

UML

Lenguaje Unificado de Modelado

¿Que es el UML?

El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. (Rumabaugh, Jacobson, & Booch, 2000)

UML es un lenguaje que nos ayuda a interpretar grandes sistemas mediante gráficos o mediante texto obteniendo modelos explícitos que ayudan a la comunicación durante el desarrollo ya que al ser estándar, los modelos podrán ser interpretados por personas que no participaron en su diseño (e incluso herramientas) sin ninguna ambigüedad. En este contexto, UML sirve para especificar, modelos concretos, no ambiguos y completos. (Alarcón, 2000)

Historia De UML

El lenguaje UML comenzó a gestarse en octubre de 1994, cuando Rumbaugh se unió a la compañía Rational fundada por Booch (dos reputados investigadores en el área de metodología del software). El objetivo de ambos era unificar dos métodos que habían desarrollado: el método Booch y el OMT (Object Modelling Tool).

El primer borrador apareció en octubre de 1995. En esa misma época otro reputado investigador, Jacobson, se unió a Rational y se incluyeron ideas suyas. Estas tres personas son conocidas como los “tres amigos”. Además, este lenguaje se abrió a la colaboración de otras empresas para que aportaran sus ideas. Todas estas colaboraciones condujeron a la definición de la primera versión de UML.

Esta primera versión se ofreció a un grupo de trabajo para convertirlo en 1997 en un estándar del OMG (Object Management Group <http://www.omg.org>). Este grupo, que gestiona estándares relacionados con la tecnología orientada a objetos (metodologías, bases de datos objetuales, CORBA, etc.), propuso una serie de modificaciones y una nueva versión de UML (la 1.1), que fue adoptada por el OMG como estándar en noviembre de 1997.

Desde aquella versión han habido varias revisiones que gestiona la OMG Revision Task Force. La última versión aprobada es la 1.4.

Para finales del 2002 se lanzaría la nueva versión 2.0, donde se incluyen cambios importantes. En las versiones previas del UML, se hacía un fuerte hincapié en que UML no era un lenguaje de programación.

Un modelo creado mediante UML no podía ejecutarse. En el UML 2.0, esta asunción cambió de manera drástica y se modificó el lenguaje, de manera tal que permitiera capturar mucho más comportamiento (Behavior). De esta forma, se permitió la creación de herramientas que soporten la automatización y generación de código ejecutable, a partir de modelos UML.

Construcción

Un modelo UML esta compuesto por tres clases de bloques de construcción:

- **Diagramas:** Son colecciones de elementos con sus relaciones.
- **Elementos:** Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.)
- **Relaciones:** relacionan los elementos entre sí.

Diagramas

Los diagramas representan perspectivas diferentes de un sistema, permitiendo modelar su estructura y comportamiento. Entre los diagramas mas conocidos se encuentran:

Diagrama	Descripción	Prioridad
Diagrama de Clases	Muestra una colección de elementos de modelado declarativo (estáticos), tales como clases, tipos y sus contenidos y relaciones.	Alta
Diagrama de Componentes	Representa los componentes que componen una aplicación, sistema o empresa. Los componentes, sus relaciones, interacciones y sus interfaces públicas.	Media
Diagrama de Estructura de Composición	Representa la estructura interna de un clasificador (tal como una clase, un componente o un caso de uso), incluyendo los puntos de interacción de clasificador con otras partes del sistema.	Baja
Diagrama de Objetos	Un diagrama que presenta los objetos y sus relaciones en un punto del tiempo. Un diagrama de objetos se puede considerar como un caso especial de un diagrama de clases o un diagrama de comunicaciones.	Baja
Diagrama de Paquetes	Un diagrama que presenta cómo se organizan los elementos de modelado en paquetes y las dependencias entre ellos, incluyendo importaciones y extensiones de paquetes.	Baja
Diagrama de Actividades	Representa los procesos de negocios de alto nivel, incluidos el flujo de datos. También puede utilizarse para modelar lógica compleja y/o paralela dentro de un sistema.	Alta
Diagrama de Revisión de la Interacción	Los Diagramas de Revisión de la Interacción enfocan la revisión del flujo de control, donde los nodos son Interacciones u Ocurrencias de Interacciones.	Baja
Diagrama de Comunicaciones (anteriormente: Diagrama de colaboraciones)	Es un diagrama que enfoca la interacción entre líneas de vida, donde es central la arquitectura de la estructura interna y cómo ella se corresponde con el pasaje de mensajes. La secuencia de los mensajes se da a través de un esquema de numerado de la secuencia.	Baja
Diagrama de Secuencias	Un diagrama que representa una interacción, poniendo el foco en la secuencia de los mensajes que se intercambian, junto con sus correspondientes ocurrencias de eventos en las Líneas de Vida.	Alta
Diagrama de Máquinas de Estado	Un diagrama de Máquina de Estados ilustra cómo un elemento, muchas veces una clase, se puede mover entre estados que clasifican su comportamiento, de acuerdo con disparadores de transiciones, guardias de restricciones y otros aspectos de los diagramas de Máquinas de Estados, que representan y explican el movimiento y el comportamiento.	Media
Diagrama de Tiempos	El propósito primario del diagrama de tiempos es mostrar los cambios en el estado o la condición de una línea de vida (representando una Instancia de un Clasificador o un Rol de un clasificador) a lo largo del tiempo lineal. El uso más común es mostrar el cambio de estado de un objeto a lo largo del tiempo, en respuesta a los eventos o estímulos aceptados. Los eventos que se reciben se anotan, a medida que muestran cuándo se desea mostrar el evento que causa el cambio en la condición o en el estado.	Baja
Diagrama de Casos de Uso	Un diagrama que muestra las relaciones entre los actores y el sujeto (sistema), y los casos de uso.	Media

Elementos

Elementos Estructurales

Los elementos estructurales en UML, es su mayoría, son las partes estáticas del modelo y representan cosas que son conceptuales o materiales.

- Clases
- Interfaz
- Colaboración
- Casos de Uso
- Clase Activa
- Componentes
- Nodos

Elementos de comportamiento

Los elementos de comportamiento son las partes dinámicas de un modelo: Se podría decir que son los verbos de un modelo y representan el comportamiento en el tiempo y en el espacio.

- Interacción
- Maquinas de estados

Elementos de agrupación

Los elementos de agrupación forman la parte organizativa de los modelos UML. El principal elemento de agrupación es el paquete, que es un mecanismo de propósito general para organizar elementos en grupos, representándolo gráficamente como una carpeta.

Elementos de anotación

Los elementos de anotación son las partes explicativas de los modelos UML. Son comentarios que se pueden aplicar para describir, clasificar y hacer observaciones sobre cualquier elemento de un modelo.

Relaciones

Dependencia ----->	Es una relación semántica entre dos elementos, en la cual un cambio a un elemento (el elemento independiente) puede afectar a la semántica del otro elemento (elemento dependiente).
Asociación _____	Es una relación estructural que describe un conjunto de enlaces, los cuales son conexiones entre objetos.
Generalización _____>	Es una relación de especialización/generalización en la cual los objetos del elemento especializados (el hijo) pueden sustituir a los objetos del elemento general (el padre). De esta forma, el hijo comparte la estructura y el comportamiento del padre.
Realización ----->	Es una relación semántica entre clasificadores, donde un clasificador especifica un contrato que otro clasificador garantiza que cumplirá.
Agregación _____◇	Es un tipo especial de asociación, un "todo/parte" la relación dentro de la cual uno o más clases son partes de un todo más grande.
Composición _____◆	Es una "fuerte" forma de agregación, donde se comparte la línea de vida y las partes solo pueden pertenecer a un todo.

Bibliografía

Alarcón, R. (2000). UML Diseño Orientado a Objetos con UML. Madrid, España: Grupo Eidos S. L.

Pilone, D., & Pitman, N. (2005). UML 2.0 in a Nutshell. United States of America.: O'Reilly Media Inc.

Rumabaugh, J., Jacobson, I., & Booch, G. (2000). El lenguaje Unificado de Modelado. Manual de Referencia. Madrid, España: Pearson Educación S. A.

Scott, K. (2004). Fast Track UML 2.0. United States of America: Apress.

Hernández Orallo, E. (Octubre de 2002). El Lenguaje Unificado de Modelado (UML). Recuperado el 15 de Octubre de 2011, de <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>

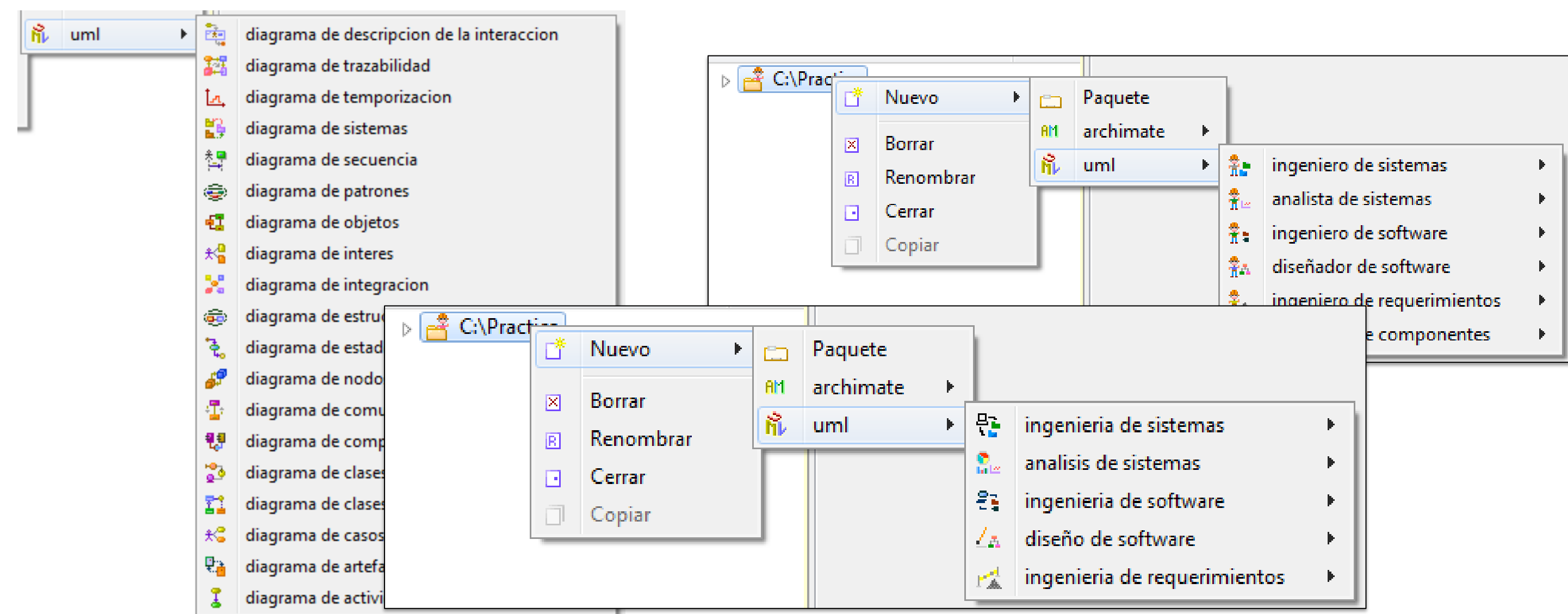
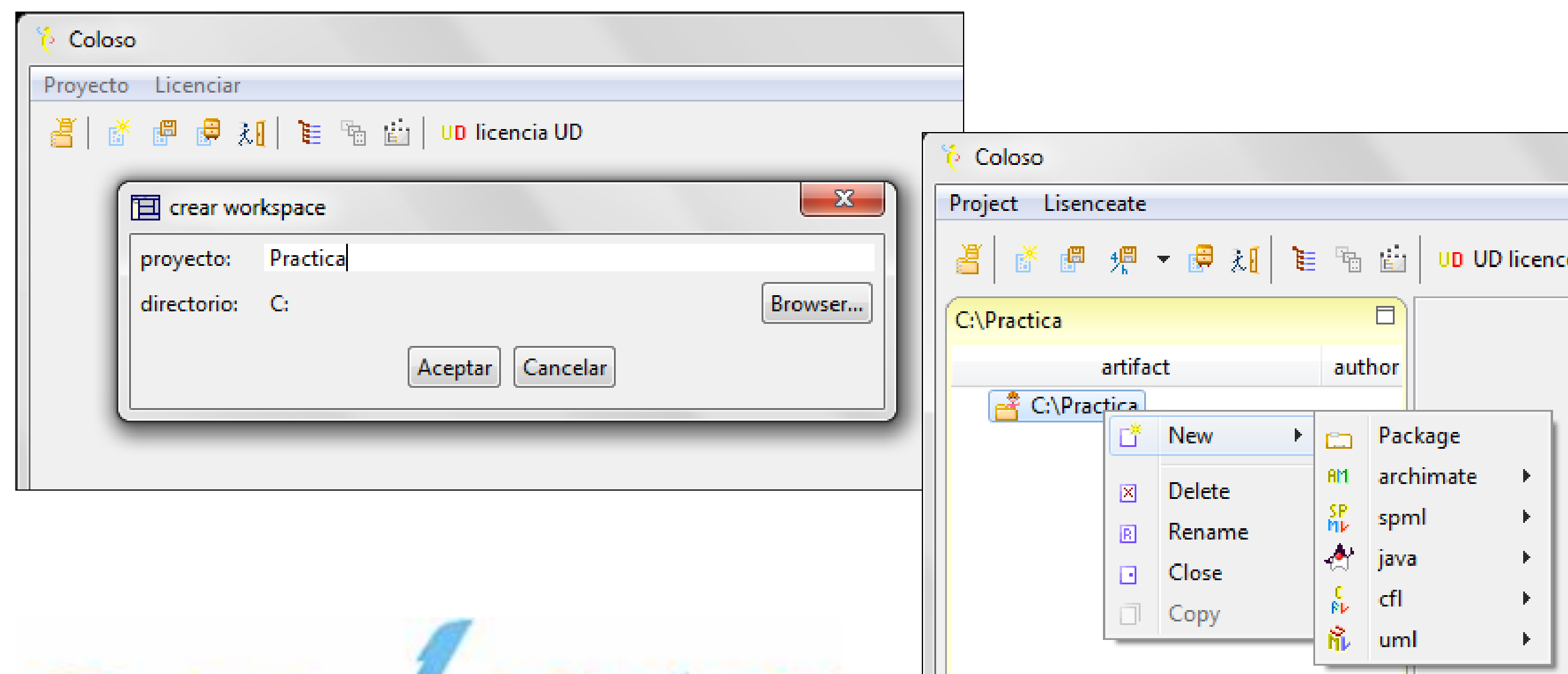
Quirón. (04 de Marzo de 2005). INTRODUCCIÓN A UML 2.0. Recuperado el 15 de Octubre de 2011, de La evolución de la programación hacia la ejecución y validación automática de modelos: http://www.epidataconsulting.com/tikiwiki/tiki-read_article.php?articleId=15

COLOSO & UML

Modelos UML

Construcción de Modelos UML

Coloso en sus diferentes versiones, se encuentra equipado con una serie de herramientas para visualizar, definir, detallar y documentar sistemas de software, a través de los ventajosos diagramas UML. Cada diagrama soportado, trae incorporado los diferentes elementos característicos para su construcción, mas una serie de elementos de apoyo, como: notas, enlaces a otros diagramas, cuadrícula, entre otras. Todo esto, a través de una interfaz amable y atractiva, que permite visualizar los diagramas de acuerdo al rol de quien la aplicara o de acuerdo a la disciplina en la que se usará.



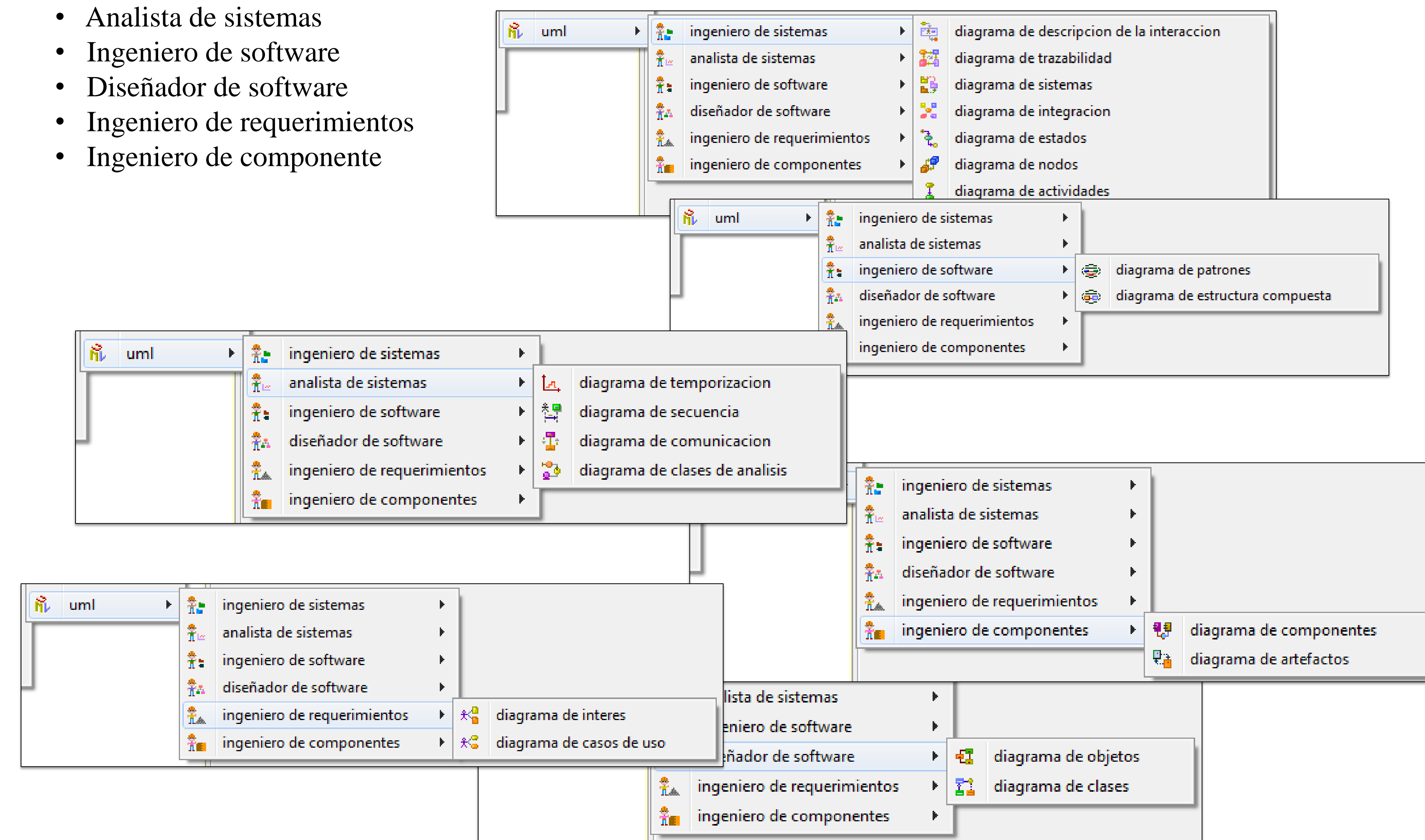
Bibliografía

- Alarcón, R. (2000). UML Diseño Orientado a Objetos con UML. Madrid, España: Grupo Eidos S. L.
- Pilone, D., & Pitman, N. (2005). UML 2.0 in a Nutshell. United States of America.: O'Reilly Media Inc.
- Rumabaugh, J., Jacobson, I., & Booch, G. (2000). El lenguaje Unificado de Modelado. Manual de Referencia. Madrid, España: Pearson Educación S. A.
- Scott, K. (2004). Fast Track UML 2.0. United States of America: Apress.
- Hernández Orallo, E. (Octubre de 2002). El Lenguaje Unificado de Modelado (UML). Recuperado el 15 de Octubre de 2011, de <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>
- Quirón. (04 de Marzo de 2005). INTRODUCCIÓN A UML 2.0. Recuperado el 15 de Octubre de 2011, de La evolución de la programación hacia la ejecución y validación automática de modelos: http://www.epidataconsulting.com/tikiwiki/tiki-read_article.php?articleId=15

Roles

La clasificación de diagramas UML, de acuerdo al rol de las personas que lo aplican, tiene en cuenta a los profesionales que participan en la construcción del software desde la el levantamiento de requerimientos hasta el desarrollo del mismo. Coloso clasifica los diagramas tomando en cuenta los siguientes roles:

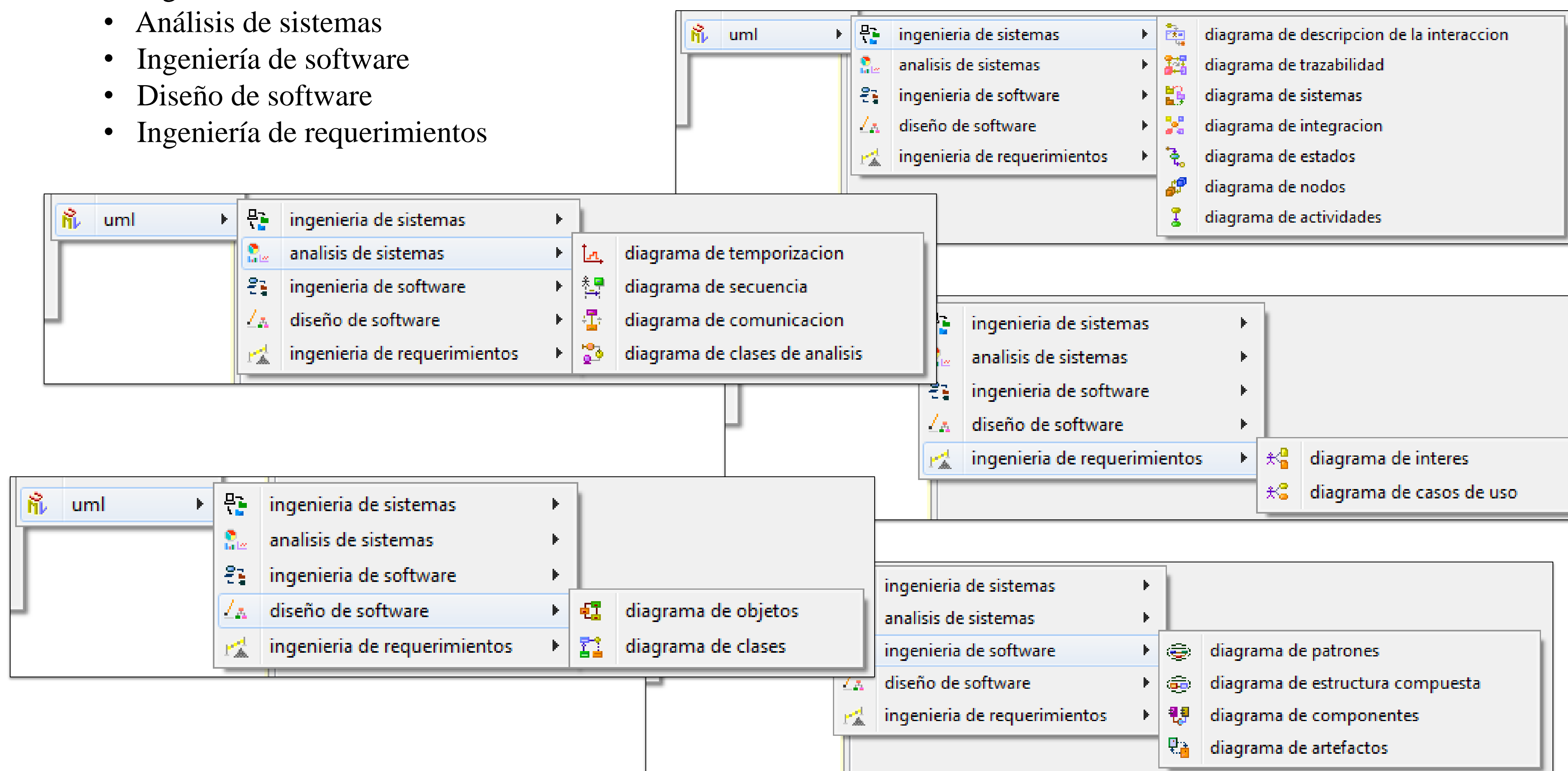
- Ingeniero de sistemas
- Analista de sistemas
- Ingeniero de software
- Diseñador de software
- Ingeniero de requerimientos
- Ingeniero de componente



Disciplinas

Las disciplinas hacen referencia a las áreas específicas que se han formado para abordar el desarrollo de software desde diferentes perspectivas. Cada una de estas se encarga de analizar partes diferentes del proceso de desarrollo, aportando nuevos elementos al sistema. Coloso presenta una clasificación de los diagramas de acuerdo las siguientes disciplinas:

- Ingeniería de sistemas
- Análisis de sistemas
- Ingeniería de software
- Diseño de software
- Ingeniería de requerimientos



Diagramas

Coloso cuenta con las herramientas para construir 19 tipos de diagramas diferentes con todos los elementos propios del lenguaje UML mas herramientas adicionales (como: notas, enlaces a otros diagramas y espacio elementos de código), facilitando el diseño y desarrollo de cualquier aplicación de software.

A continuación se encuentran algunos ejemplos de los diagramas que se pueden elaborar en Coloso:

