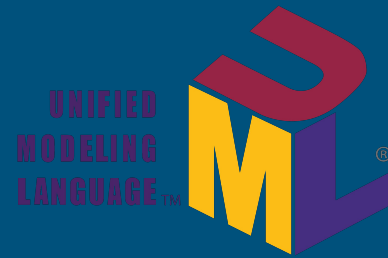


# Metodología de Ingeniería de Software para web IFCD036PO

METODOLOGÍA ORIENTADA A  
OBJETOS

# U.M.L

---



- ¿Qué es?

- UML, o Unified Modeling Language (Lenguaje Unificado de Modelado), es un lenguaje gráfico que se utiliza para ver, construir y documentar un sistema de software.

- ¿Por qué “lenguaje gráfico”?
- Porque utiliza diagramas y símbolos visuales para representar diferentes aspectos de un sistema de software, incluyendo su estructura, comportamiento, e incluso los patrones de interacción entre objetos o componentes dentro del sistema.

- Origen

- Se remonta a la década de 1990. En ese momento había muchos tipos de modelado de sistemas de software, cada uno con su propia notación y metodología. Esto causó una gran cantidad de confusión y dificultades de interoperabilidad.
- Para abordar este problema, Grady Booch, James Rumbaugh e Ivar Jacobson, conocidos como "Los Tres Amigos", comenzaron a trabajar en unificar sus respectivas metodologías.



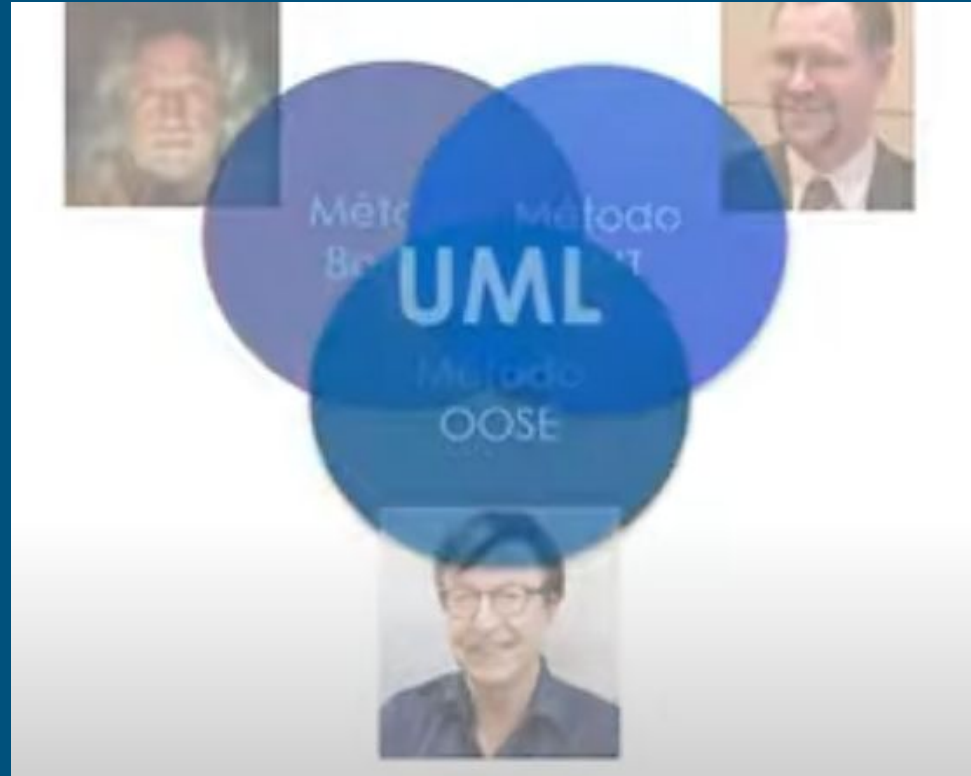
Método  
Booch



Método  
OMT

Método  
OOSE

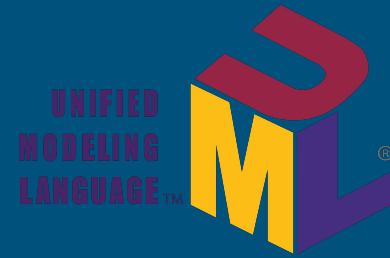




- Lenguaje visual con el objetivo de:
  - Facilitar la creación de código
  - Describir la estructura de una aplicación
  - Describir la estructura de un sistema
  - Mostrar el comportamiento de un sistema
  - Crear modelos de datos



# Bloques de construcción UML

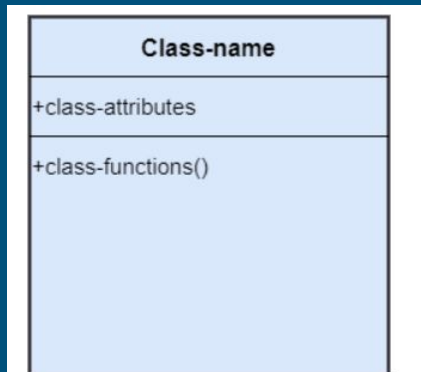


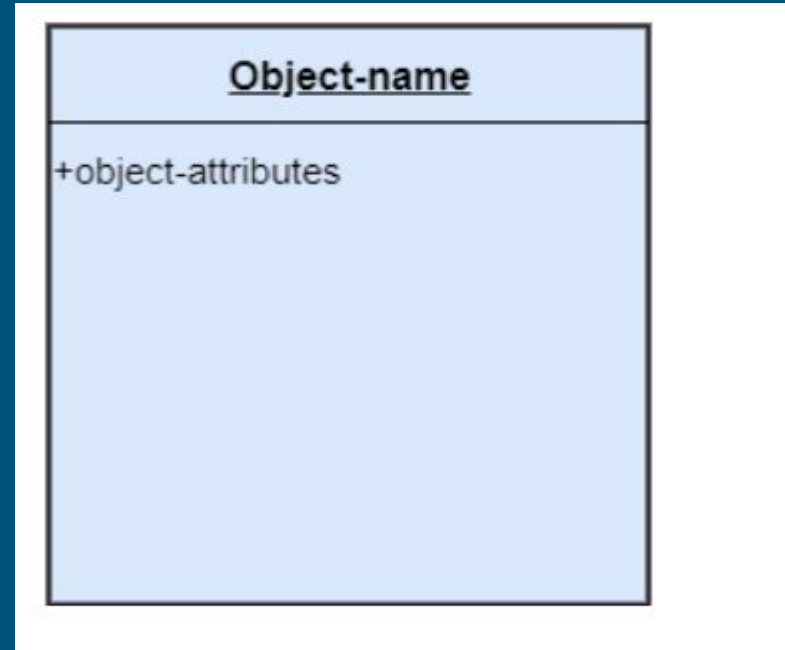
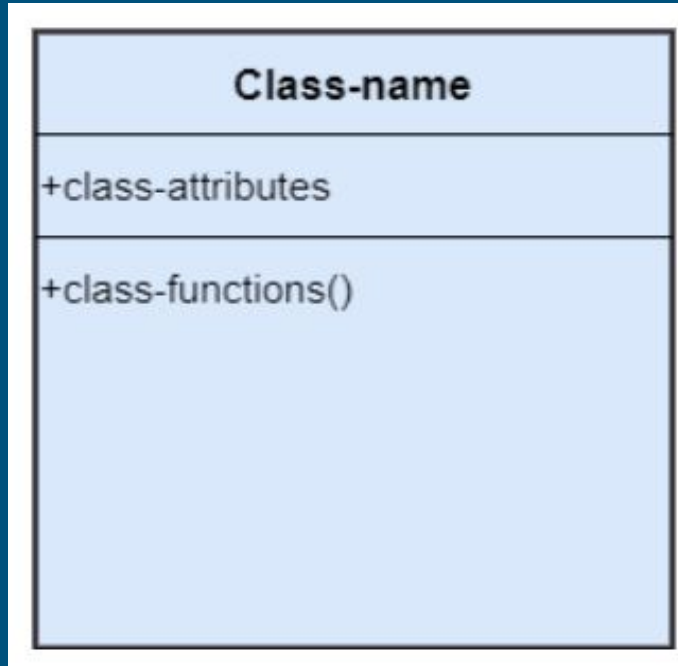
UML se compone de tres bloques de construcción principales, es decir, cosas, relaciones y diagramas. Los bloques de construcción generan un diagrama de modelo UML completo girando alrededor de varios bloques diferentes. Desempeña un papel esencial en el desarrollo de diagramas UML. Los bloques de creación básicos de UML se enumeran a continuación:

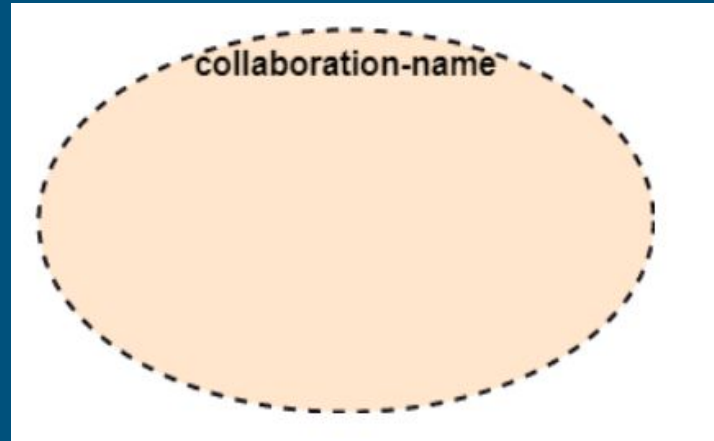
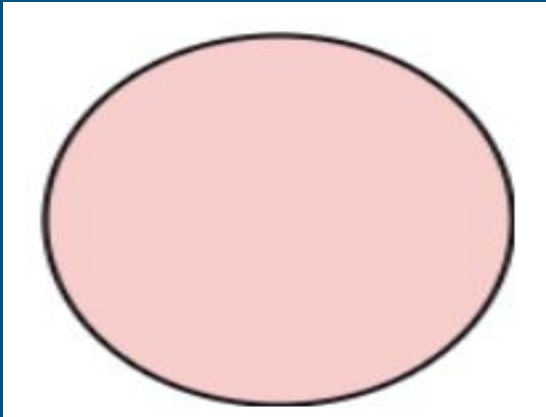
1. Cosas
2. Relaciones
3. Diagramas

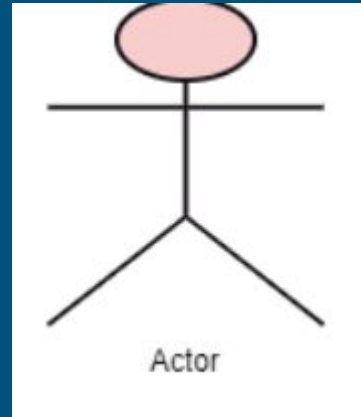
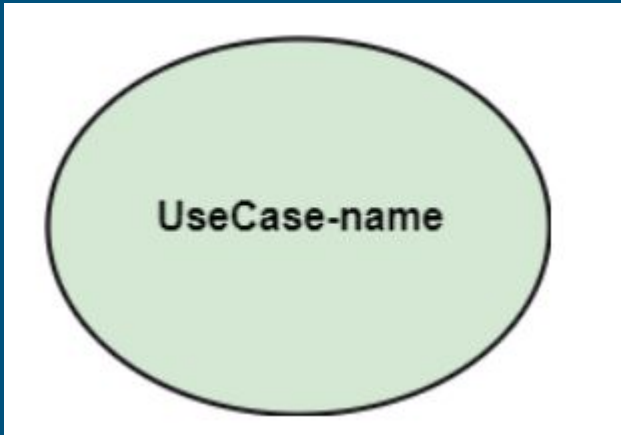
## Cosas estructurales

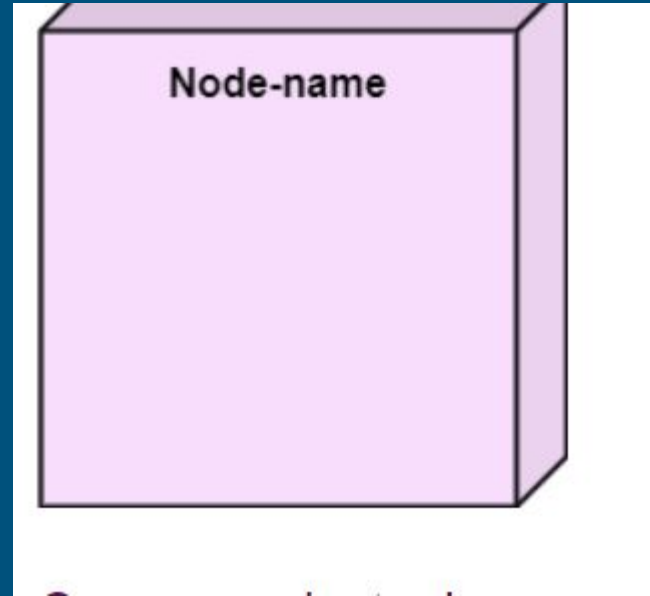
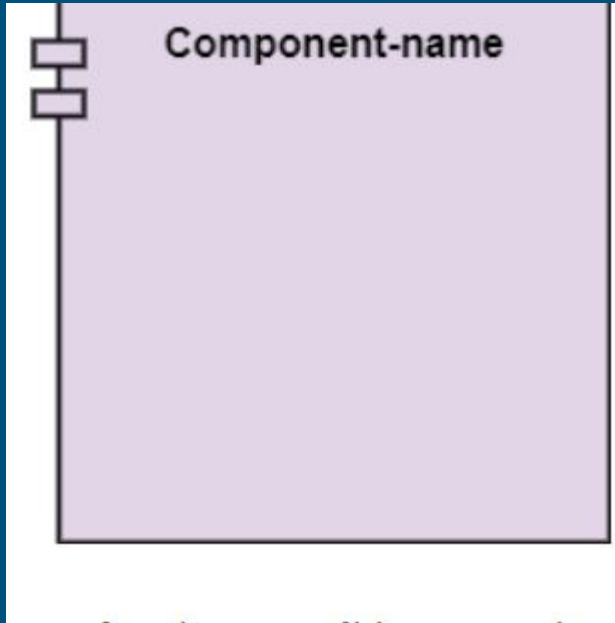
Los sustantivos que representan el comportamiento estático de un modelo se denominan cosas estructurales. Muestran los componentes físicos y conceptuales. Incluyen la clase, el objeto, la interfaz, el nodo, la colaboración, el componente y un caso de uso.

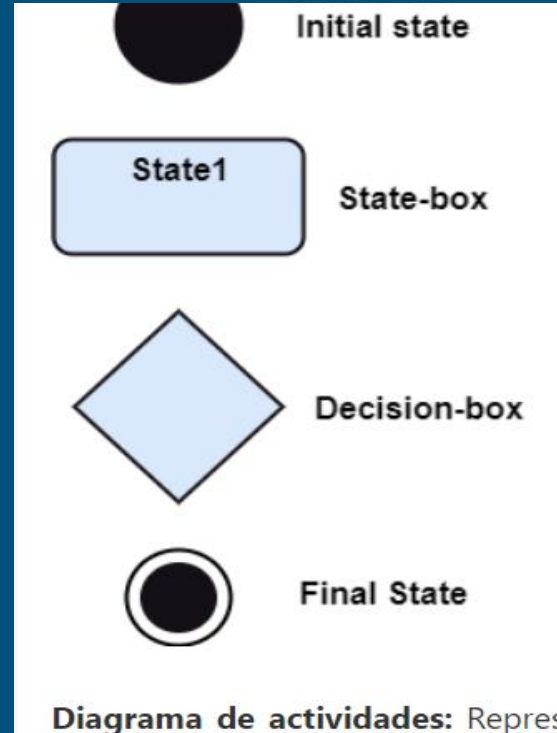
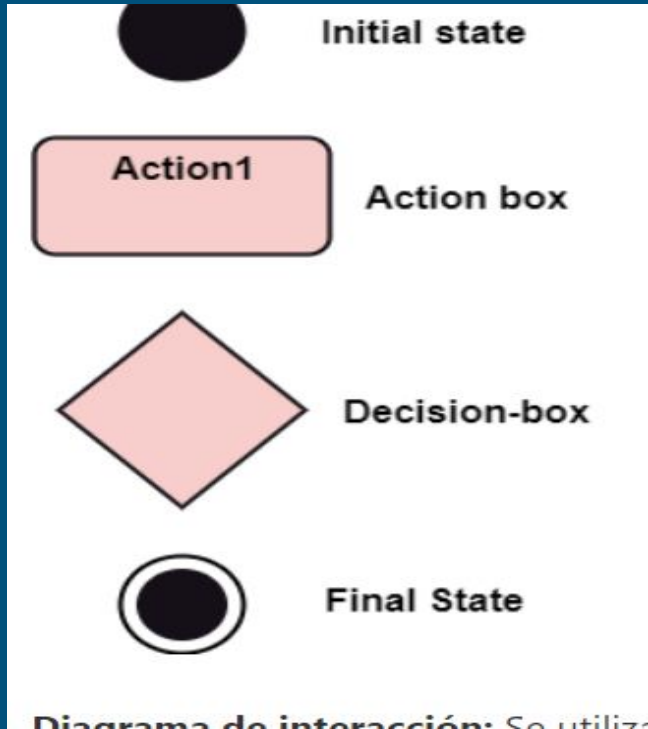




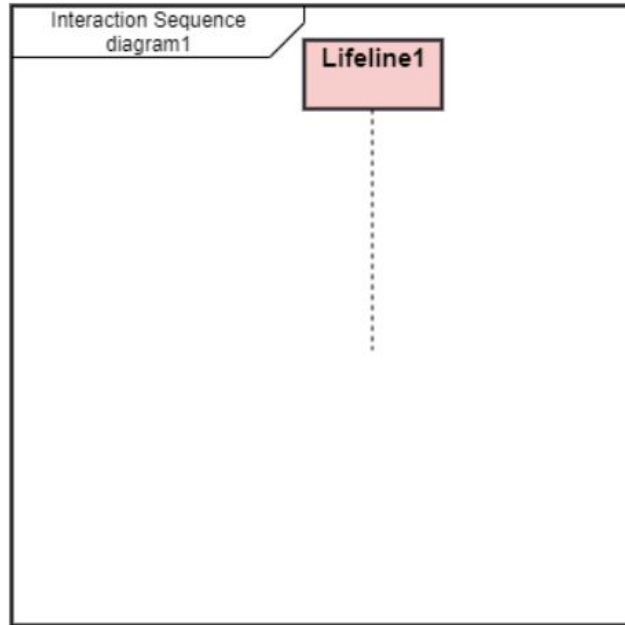




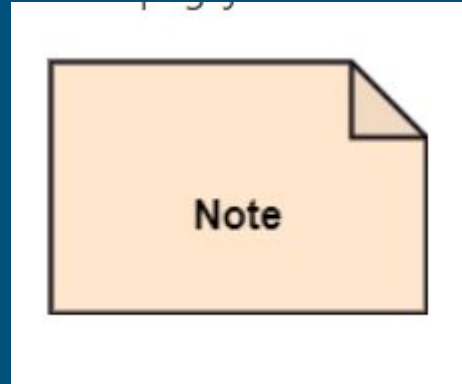
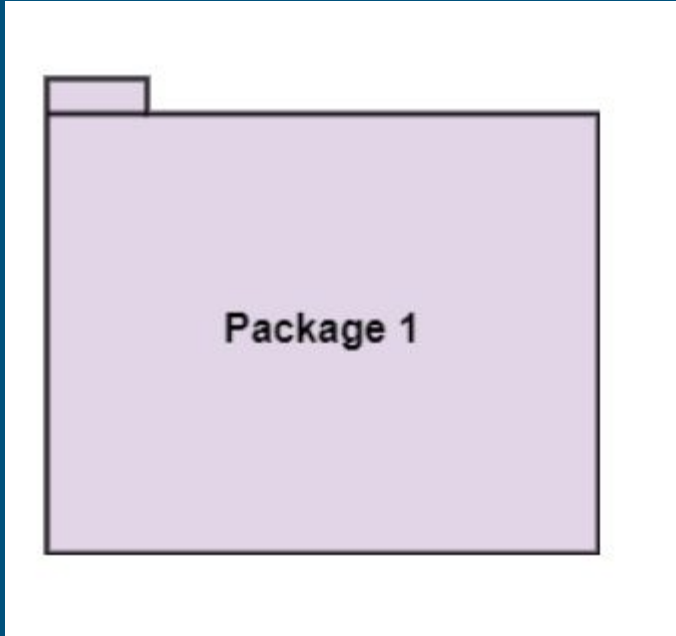




**Diagrama de interaccion:** Se utiliza para visualizar el flujo de mensajes entre varios componentes de un sistema.



Agrupar cosas



---

----- Dependency ----->

<----- Association ----->

----- Generalization ----->

----- Realization ----->