



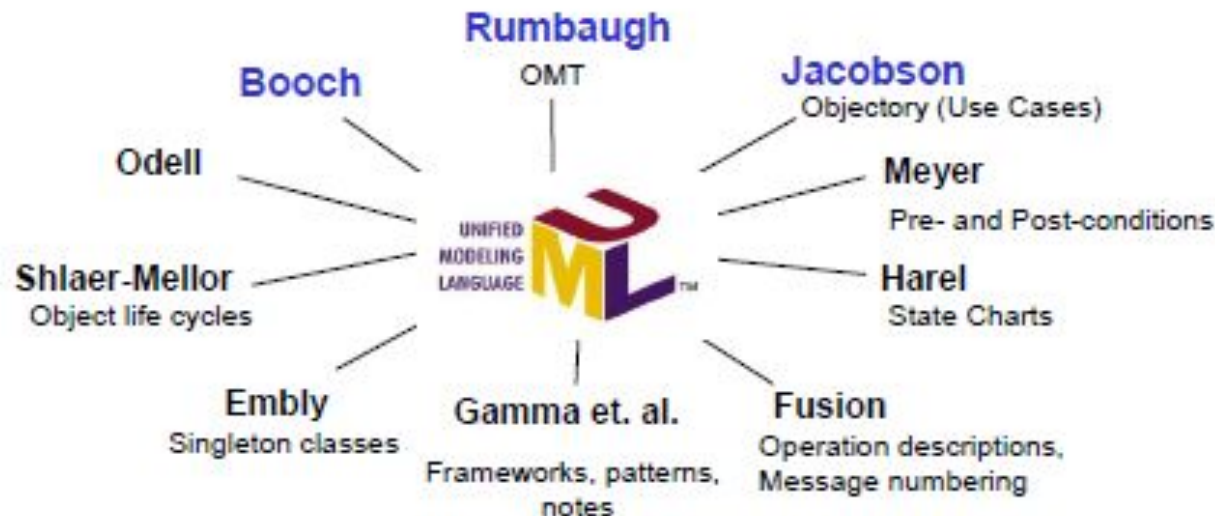
# U.M.L “LENGUAJE UNIFICADO DE MODELADO”

Prof. Valeria Ortiz Quiroz

# ¿QUÉ ES UML?

**UML = Unified Modeling Language**

- Es un lenguaje de modelado visual de propósito general orientado a objetos, para escribir “planos” de software..
- Impulsado por el **Object Management Group** [www.omg.org](http://www.omg.org)
- Estándar: UML es un Lenguaje independiente del proceso de desarrollo aunque para que sea utilizado óptimamente se debe utilizar un proceso que sea dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental.
- Agrupa todos los métodos de programación orientada a objetos.



# SURGIMIENTO DE UML

- Los lenguajes de modelado orientados a objetos aparecieron entre la mitad de los años 70 y finales de los 80, por el uso de los lenguajes de programación O.O.
- Entre fines de los 80 y mediados de los 90 surgen muchos nuevos métodos O.O. Autores mas destacados: Booch, Jacobson (OOSE), Rumbaugh (OMT).
- En los 90 Booch (Rational Software Corporation), Jacobson (Objectory), Rumbaugh (General Electric) comienzan a unificar sus ideas.
- En 1997 se presenta la versión 1.0 de UML a OMG (Object Management Group), que se fue perfeccionando hasta la 1.3.
- Actualmente versión 2.5.1 de UML.  
(<http://https://www.omg.org/spec/UML/>)



# UML SE UTILIZA PARA...

- ▣ **Visualizar:** UML es un lenguaje gráfico y es algo más que un conjunto de símbolos gráficos. Detrás de cada símbolo en la notación UML hay una semántica bien definida.
- ▣ **Especificar:** tiene que ver con la construcción de modelos precisos, no ambiguos y completos. UML cubre la especificación de todas las decisiones de análisis, diseño e implementación que deben realizarse al desarrollar y desplegar un sistema con gran cantidad de software.
- ▣ **Construir:** UML no es un lenguaje de programación visual, pero sus modelos pueden conectarse de forma directa a una gran variedad de lenguajes de programación, esto permite realizar ingeniería directa, es decir, la generación de código a partir de los modelos.
- ▣ **Documentar:** UML permite generar una serie de artefactos además del código fuente.



# VENTAJAS E INCONVENIENTES DE UML

## ▣ Ventajas de UML

- Es **estándar** => Facilita la comunicación
- Está basado en **metamodelo** con una semántica bien definida
- Se basa en una **notación gráfica** concisa y fácil de aprender y utilizar
- Se puede utilizar para modelar **sistemas software** en diversos dominios:
  - Sistemas de información empresariales, Sistemas WEB, sistemas críticos y de tiempo real, etc.
  - Incluso en sistemas que no son software
- Es fácilmente **extensible**

## ▣ Inconvenientes de UML

- No es una metodología. Además de UML, hace falta una metodología OO
- No cubre todas las necesidades de especificación de un proyecto software
  - No define los documentos textuales o el diseño de interfaces de usuario
- Faltan ejemplos elaborados en la documentación
- Puede resultar complejo alcanzar un conocimiento completo del lenguaje
  - Sin embargo => Regla del 80 - 20



# MODELO CONCEPTUAL DE UML

## ▣ Bloques de Construcción de UML

- ▣ Elementos
- ▣ Relaciones
- ▣ Diagramas

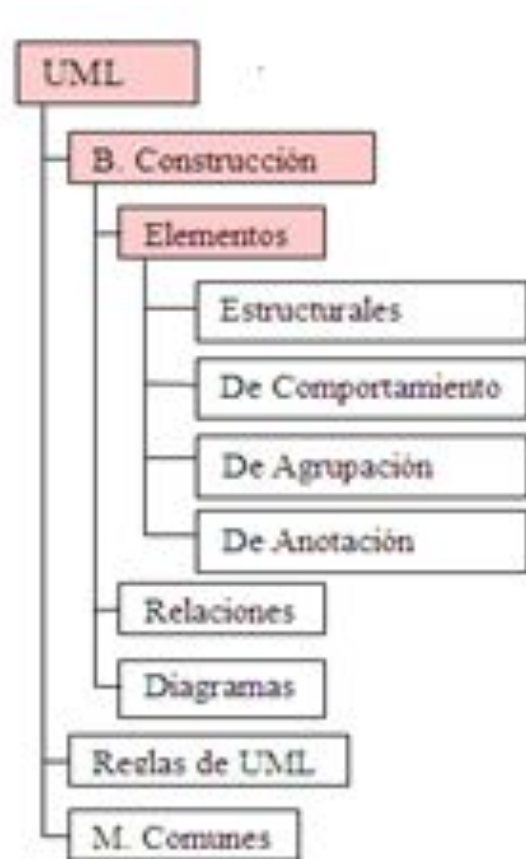
## ▣ Reglas de UML

- ▣ Reglas Semánticas
- ▣ Para la Construcción de Modelos

## ▣ Mecanismos Comunes de UML



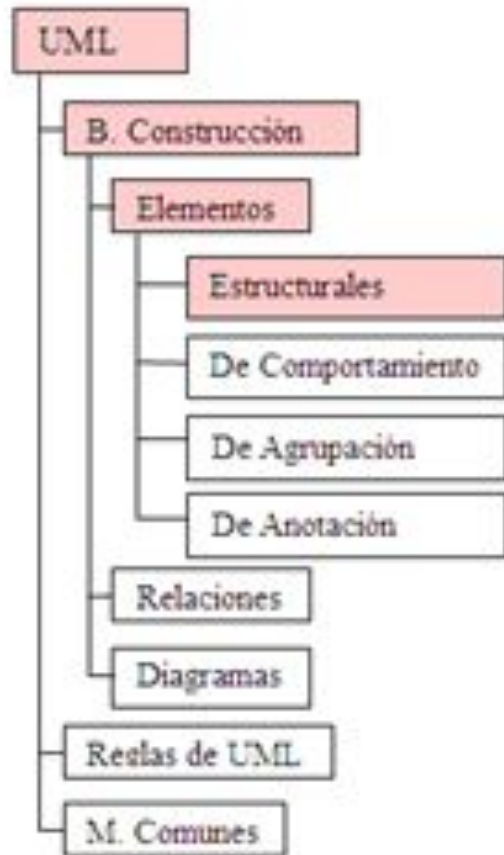
# BLOQUES DE CONSTRUCCIÓN: ELEMENTOS



- Estructurales
- De Comportamiento
- De Agrupación
- De Notación



# ELEMENTOS: ESTRUCTURALES

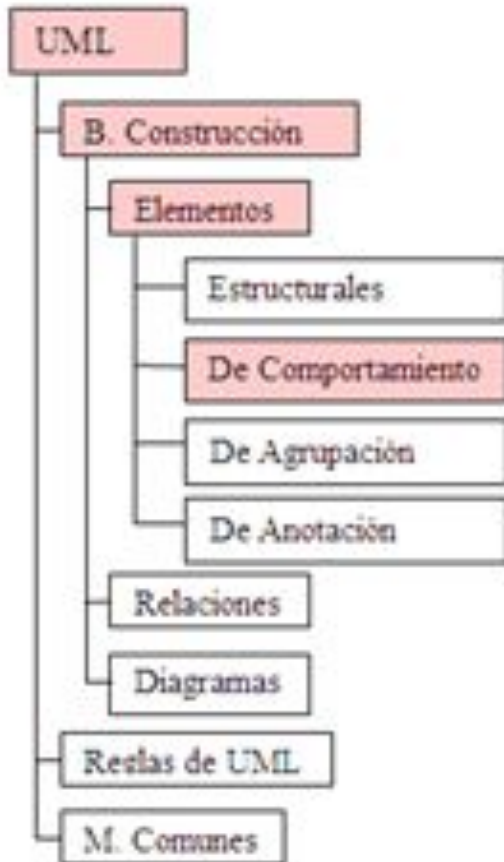


- Clase
- Interfaz
- Colaboración
- Caso de Uso
- Clase Activa
- Nodo



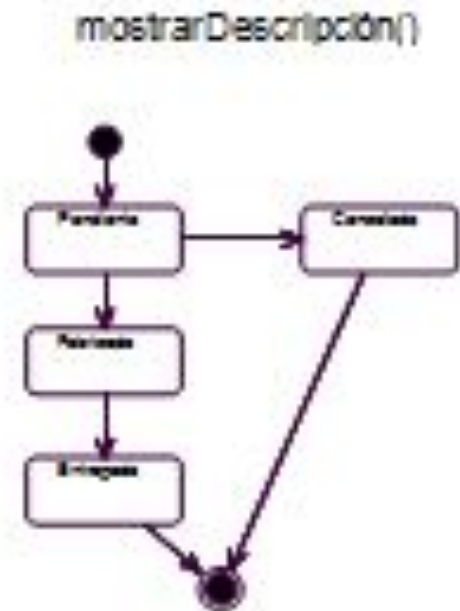


# ELEMENTOS: DE COMPORTAMIENTO

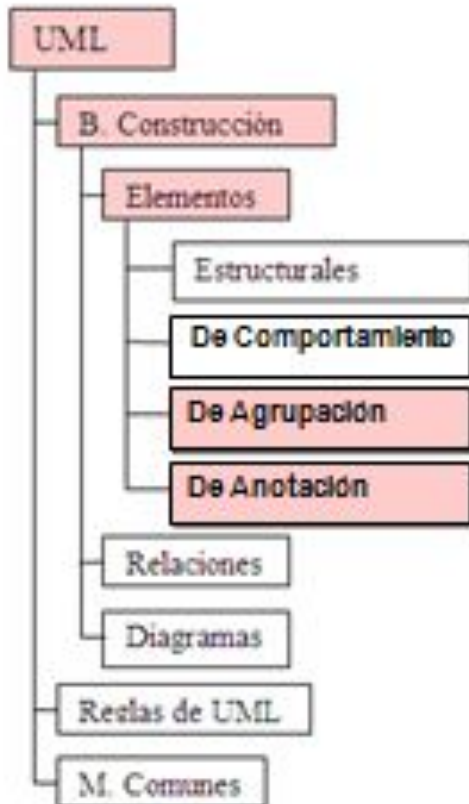


□ Interacción

□ Máquina de estados



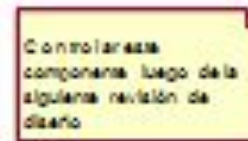
# ELEMENTOS DE AGRUPACIÓN Y NOTACIÓN



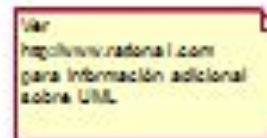
## Paquetes



## Notas

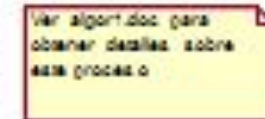


texto simple

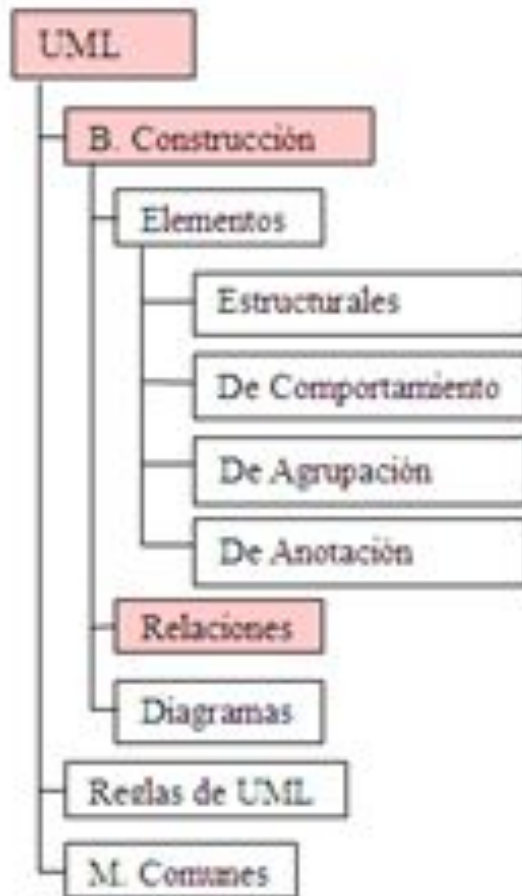


URL embebida

link a un documento



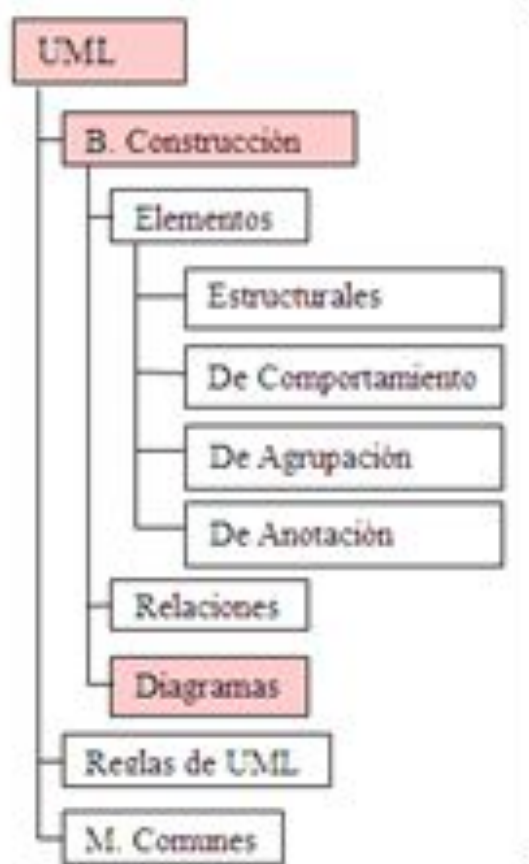
# BLOQUES DE CONSTRUCCIÓN: RELACIONES



- Dependencia
- Asociación
- Generalización
- Realización



# BLOQUES DE CONSTRUCCIÓN: DIAGRAMAS



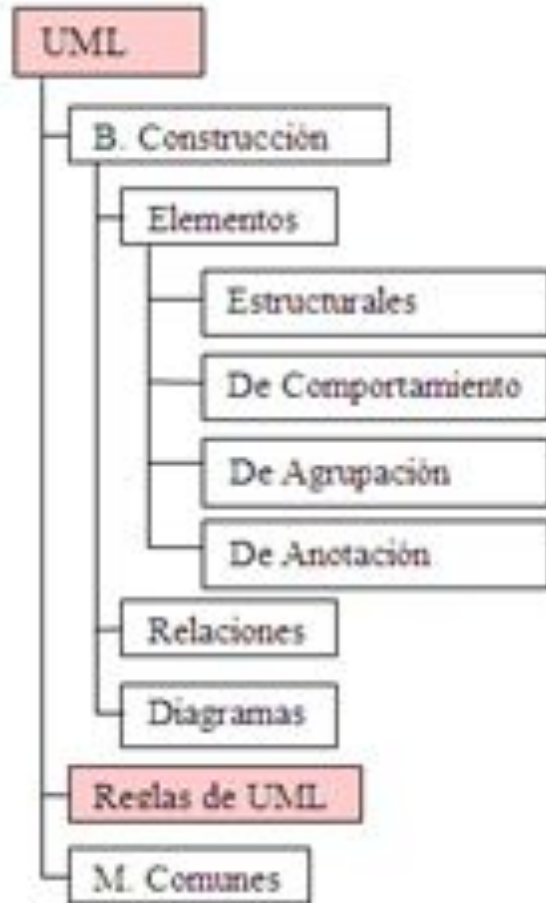
- Clases
- Objetos
- Casos de Uso
- Componentes
- Despliegue
- Interacción
  - Colaboración
  - Secuencia
- Estados
- Actividades

Cubren la vista estática de un sistema.

Cubren la vista dinámica de un sistema



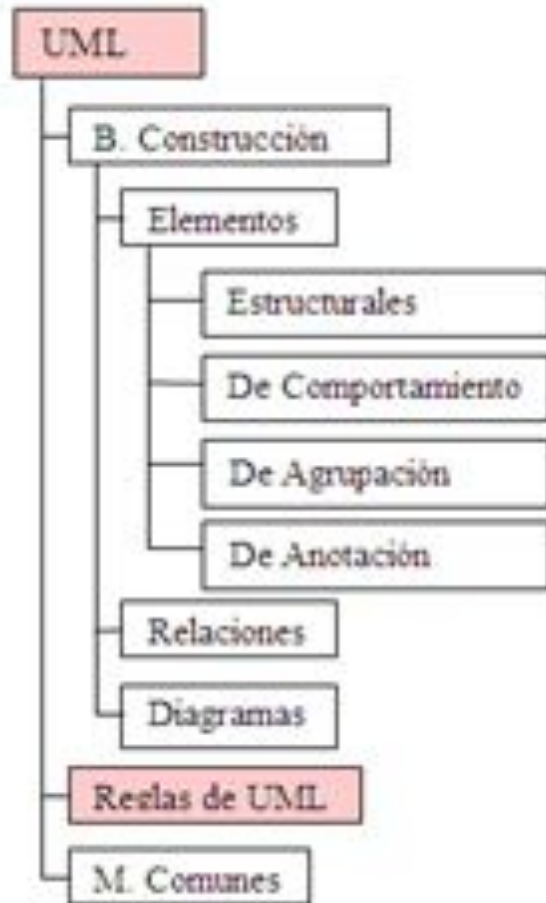
# REGLAS DE UML: SEMÁNTICAS



- ▣ **Nombres:** Cómo llamar a los elementos, relaciones y diagramas.
- ▣ **Alcance:** El contexto que da un significado específico a un nombre.
- ▣ **Visibilidad:** Cómo se pueden ver y utilizar esos nombres por otros.
- ▣ **Integridad:** Cómo se relacionan apropiada y consistentemente unos elementos con otros.
- ▣ **Ejecución:** Qué significa ejecutar o simular un modelo dinámico.



# REGLAS DE UML: CONSTRUCCIÓN DE MODELOS



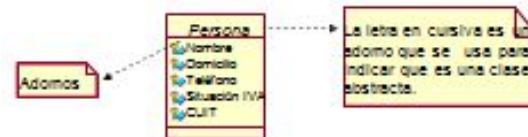
- ❑ **Abreviados:** Ciertos elementos se ocultan para simplificar la vista.
- ❑ **Incompletos:** Pueden estar ausentes algunos elementos, por lo menos en los tramos iniciales del desarrollo.
- ❑ **Inconsistentes:** En estos casos no se garantiza la integridad del modelo.

# MECANISMOS COMUNES



- ❑ **Especificaciones:** UML es más que un lenguaje gráfico. Detrás de cada elemento de notación gráfica hay una especificación que proporciona una explicación textual de la sintaxis y semántica de ese bloque de construcción.

- ❑ **Adornos:**

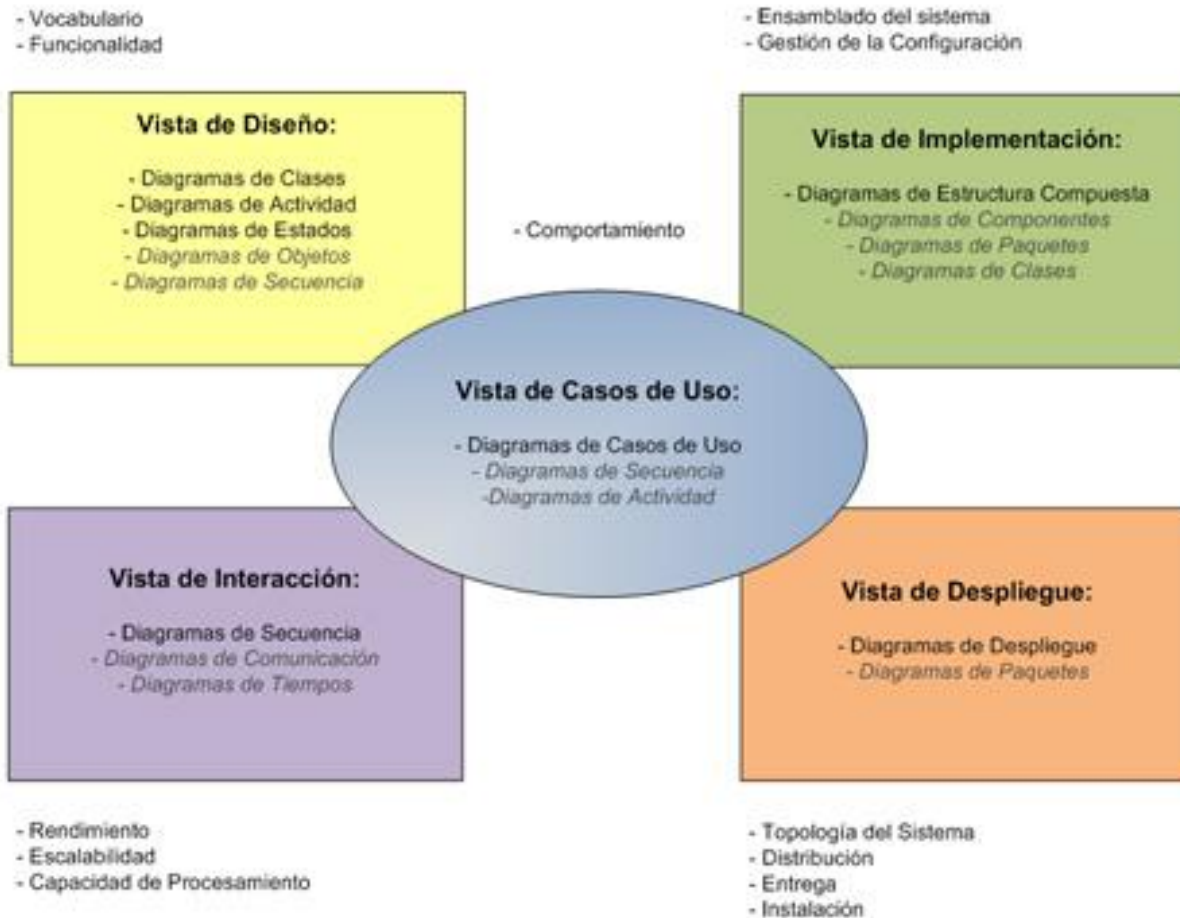


- ❑ **Divisiones Comunes:** Al modelar sistemas orientados a objetos las cosas pueden dividirse al menos en un par de formas
- ❑ **Mecanismos de extensibilidad:** Estereotipos, Valores etiquetados, Restricciones

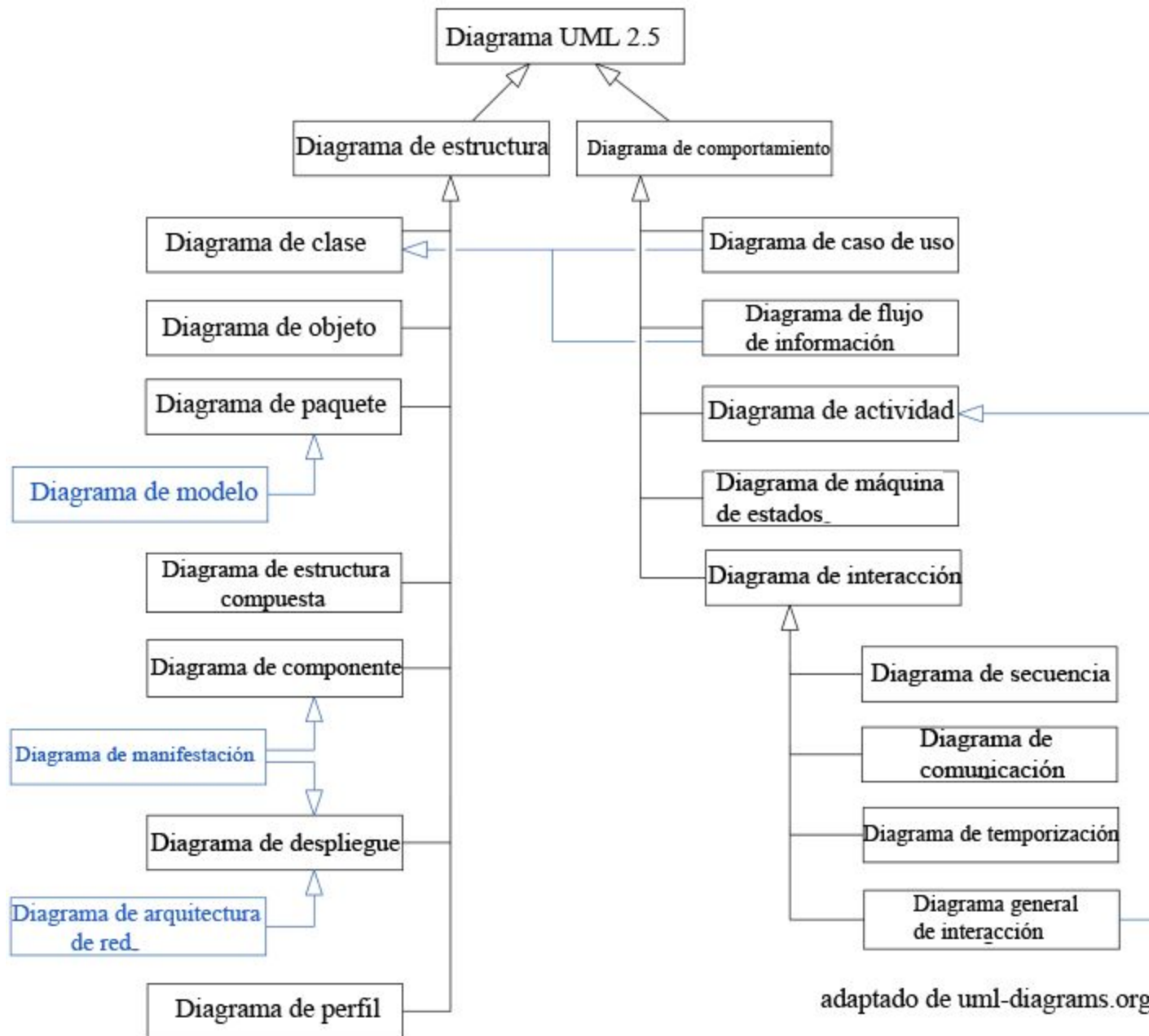


# VISTAS DE UML (PHILIPPE KRUTCHEN)

## ARQUITECTURA DE UN SISTEMA SOFTWARE (UML 2.0)







adaptado de [uml-diagrams.org](http://uml-diagrams.org)