



**Lectura aprendiendo UML en 24 horas**

**UNIDAD 2**

**Presenta: Mauricio Uriel Puga Martínez**

**Docente: Eduardo Flores Gallegos**

**Desarrollo, Aplicación y Consultoría de sistemas**

---

## Aprendiendo UML en 24 horas

### Contenido

Hora 1 Introducción al UML.....	2
Hora 2 Orientación a objetos.....	3
Hora 3 uso de la orientación a objetos.....	3
Hora 4 uso de relaciones.....	4
Hora 5 Agregación, composición interfaces y realización .....	4
Hora 6 introducción a los casos de uso.....	5
Hora 7 Diagramas de caso de uso.....	5
Hora 8 diagramas de estados.....	5
Hora 9 diagramas de secuencias.....	6
Hora 10 diagrama de colaboraciones. ....	6
Hora 11 diagrama de actividades .....	7
Hora 12 diagrama de componentes. ....	7

### Hora 1 Introducción al UML.

UML es una herramienta de desarrollo de sistemas, la cual es una de las más populares ya que transmite las ideas de los desarrolladores de forma convencional y de manera fácil para la comprensión de otras personas.

UML fue un gran cambio con respecto al cómo desarrollar un proyecto, ya que la forma en que se así esto era crear diseños o bosquejos en papel o simplemente diseñar y programar lo que se iba necesitando. UML dio a solución al mal entendimiento entre desarrollador y cliente, ya que con sus diagramas le da facilidad al desarrollador de explicarle al cliente cómo será el desarrollo del software y el desarrollador podrá comprender los requisitos que necesita el cliente.

UML utiliza diferentes diafragmas para saber cómo implementar un sistema, algunos ejemplos son

- Diagrama de clases: el cual describe clases (las cuales se pueden interpretar como categorías) con sus respectivos atributos (ejemplo marcas, modelos, números de serie, etc.) y con sus respectivas acciones.
- Diagrama de objetos: un objeto es una entidad con valores específicos de los atributos y acciones.
- Diagrama de caso de uso: describe las acciones que puede realizar un usuario con el sistema

- Diagrama de estados: describe la situación en la que está el sistema (por ejemplo: iniciado, cargando, analizando, mostrando y finalizando).
- Diagrama de secuencias: muestra las interacciones que se dan entre clases con acciones.
- Diagrama de actividades: describe las actividades que ocurren dentro de un caso de uso o del comportamiento de un objeto.
- Diagrama de componentes: el moderno desarrollo de software se realiza mediante componentes.
- Diagrama de distribución: muestra la arquitectura física de un sistema informático el cual muestra la interconexiones, equipos y dispositivos.

## **Hora 2 Orientación a objetos.**

La programación orientada a objetos es un paradigma de programación. Un objeto se ve como una instancia de una clase y una clase es una categoría genérica de objetos la cual tiene los mismos atributos y acciones.

Una de las características de la orientación a objetos es la herencia, la cual consiste en heredar tanto atributos y operaciones de su clase.

El polimorfismo consiste en las acciones que pueden tener el mismo nombre en diferentes clases, pero realizar tareas diferentes.

El ocultamiento de funcionalidades es aplicado en la orientación a objetos para aprovechar la funcionalidad de los distintos objetos del sistema.

Los objetos pueden trabajar en conjunto a través de mensajes entre ellos, los cuales son solicitudes para realizar operaciones.

Los objetos tienden asociarse entre sí y estas asociaciones pueden ser de distintos tipos y un objeto puede asociarse con cualquier cantidad de objetos distintos de otras clases.

La agregación consiste en la agregación de un objeto con un conjunto de otros objetos, esto es un tipo de agregación especial.

## **Hora 3 uso de la orientación a objetos**

Para representar una clase de modo simbólica en UML se utiliza un rectángulo, dentro del mismo rectángulo y dividido en secciones van atributos, operaciones y responsabilidades de la clase. Se puede utilizar estereotipos para organizar las listas de atributos y operaciones, además de abreviar una clase al mostrar solo un subconjunto de atributos y contraseñas.

También se puede mostrar el tipo de atributo, su valor inicial y enseñar los valores con funciona una operación, así como sus tipos.

Ambigüedad se recomienda el uso de restricciones. En UML se permite utilizar notas adjuntas al rectángulo para mostrar más información.

Las conversaciones con los clientes dejan sustantivos que pueden utilizarse para crear clases y los verbos se pueden convertir en operaciones los cuales pueden convertirse en un diagrama.

## **Hora 4 uso de relaciones**

Las relaciones dentro del modelo UML permiten un mejor conocimiento de la interacción de las clases y como estas van ligadas. Cada clase en una asociación juega un rol, y la multiplicidad especifica el número de objetos de una clase asociada. Una asociación se representa como una línea entre los rectángulos de clases con los papeles y multiplicidades en cada extremo; de igual manera una asociación puede contener atributos y operaciones como las clases.

Las clases heredan atributos y operaciones, la clase heredada es secundaria de la principal (pero esta puede ser una principal para otra heredada), para identificar las clases heredadas hay que ver el modelo inicial y revisar las clases que tengan atributos y operaciones en común. Las clases abstractas no proporcionan objetos por si mismas ya que se proyectan como base de herencia. La herencia se representa como una línea entre clase principal y secundaria, con un triangulo sin rellenar que apunta a la clase principal.

Las dependencias utilizan otras clases, el uso más común de estas es mostrar que una firma en la operación de una clase utiliza a otra clase, se representan como una línea discontinua que reúne a las dos clases en la dependencia, con una punta de flecha en forma de triangulo sin relleno que adjunta la clase de la que depende.

## **Hora 5 Agregación, composición interfaces y realización**

“Una agregación establece una asociación para conformar un todo; una clase <<todo>> se genera de las clases que lo componen. Un componente en una agregación puede ser parte de mas de un todo. Una composición es una conformación muy íntimamente ligada con agregación en el sentido que un componente de una composición puede ser parte solamente de un todo. La representación del UML de la agregación es similar a la representación de las composiciones. La línea de asignación que une la parte con un todo tiene un rombo. En una agregación el rombo no esta relleno, en tanto que en una composición si lo está.”

Un diagrama de contexto enfoca la atención en una clase especifica dentro de un sistema. Un mapa de contexto se podría describir como un mapa que describe un mapa mayor de forma detallada. Un diagrama de contexto de sistema muestra como un diagrama de clases compuestas se relacionan con otros objetos del sistema.

Una realización es la unión entre una clase y una interfaz, una colección de operaciones que cierta cantidad de clases podrían utilizar y una interfaz se representa como una clase sin atributos. Para poder distinguirla de una clase cuyos atributos hayan sido omitidos de un diagrama se utiliza el estereotipo <<interfaz>> mostrado encima del nombre de la interfaz.

Las interfaces son publicas de modo que cualquier clase puede utilizarla. Los otros dos estados son <<protegido>>, que se refiere a la funcionalidad que hay entre clases secundarias que contienen atributos y operaciones, y el otro estado es privado que se refiere a la utilización de atributos y operaciones dentro de la clase que los contiene. El signo más representa una visibilidad publica (+), el signo de numero (#) la protegida y el guion (-) la privada.

“el ámbito es otro aspecto de los atributos y operaciones. En un ámbito de instancia, cada objeto de una clase cuenta con su propio valor en un atributo u operación.”

## **Hora 6 introducción a los casos de uso**

El caso de uso se utiliza para describir la forma en la que se diseñara un sistema de forma que sea atractiva para los usuarios. Los casos de uso buscan identificar actores y los posibles escenarios en los que puede encontrarse ante el sistema.

Los casos de uso se pueden reutilizar incluyendo pasos utilizados en otros casos de uso como parte la secuencia de pasos de otro caso de uso, otra forma es crear un caso de uso mediante la utilización de pasos de un caso de uso ya existente.

Las entrevistas son la mejor herramienta para determinar los posibles casos de uso, una vez se le hace la entrevista el usuario para determinar lo que tiene que realizar el sistema, el siguiente paso es entrevistar a los posibles usuarios para determinar los posibles escenarios.

## **Hora 7 Diagramas de caso de uso**

Los diagramas de caso de uso son herramientas las cuales permite recabar los requerimientos funcionales; son grandiosas herramientas ya que facilitan la comunicación entre analistas y los usuarios y entre los analistas y clientes. En un diagrama, el símbolo del caso de uso es una elipse. Los actores se representan con figuras y una línea une a un actor con el caso de uso. Por lo general los casos de uso están dentro de un rectángulo que representan el confín del sistema.

Las inclusiones se representan por una línea de dependencia con un estereotipo <<incluir>>. La extensión se representa con una línea de dependencia con el estereotipo <<extender>>. Las otras dos relaciones entre caso de uso son generalización, el la que organiza un conjunto de caso de uso. La generalización se representa por la misma línea que muestra la herencia entre clases. El agrupamiento se representa por el icono de paquete.

Los diagramas de caso de uso se utilizan para el proceso de análisis empezando con entrevistas con los clientes para establecer los diagramas de clase y de esto sacar las bases para la entrevista con los usuarios. Los resultados de las entrevistas de usuario dan como resultado un diagrama de caso de uso de alto nivel que muestra los requerimientos funcionales de un sistema

## **Hora 8 diagramas de estados.**

Los objetos en los sistemas modifican sus estados como respuestas a sucesos y al tiempo; los diagramas de estado de UML se encargan de capturar estos cambios de estado. Los diagramas de estado se enfocan en los cambios de estado en un solo objeto. Un rectángulo de vértices redondeado representa a un estado, y una línea continua con punta de flecha representa una transición de un estado a otro.

El símbolo del estado contiene el nombre del mismo y pueden tener variables y actividades del estado. La respuesta a una transición se da a un suceso desencadenado, e implica una respuesta o acción, también una transición puede suceder por la actividad de un estado, las transiciones que ocurren debido a este motivo se les conoce como transiciones no desencadenada. Para que una transición se ejecute se tiene que cumplir una condición particular o condición de seguridad.

Los estados cuentan con subestados, los cuales pueden ser secuenciales o concurrentes. Los estados que tienen subestados se conocen como estados compuestos. Los estados históricos indican que un estado compuesto recordara su subestado cuando un objeto transcienda de este estado compuesto. Hay dos tipos de estados históricos, el superficial y el profundo. El estado histórico superficial recuerda solo el subestado principal y el profundo recuerda todos los noveles de los subestados.

Los mensajes enviados por objetos que desencadenan una transición en el diagrama de estado de otro objeto se consideran como señal y una señal por si misma es un objeto la cual puede crear una jerarquía de herencia de señales.

Los diagramas de estados facilitan la comprensión de los objetos del sistema a los analista, diseñadores y desarrolladores, ya que es esencial que comprenda el comportamiento que adoptaran los objetos del sistema.

### **Hora 9 diagramas de secuencias.**

Los diagramas de secuencia de UML agregan la dimensión del tiempo a las interactividades de los objetos. En el diagrama los objetos se colocan en la parte superior y el tiempo avanza de arriba hacia abajo y la línea de vida de un objeto desciende de cada uno de ellos. Un pequeño rectángulo de la línea de vida de un objeto representa la activación; puede incorporar los estados de un objeto colocándolos junto a la línea de vida.

Los mensajes se conectan a través de líneas de vida con flechas y estas se conectan entre sí. Los mensajes en la dimensión vertical representan el momento en que sucede dentro de la secuencia. Los mensajes que ocurren primero son los que están en la parte superior del diagrama, y los que suceden después en la parte inferior.

Los diagramas de secuencia se pueden mostrar un escenario de un caso de uso o pueden mostrarse todos los escenarios del caso de uso. Los diagramas de secuencia genéricos con frecuencia dan la oportunidad de representar instrucciones condicionales y ciclos “mientras”. Bordea a cada condición con corchetes, y haga lo mismo en un ciclo “mientras” pero antecede al corchete izquierdo con un asterisco.

Cuando una secuencia incluye la creación de un objeto se representará con un rectángulo de la forma acostumbrada a utilizarse. Su posición en la dimensión vertical representara el momento en el que se creó.

En algunos sistemas las operaciones pueden invocarse a sí mismas, a esto se le conoce como recursividad, la cual se representa con una flecha que sale de la activación hacia si misma y un pequeño rectángulo a la activación.

### **Hora 10 diagrama de colaboraciones.**

El diagrama de colaboraciones es otra forma de presentar un diagrama de secuencias, ya que ambos diagramas son semánticamente equivalentes y se recomienda usar ambos tipos para estimar tiempo y espacio.

El diagrama de colaboraciones muestra las asociaciones entre objetos, así como los mensajes que pasan entre ellos. El mensaje se representa con una flecha junto a la línea de asociación, y una etiqueta numerada que representa el turno y muestra el contenido del mensaje.

Las condiciones se representan como antes, mediante la colocación de la instrucción condicional entre corchetes y el ciclo “mientras” se representa colocando un asterisco antes del corchete izquierdo.

Algunos mensajes provienen de otros y los esquemas de numeración representa esto de forma similar a los manuales técnicos que muestran en sus encabezados y subtítulos con un sistema de numeración que utiliza puntos decimales para representar los niveles del anidamiento.

Los modelos de colaboración permiten modelar varios objetos receptores en una clase, ya sea recibiendo o no mensajes en un orden específico, también se puede representar objetos activos que controlen el flujo de mensajes, así como los mensajes que se sincronizan con otros.

## **Hora 11 diagrama de actividades**

El diagrama de actividades de UML es parecido a los diagramas de flujo, muestran los pasos, puntos de decisiones y bifurcaciones; estos tipos de diagramas son útiles para representar las operaciones de un objeto y los procesos de negocios.

El diagrama de actividades es una extensión del de estados. Los diagramas de actividades se enfocan en las actividades, mientras que el de estados enfoca estados, valga la redundancia, y representan actividades como flechas entre los estados. Las actividades del diagrama de actividades se representan como rectángulos con esquinas redondeadas, mas ovalados en apariencia que la representación de estados, además utiliza los mismos símbolos para los puntos de inicio y final.

Las dispersiones de rutas se representan con una línea gruesa perpendicular a las rutas, misma que se reunirá en una línea similar. En los diagramas de secuencia puede mostrar una señal cuya transmisión se representa con un pentágono convexo, y la recepción con uno cóncavo.

La responsabilidad asignada en los diagramas de actividades se representa con marcos de responsabilidad, mismos que son segmentos paralelos que corresponden a los responsables de realizar cada tarea.

La combinación de símbolos de otros diafragmas produce diagramas híbridos.

## **Hora 12 diagrama de componentes.**

El diagrama de componentes en vez de representar clases o estados representa una parte mas real; un componente del software

A través de una interfaz se puede acceder a una colección de operaciones. La relación que existe entre componente y la interfaz se llama realización.

Los componentes se representan con rectángulos con otros dos rectángulos pequeños sobrepuestos del lado izquierdo. Pueden representar interfaces de dos mane; la primera es un rectángulo que contiene información de interfaz y se conecta con el componente mediante una línea de descontinua con una flecha representada por triángulos sin relleno. La otra es un pequeño circulo conectado al componente con una line continua.

# **Bibliografía**

[1] J. Schmuller, Aprendiendo UML en 24 Horas, Juarez, Mexico: Prentice Hall.

