

## Lectura 1. Aprendiendo UML en 24 horas.

## **Lectura 1. Aprendiendo UML en 24 horas**

### **Hora 1. Introducción al UML.**

UML (Lenguaje unificado de Modelado), es una herramienta para el desarrollo de sistemas, ya que permite a los creadores de sistemas, capturar sus ideas en una forma esencial y fácil de comprender.

- ¿Por qué es necesario?

Anteriormente los programadores tenían sus ideas claras pero no una manera eficaz para plasmarlas y llevar un mejor seguimiento y orden, además de que el cliente, tenía que ser capaz de comprender lo que los desarrolladores estaban creando.

UML, proporciona una organización que permite a los clientes, desarrolladores y otras personas involucradas, comprender.

- Diagramas del UML

El UML se compone de diversos elementos gráficos que se combinan para la creación de los diagramas. UML es un lenguaje, por tanto, contiene ciertas reglas las cuales ayudan a combinar dichos elementos, así teniendo como finalidad una visualización o previsualización de un sistema, o bien, un modelo.

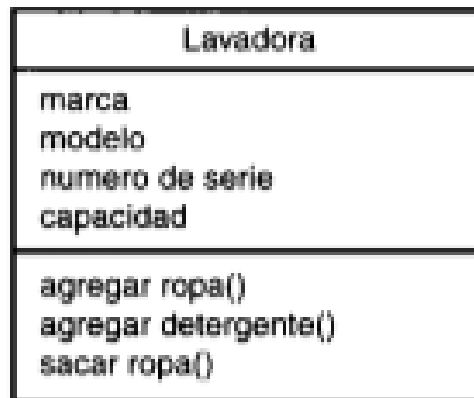
- Diagramas de clases

Entrando más a programación, es un claro ejemplo de lo que es una clase que captura atributos y acciones, como el ejemplo que nos muestra la lectura en la figura

1.1:

**FIGURA 1.1**

*El símbolo UML de una clase.*



- Diagrama de objetos

Un objeto es una instancia de una clase que contiene valores de los atributos y acciones. La siguiente figura (1.2), nos muestra la forma en la que se representa un objeto, en base al ejemplo anterior.

**FIGURA 1.2**

*El símbolo UML del objeto.*



- Diagramas de uso

Es de donde se visualiza las acciones del usuario dentro del sistema.

- Diagrama de secuencias

Representan información estática, brindando un conjunto de instrucciones parecidas a lo que se les llama algoritmos o serie de pasos a seguir.

Entre otros cuantos diagramas, nos muestra la importancia que tiene la estructura de cada uno de ellos y la manera en la que la complejidad de los problemas puede solucionarse.

## Hora 2. Orientación a objetos.

- Objetos, objetos por doquier.

Como se menciona anteriormente en la hora 1, hablaba de objetos, que son la instancia de una clase y que su propósito es desarrollar modelos que reflejen el mundo real.

- Abstracción

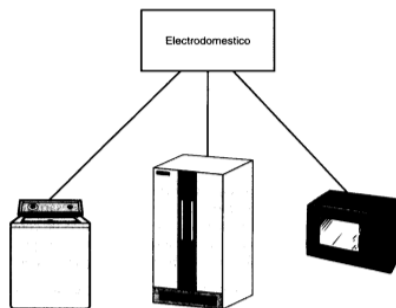
Se refiere a dejar únicamente a dejar sólo las propiedades más importantes de un objeto o bien, las que sólo son necesarias.

- Herencia

Un objeto, no solo hereda una clase, sino que una clase también puede heredar a otra. Un claro ejemplo que nos proporciona la lectura con la figura 2.3

**FIGURA 2.3**

*Los electrodomésticos heredan los atributos y acciones de la clase Electrodomestico. Cada electrodoméstico es una subclase de la clase Electrodomestico. La clase Electrodomestico es una superclase de cada subclase.*



- Polimorfismo

No solo se trata de los modeladores, también es importante para los desarrolladores, ya que les permite tener una comunicación abierta con el cliente para así poder comprender el modelado.

- Encapsulamiento

Se trata de ocultar la funcionalidad interna de los aparatos o programas para que el cliente pueda solo ver lo que se pide.

- Envío de mensajes

Un objeto, envía a otro un mensaje para otro para realizar una operación, y el objeto receptor, ejecutará la operación.

- Asociaciones

Los objetos se relacionan entre sí, de alguna manera. Un ejemplo sería la televisión, que está recibiendo un mensaje o una indicación del cliente (Control remoto) y ésta, está ejecutando la operación solicitada.

- Agregación

Simple, es un objeto que se conforma de una combinación de diversos tipos de objetos.

- La recompensa

Lo importante aquí es la comunicación que se tiene con los clientes, para así poder superarse y cumplir con lo solicitado, teniendo así la recompensa del negocio, gracias a la comunicación y el modelado.

### Hora 3. Uso de la orientación a objetos

- Concepción de una clase

Dependiendo de la estructura de la clase, se va definiendo su comportamiento.

- Atributos

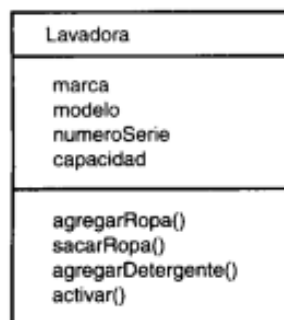
Es una característica de una clase y describe un rango de valores de las propiedades.

- Operaciones

Una operación es algo que la clase puede realizar. Teniendo así, que el nombre de un atributo teniendo cuidado con la manera en la que se escribe.

**FIGURA 3.7**

*La lista de operaciones de una clase aparece debajo de una línea que las separa de los atributos de la clase.*



- ¿Qué es lo que hacen las clases y cómo encontrarlas?

Conforme se vaya hablando con los clientes, se va analizando la situación, su área de conocimiento y el diseño del sistema que resolverá los problemas identificados por el cliente y es ahí en donde se lleva a cabo el modelado en UML.

Se tiene que prestar suma atención con los clientes, para entender las entidades de sus negocios. En pocas palabras la atención cuenta mucho, porque gracias a todo lo que se escuche, es como se va imaginando el modelado del sistema, antes de ser plasmado en UML.

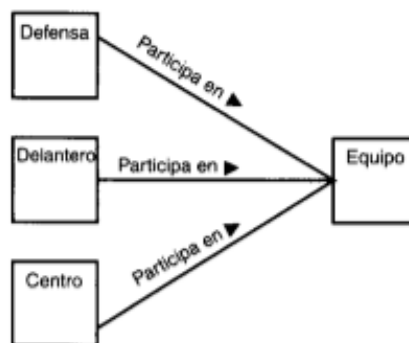
#### **Hora 4. Uso de relaciones**

- Asociaciones

Se conoce como asociación cuando las clases se conectan entre sí de forma conceptual y cuando una clase se asocia con otra cada una de ellas juega un papel importante. Esta asociación puede funcionar en dirección inversa.

Y se pueden volver más complejas como que tan solo una clase conecte a otras.

**FIGURA 4.4**  
*Pueden asociarse  
diversas clases con  
una en particular.*



- Restricciones en las asociaciones

Una asociación entre dos clases debe seguir una regla, la cual, se indica al establecer una restricción junto a la línea de asociación.

Otro tipo de restricción es la relación O (Distinguida como {Or}) en una línea discontinua que conecta a otras dos líneas de asociación.

- Clases de asociación

Aquí se modelan los atributos y operaciones de una asociación. Se conecta a otra mediante una línea discontinua y así puede asociarse a otra clase.

- Vínculos

Es la instancia de una asociación a los objetos en vez de las clases.

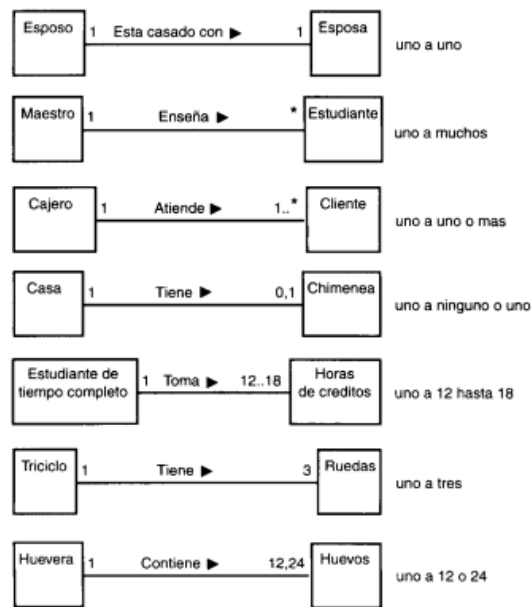
- Multiplicidad

Sugiere que las dos clases tengan una relación de uno a uno, pero cuando se analiza la situación nos daremos cuenta que, no será totalmente necesario irnos por esa opción-

Señala la cantidad de objetos de una clase que pueden relacionarse con un objeto de una clase asociada.



**FIGURA 4.10**  
Posibles multiplicidades y cómo representarlas en el UML.



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

Una agregación establece, una asociación para conformar una clase y se generen las clases que la componen.

La composición es una conformación ligada con la agregación. La representación del UML en las agregaciones es similar a la de las composiciones.

Una realización es una asociación entre una clase y una interfaz, una colección que solo es exclusiva para cierta cantidad de clases que posteriormente serán utilizadas.

Una interfaz es una clase sin atributos y en términos de visibilidad, todas las operaciones en una interfaz son públicas, de modo que cualquier otra clase pueda utilizarlas.

## **Hora 6. Introducción a los casos de uso.**

- ¿Qué son los casos de uso?

La forma en la que se utiliza un sistema se determina en el uso de su estructura. Con una colección de casos de usos en donde se hace un bosquejo de un sistema en términos de lo que los usuarios intenten hacer con él.

- Importancia de los casos de uso

El uso es una herramienta para estimular a que los usuarios potenciales hablen de un sistema, desde sus propios puntos de vista. No es fácil para los usuarios explicarles cómo utilizar el sistema. Lo mejor es involucrar a los usuarios en las etapas iniciales del análisis y diseño del sistema. Por lo que aumenta la probabilidad de que el sistema sea de mayor provecho para la gente a la que supuestamente ayudará.

## **Hora 7. Diagramas de casos de uso**

Los diagramas de casos de uso son herramientas para obtener requerimientos funcionales, debido a que facilitan la comunicación entre analistas y usuarios y entre analistas y clientes.

La inclusión se representa por una línea de dependencia con un estereotipo “incluir”. Para generalizar se representa con la misma línea que muestra la herencia entre las clases.

Los diagramas de casos de uso figuran con fuerza en el proceso de análisis. Se empieza con la comunicación con los clientes, así dando como resultado el diagrama.

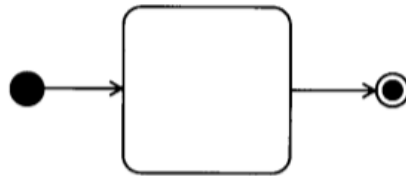
## Hora 8. Diagramas de estado

También son conocidos como motor de estado. Estos caracterizan un cambio en un sistema es decir que los estados que lo componen modificaron su estado como respuesta a los sucesos y al tiempo.

- Simbología

**FIGURA 8.1**

*Los símbolos UML en un diagrama de estados. El icono para el estado es un rectángulo de vértices redondeados, y el símbolo de una transición es una línea continua y una punta de flecha. Un círculo relleno se interpreta como el punto inicial de una secuencia de estados, y una diana representa al punto final.*



- Sucesos y acciones

Pueden indicar transiciones y a la actividad de cómputo que se ejecute y haga que suceda la modificación.

En ocasiones se verán eventos sin que la acción sea asociada y algunas veces sucederá que un estado finalice una actividad.

Cuando un estado consta de un subestado, pueden ser secuenciales o concurrentes. A esto se le conoce como estado compuesto. En cambio un estado histórico indica que un estado compuesto recordará su subestado cuando el objeto trascienda de este estado compuesto y puede ser superficial o profundo.

Cuando un objeto envía un mensaje que desencadena el diagrama en estados de otro objeto, el mensaje es una señal.

Es necesario contar con diagramas de estado porque facilitan la comprensión de los objetos de un sistema.

## **Hora 9. Diagramas de secuencias**

El diagrama de secuencias UML, los objetos se colocan en la parte superior y el tiempo avanza de arriba abajo y la vida de línea depende de cada uno de ellos.

Los mensajes son flechas que conectan a una de vida con otra. La ubicación del mensaje representa el momento que sucede dentro de la secuencia.

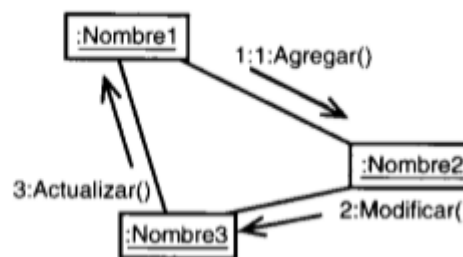
Un diagrama de secuencias puede mostrar una instancia de un caso de uso o puede ser genérico, que dan la oportunidad de representarlas con ciclos.

En algunos sistemas, una operación puede invocarse a sí misma. A esto se le conoce como recursividad. Se representa con una flecha que sale de la activación hacia sí misma y un rectángulo sobrepuesto.

## Hora 10. Diagramas de colaboraciones

Muestra los objetos como tales y sus reacciones, además de eso, muestra los mensajes que se envían entre sí.

**FIGURA 10.1**  
*Simbología del  
diagrama de  
colaboraciones.*



Las condiciones se representan como antes, mediante la colocación de la instrucción condicional entre corchetes. Para representar un ciclo.

Los diagramas de colaboración le permiten modelar varios objetos receptores en una clase, va dependiendo de que los objetos reciban o no los mensajes en un orden específico.

### Bibliografía:

Por S. JOSEPH, Tomado de [https://trello-attachments.s3.amazonaws.com/5e1647e1e290f274a873acd2/5e58644b2de3593d7b9117a1/ba5b1a9962d815a291f51e93140cf4ea/Prentice\\_Hall\\_Aprendiendo\\_UML\\_en\\_24\\_horas.pdf](https://trello-attachments.s3.amazonaws.com/5e1647e1e290f274a873acd2/5e58644b2de3593d7b9117a1/ba5b1a9962d815a291f51e93140cf4ea/Prentice_Hall_Aprendiendo_UML_en_24_horas.pdf)

