



TECNOLOGICO
NACIONAL DE MÉXICO



INSTITUTO TECNOLÓGICO
de Pabellón de Arteaga

ITEC

INSTITUTO TECNOLÓGICO DE PABELLÓN DE ARTEGA, AGS.

INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y LA
COMUNICACIÓN.

4º SEMESTRE

INGENIERÍA DE SOFTWARE

Mtro. Eduardo Flores Gallegos

Alumna. Stephany Joanna Meléndez

“Aprendiendo UML en 24 horas”

*Si sólo tiene tiempo
para las respuestas
24 lecciones en las que invertirá
una hora por lección*



Índice

Hora 1. Introducción al UML.

Hora 2. Orientación a objetos.

Hora 3. Uso de la Orientación a Objetos.

Hora 4. Uso de relaciones.

Hora 5. Agregación, composición, interfaces y realización.

Hora 6. Introducción a los casos de uso.

Hora 7. Diagramas de casos de uso.

Hora 8. Diagramas de estados.

Hora 9. Diagramas de secuencia.

Hora 10. Diagramas de colaboraciones.

Hora 11. Diagramas de actividades.

Hora 12. Diagrama de componentes.

Hora 13. Diagrama de distribución.

Hora 1.

- ✚ El Lenguaje Unificado de Modelado, mejor conocido como UML. Es una aplicación de software que utilizan los analistas y programadores informáticos, cuya finalidad es facilitar el trabajo con todo tipo de diagramas y es representado con un rectángulo. Como bien lo dice es de suma importancia debido a que gracias a este lenguaje es posible establecer una infinidad de series para poder plasmar estructuras o requerimientos necesarios para el software previo. Es importante mencionar que es creado por Grady Booch, James Rumbaugh e Ivár Jacobson. Mejor conocidos como los “tres amigos”. Ellos realizaron un excelente trabajo en equipo, cuya finalidad fue circular en la industria y crecer mayormente. Los diagramas son de suma importancia en la UML, debido que satisface necesidades y poseen ciertas características, entre ellas está la facilidad de comunicación en los programadores y analistas, facilitan el proceso de desarrollo de software, determinan de una mejor manera la situación a presentar, entre muchas más. Es por eso que es de suma importancia conocer que es lo que nos brinda el UML.

Hora 2.

- ✚ La orientación a objetos es un paradigma de programación como bien sabemos se usa en los objetos para crear interacciones, diseñar apps y programas informáticos. En ella se conlleva varias técnicas como lo es la herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento, que son muy importantes para la orientación a objetos. Es preciso mencionar que este paradigma lo podemos presenciar en nuestro día a día. Es importante utilizar los conceptos de orientación a objetos para ayudarnos a comprender de una manera más útil el área de conocimiento de nuestro cliente. También es importante saber que existen ventajas que son la facilidad de mantenimiento, suele acoplarse bien en la utilización de base de datos, los componentes se pueden reutilizar, entre muchas más. Y por lo contrario tiene desventajas la cual es que no hay manera única de poder resolver problemas y algunas limitaciones para el programador. Es importante saber que la orientación a objetos es una manera o forma especial de pensar para poder modelar nuestra realidad en el entorno.

Hora 3.

✚ Es necesario tener en cuenta que la representación debe llevar cierto aspecto para poder llevarse a cabo cierta tarea, es decir una palabra con la primera letra mayúscula y normalmente es colocada en la parte superior del rectángulo, si consta de dos palabras se unen e inician con mayúsculas ambas palabras. Es preciso mencionar que un UML representa como una carpeta. De cierta manera un atributo es coloquial en el uso de la orientación a objetos debido a que son propiedades o características de una clase, es decir, que describe a los objetos. Y una clase puede tener varios o ningún atributo. El UML indica mayor información respecto a una clase, mediante notas adjuntas, para poder facilitar la información más rápido posible. De esta manera el uso en la orientación de objetos es de suma importancia, ya que con ello podrá realizar, utilizar diagramas y poder evidencia de la información adicional.

Hora 4.

✚ Como bien sabemos las relaciones de uso le muestran como o de qué manera se pueden conectar los términos del vocabulario. Podemos darnos cuenta que cada clase juega un papel importante y diferente. Cabe destacar que existen muchos tipos de multiplicidad, en la cual podemos representar asociaciones mediante una línea entre los rectángulos de clases y multiplicidades en cada extremo. Así mismo podemos darnos cuenta que una asociación también puede contener atributos y operaciones. La multiplicidad señala la cantidad de objetos de una clase que puedan relacionarse con un objeto de una clase asociada, y se representan de uno a uno, uno a muchos, muchos a muchos, etc. También un calificador resuelve problemas de búsqueda. Las jerarquías de herencia con clases son abstractas dependiendo en la forma en que lo plantees. Es importante saber que dos o más clases tienen ciertos atributos y operaciones en común. Así como lo hace mención en el juego del baloncesto muestra claramente que un cronometro controla el tiempo restante del partido. Por otro lado, están las dependencias en las cuales se utilizan para que una clase utilice a otra. Es de suma importancia tener presente y claros los conceptos de cada uno.

Hora 5.

✚ Es preciso mencionar que las relaciones entre clases tienen otras maneras de poder representarse u otros conceptos en los cuales pueda ser entendible de una mejor manera la agregación y composición de tal manera que son casos muy particulares de asociación cuya relación esta entre un todo y sus partes, su representación gráfica es una asociación con un rombo en los extremos. Los contextos en cual enfoca su atención en un agrupamiento, las composiciones figuran en gran medida dentro de los diagramas de contexto. Cabe destacar que un diagrama de contexto de composición es como un mapa más detallado. Es preciso saber la forma en que se representa de tal manera que es más fácil para el programador, ya que así puede hacer su trabajo más rápido y no puede que haya ningún conflicto en su realización. Como bien nos pudimos dar cuenta la interfaz es un conjunto de operaciones que especifica cierto aspecto de funcionalidad de tal manera que sea más entendible para el programador.

Hora 6.

✚ Cabe destacar que los casos de uso en el software más que nada pretenden ser herramientas simples para poder describir el comportamiento del mismo software o del sistema, cabe destacar que es mejor conocido como un diagrama de caso de uso. Es de suma importancia debido a que no siempre es fácil de explicar a los usuarios como se pretende utilizar el sistema, y así puedan darse una idea más clara y precisa de cómo se llevara a cabo. Un caso de uso debería de dar resultado para el que lo inicio o para alguien mas. Es preciso mencionar que la mejor manera de realizar casos de uso es entrevistando de manera directa a los usuarios, es importante destacar las condiciones para iniciar. Incluir y extender un caso de uso son de importancia debido a que uno hace que sea más claro los casos de uso y otro que se pueda volver a utilizar un caso de forma distinta. La estructura se vuelve útil cuando tiene que llevar resultados de sus entrevistados y comunicarlos a los desarrolladores. Es de suma importancia saber realizar caso de uso debido a que tienden ayudar a la realización del software de una mejor manera.

Hora 7.

✚ En base a lo que hemos revisado anteriormente nos damos cuenta que existen diferentes metodologías que nos permiten desarrollar software de calidad enfocadas a las necesidades que se tengan. Dentro de UML se pueden encontrar diversos diagramas que permiten representar las diversas perspectivas de un sistema, a las cuales se les conoce como modelo que es una representación simplificada de la realidad. Es una herramienta valiosa dado que es una técnica de aciertos y errores para obtener los requerimientos del sistema, justamente desde el punto de vista del usuario. El rectángulo representa los límites del sistema que contiene los casos de uso. Se representan con óvalos. La etiqueta en el óvalo indica la función del sistema. Un diagrama de caso de uso contiene los símbolos del actor y del caso de uso, junto con líneas conectoras. Los actores son similares a las entidades externas. Un diagrama de caso de uso muestra la extensión y la inclusión. Las clases pueden heredarse entre sí y también aplica en los casos de uso es decir que puede heredar el sentido y comportamiento de otro. Son muy importantes debido a que puedes darte la idea de lo que un software puede contener y de qué manera lo puedes realizar.

Hora 8.

✚ Los diagramas de estado son elementos que muestra como modificar los procedimientos con el tiempo. Los diagramas de estado son útiles para modelar la vida de un objeto. En esta muestra el control entre estados, es decir, en que estados puede estar algo y como puede producir los cambios de dicho estado. También se conoce como un motor de estado. Y de igual manera tiene una simbología, en la cual el icono es un rectángulo de vértices redondos y el símbolo de una transición es una línea continua. Cabe destacar que en este da ciertas opciones agregar detalles es decir permite la edición de ciertos puntos. Aquí mismo podemos darnos cuenta que una máquina de fax es un buen ejemplo de un estado con variables y actividades. Ciertos puntos como la inicialización, operación y apagar, pueden asumirse en el GU I(interfaz gráfica de usuarios). Es de suma importancia saber que cuenta con ciertas condiciones de seguridad y así evitar problemas al elaborar cierto procedimiento o actividad. Son muy importantes los diagramas de estado debido a que nos proporciona una gran variedad de símbolos e ideas. Lo cual permite a los desarrolladores comprender el comportamiento de los objetos de un sistema.

Hora 9.

🚦 Cabe mencionar que en los diagramas de secuencia UML se va agregando dimensiones de tiempo al momento de estar interactuando con los objetos de la misma. Se pueden incorporar estados de cada objeto junto a una línea. Hace mención sobre los tipos de mensaje ya sea simple, sincrónicos y asincrónicos que son flechas que conectan líneas con otras. También un diagrama de secuencia tiene la funcionalidad de mostrar instancias e incorporar los escenarios necesarios para el caso de uso. Más que nada estos sirven para representar aspectos de estructura o de comportamiento de un sistema que existe o que está en proceso. De igual manera modela la lógica de un método y la de un servicio. Un claro ejemplo es un cine cuya estructura es a la de un diagrama de secuencia debido a que se compone de salas de cine, cada una con una capacidad y número de sala. Así mismo son de suma importancia este tipo de diagramas secuenciales debido a que podemos obtener una mejor organización e interactuar de una mejor manera con el software o trabajo que se realice.

Hora 10.

🚦 Un diagrama de colaboración más que nada es otra forma de poder representar información como en el diagrama de secuencia, en pocas palabras van de la mano. Por lo contrario, un diagrama de colaboración muestra las asociaciones entre objetos, así como los mensajes que pasen de un objeto a otro. El único objetivo de estos es describir el comportamiento dinámico del sistema de información, mostrando cómo interactúan entre sí o es decir con que otros objetos tienen vínculos o así mismo intercambia mensajes. Más que nada consiste en como mostrar las instancias específicas de la clase, cabe destacar que existen muchos elementos que lo conforman con la única finalidad de que sea entendible y se pueda tener una buena interacción dentro de estos diagramas. Los mensajes en un diagrama de colaboración se representan con flechas que van junto a un enlace que circulan y con el nombre del mensaje y los parámetros (si los tiene) entre paréntesis. Cada mensaje lleva un número de secuencia que denota cual es el mensaje que procede, excepto el mensaje que inicia el diagrama. De igual manera se puede representar objetos activos que controlen el flujo de los mensajes, así como los mensajes que sincronizan con otros.

Hora 11.

✚ Y si los diagramas de actividades son muy parecidos a los diagramas de flujo debido a que muestran los pasos, puntos de decisión, etc. Estos tipos de diagramas son muy útiles para representar operaciones de un objeto y los procesos que un negocio/empresa tiene.

Básicamente un diagrama de actividad se podría decir que tiene estados de actividad, estados para hacer algo es decir de acción, transiciones u objetos. Desde un punto de vista conceptual un diagrama de actividad muestra como fluye el control de unas clases a otras con la finalidad de culminar con un flujo de control total que se corresponde con la secuencia de un proceso más complejo. Por este motivo en un diagrama de actividad siempre aparecerán acciones correspondientes y actividades a distintas clases. Cabe destacar que es importante mencionar que en este tipo de diagramas puede representarse las actividades de acuerdo con la responsabilidad asignada. En conclusión, son de gran utilidad ya que ayudan a poder llevar un control sobre todo lo que lleve tu empresa, negocio, objeto.

Hora 12.

✚ Lo que distingue a un diagrama de componentes de otros tipos de diagramas es su contenido. Normalmente contienen componentes, interfaces y relaciones entre ellos. Y como todos los diagramas, también puede contener paquetes utilizados para agrupar elementos del modelo. Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes. Dado que los diagramas de componentes muestran los componentes software que constituyen una parte de sus interfaces, y sus interrelaciones, en muchos aspectos se puede considerar que un diagrama de componentes es un diagrama de clases a gran escala. Cada componente en el diagrama debe ser documentado con un diagrama de componentes más detallado, un diagrama de clases, o un diagrama de casos de uso.

Hora 13.

✚ Por ultimo un diagrama de distribución del UML es más que nada ilustrar la forma en la que luce un sistema físicamente cuando sea conjugado. Los Diagramas de Distribución muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. Un nodo es un elemento físico que existe en tiempo de ejecución y representa un recurso computacional, que generalmente tiene algo de memoria y, a menudo, capacidad de procesamiento. Los nodos se utilizan para modelar la topología del hardware sobre el que se ejecuta el sistema. Representa típicamente un procesador o un dispositivo sobre el que se pueden desplegar los componentes.