

# MODULO IV



## Análisis y Diseño de Sistemas de Información

**INF-162**

### IV. UML

#### 4.1 Introducción

**Facilitador: Miguel Cotaña**

**Mayo de 2014**



# ¿QUÉ ES UML?

Un diagrama UML es una representación gráfica parcial (vista) de un modelo de un sistema.

Es una herramienta que permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional y fácil de comprender y así poder comunicárselas a otras personas.



## UML = Unified Modeling Language

Un lenguaje de propósito general para el modelado orientado a objetos. Impulsado por el Object Management Group (OMG, [www.omg.org](http://www.omg.org)). Se encarga de la definición y mantenimiento de estándares para aplicaciones de la industria de la computación



UML combina notaciones provenientes desde:

- ✚ Modelado Orientado a Objetos;
- ✚ Modelado de Datos;
- ✚ Modelado de Componentes;
- ✚ Modelado de Flujos de Trabajo (Workflows).



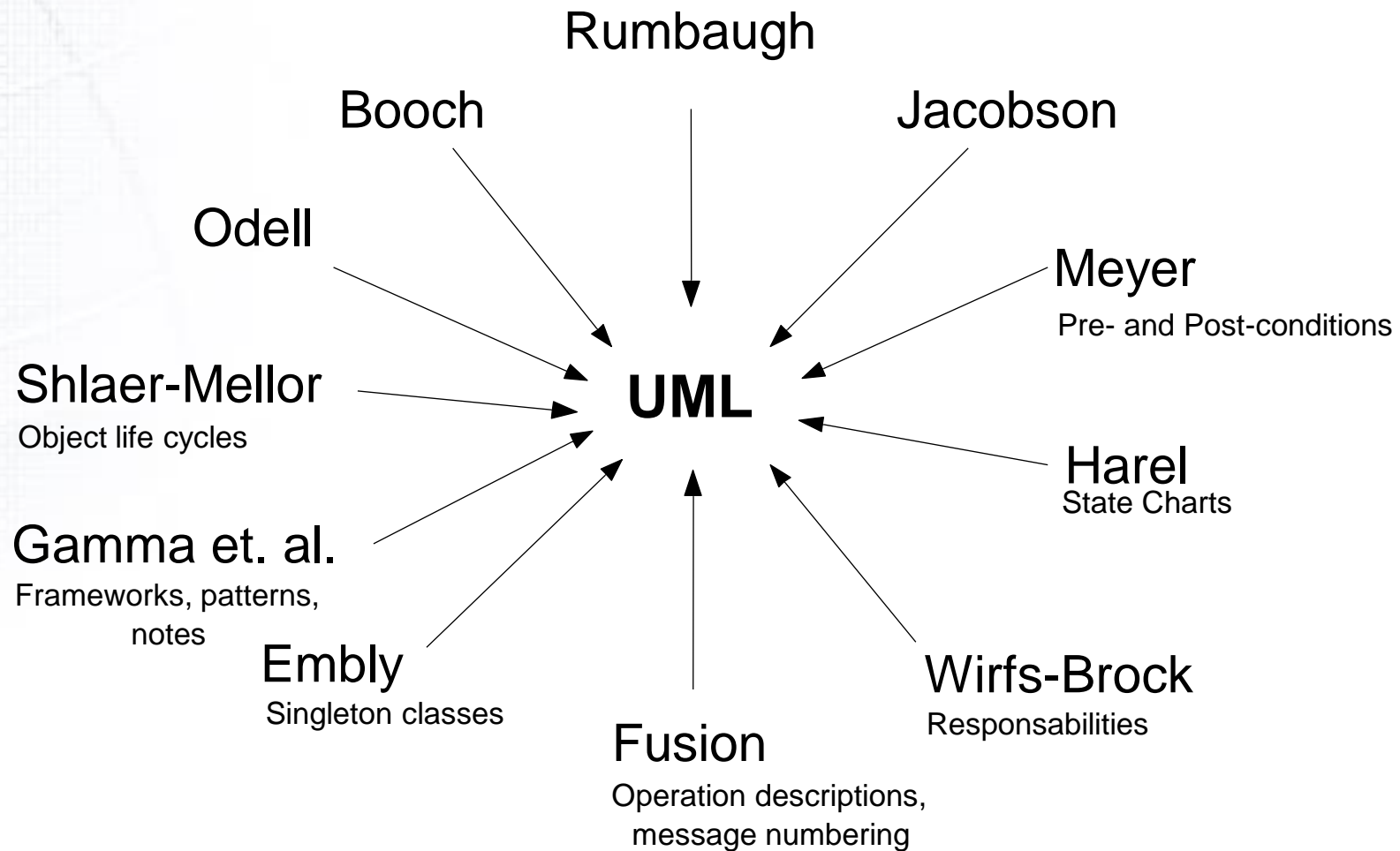
# HISTORIA

Entre la guerra de los métodos, aparecieron los siguientes:

- + Booch (Rational Software);
- + OOSE (Objet-Oriented Software Engineering) de Jacobson (Objectory: casos de uso);
- + OMT (Object Modeling Technique) de Rumbaugh (G&E);
- + Fusión;
- + Shlaer-Mellor;
- + Coad-Yourdon.



# UML "aglutina" enfoques OO







UML, es un lenguaje visual para especificar, construir y documentar sistemas.

**Unified:** Aporte de muchos métodos y notaciones. Independiente de implementación, plataforma y lenguajes.

**Modeling:** Los modelos son utilizados en todas las ingenierías.

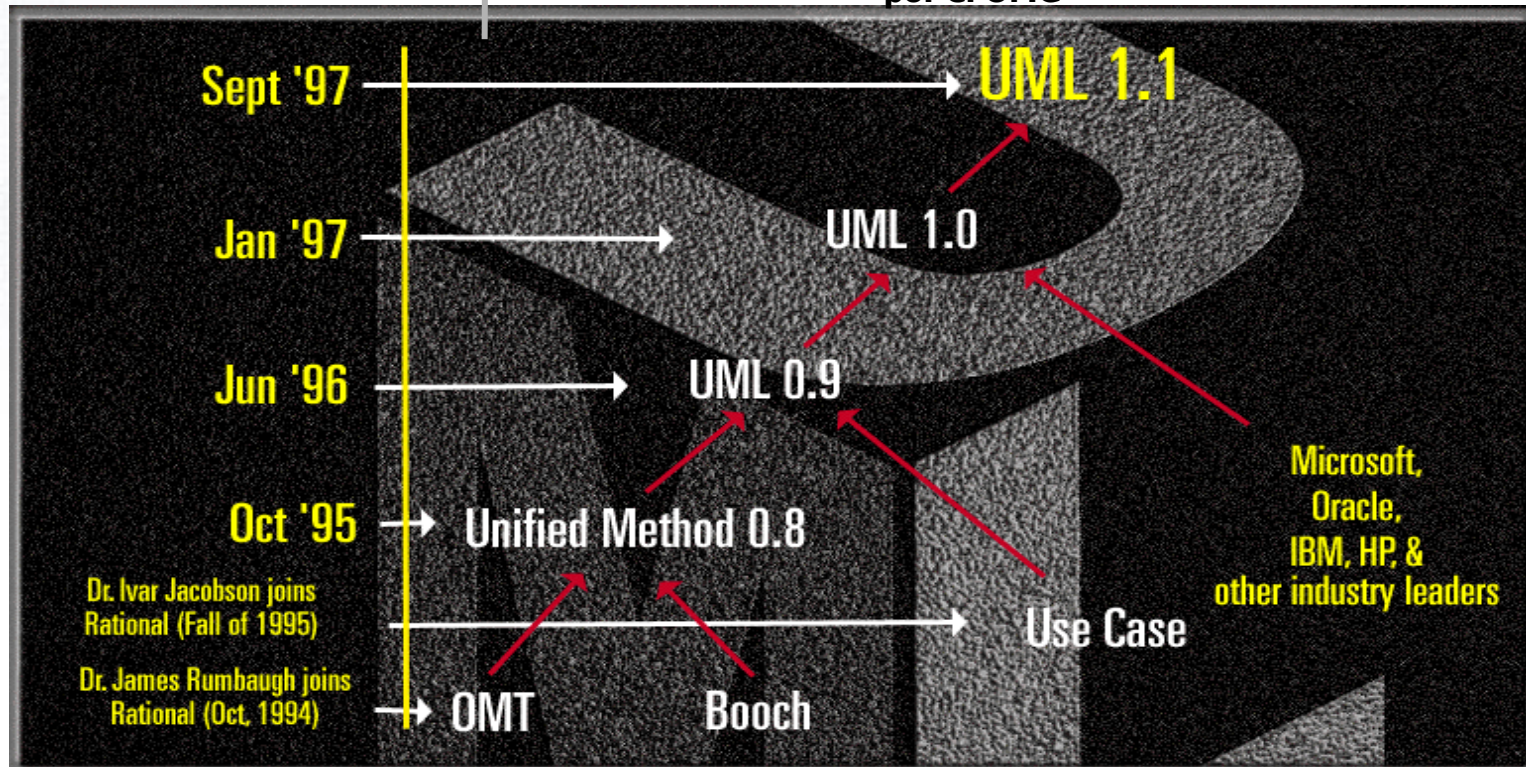
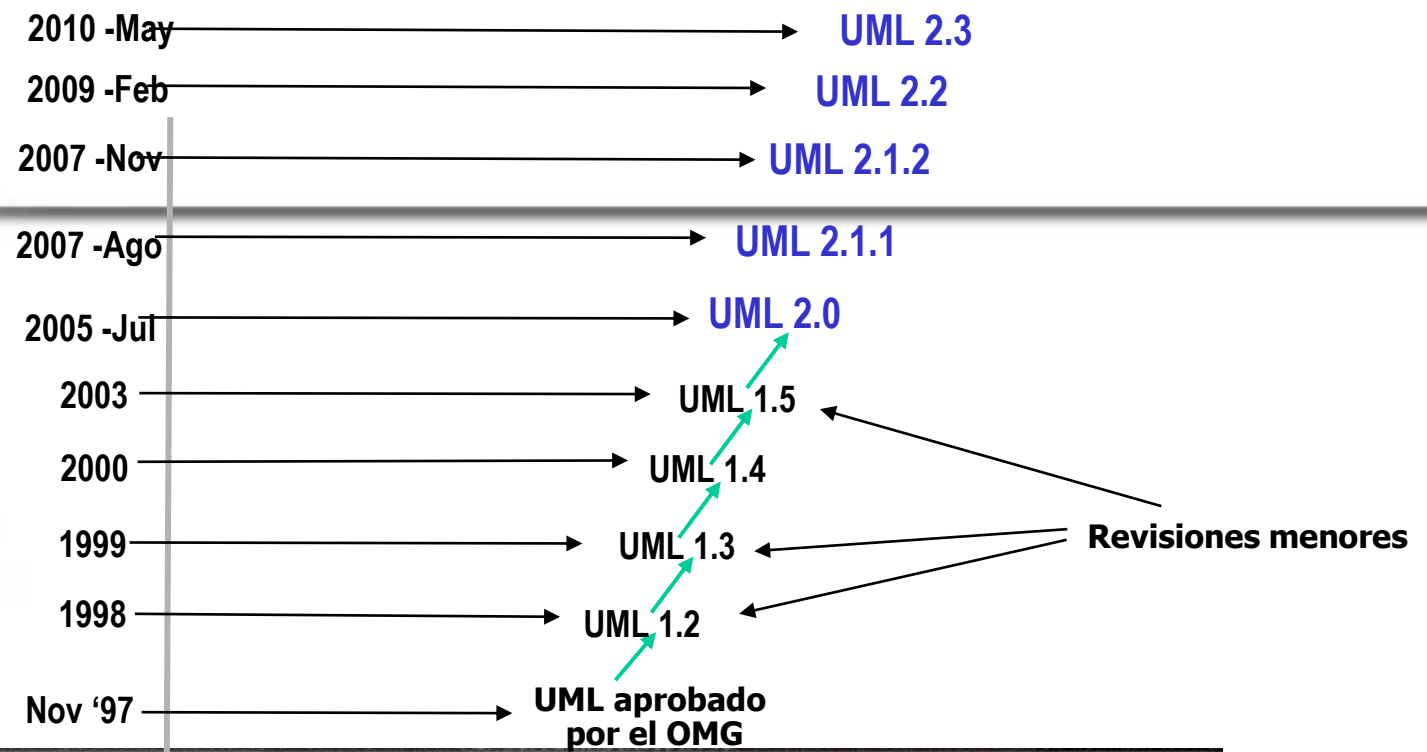
**Language:** Si hay gente, requieren comunicarse. Si se tienen que comunicar, se tienen que entender. Para entenderse necesitan un lenguaje común.



UML, es un lenguaje de modelado, y no un método. La mayor parte de los métodos consisten, al menos al principio, en un lenguaje y en un proceso para modelar.

El lenguaje de modelado es la notación (principalmente gráfica) de que se valen los métodos para expresar los diseños. El proceso es la orientación que nos dan sobre los pasos a seguir para hacer el diseño. <sup>8</sup>







UML, define una notación y un metamodelo:

**Notación:** es el material gráfico que se ve en los modelos; es la sintaxis del lenguaje de modelado;

**Metamodelo:** Modelo que define otros modelos (un diagrama, usualmente un diagrama de clases, que defina la notación)



UML, es un lenguaje estándar para escribir planos de software;

UML, es un lenguaje expresivo;

UML es un lenguaje para:

- + Visualizar;
- + Especificar;
- + Construir;
- + Documentar.



# Análisis y Diseño de Sistemas de Información

**INF-162**

## IV. UML

### 4.2 Modelado básico

**Facilitador: Miguel Cotaña**

**08 de Noviembre 2012**





## MODELO CONCEPTUAL DE UML

Para comprender UML, se necesita adquirir un modelo conceptual del lenguaje, que comprende:

- + Los bloques básicos de construcción;
- + Las reglas que dictan cómo se pueden combinar esos bloques básicos;
- + Algunos mecanismos comunes que se aplican a través de UML.



## BLOQUES DE CONSTRUCCION

El vocabulario de UML incluye 3 clases de bloques de construcción:

- 1.Elementos;
- 2.Relaciones;
- 3.Diagramas.

Los elementos son abstracciones que son ciudadanos de primera clase en un modelo; Las relaciones ligam los elementos; los diagramas agrupan elementos.





# 1. Elementos

Existen 4 tipos:

1. Elementos estructurales;
2. Elementos de comportamiento;
3. Elementos de agrupación;
4. Elementos de anotación.

Estos elementos son los bloques básicos de construcción OO de UML.



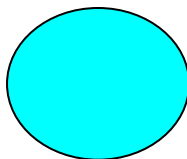
## 1.1 Elementos estructurales

Son los nombres de los modelos UML. En su mayoría son partes estáticas de un modelo y representan cosas que son conceptuales o materiales. Existen 7 tipos:

**Clase:** conjunto de objetos que comparten atributos, operaciones, relaciones y semántica;



**interfaz:** colección de operaciones que especifican un servicio de una clase o componente. Una interfaz describe el comportamiento visible externamente de ese elemento. Una interfaz puede representar el comportamiento completo de una clase o componente o sólo una parte de ese comportamiento;





**colaboración:** define una interacción y es una sociedad de roles y otros elementos que colaboran para proporcionar un comportamiento cooperativo mayor que la suma de los comportamientos de sus elementos;

**Cadena de  
responsabilidad**



**Caso de uso:** es una descripción de un conjunto de secuencias de acciones que un sistema ejecuta y que produce un resultado observable. Se utiliza para estructurar los aspectos de comportamiento en un modelo. Un caso de uso es realizado por una colaboración;

**Realizar pedido**



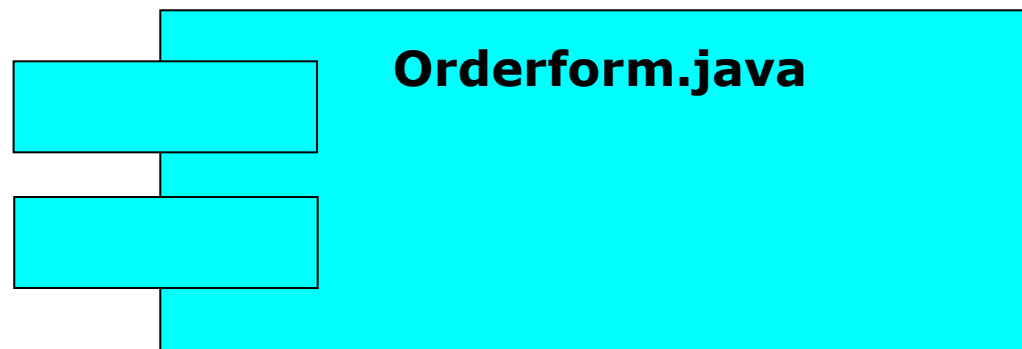
**Clase activa:** es una clase cuyos objetos tienen uno o más procesos o hilos de ejecución. Es igual que una clase, excepto en que sus objetos representan elementos cuyo comportamiento es concurrente con otros elementos;

GestorEventos
Suspender() VaciarCola()



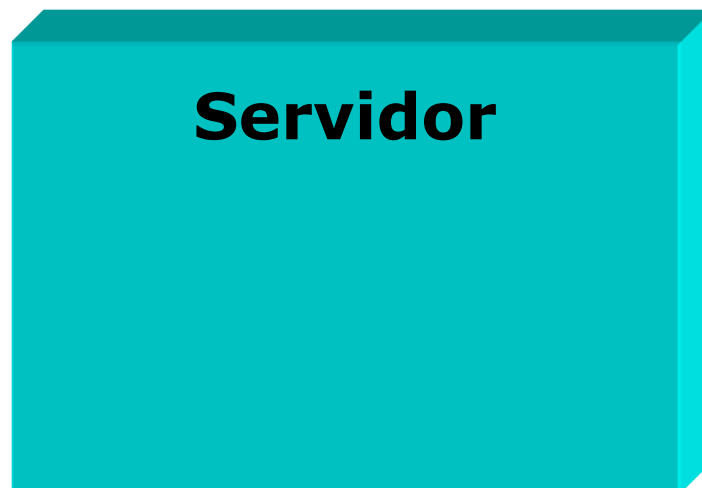


**componente:** es una parte física y reemplazable de un sistema que conforma con un conjunto de interfaces y proporciona la implementación de dicho conjunto. Representa típicamente el empaquetamiento físico de diferentes elementos lógicos;





**nodo:** elemento físico que existe en tiempo de ejecución y representa un recurso computacional, que por general dispone de memoria y capacidad de procesamiento.



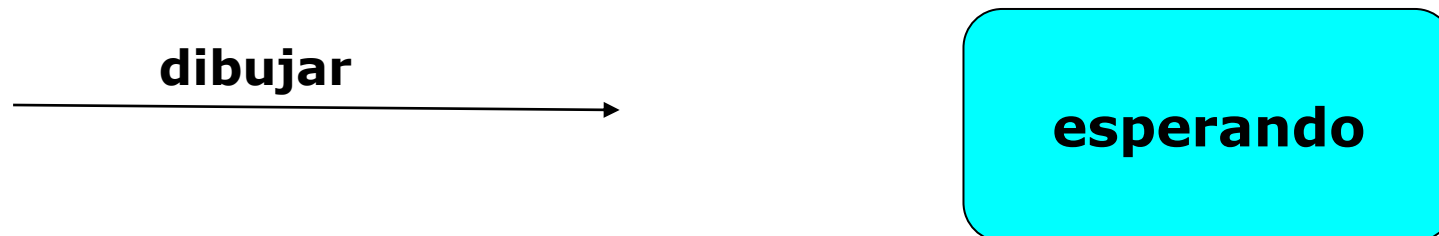


## 1.2 Elementos de comportamiento

Son las partes dinámicas de los modelos UML. Hay 2 tipos:

**interacción:** conjunto de mensajes;

**Máquina de estados:** especifica la secuencia de estados por las que pasa un objeto

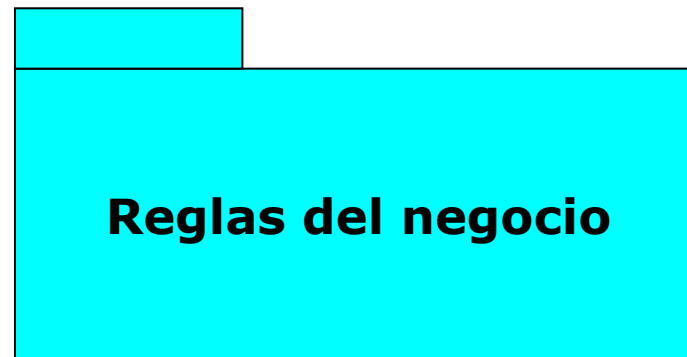




## 1.3 Elementos de agrupación

Son las partes organizativas. Son cajas en las que puede descomponerse un modelo:

**paquete:** organiza elementos en grupo. Es puramente conceptual (sólo existe en tiempo de desarrollo).





## 1.4 Elementos de anotación

Son las partes explicativas. Son comentarios que se pueden aplicar para describir, clarificar y hacer observaciones sobre cualquier elemento de un modelo:

**nota:** se utilizarán para adornar los diagramas con restricciones o comentarios

**Devuelve una  
copia del objeto  
receptor**



## 2. Relaciones

Existen 4 tipos:

- 1. Dependencia; .....>
- 2. Asociación; \_\_\_\_\_
- 3. Generalización; \_\_\_\_\_▶
- 4. Realización. ....▶

Estos relaciones son los bloques básicos de construcción para relaciones UML.





### 3. Diagramas

Es la representación gráfica de un conjunto de elementos.

Diagrama de Casos de Uso

Diagrama de Clases

Diagrama de Objetos

Diagramas de Comportamiento

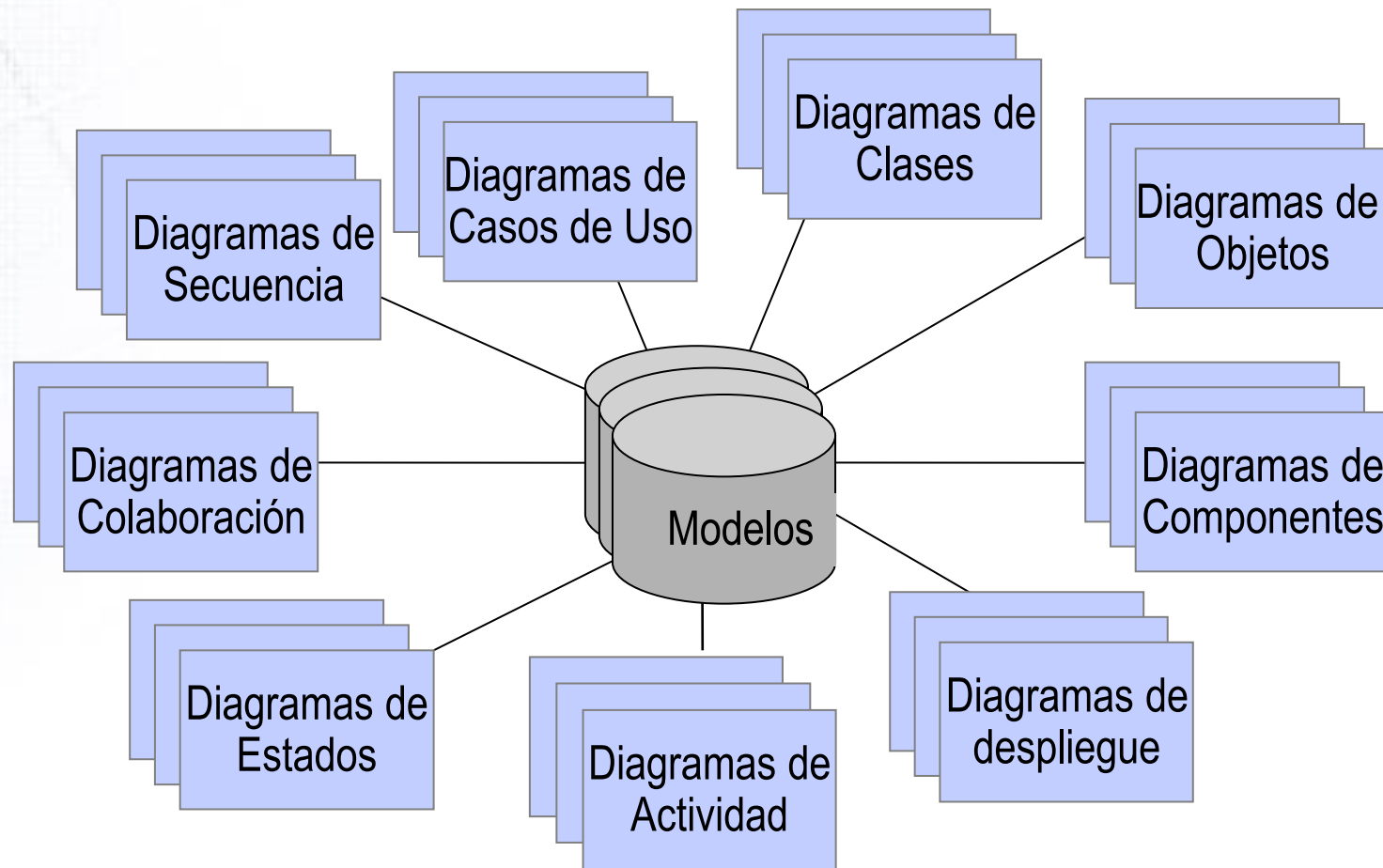
- Diagrama de Estados
- Diagrama de Actividad

Diagramas de Interacción

- Diagrama de Secuencia
- Diagrama de Colaboración

Diagramas de implementación

- Diagrama de Componentes
- Diagrama de Despliegue



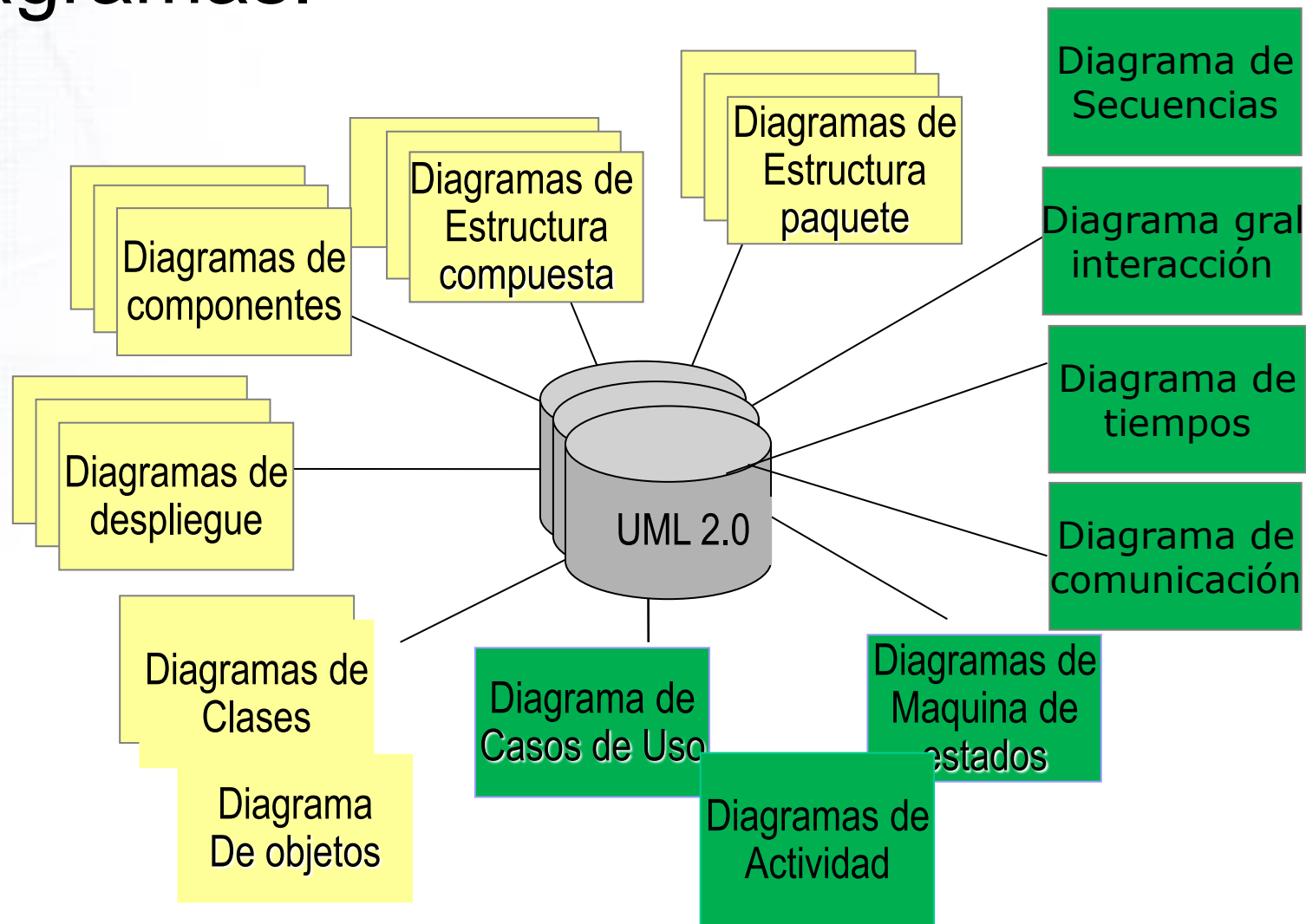


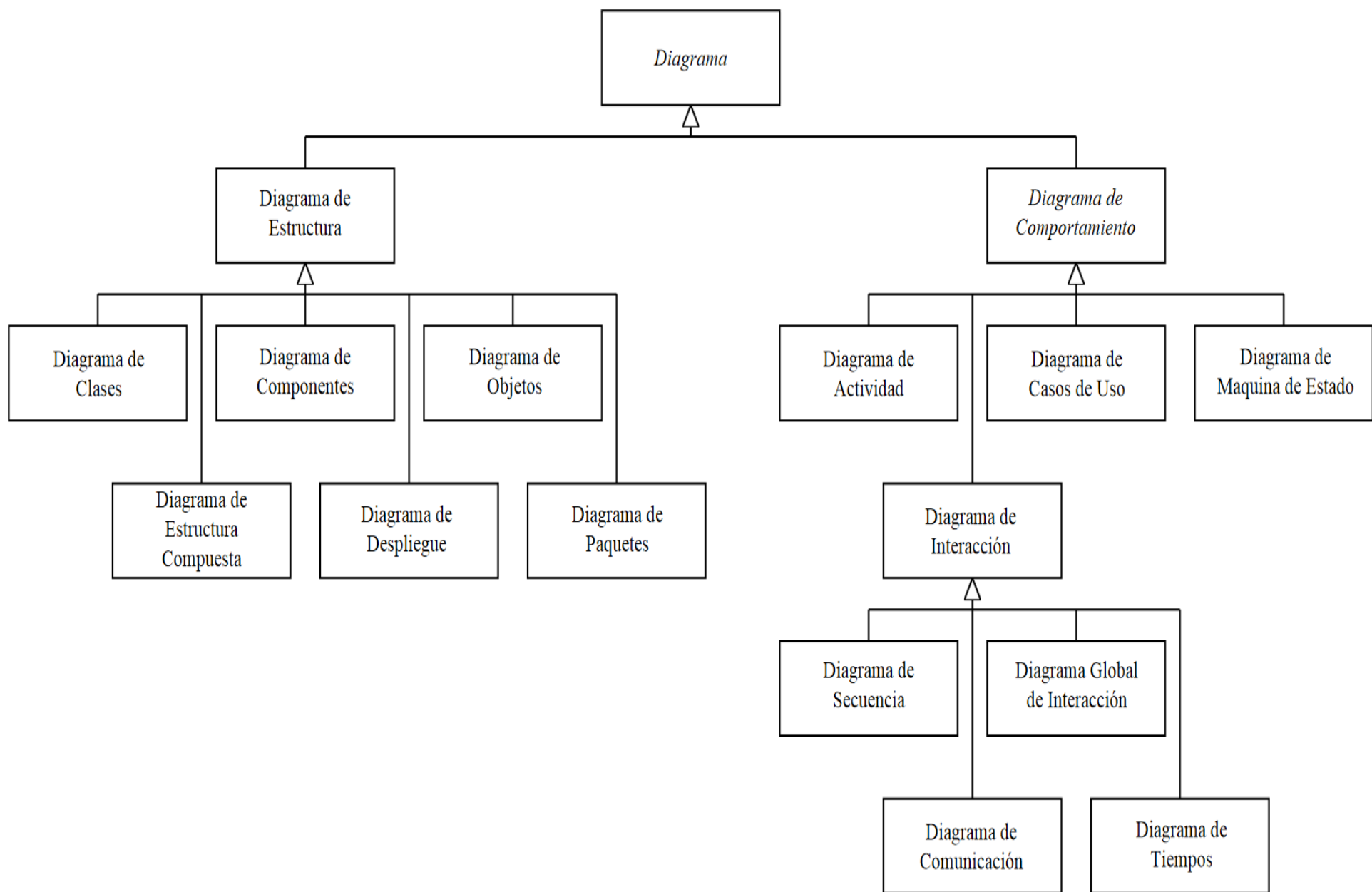
## UML 2.0

En OMG UML 2.0 se definen una serie de diagramas adicionales a los establecidos en OMG UML 1.x. El conjunto de diagramas se encuentra organizado en torno a dos categorías: **diagramas estructurales** (representados en amarillo) y **diagramas dinámicos** o de comportamiento (representados en verde)



# En UML 2.0 hay 13 tipos diferentes de diagramas.







**Diagrama de Estructura Compuesta.** Se emplea para visualizar de manera gráfica las partes que definen la estructura interna de un clasificador. Cuando se utiliza en el marco de una clase, este diagrama permite elaborar un diagrama de clases donde se muestran los atributos y las clases, indicando asociaciones de agregación o de composición.





## **Diagrama General de Interacción.**

Se emplea fundamentalmente para representar las interacciones, a través de diagramas o fragmentos de diagramas de secuencias, entre los actores y el sistema como una gran caja negra, y de diagramas de actividades en los que aparecen dichos fragmentos.



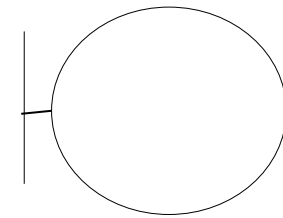
**Diagramas de Tiempos.** Empleados para mostrar las interacciones donde el propósito fundamental consiste en razonar sobre la ocurrencia de eventos en el tiempo que provocan el cambio de estados de un elemento estructural.

**Diagrama de Comunicación.** Equivalente al diagrama de colaboración, los diagramas aparecen dentro de un frame que posee una etiqueta para indicar el tipo de diagrama.

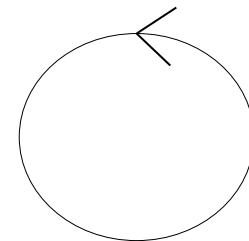


## Diagrama de Comunicación de análisis y diseño:

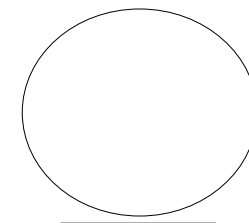
- + Diferente granularidad y nivel de detalle;
- + Estereotipos específicos para el análisis,



**interfaz**



**control**



**entidad**



## **Estructural :**

- + pkg Diagrama de Paquete
- + cmp Diagrama Componentes

## **Dinámica o Comportamiento**

- + uc Diagrama de Casos de Uso
- + act Diagrama de Actividad
- + stm Diagrama de Máquina de Estados
- + sd Diagrama de Secuencia



**El Diagrama de Casos de Uso** permiten, entre otras cosas, refinar el MCU a través de las asociaciones de: **<<incluye>>).** Permite incorporar el flujo de eventos de un caso de uso pequeño dentro de un caso de uso base de la aplicación. **<<extend>>).** Permite incorporar el flujo de eventos de un caso de uso pequeño bajo la ocurrencia de una determinada condición, cuando la misma evalúa verdadero.



**El Diagrama de Clases**, no ha sufrido cambios radicales en OMG UML 2.0.

**El Diagrama de Secuencia**, se le ha incorporado:

- ✚ **opt** : Indica que el fragmento de diagrama es opcional;
- ✚ **alt** : Indica que el fragmento de diagrama es una alternativa;
- ✚ **loop**: Indica que el fragmento de diagrama se ejecuta repetidas veces;
- ✚ **par**: Indica que el fragmento de diagrama incluye hilos de ejecución paralelo;
- ✚ **critical**: Indica una secuencia que no puede ser interrumpida por otro proceso;
- ✚ **sd**: Representa un diagrama de secuencia.





**El Diagrama de Clases de diseño.**

**El Diagrama de Componentes**, uno de los elementos incorporados consiste en la definición de puertos a través de los cuales cada componente software entrega un conjunto de servicios a través de interfaces proveídas .

**El Despliegue de la Solución sobre la Infraestructura TI**, A través del diagrama de despliegue se combina la Arquitectura de TI con la Arquitectura de Aplicación o Software.<sup>39</sup>



## ***Diagramas de Estructura:***

- ✚ Diagrama de clases
- ✚ Diagrama de componentes
- ✚ Diagrama de objetos
- ✚ Diagrama de estructura compuesta (UML 2.0)
- ✚ Diagrama de despliegue
- ✚ Diagrama de paquetes

## ***Diagramas de Comportamiento:***

- ✚ Diagrama de actividades
- ✚ Diagrama de casos de uso
- ✚ Diagrama de estados

## ***Diagramas de Interacción:***

- ✚ Diagrama de secuencia
- ✚ Diagrama de colaboración
- ✚ Diagrama de tiempos (UML 2.0)
- ✚ Diagrama de vista de interacción (UML 2.0)



**Diagramas de estructura:** muestra la **estructura estática** del sistema y sus partes en la abstracción y diferentes **niveles de** aplicación y cómo estas se relacionan entre sí.

**Esquemas de comportamiento:** muestran el comportamiento dinámico de los objetos en un sistema, que puede ser descrito como una serie de cambios en el sistema con el **tiempo**.

