

Sesión 5

Introducción al Unified Model Language (UML)

Unidad 2

**Introducción al Unified Model
Language (UML)**

Mg. Gustavo G. Delgado Ugarte

Historia del UML

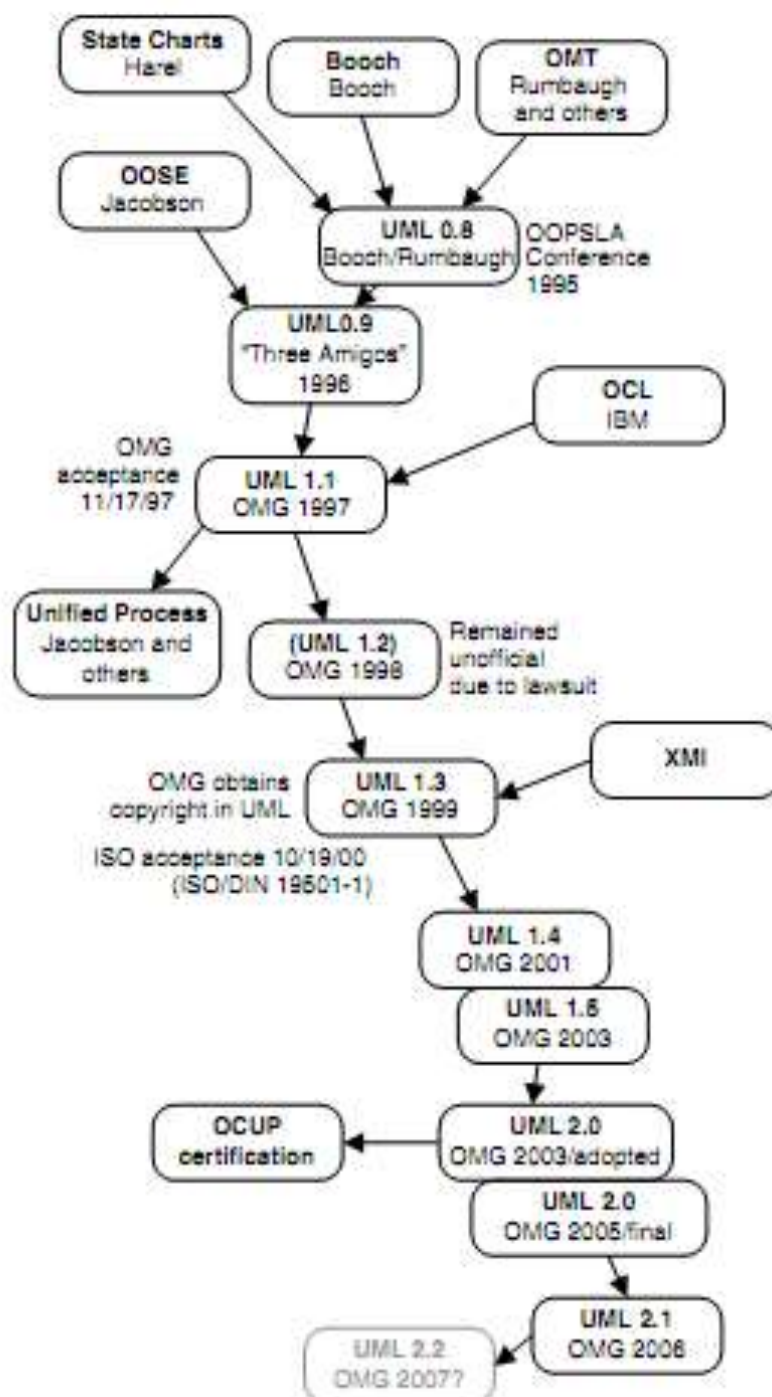
- El UML es la creación de Grady Booch, James Rumbaugh, e Ivar Jacobson (“Los Tres Amigos”)
 - Trabajaron en organizaciones separadas durante los años 1980 y principios de 1990, elaborando cada uno su propia metodología de análisis y diseño orientado a objetos
 - A mediados de la década de 1990, empezaron a pedirse prestadas ideas uno de otros, así que decidieron desarrollar su trabajo juntos
 - En 1994 Rumbaugh se unió a Rational Software Corporation, donde Booch ya estaba trabajando. Jacobson ingresó a Rational, un año después

Historia del UML

- Borradores de UML comenzaron a circular en toda la industria del software, y la respuesta resultante trajo cambios sustanciales
- Muchas empresas sintieron que UML serviría a sus fines estratégicos, surgió UML consortium (consorcio UML) surgido
 - Miembros: DEC, Hewlett-Packard, Intellicorp, Microsoft, Oracle, Texas Instruments, Rational, otros.
- **1997** el consorcio produjo la versión 1.0 de UML y lo presentó al Object Management Group (OMG) en respuesta a la petición de la OMG publicó una solicitud de propuesta de un lenguaje de modelado estándar

Historia del UML

- El consorcio, generó la versión 1.1, y lo presentó a los OMG, que lo aprobó a finales de **1997**.
- OMG se hizo cargo del mantenimiento de UML y produjo dos revisiones más en **1998**.
- UML se ha convertido en un estándar de facto en la industria del software, y sigue evolucionando
 - Versiones 1.3, 1.4, 1.5 ,2.0
- Además de las ideas de Booch, Rumbaugh y Jacobson, incluyen otras, por ejemplo, diagramas de estado de Harel (Harel, 1987)



Tecnologías base del UML

- Tecnología de Objetos
 - Métodos Orientado a Objetos
 - OMT (Object-modeling technique) de Rumbaugh
 - Método Booch de Grady Booch
 - Método OOSE (Object-Oriented Software Engineering) de Jacobson
 - Notación de Casos de Uso del Objectory
 - Notación de componentes de Booch
 - Diagramas de Estado de Harel
 - Conceptos de muchos otros métodos OO
 - Ej. Bran Selic, Conrad Bock, James Odell, etc.

OMG (Object Management Group)

- El Object Management Group, una organización internacional de la industria en la cual todas las empresas de TI importantes son miembros, además de los que desarrollaron y estandarizaron UML
- Las Empresas miembros del OMG cooperan en el mantenimiento y la aplicación del estándar UML
- Los miembros del OMG incluyen
 - Grandes empresas internacionales: IBM, Hewlett-Packard, Sun Microsystems, Telelogic, Boeing, Adobe y DaimlerChrysler
 - Empresas innovadoras de tamaño medio: las empresas alemanas Prostep AG, b + m Informatik, y OOSE GmbH

Sub especificaciones UML

- UML, versión 2.0 ha sido formalmente dividido en las sub especificaciones siguientes:
 - **Infraestructura:** Núcleo de la arquitectura, los perfiles, y los estereotipos
 - **Superestructura:** elementos de los modelos estáticos y dinámicos
 - **Object Constraint Language (OCL):** Un lenguaje formal utilizado para describir expresiones en los modelos UML
 - **Diagrama de intercambio:** El formato de intercambio de diagramas UML.

Metamodelo UML

- UML se define formalmente empleando un metamodelo – un modelo de las estructuras del UML
- El metamodelo está expresado en UML

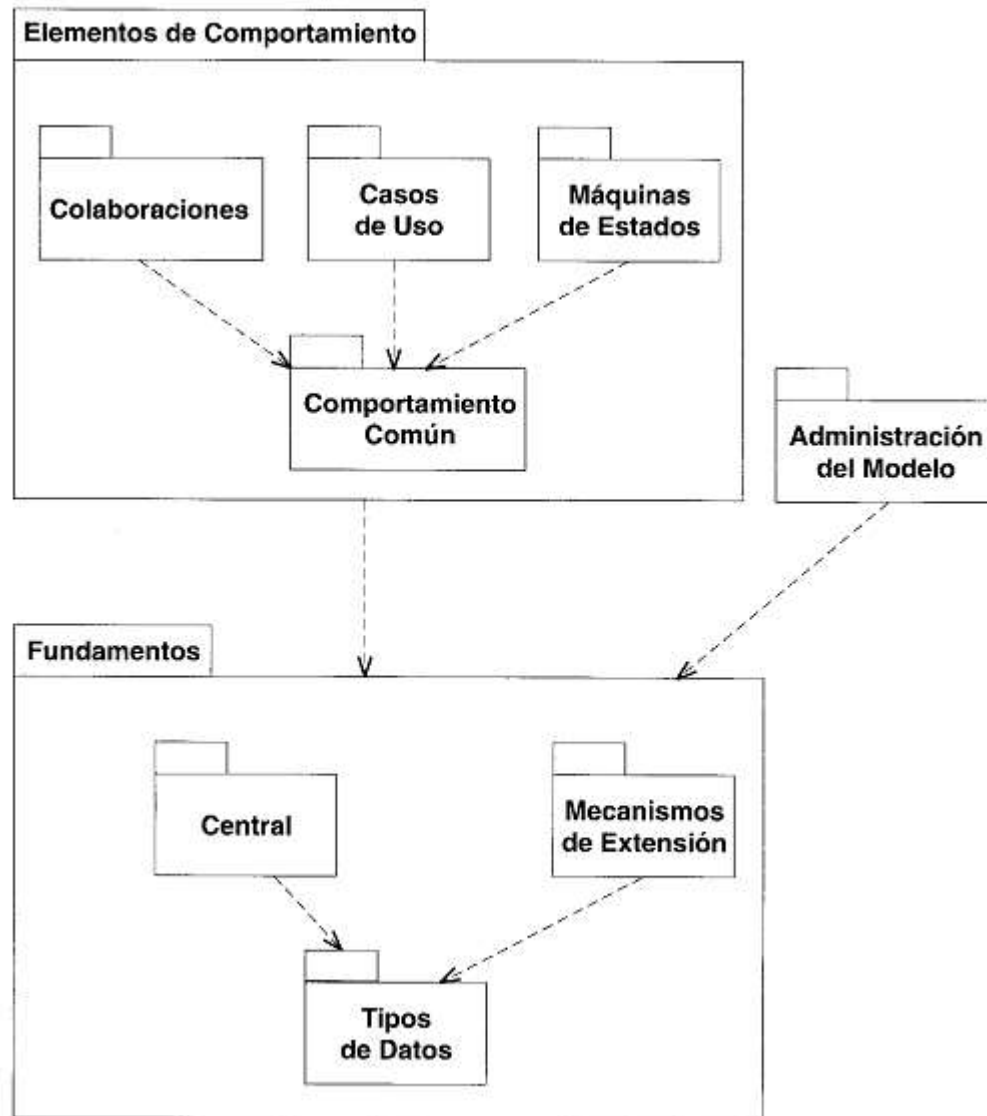
Metamodelo UML

- Cada sección de documento semántico contiene:
 - Un diagrama de clases que muestra una porción del metamodelo
 - Una descripción textual de las clases del metamodelo definidas en esa sección, con atributos y relaciones
 - Una lista de restricciones aplicables a los elementos del modelo expresado en lenguaje natural y en OCL
 - Una descripción de la semántica dinámica de las estructuras de UML definidas en la sección (Informal)

Estructura del Metamodelo

- El metamodelo está dividido en 3 paquetes principales
 - **El paquete fundamentos** define la estructura estática de UML
 - **El paquete de elementos de comportamiento** define la estructura dinámica del UML
 - **El paquete de administración del modelo** define la estructura organizativa de los modelos de UML

Estructura del Metamodelo



Paquete de Fundamentos

- Núcleo(Central)
 - Describe las principales estructuras estáticas del UML
 - Clasificadores
 - Contenido.- incluye atributo, operación, método y parámetro
 - Relaciones.- incluyen generalización, asociación y dependencia
 - Se definen varias metaclases abstractas
 - Elemento generalizable
 - Espacio de nombres
 - Elemento del modelo
 - Define plantilla , tipos de subclases de dependencia, componente, nodo y comentario

Paquete de Fundamentos

- Tipo de Dato
 - Describe las clases de tipos de datos que utiliza el metamodelo
- Mecanismos de extensión
 - Describe los mecanismos
 - Restricción
 - Estereotipo
 - Valor etiquetado

Paquete Elementos de Comportamiento

- Tiene un subpaquete por cada vista principal
- Tiene un paquete para estructuras de comportamiento que comparten las tres vistas principales

Paquete Elementos de Comportamiento

- Comportamiento común
 - Describe señal, operación y acción
 - Describe instancias de clases correspondientes a diferentes descriptores
- Colaboraciones
 - Describe colaboración, interacción, mensaje, rol de clasificador y asociación

Paquete Elementos de Comportamiento

- Casos de uso
 - Describe actor y caso de uso
- Maquinas de estado
 - Describe la estructura de una máquina de estados
 - Estado
 - Pseudoestado
 - Evento
 - Señal
 - Transición
 - Condición de guarda
 - Describe estructuras adicionales para modelos de actividad
 - Estado de acción
 - Estado de actividad
 - Estado de flujo de objeto

Paquete de administración de modelos

- Describe los paquetes, modelos y subsistemas
- Describe las propiedades de propiedad y visibilidad de los espacios de nombre y de los paquetes
- No posee subpaquetes

Arquitectura 4+1

- Una de las herramientas para representar modelos de arquitectura anteriores al UML es la denominada 4+1 vistas propuesta por Kruchten
- El modelo describe la arquitectura de software del sistema a través de 5 vistas concurrentes

Arquitectura 4+1

- **La vista lógica**
 - Trata de clases y subsistemas, tiene las siguientes particularidades:
 - Soporta los requerimientos funcionales
 - Identifica mecanismos y diseña elementos comunes a través del sistema
 - Utiliza los diagramas de clases y la notación de Booch además de utilizar el estilo arquitectónico orientado a objetos

Arquitectura 4+1

- **La vista concurrente o de procesos**
 - Describe el diseño de concurrencia y aspectos de sincronización
 - Especifica las líneas de mando que ejecutan cada operación en cada una de las clases señaladas en la vista lógica
 - Los diseñadores realizan esta vista en varios niveles de abstracción, además de dividir el software en conjuntos independientes de tareas, es decir, se empaqueta en pequeños programas o librerías del subsistema
 - Los estilos arquitectónicos más usados son los de tuberías y filtros o el de cliente/servidor.

Arquitectura 4+1

- **La vista de componentes o de desarrollo**
 - Describe la organización estática de software en los ambientes de desarrollo
 - Es una importante característica de lógica en casos de vistas, autónomas, persistentes y de distribución, describe la participación en diferentes operaciones, determina si existe persistencia entre un objeto y otro, además determina el estado de los objetos y operaciones de accesibilidad por muchos nodos
 - Es recomendable usar el estilo de arquitectura por capas.

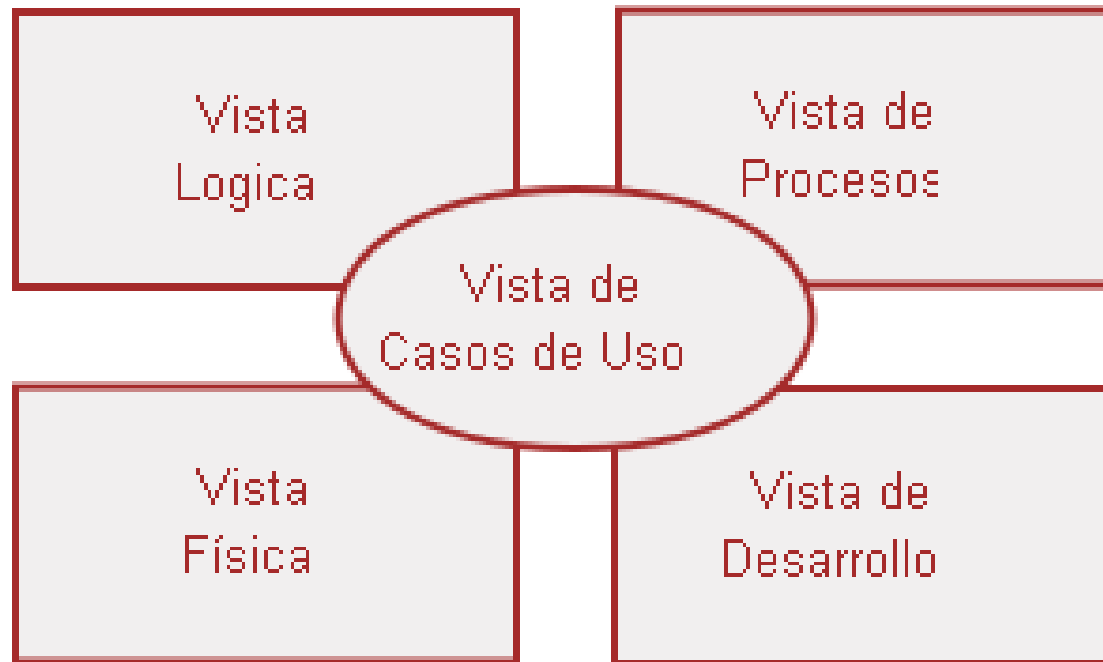
Arquitectura 4+1

- **La vista distribuida o física**
 - Se refiere a la implementación en módulos y fragmentación en muchas capas
 - Colecciona las categorías de clases y grupos
 - Describe el mapeo del software en el hardware y toma en cuenta los requerimientos funcionales del sistema, tales como confiabilidad, respuesta y escalabilidad

Arquitectura 4+1

- **La vista de casos de uso o escenarios**
 - Gobierna los requerimientos que son necesarios para el usuario final, y construye elementos comunes a través del sistema
 - Esta vista es redundante relacionado con el conjunto que forman las anteriores de ahí que se le denomina +1, pero su inclusión es vital ya que desempeña dos roles importantes
 - Actúa como indicador que ayuda al diseñador a descubrir los elementos de la arquitectura durante su diseño
 - Valida e ilustra el diseño de la misma
 - La notación es similar a la que se utiliza en la vista lógica, a excepción de que usa los conectores de la vista de procesos para indicar la interacción entre objetos

Arquitectura 4+1



Estándares relacionados con UML y con la orientación a objetos

- Entornos de desarrollo integrado
- Herramientas CASE
- Técnicas de Modelado a Objetos
- Programación orientada a objetos
- **XMI**, un formato estándar basado en XML para el intercambio de modelos UML.
- **OCL**, Lenguaje de especificación para los diferentes modelos en UML.
- **Webml**, Metodología para el diseño de Sistemas de Información Web.
- **Business Process Modeling Notation** o **BPMN**, notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo (workflow), es actualmente mantenida por el OMG
- **CORBA** (*Common Object Request Broker Architecture* — arquitectura común de intermediarios en peticiones a objetos); es un estándar que establece una plataforma de desarrollo de sistemas distribuidos facilitando la invocación de métodos remotos bajo un paradigma orientado a objetos. Fue definido y está controlado por el OMG