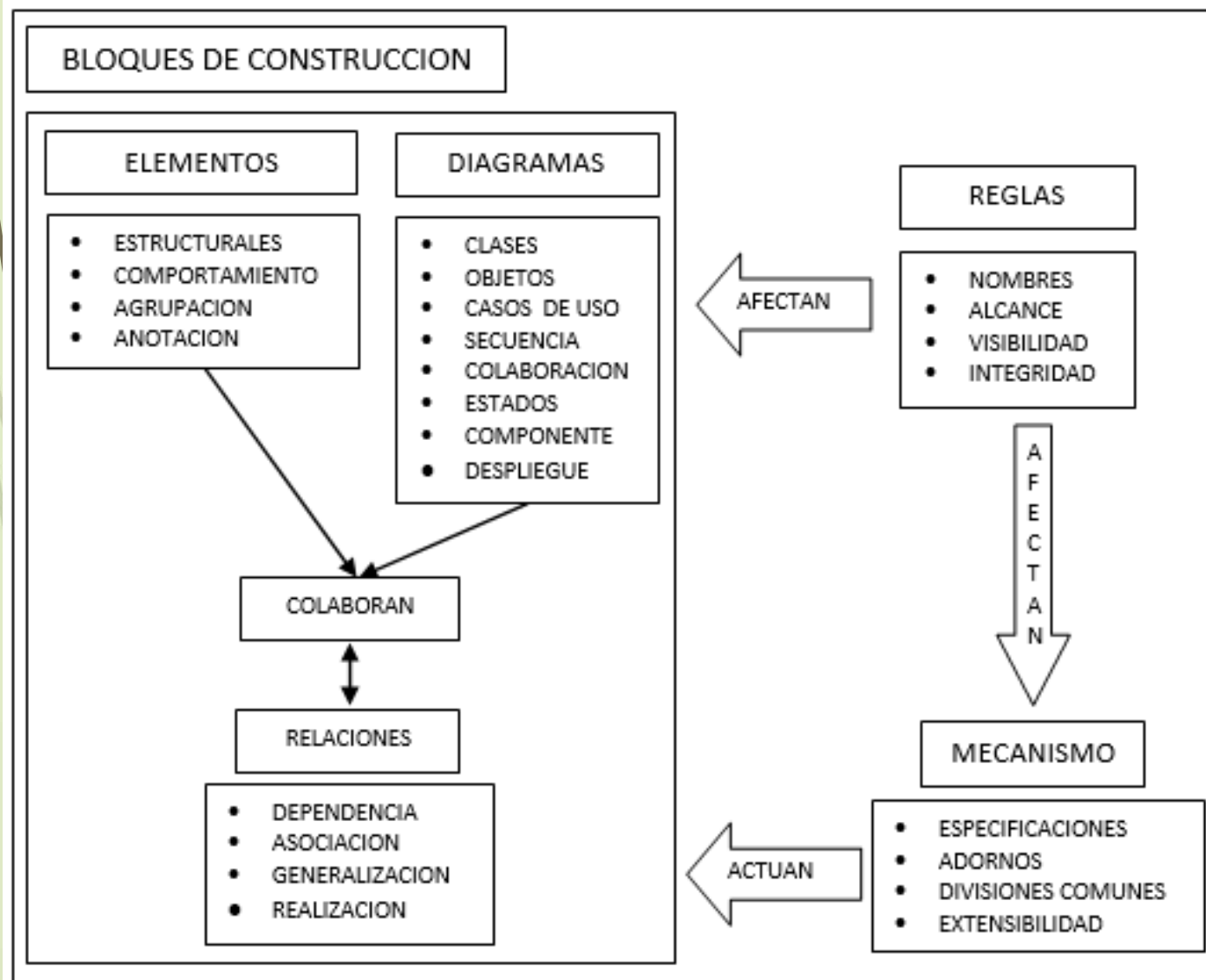
## ► Elementos de Agrupación:

- Paquetes

## ► Elementos de Anotación:

- Notas

## ► Bloque de Construcción **RELACIONES**



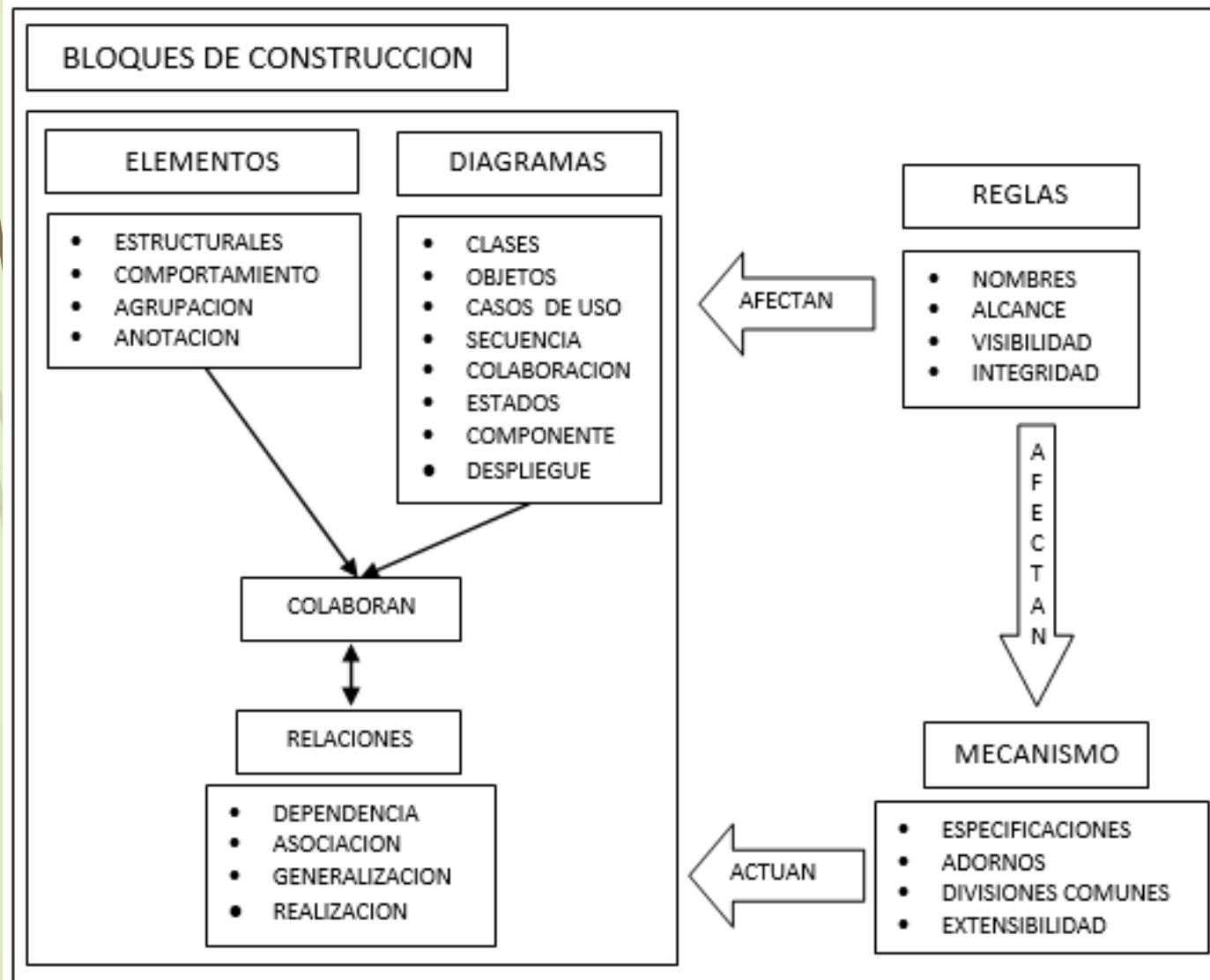
### ► Relaciones:

- Dependencia
- Generalización
- Asociación
- Realización



# UML

## ► Bloque de Construcción **DIAGRAMAS**



### ► Diagramas:

- De clases
- De objetos
- De casos de uso
- De secuencia
- De colaboración
- De estados
- De componentes
- De despliegue

# UML

## ➤ Elementos **ESTRUCTURALES: OBJETOS**

**Objetos:** Donde se indica su identidad, estado y comportamiento

nombre\_de\_objeto: nombre\_de\_clase

Nombre del  
objeto

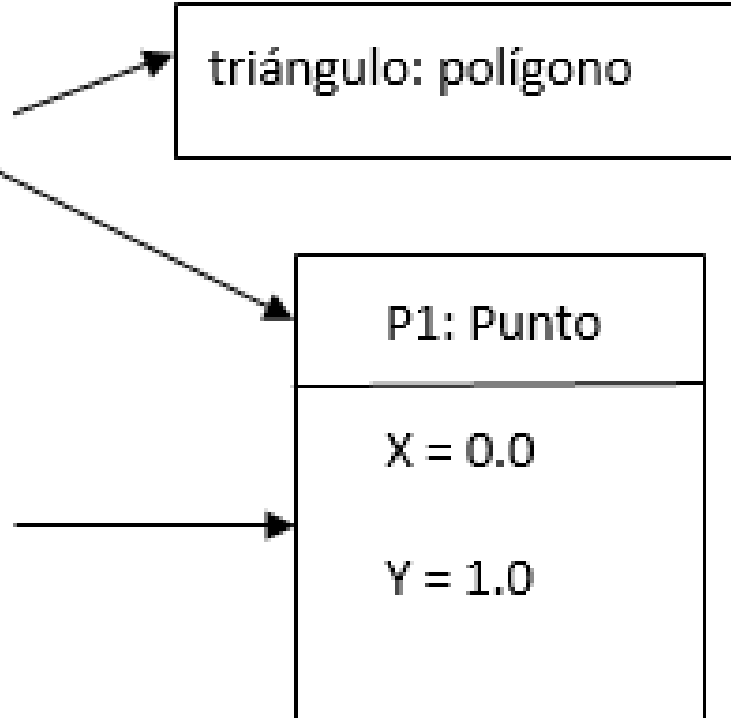
triángulo: polígono

Valores de  
atributo

P1: Punto

X = 0.0

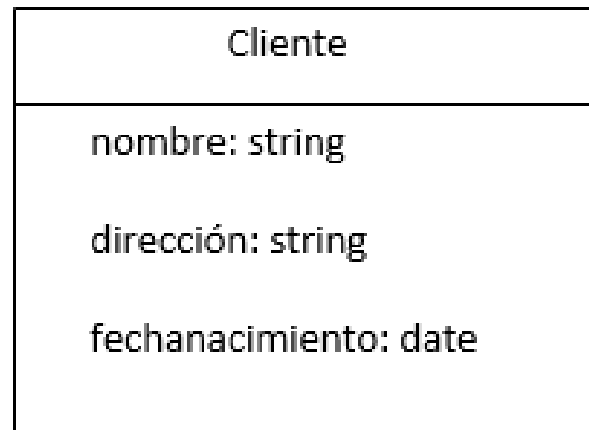
Y = 1.0



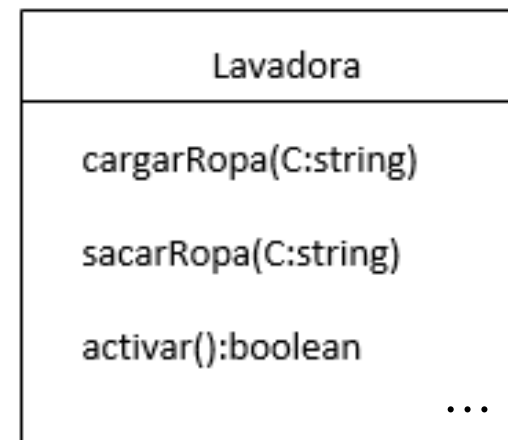
# UML

## ➤ Elementos **ESTRUCTURALES: CLASES**

Se pueden indicar la clase y sus atributos:



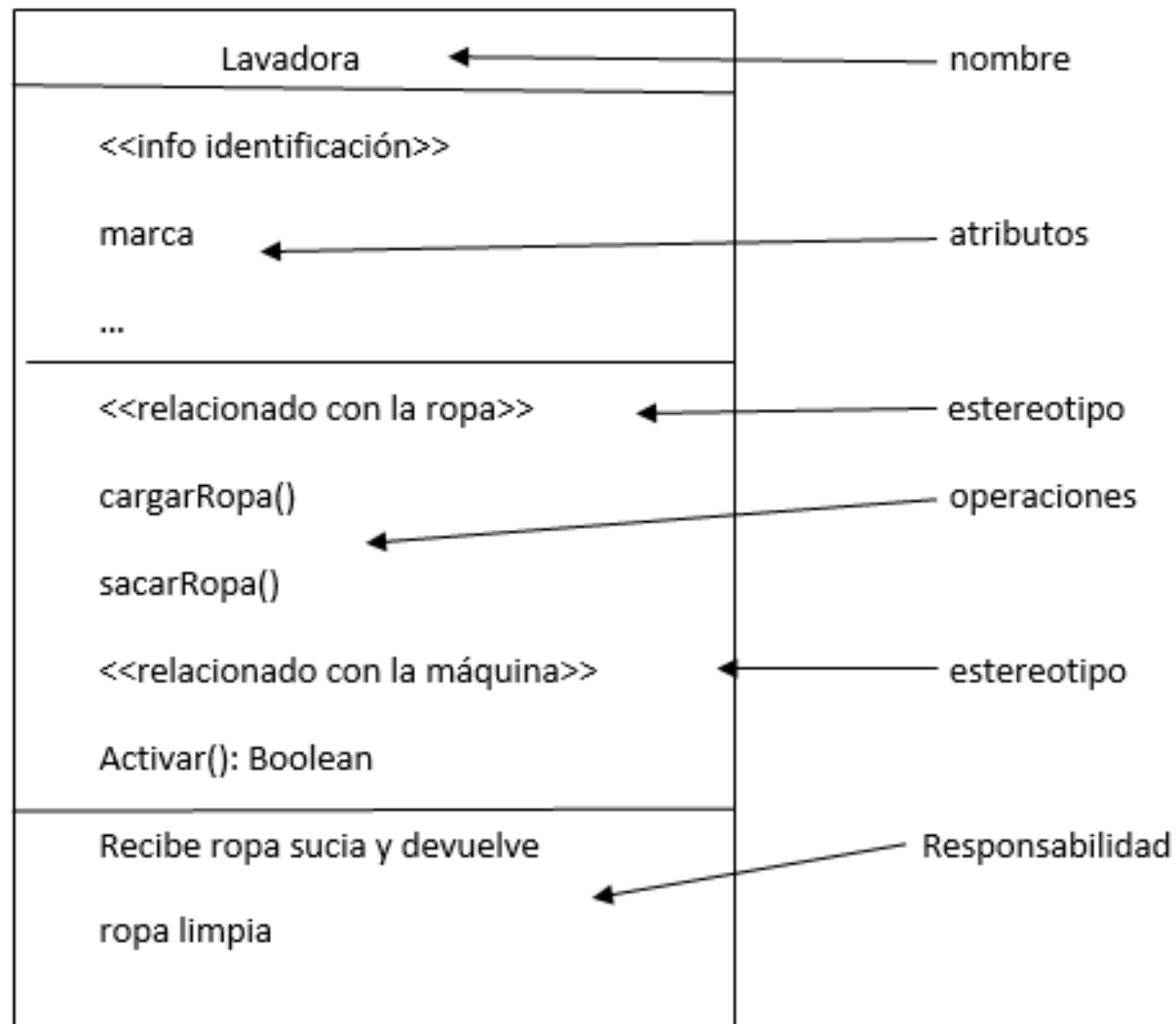
La clase y sus operaciones:



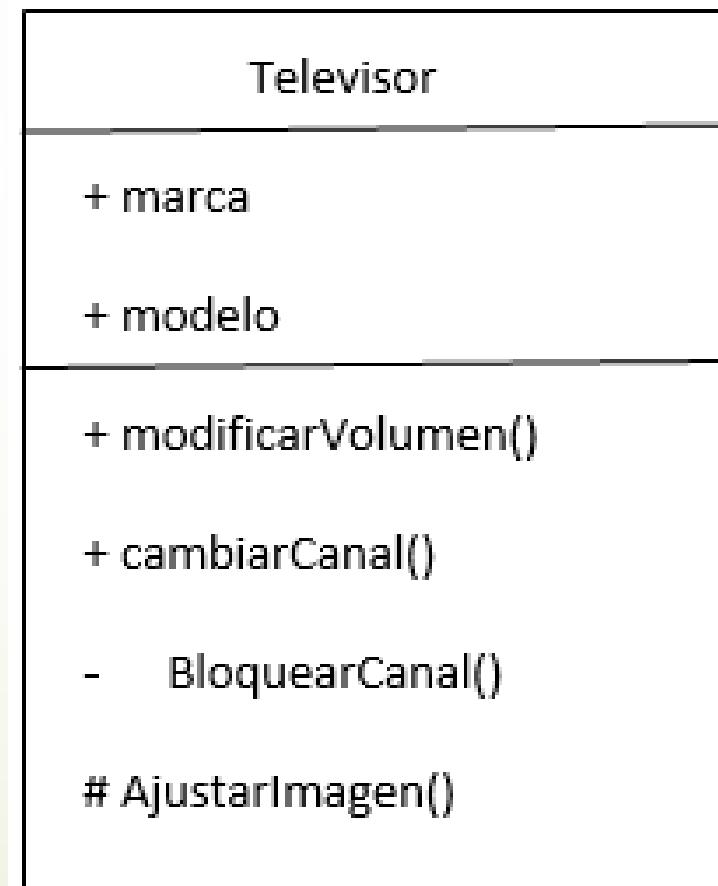
# UML

## Elementos **ESTRUCTURALES: CLASES**

Se puede usar estereotipos

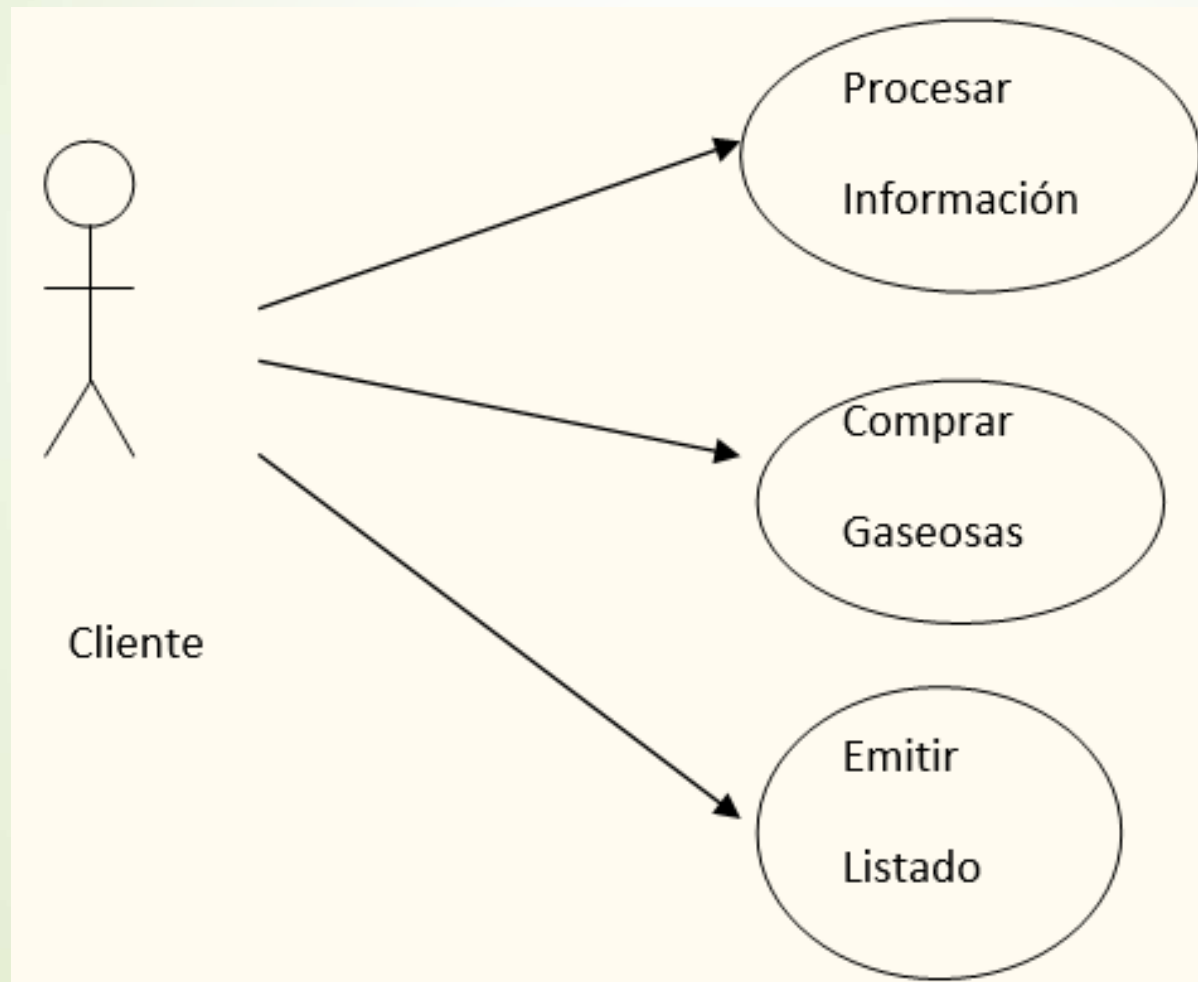


Se puede usar un marcador para indicar visibilidad: **+**(público), **-**(privado) y **#**(protegido).



# UML

## Elementos **ESTRUCTURALES**: CASOS DE USO



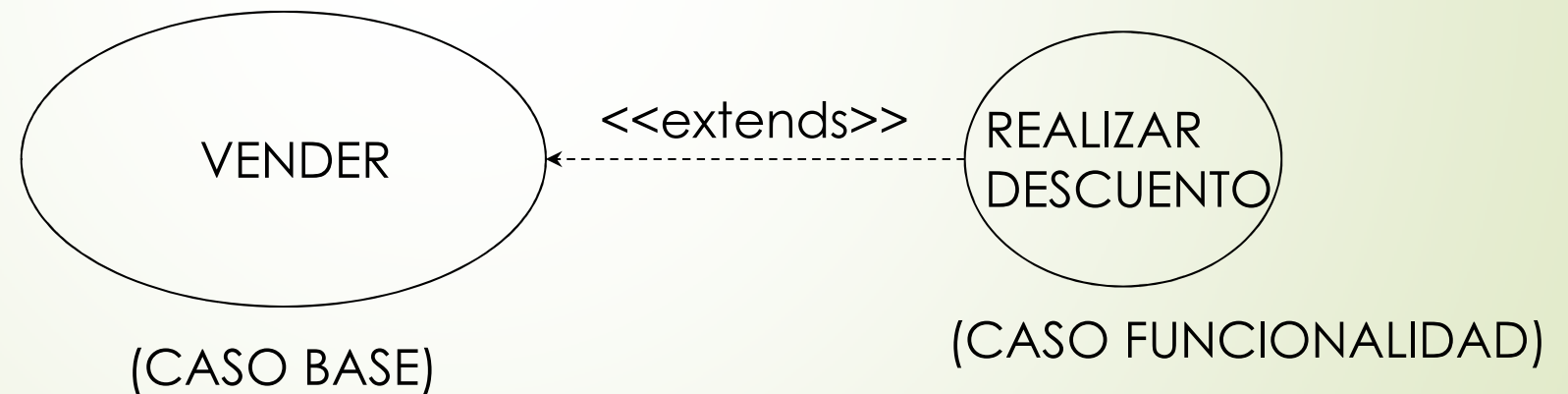


# UML

## ■ Elementos **ESTRUCTURALES: CASOS DE USO**

### • **Relación de EXTENSION:**

- Extiende funcionalidad.
- No es esencial

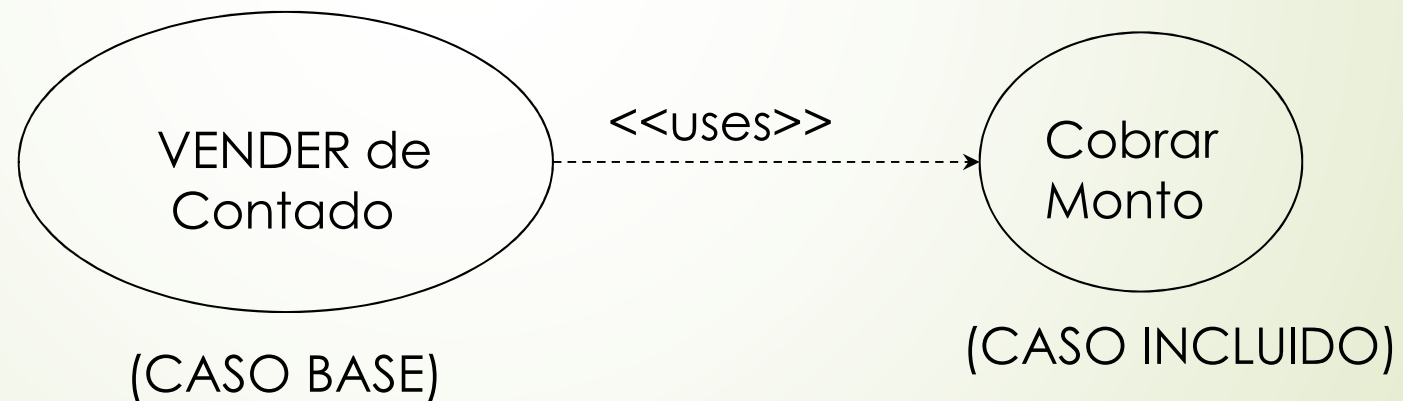


# UML

## ■ Elementos **ESTRUCTURALES: CASOS DE USO**

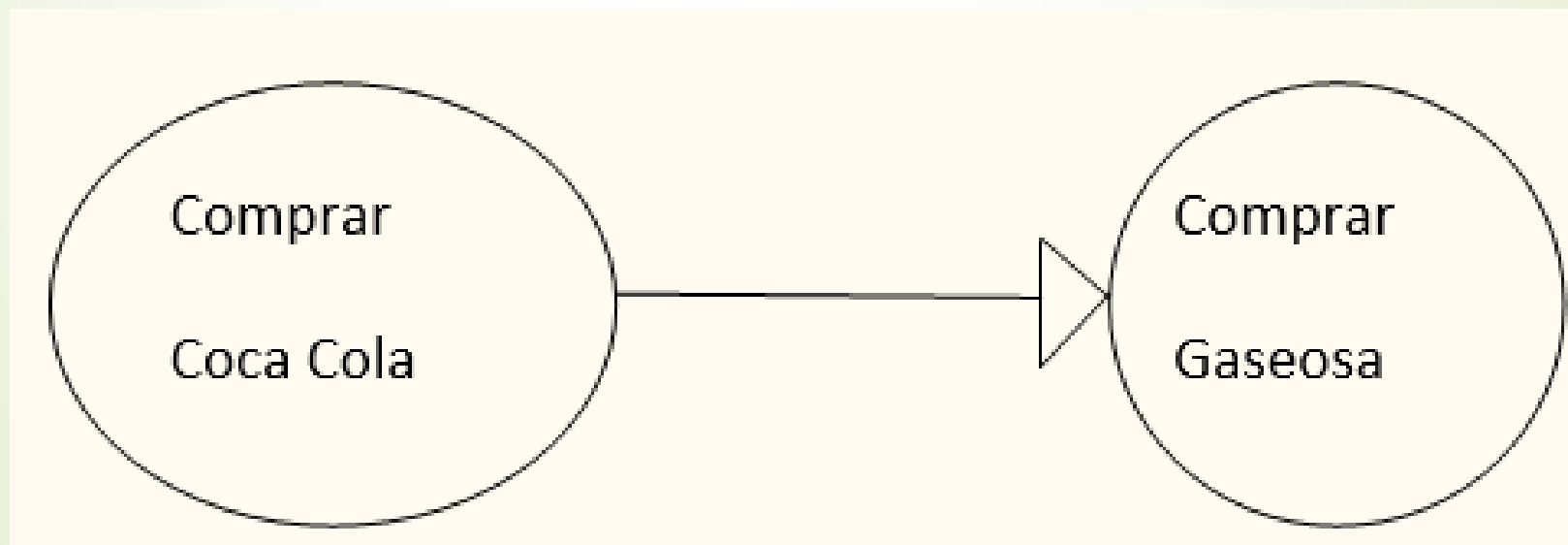
### • **Relación de INCLUSION:**

- Es esencial, el 1º no podría existir sin el caso de uso incluido.



# UML

- Elementos **ESTRUCTURALES: CASOS DE USO**
- **Relación de GENERALIZACION:**
  - Especifica una relación de generalización/especialización (HERENCIA).

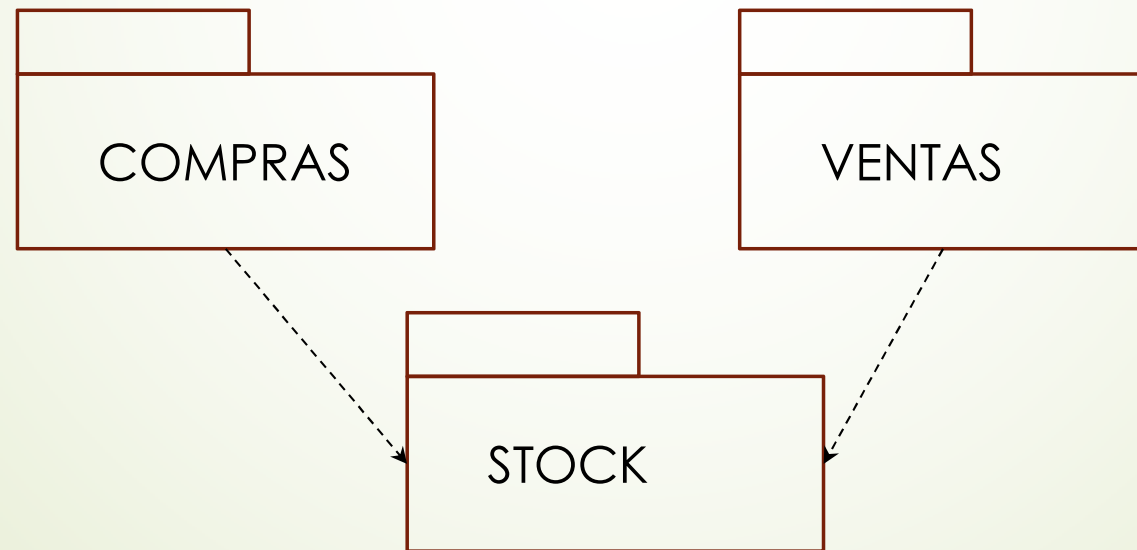


# UML

## ➤ Elementos **ESTRUCTURALES: CASOS DE USO**

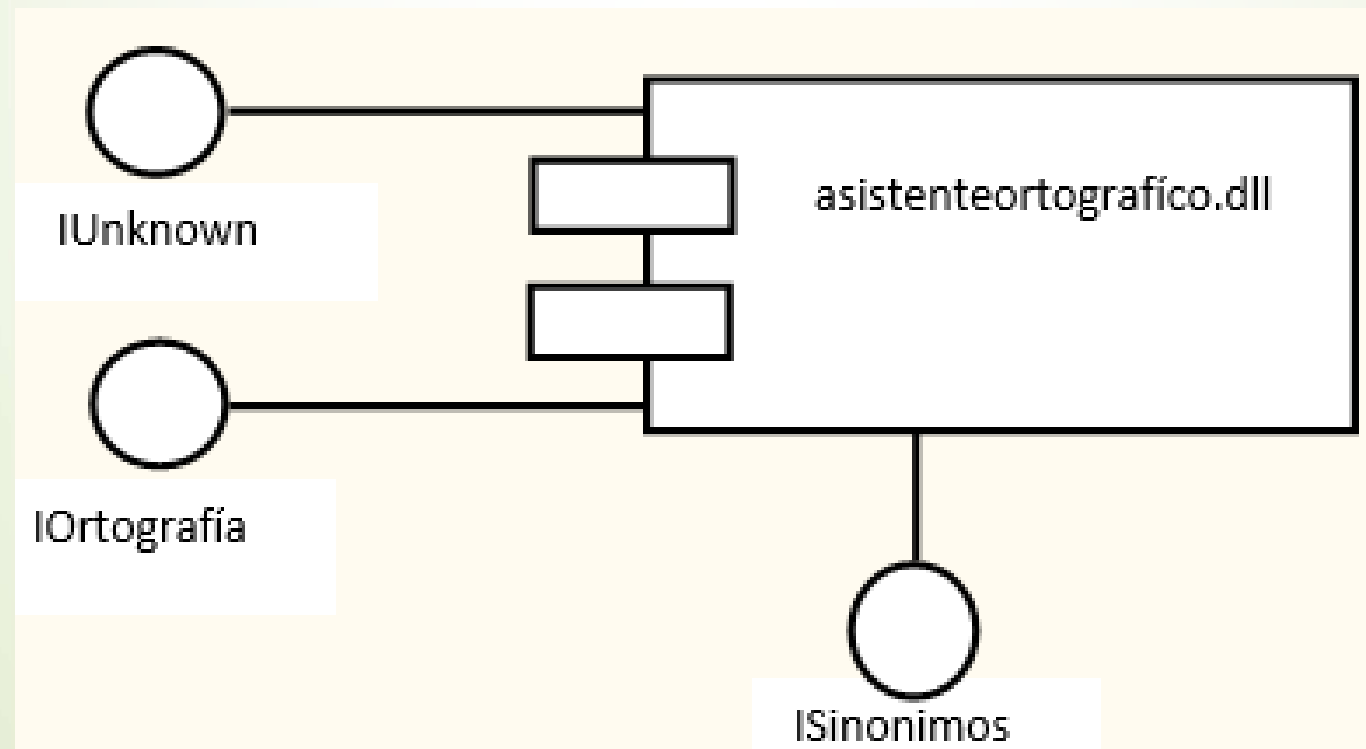
### • **Relación de AGRUPAMIENTO:**

- Los casos de uso se pueden AGRUPAR en paquetes de casos de uso relacionados.



# UML

## ➤ Elementos **ESTRUCTURALES: INTERFACES**





# UML

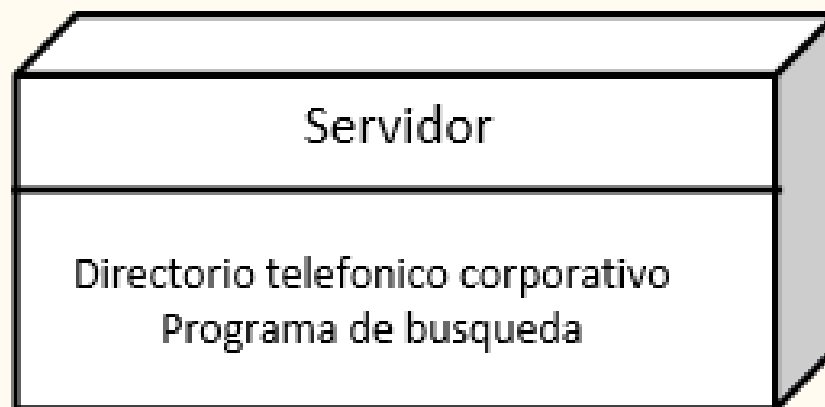
## ■ Elementos **ESTRUCTURALES: COMPONENTES**

- Un componente representa la **parte física y reemplazable** de un sistema. (Tablas, archivos, etc).
- Pueden encontrarse componentes que sean **artefactos** del proceso de desarrollo.
- En UML todos los componentes físicos se modelan como componentes.



# UML

- Elementos **ESTRUCTURALES: NODO**
  - Es un elemento físico que existe en **tiempo de ejecución** y representa un recurso computacional.
  - Se representa como un cubo con un nombre y puede incluir los componentes que residen en él.



# UML

## ➤ Elementos **ESTRUCTURALES: NODO**

- Es un elemento físico que existe en **tiempo de ejecución** y representa un recurso computacional.
- Se representa como un cubo con un nombre y puede incluir los componentes que residen en él.

