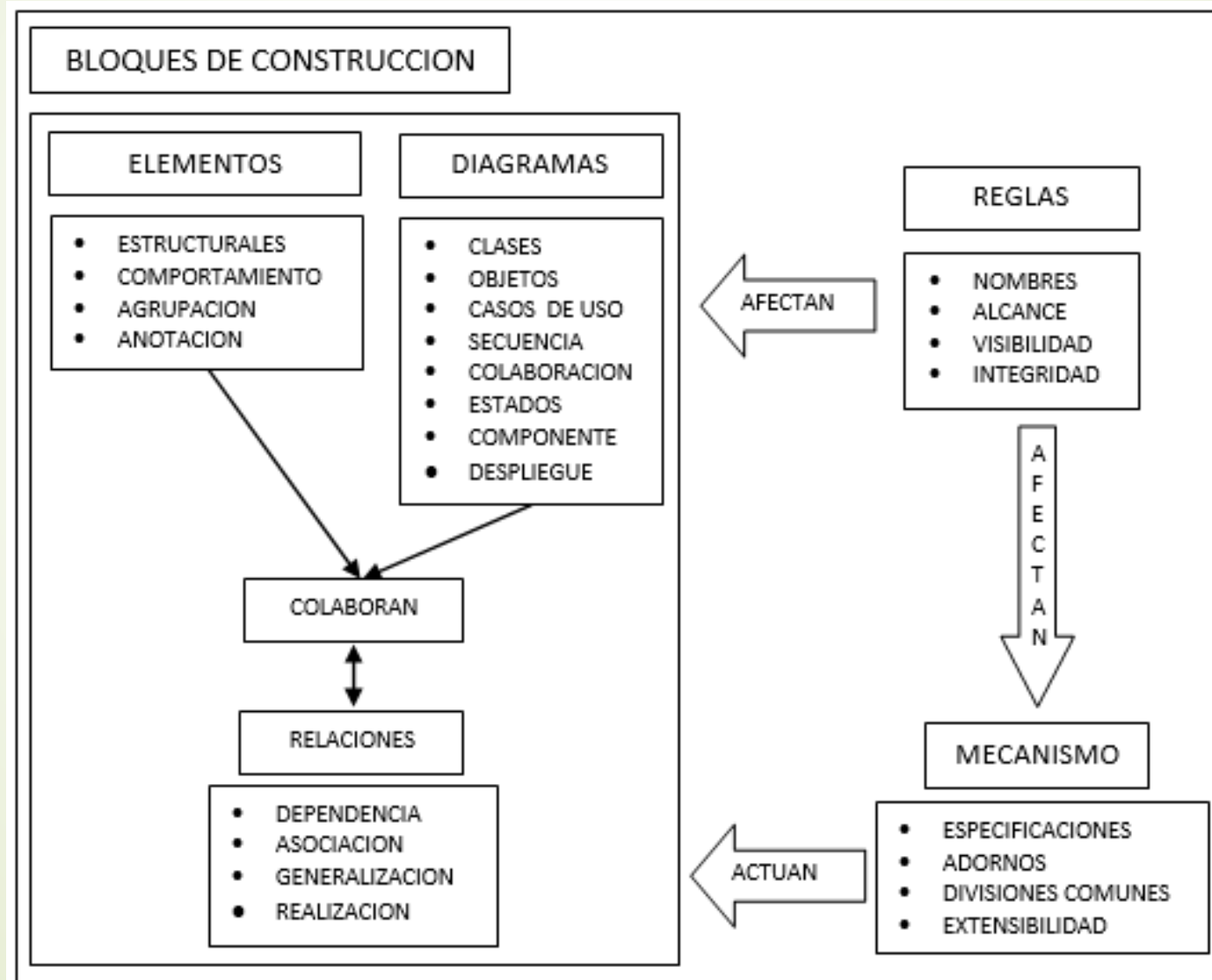


# UML

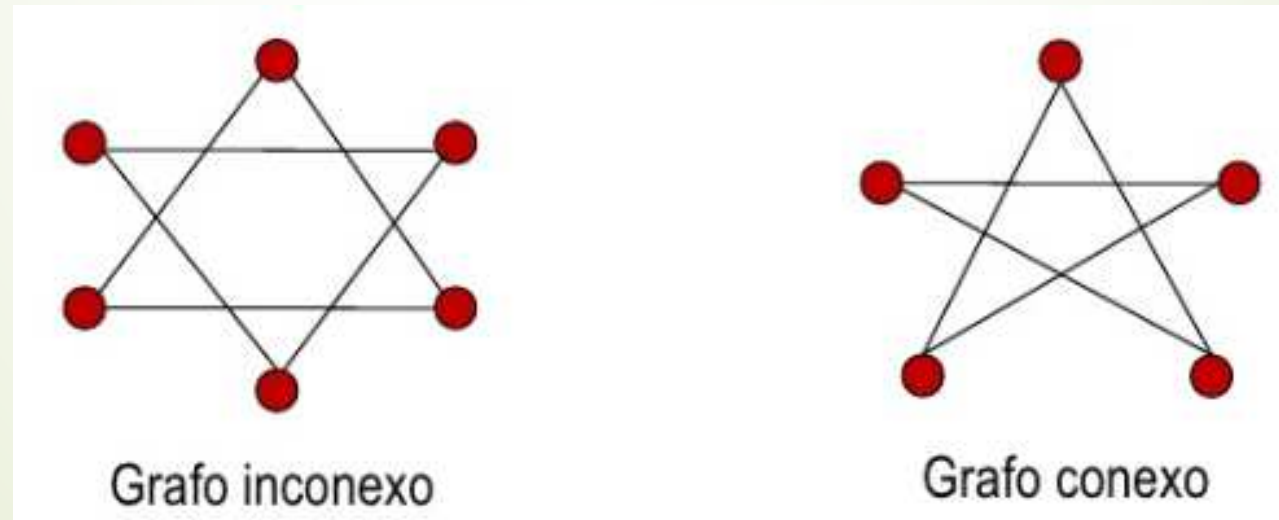
## Elementos Básicos de UML



# UML

## ► Diagramas

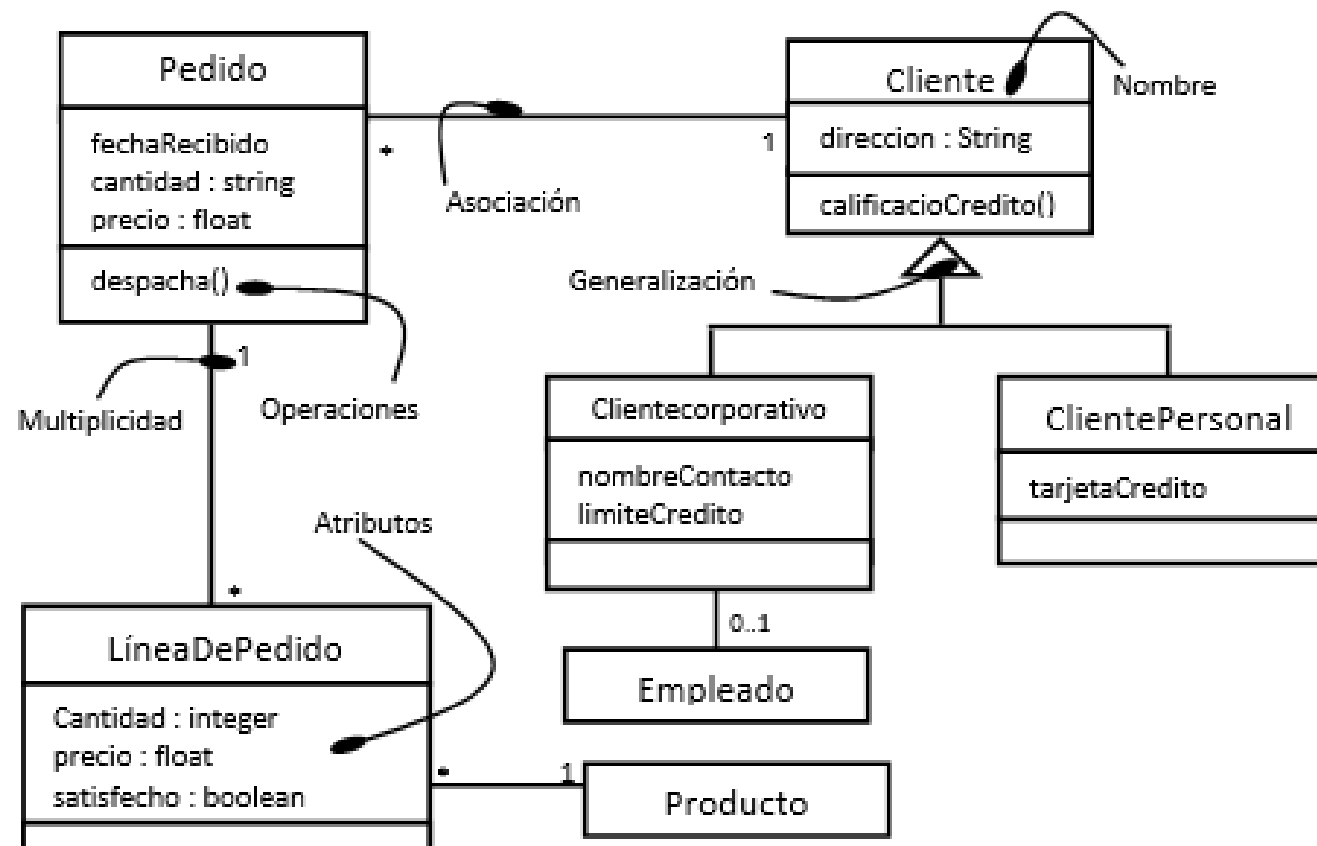
- Combinan diversos elementos gráficos visualizados la mayoría de las veces como un grafo conexo de nodos(elementos) y arcos (relaciones).
- Los modelos describen lo que hará un sistema pero no como implementarlo.



# UML

## Diagrama de **CLASES**

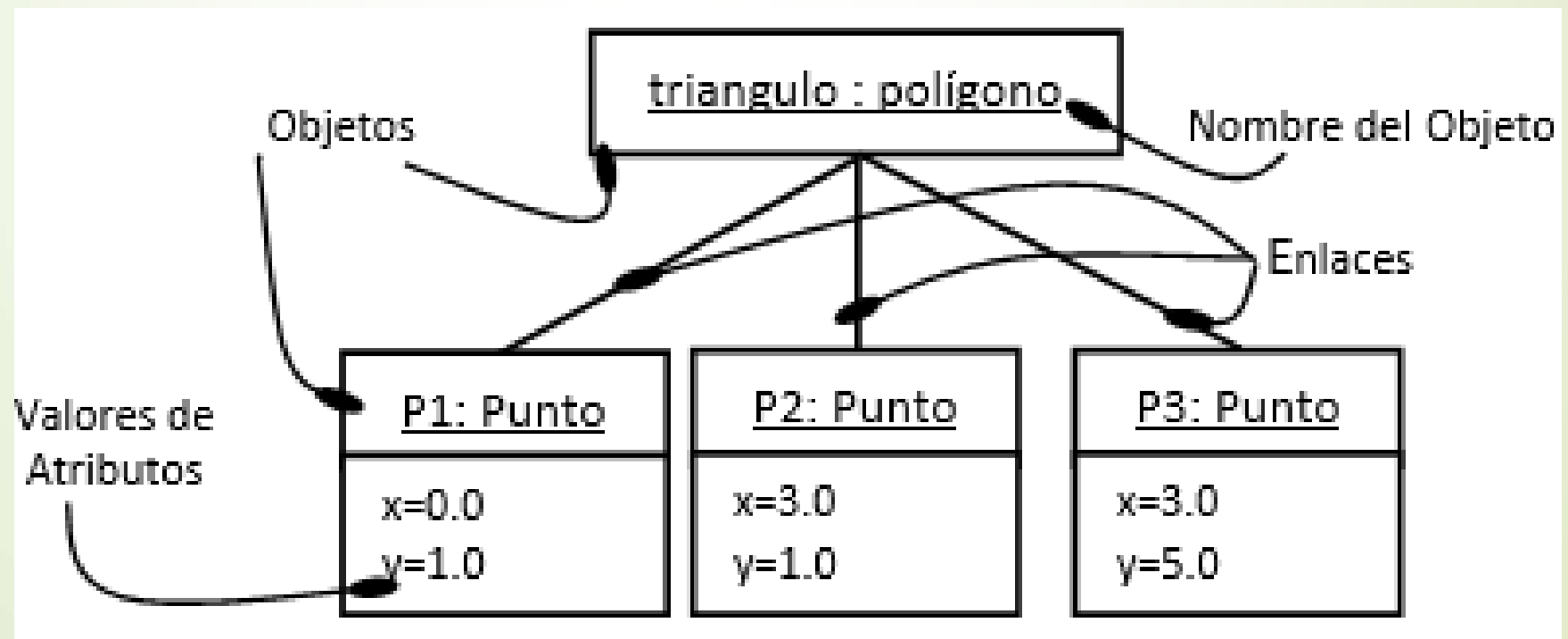
- Facilitan las representaciones del sistema para luego poder ser discutido con el cliente durante el proceso de análisis.
- Describen la vista de diseño estática de un sistema.



# UML

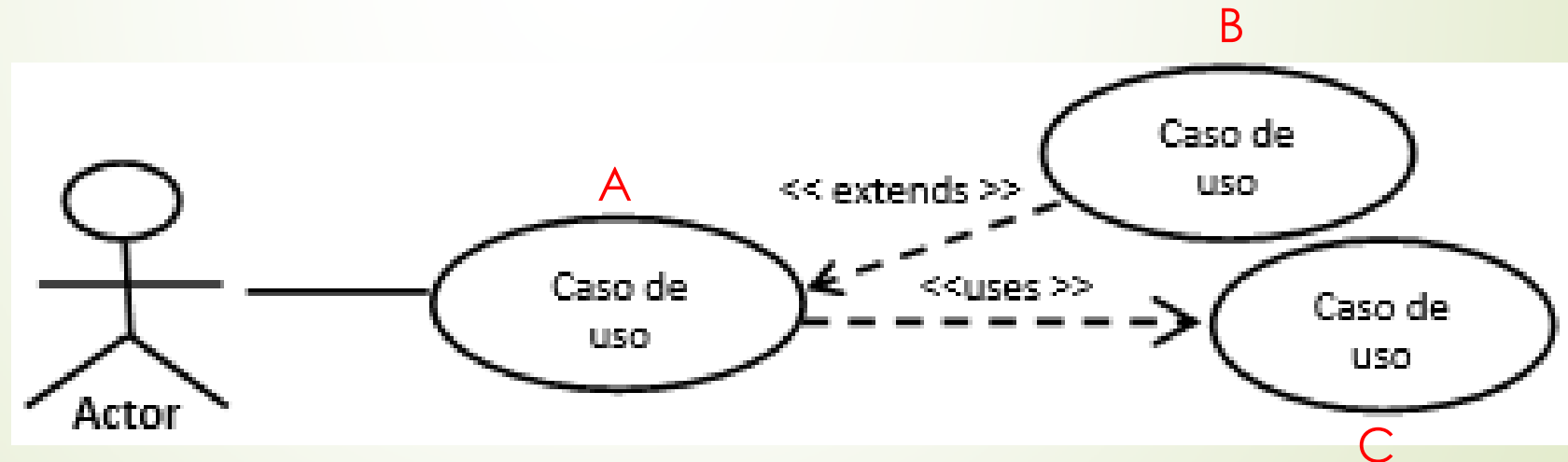
## Diagrama de **OBJETOS**

- Un objeto es una instancia de una clase.
- Los diagramas de objetos muestran un conjunto de objetos y sus relaciones. Estos diagramas representan “**fotografías instantáneas**” de instancias de los elementos que se encuentran en un diagrama de clase.



# UML

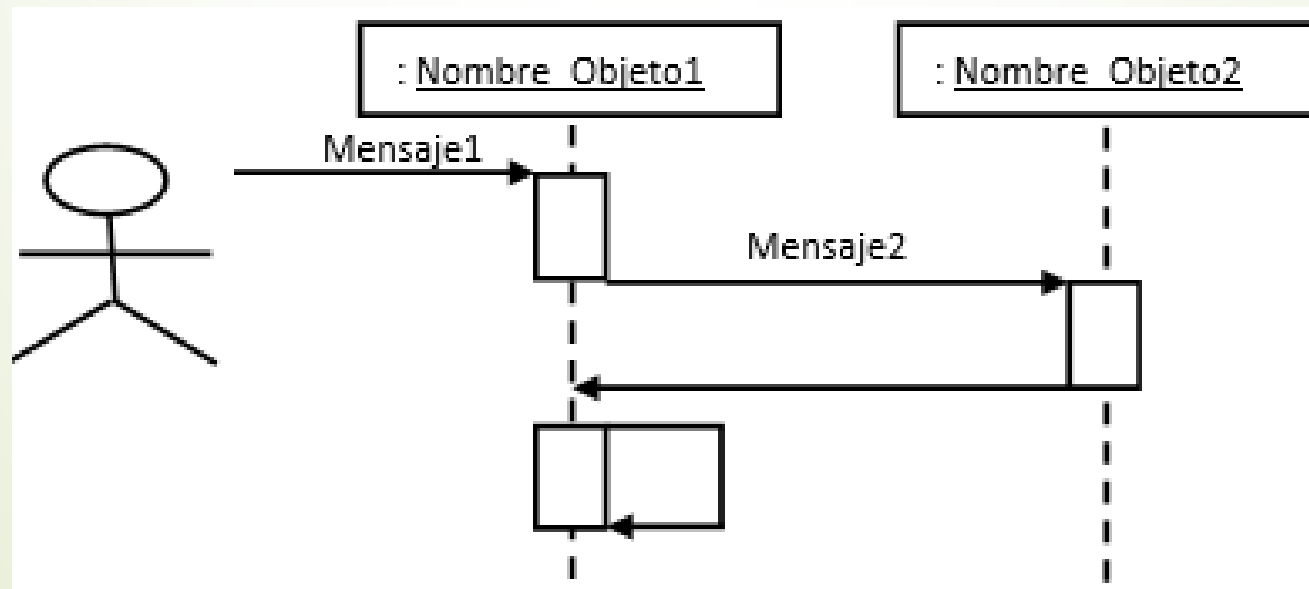
- Diagrama de **CASOS DE USO**
- Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario.



# UML

## ► Diagrama de **SECUENCIAS**

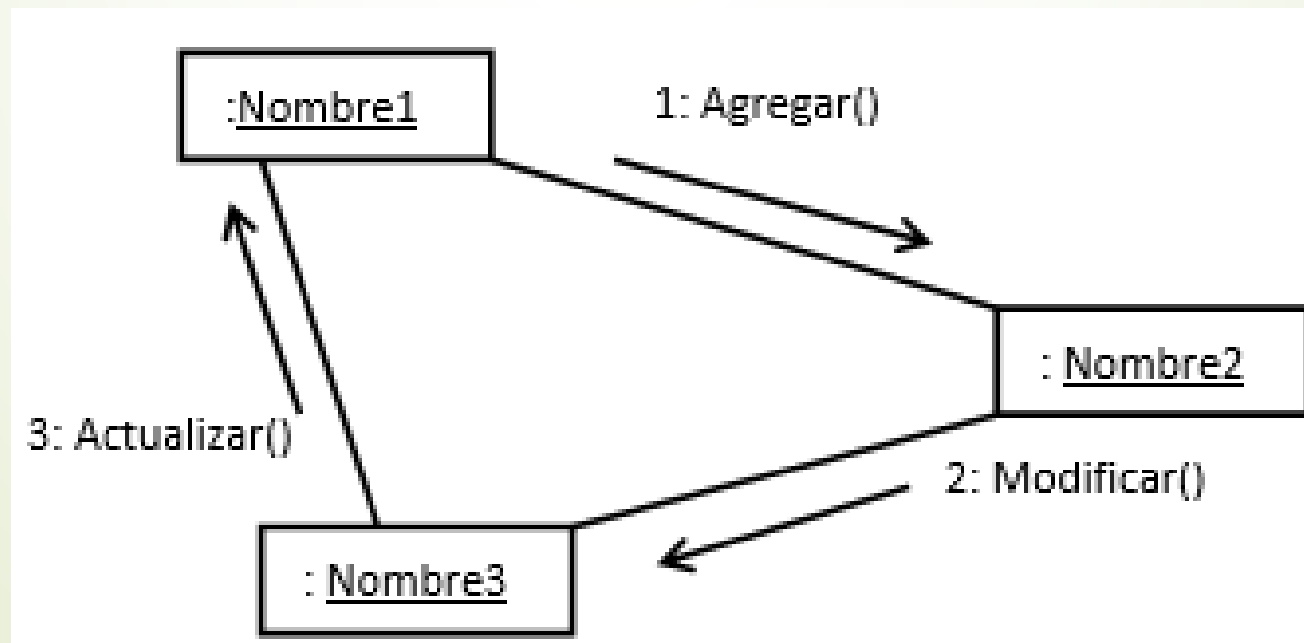
- Los diagramas de secuencias muestran la mecánica de la interacción con base en tiempo, haciendo uso de mensajes y representando acciones.



# UML

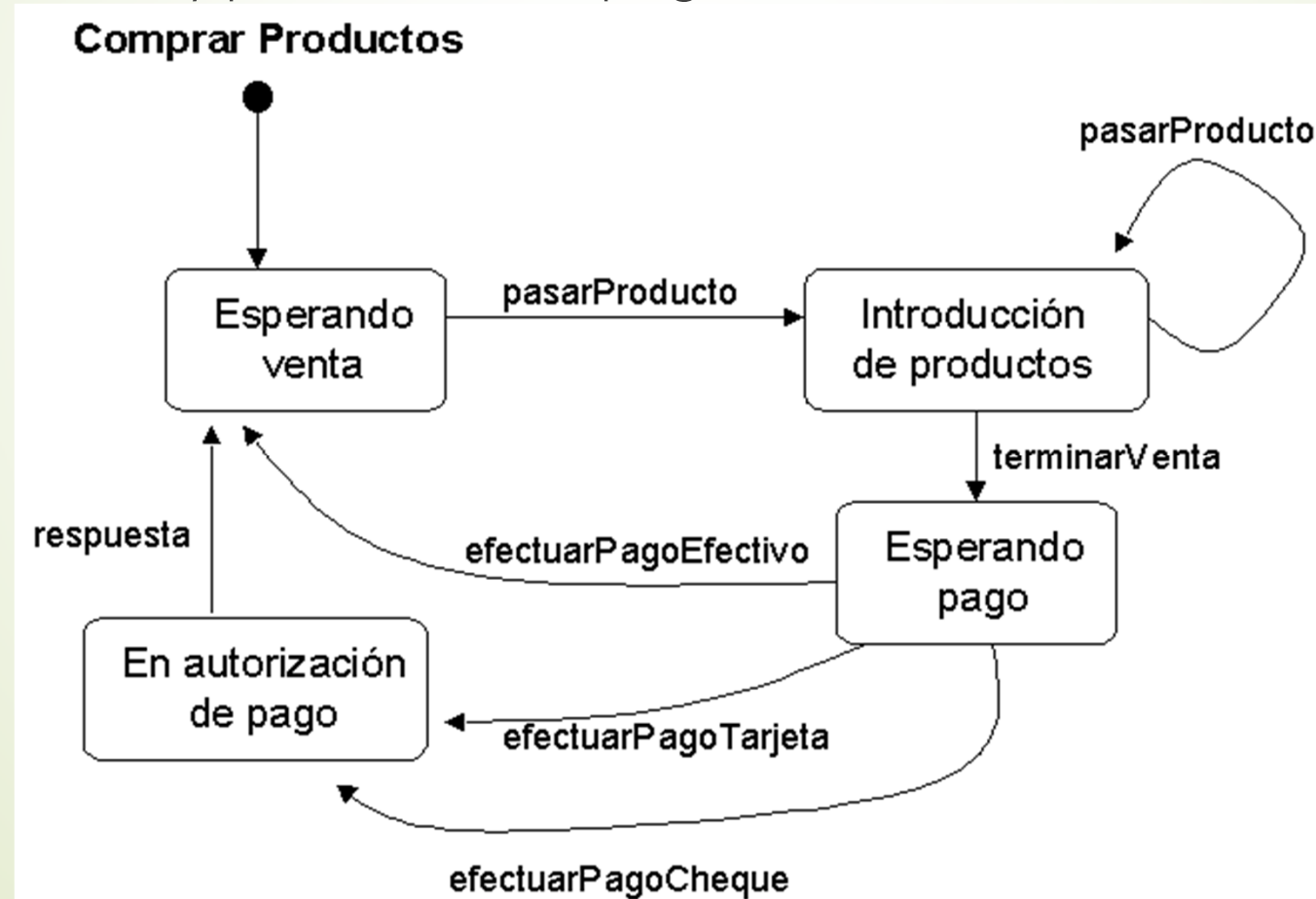
## ► Diagrama de **COLABORACION**

- Los **diagramas de secuencias** y los de **colaboración** son un tipo de diagrama de interacción.
- Los diagramas de secuencia y los de colaboración son **isomórficos**, es decir, se pueden transformar el uno en el otro.



## Diagrama de **ESTADOS**

- **Ejemplo:** Un cliente Compra productos, los pasa por sistema, termina la venta, y posteriormente paga.

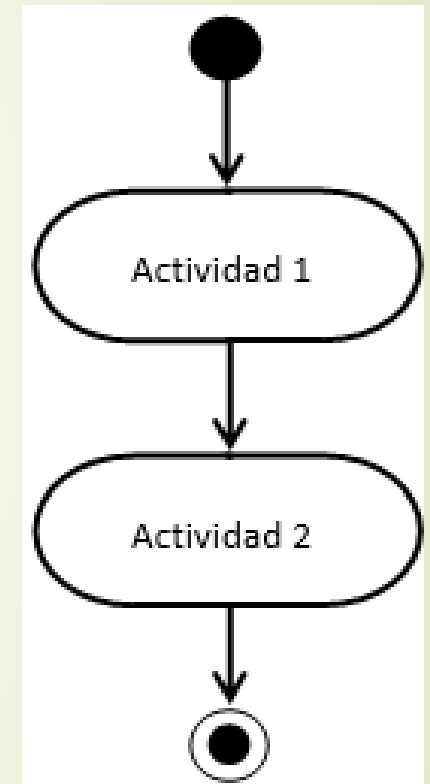




# UML

## ► Diagrama de **ACTIVIDADES**

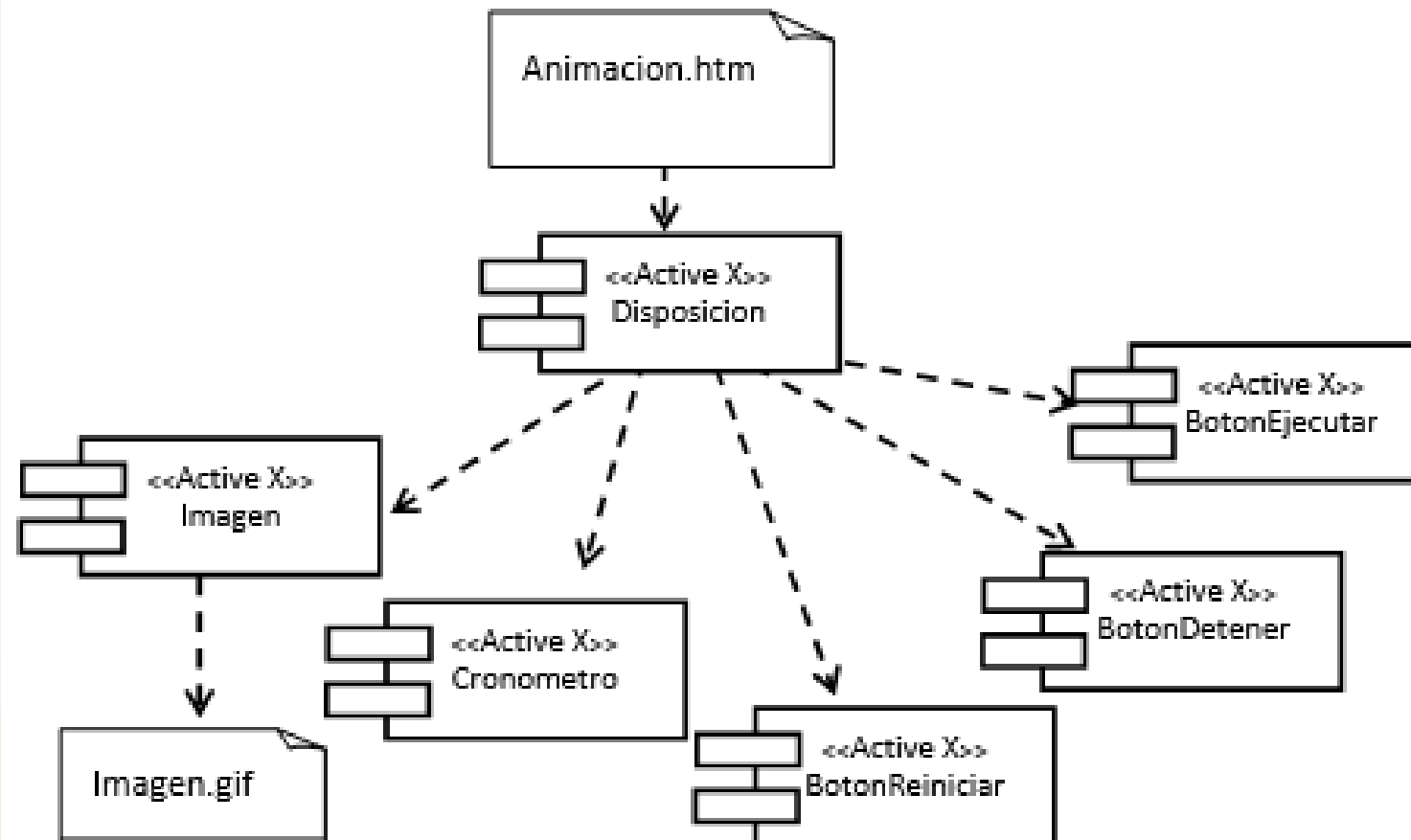
- Un diagrama de actividades es un tipo especial de diagrama de estados que muestra el flujo de actividades dentro de un sistema.
- Enfatizan el flujo de control entre los objetos.



# UML

## ► Diagrama de **COMPONENTES**

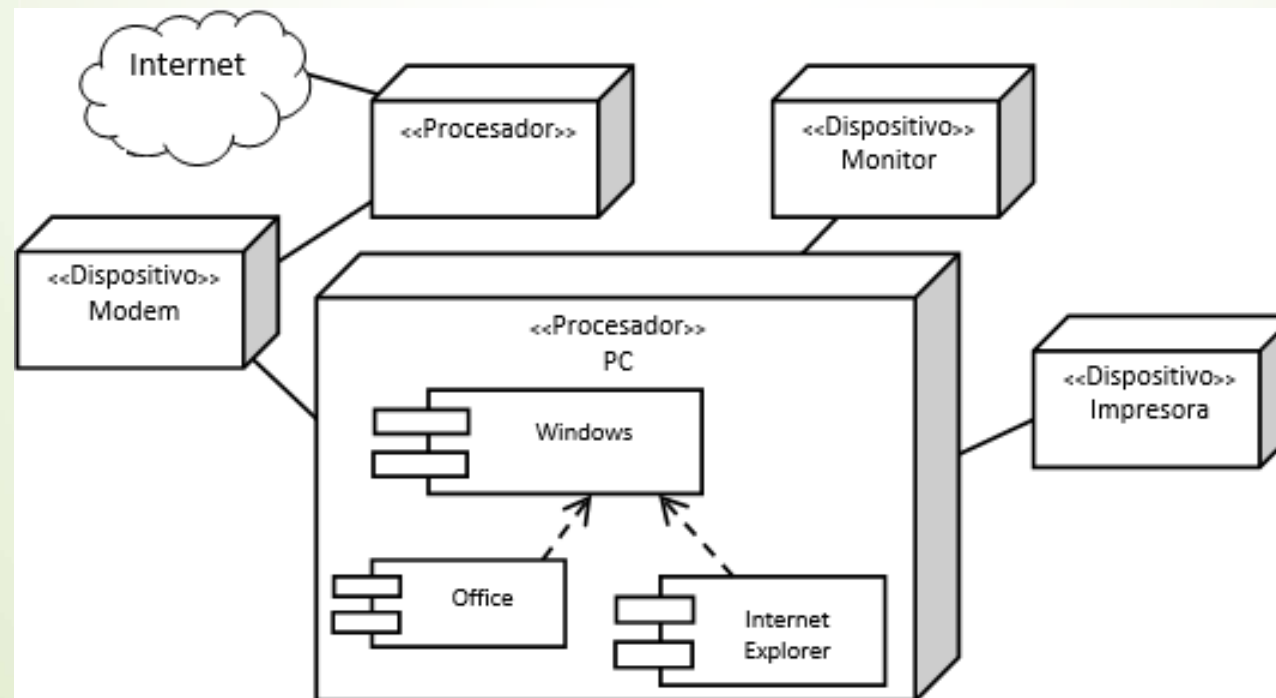
- Muestra las organizaciones y dependencias entre un conjunto de componentes.



# UML

## ► Diagrama de **DESPLIEGUE**

- Muestra la **arquitectura física** de un **sistema informático**.
- Muestra la configuración de los nodos que se procesan en tiempo de ejecución y dentro de ellos los componentes que participan.



# UML

## ➤ Reglas

- Los bloques de construcción del UML no se pueden combinar de forma aleatoria.
- Un **modelo bien formado** es aquel que es semánticamente auto consistente y está en armonía con el resto de los modelos con los que se relaciona.
- El UML posee reglas semánticas para:
  - **Nombres:** ¿Cómo llamar a los elementos, relaciones y diagramas?
  - **Ámbito:** ¿Cuál es el contexto que da un significado específico a un nombre?
  - **Visibilidad:** ¿Cómo esos nombres van a ser vistos y usados por otros?
  - **Integridad:** ¿Cómo los elementos se relacionan con otros apropiada y consistentemente?
  - **Ejecución:** ¿Qué significa realmente ejecutar o simular un modelo dinámico?

# UML

## ► Reglas

- Pero el equipo de desarrollo no sólo construye modelos bien formados, sino que también realiza modelos con las siguientes características:
  - **Abreviados:** Ciertos elementos se ocultan para simplificar la vista
  - **Incompletos:** Pueden estar ausentes ciertos elementos
  - **Inconsistentes:** No se garantiza la integridad del modelo

# UML

## ► Mecanismos Comunes

- La construcción de los bloques del UML resulta más sencilla y más armoniosa, si se realiza de acuerdo a un **patrón de características comunes**.
- En UML se aplican de forma consistente 4 mecanismos comunes:
- **Especificaciones:**
  - Se puede obtener detalles de un sistema a partir de su representación grafica.

# UML

## ► Mecanismos Comunes

- **Adornos:**

- Es una representación visual de los aspectos mas relevantes de un elemento.

- **Divisiones Comunes:**

- Dicotomía Clase/Objeto
- Dicotomía Interfaz/Implementación



# UML

## ► Mecanismos Comunes

- **Mecanismos de Extensibilidad:**

- UML proporciona un lenguaje estándar para escribir modelos de software, pero no es posible para un lenguaje cerrado expresar todos los detalles de todos los modelos en todos los dominios a lo largo de todo el tiempo.
- Se presentan 3 mecanismos de extensibilidad:
  - Estereotipo
  - Valores etiquetados
  - Restricción



# UML

## ► Mecanismos Comunes

- **Mecanismos de Extensibilidad:**

- UML proporciona un lenguaje estándar para escribir modelos de software, pero no es posible para un lenguaje cerrado expresar todos los detalles de todos los modelos en todos los dominios a lo largo de todo el tiempo.
- Se presentan 3 mecanismos de extensibilidad:
  - Estereotipo



Diagram illustrating a UML metaclass element. It consists of a rectangular box with a double border. Inside the box, the text is organized into two lines: the top line contains the UML metaclass notation <<metaclass>> and the bottom line contains the name ElementoDelModelo.

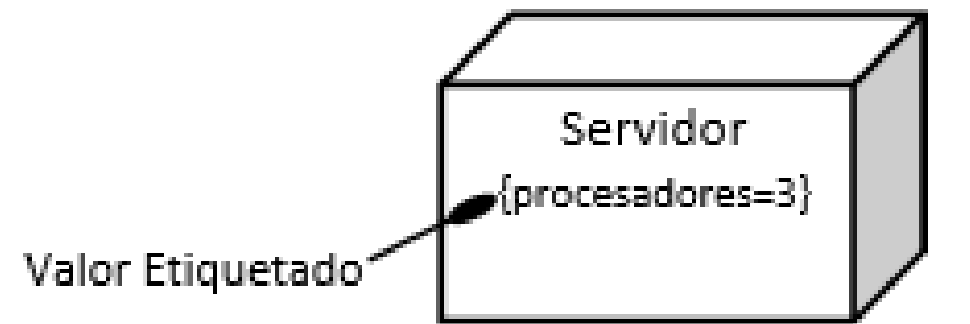
```
<<metaclass>>  
ElementoDelModelo
```

# UML

## ➤ Mecanismos Comunes

- **Mecanismos de Extensibilidad:**

- Se presentan 3 mecanismos de extensibilidad:
  - Valores etiquetados



- Restricción

