	Manual de prácticas del Laboratorio de Programación orientada a objetos	Código:	MADO-22
		Versión:	01
		Página	141/208
		Sección ISO	8.3
		Fecha de emisión	20 de enero de 2017
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Guía práctica de estudio 09: UML




Elaborado por:

M.C. M. Angélica Nakayama C.
Ing. Jorge A. Solano Gálvez

Autorizado por:

M.C. Alejandro Velázquez Mena

	Manual de prácticas del Laboratorio de Programación orientada a objetos	Código:	MADO-22
		Versión:	01
		Página	142/208
		Sección ISO	8.3
		Fecha de emisión	20 de enero de 2017
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Guía práctica de estudio 09: UML

Objetivo:

Utilizar UML como herramienta para diseñar soluciones de software para un lenguaje de programación orientado a objetos.

Actividades:

- Elegir el o los diagrama(s) necesarios para mostrar la solución de un problema.
- Crear diagramas UML para mostrar la solución de un problema.

Introducción

El **Lenguaje de Modelado Unificado (UML - Unified Modeling Language)** es un lenguaje gráfico que permite visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software.

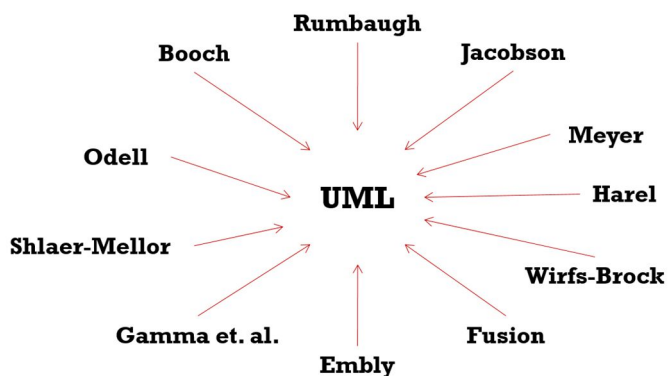



Figura 1. Representación de los lenguajes que componen a UML.

	Manual de prácticas del Laboratorio de Programación orientada a objetos	Código:	MADO-22
		Versión:	01
		Página	143/208
		Sección ISO	8.3
		Fecha de emisión	20 de enero de 2017
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

UML

Los **diagramas UML** permiten modelar aspectos conceptuales como procesos de negocios o funcionalidades de un sistema, cumpliendo con los siguientes objetivos:

- *Visualizar*: expresar de forma gráfica la solución y/o flujo del proceso o sistema.
- *Especificar*: mostrar las características del sistema.
- *Construir*: generar soluciones de software.
- *Documentar*: especificar la solución implementada.

UML está compuesto por tres bloques generales de formas:

- *Elementos*: son representaciones de entes reales (usuarios) o abstractos (objetos, acciones, clases, etc.).
- *Relaciones*: es la unión e interacción entre los diferentes elementos.
- *Diagramas*: muestra a todos los elementos con sus relaciones.

Diseño estático o de estructura

Los **diagramas estáticos o estructurales** aportan una visión fija del sistema, los diagramas que permiten modelar estas características para un lenguaje orientado a objetos son:

- Diagrama de casos de uso
- Diagrama de clases
- Diagrama de objetos

Diseño dinámico o de comportamiento

Los **diagramas dinámicos o de comportamiento** permiten visualizar la comunicación entre elementos del sistema para un proceso específico, los diagramas que permiten modelar estas características para un lenguaje orientado a objetos son:

- Diagrama de estados
- Diagrama de actividades
- Diagrama de interacción


	Manual de prácticas del Laboratorio de Programación orientada a objetos	Código:	MADO-22
		Versión:	01
		Página	144/208
		Sección ISO	8.3
		Fecha de emisión	20 de enero de 2017
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Diagrama de casos de uso

Un **diagrama de caso de uso** define la manera en la que el usuario del sistema interactúa con éste. Los diagramas de casos de uso permiten modelar el comportamiento de un sistema, un subsistema o, incluso, una clase. Está compuesto por 3 elementos básicos:

- Actor(es).
- Caso(s) de uso.
- Relación(es) (uso, herencia y comunicación).

Por lo tanto, para generar un diagrama de casos de uso hay que identificar a los **actores** (o roles) que van a interactuar con el sistema, las **actividades** dentro del sistema, así como los **permisos** (relaciones) que tendrá cada actor con los casos de uso.

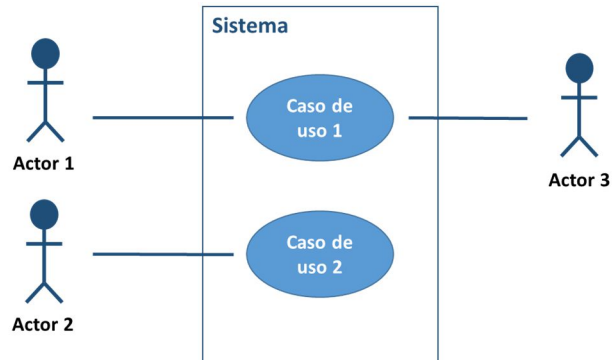


Figura 2. Diagrama de casos de uso.


	Manual de prácticas del Laboratorio de Programación orientada a objetos	Código:	MADO-22
		Versión:	01
		Página	145/208
		Sección ISO	8.3
		Fecha de emisión	20 de enero de 2017
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Diagrama de clases

Un **diagrama de clases** permite modelar las características de las clases que componen al sistema. Dentro de cada clase se pueden visualizar los atributos y métodos que contiene la clase. El conjunto de clases permite observar las relaciones que existen entre ellas dentro del sistema.

Los diagramas de clase están compuestos por 3 elementos básicos:

- Clase(s)
- Cardinalidad
- Relación(es)

Los diagramas de clase permiten visualizar un **panorama general** del sistema, así como de la **comunicación** que se requiere entre las diferentes clases.

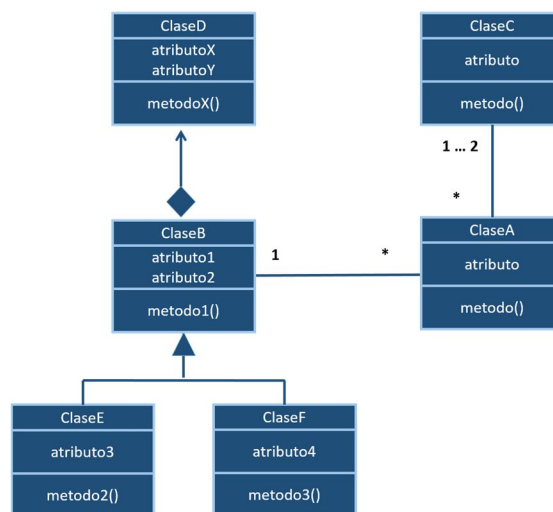


Figura 3. Diagrama de clases.


	Manual de prácticas del Laboratorio de Programación orientada a objetos	Código:	MADO-22
		Versión:	01
		Página	146/208
		Sección ISO	8.3
		Fecha de emisión	20 de enero de 2017
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Diagrama de objetos

Los **diagramas de objetos** modelan las instancias generadas a través de las clases y se utilizan para describir al sistema en un instante de tiempo (o acción) en particular.

Permiten mostrar los objetos y las relaciones entre ellos en un momento dado, por lo tanto, representa la parte estática de la interacción entre objetos (una situación específica en un momento determinado).

Los elementos utilizados para generar este tipo de diagramas son:

- Objeto(s)
- Asociación(es)

Los diagramas de objetos permiten conocer los **valores** (estado) que pueden tener los objetos en un instante de tiempo, así como su **relación** con otros objetos.

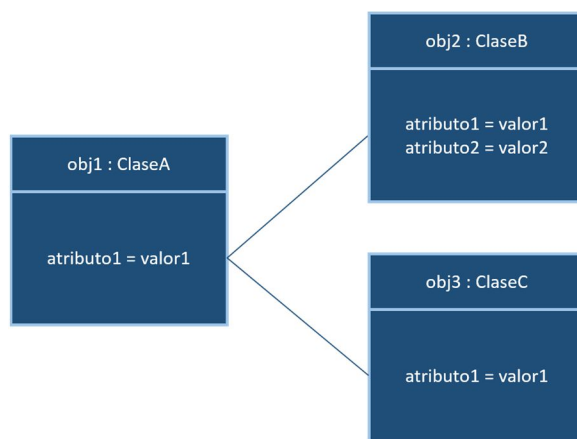


Figura 4. Diagrama de objetos.


	Manual de prácticas del Laboratorio de Programación orientada a objetos	Código:	MADO-22
		Versión:	01
		Página	147/208
		Sección ISO	8.3
		Fecha de emisión	20 de enero de 2017
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Diagrama de estados

Los **diagramas de estados** describen las transiciones por las que puede pasar un objeto durante su tiempo de vida en la aplicación, así como la manera en la que cambia de estado el objeto.

Los elementos de un diagrama de estados son:

- Nodos de entrada y salida.
- Estado(s) del objeto.
- Transición(es) entre estados.

Un diagrama de estados permite modelar el **comportamiento** de un objeto de manera automática, viendo todas las posibles **transiciones** que puede generar.

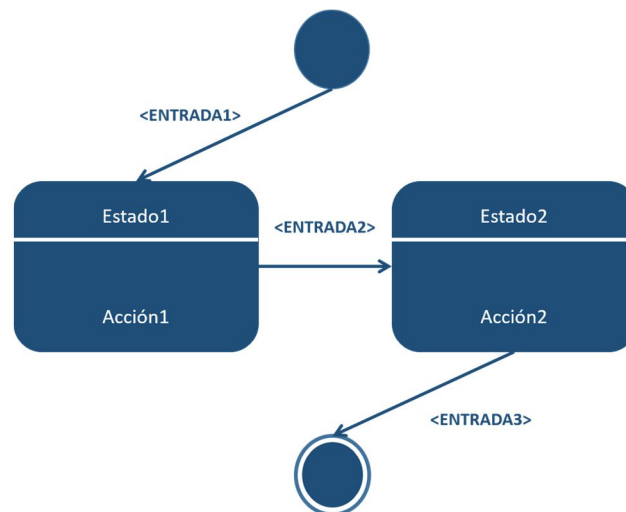


Figura 5. Diagrama de estados.


	Manual de prácticas del Laboratorio de Programación orientada a objetos	Código:	MADO-22
		Versión:	01
		Página	148/208
		Sección ISO	8.3
		Fecha de emisión	20 de enero de 2017
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Diagrama de actividades

El **diagrama de actividades** muestra el flujo de acciones (operaciones que se ejecutan) y los objetos involucrados. Permite visualizar el orden en el que se van realizando tareas dentro de un sistema, así como los objetos involucrados en la comunicación.

Los componentes básicos de un diagrama de actividades son:

- Nodos de entrada y salida.
- Actividad(es).
- Transiciones.

Un diagrama de actividades permite ver la **comunicación** que tienen de manera interna las **actividades** que puede realizar un objeto ya sea con métodos propios del objeto (secuencial) o con otros objetos.

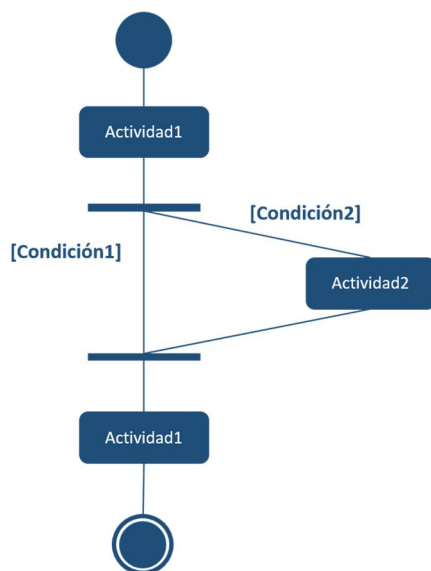


Figura 6. Diagrama de actividades.


	Manual de prácticas del Laboratorio de Programación orientada a objetos	Código:	MADO-22
		Versión:	01
		Página	149/208
		Sección ISO	8.3
		Fecha de emisión	20 de enero de 2017
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Diagrama de interacción

Estos diagramas representan la comunicación que se lleva a cabo entre un cliente (actor) o un objeto (clase) cuando se ejecuta una acción en el sistema.

Los elementos básicos de los diagramas de interacción son:

- Objeto o actor.
- Enlace(s).
- Mensaje(s).

Los diagramas de interacción, a su vez, se dividen en dos tipos:

- Diagrama de secuencia.
- Diagrama de colaboración.

Diagrama de secuencia

Los **diagramas de secuencia** muestran una interacción ordenada de eventos, visualizando los objetos participantes en cada interacción, así como los mensajes que intercambian entre ellos.

Los diagramas de secuencia permiten ver la **interacción** que se genera al ejecutar una **acción**, así como los **objetos involucrados** en el proceso.

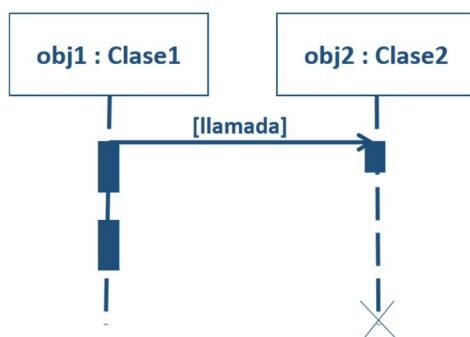


Figura 7. Diagrama de secuencia.


	Manual de prácticas del Laboratorio de Programación orientada a objetos	Código:	MADO-22
		Versión:	01
		Página	150/208
		Sección ISO	8.3
		Fecha de emisión	20 de enero de 2017
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Diagrama de comunicación

Los **diagramas de comunicación** muestran la interacción entre varios objetos y el orden que existen entre ellos. La secuencia de los mensajes y los flujos de ejecución concurrentes se determinan mediante números secuenciales.

Un diagrama de comunicación permite visualizar la **interacción** que se presenta en un **proceso** en particular, permite identificar los objetos involucrados y el orden de invocación.

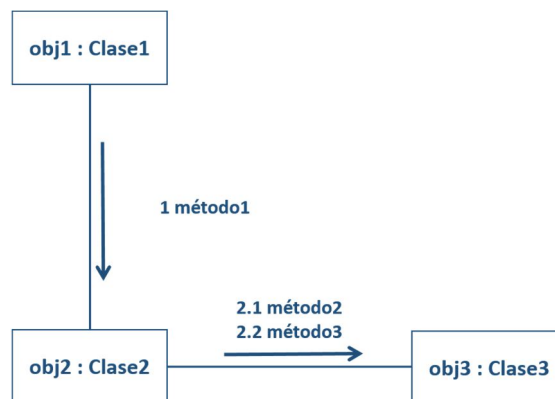



Figura 8. Diagrama de comunicación.

	Manual de prácticas del Laboratorio de Programación orientada a objetos	Código:	MADO-22
		Versión:	01
		Página	151/208
		Sección ISO	8.3
		Fecha de emisión	20 de enero de 2017
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Bibliografía

MILES, Russ, HAMILTON, Kim

Learning UML 2.0

Boston

O Reilly Media, 2006

GOMAA, Hassan

Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures

Washington

Cambridge University Press, 2011

Schmuller, Joseph

Aprendiendo UML En 24 Horas

México

Prentice-Hall, 2000