



INSTITUTO TECNOLÓGICO
de Pabellón de Arteaga

ATEC

LECTURA

Unidad 2



TECNOLÓGICO
NACIONAL DE MÉXICO

Tristán Huerta

Hora 1 – Introducción UML

En esta hora vi el Lenguaje Unificado de Modelado, mejor conocido como UML. Este 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. Se dice que esto es de suma importancia, ya que, gracias a este lenguaje, se puede crear una infinidad de series para capturar las estructuras o requisitos requeridos por el software anterior. Es importante mencionar que fue creado por Grady Booch, James Rumbaugh e Ivár Jacobson. Más conocido como los "tres amigos". Hicieron un excelente trabajo en equipo, cuyo propósito era circular y crecer más en la industria. Los diagramas son de suma importancia en UML porque satisfacen las necesidades y, entre otras cosas, tienen ciertas propiedades. La comunicación simple entre programadores y analistas simplifica el proceso de desarrollo de software y determina, entre otras cosas, la presentación de la situación. Por lo tanto, es extremadamente importante saber qué nos ofrece UML.

Hora 2 – Orientación a objetos

Orientado a objetos, el sistema es más fácil de usar y comprender tanto para desarrolladores como para clientes. Un objeto es la instancia de una clase y también consta de acciones y atributos. Del mismo modo, tiene varias partes que son fundamentales, por ejemplo, la abstracción, que prioriza solo las propiedades y acciones del objeto que son necesarias, ya que no todos los problemas son igualmente complejos y no requieren el mismo desarrollo. Otro aspecto importante es la herencia, ya que las acciones y los atributos se pueden heredar de un objeto a otros objetos. Esto ahorra tanta documentación como sea posible y también simplifica la implementación del sistema. El polimorfismo es el que realiza las operaciones dentro de las acciones. Deriva lo que debe llevarse a cabo en cada acción. Una operación puede ser la misma para diferentes objetos, pero funciona de manera diferente para todos. La información está oculta en la encapsulación, por ejemplo, podemos ver una imagen agradable y diseñada en una computadora,

pero detrás de ella hay código, objetos, etc. La encapsulación se puede llamar una interfaz.

Hora 3 – Uso de la orientación a objetos

En esta tercera hora vi que debe tenerse en cuenta que la representación debe tener un cierto aspecto para llevar a cabo una determinada tarea, es decir, una palabra con la primera letra mayúscula, que normalmente está en la parte superior del rectángulo si consta de dos palabras con las que están conectadas y con las cuales comienzan a capitalizar ambas palabras. Cabe mencionar que un IJML representa como una carpeta. En cierto modo, un atributo es coloquial cuando se usa la orientación a objetos, porque se refiere a propiedades o características de una clase, es decir, objetos. Y una clase no puede tener atributos más que 0. El IJML muestra información adicional sobre una clase con notas adjuntas para proporcionar la información lo más rápido posible. De esta manera, su uso al alinear objetos es de suma importancia porque le permite crear, usar y demostrar la información adicional.

Hora 4 – Uso de relaciones

La asociación es una conexión entre clases, estas asociaciones se utilizan para saber qué clases se conectarán a otras, a veces solo se pueden conectar a una. La asociación está representada por una flecha. Para saber en qué orden funcionarán las clases, en qué momento funcionarán, esto se escribe cerca del diagrama de clase y para determinar si es una restricción encerrada entre llaves. Los enlaces son aquellos que están asociados con objetos en lugar de clases, y el nombre está subrayado para distinguirlos. La multiplicidad es el número de repeticiones o el número de clases y objetos que pueden tener, por ejemplo, de uno a tres neumáticos en la clase de triciclo. También de uno a uno, de uno a muchos, de uno a ninguno o uno, de uno a uno o más. Dependiendo de la dependencia, algunas clases están relacionadas. En la jerarquía de herencia, se

divide en niveles, es decir, un objeto surge de cada clase y más surge de sí mismo.

Hora 5 – Agregación, composición, interface y realización

Aquí pude ver que las relaciones entre clases tienen otras formas de representarse a sí mismas u otros conceptos en los que la agregación y la composición se pueden entender mejor, de modo que son casos de asociación muy específicos, la relación entre un todo y sus partes. existe La representación gráfica es una asociación con un diamante en los extremos. En los contextos en los que enfoca su atención en una agrupación, las composiciones están en gran parte contenidas en los diagramas de contexto. Tenga en cuenta que un diagrama de contexto compositivo se asemeja a un mapa más detallado. Es necesario saber cómo presentarlo de tal manera que sea más fácil para el programador, porque de esta manera puede hacer su trabajo más rápido y no puede haber conflictos en su implementación. Como pudimos ver, la interfaz es una serie de operaciones que especifican un aspecto particular de la funcionalidad para que el programador pueda entenderla.

Hora 6 – Introducción a los casos de uso

Los casos de uso son estructuras que muestran cómo debe diseñarse un sistema. Debido a estos casos de uso, es más fácil para un cliente aprobar el proyecto o no, y para los desarrolladores mostrar una implementación fácil del software. Estos se basan en entrevistas personales con los clientes, las preguntas son comprensiblemente posibles y la redacción de las preguntas también se tiene en cuenta, ya que un cliente no puede tener el mismo dialecto que el desarrollador, este aspecto es importante, de lo contrario esto podría ser una pérdida de tiempo y la implementación de Proyecto. Es importante tener una buena estructura de casos de uso al entrevistar al cliente, porque dependiendo de las necesidades del cliente, es posible deducir entre los desarrolladores si el software requerido para el cliente no es complejo o si en este caso lleva mucho tiempo Se pueden proponer

algunas estructuras al cliente que satisfagan sus necesidades, para que podamos trabajar con ellos y ayudarlos a resolver su problema. La inclusión de casos de uso menciona que se trata de usar otros casos, reutilizar acciones, por así decirlo, o no usar muchos casos de uso para unirlos siempre que lo hagan en ambos casos.

Hora 7 – Diagramas de casos de uso

En base a IO 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 IJML se pueden encontrar diversos diagramas que permiten representar las diversas perspectivas de un sistema, conocido como modelo, que es una representación simplificada de la realidad. Es una herramienta valiosa, ya que es una técnica de éxito y fracaso para obtener los requisitos del sistema exactamente desde el punto de vista del usuario. El rectángulo representa los límites del sistema que contienen los casos de uso. Se muestran con óvalos. La etiqueta en el óvalo muestra la función del sistema. Un diagrama de casos de uso contiene el actor y los símbolos de casos de uso, así como las líneas de conexión. Los actores son como entidades externas. Un diagrama de caso de uso muestra el alcance y la inclusión. Las clases pueden heredarse entre sí y también se aplica en casos de uso, es decir, puede heredar el significado y el comportamiento de otro. Son muy importantes porque le dan una idea de qué software puede contener y cómo puede hacerlo.

Hora 8 - Diagramas de estados

El diagrama de estado es usado para organizar el comportamiento de clases y objetos. Representa la secuencia de estados que atraviesa un objeto de clase a lo largo de su vida. Tiene un principio y un final, y cada evento representa algo que hace que el objeto sea cambiante. El rectángulo con esquinas redondeadas representa el estado, el círculo sólido representa el comienzo, las flechas son

líneas de transición que representan el movimiento que tiene lugar de un estado a otro, es decir, el evento que causa la transición y representa un objetivo El punto final. Así como las clases, los nombres, los atributos y las acciones pueden subdividirse, estos diagramas de estado también pueden subdividirse en nombres, variables de estado y actividades. Las variables de estado son el tiempo y las actividades son acciones y eventos y utilizan entrada, salida y ejecución. Se pueden agregar detalles y eventos a los detalles, lo que provoca un cambio de estado. Las condiciones de seguridad están asociadas con esto, porque dependiendo de lo que se está trabajando y en qué estado se cambia, se pueden tomar decisiones para mejorar esto y así lograr un mejor resultado.

Hora 9 – Diagramas de secuencias

Vale la pena señalar que en los diagramas de secuencia UML, se agregan dimensiones de tiempo al interactuar con sus objetos. Los estados de la década del objeto se pueden insertar al lado de una línea. Se mencionan los tipos de mensajes, ya sean simples, sincrónicos o asincrónicos, que son flechas que conectan líneas con otros. Un diagrama de secuencia también ofrece la funcionalidad para mostrar instancias e incluir los escenarios necesarios para la aplicación. Sobre todo, sirven para representar aspectos de la estructura o el comportamiento de un sistema existente o en proceso. Del mismo modo, modela la lógica de un método y la de un servicio. Un claro ejemplo es un cine, cuya estructura es la de un diagrama de secuencia, ya que consta de cines cada uno con capacidad y número de salas. Del mismo modo, este tipo de diagrama secuencial es de suma importancia porque podemos lograr una mejor organización e interactuar mejor con el software o el trabajo que se realiza.

Hora 10 – Diagramas de colaboraciones

Los diagramas de colaboraciones también muestran cómo los objetos funcionan juntos y los mensajes que se intercambian entre ellos. Estos enfatizan el contexto

y la organización de los objetos. Este diagrama se basa en el espacio y generalmente evita una multitud. Con estos diagramas, puede sincronizar el envío de mensajes de un objeto a otro para recibir un pedido y cumplir con los plazos de entrega. Un objeto activo envía mensajes a objetos pasivos para que pueda interactuar con objetos activos. Cuando un objeto envía un mensaje a un grupo de objetos de la misma clase, está representado por varios rectángulos para el grupo de objetos. Se utiliza una condición entre paréntesis y un asterisco para saber que el mensaje se enviará a todos los objetos. Con estos diagramas puede identificar el espacio de almacenamiento utilizado. También puede enviar mensajes desde un objeto a diferentes objetos para sincronizar y organizar los mensajes.

Hora 11 – Diagramas de actividades

Y si los diagramas de actividad son muy similares a los diagramas de flujo porque muestran los pasos, puntos de decisión, etc. Estos tipos de diagramas son muy útiles para representar las operaciones de un objeto y los procesos de una empresa. Básicamente, se puede decir que un diagrama de actividad tiene estados de actividad, estados para hacer algo, es decir, acciones, transiciones u objetos. Conceptualmente, un diagrama de actividad muestra cómo el control fluye de una clase a otra para culminar en un flujo de control completo que sigue la secuencia de un proceso más complejo. Por esta razón, un diagrama de actividad siempre muestra las acciones y actividades correspondientes para diferentes clases. Cabe señalar que es importante tener en cuenta que en este tipo de diagrama las actividades se pueden mostrar de acuerdo con la responsabilidad asignada. En resumen, son muy útiles porque ayudan a realizar un seguimiento de todas las E / S que lleva su empresa, empresa o propiedad.

Hora 12 – Diagramas de componentes

Un componente es una parte física que forma un sistema. Los componentes son parte de una interfaz. Si se debe mostrar un objeto, esto se hace a través de una

interfaz que solo ejecuta la acción y no muestra lo que hay detrás. La relación entre una clase y su interfaz se llama implementación. Un componente puede usar los servicios de otro componente. La reutilización de un componente puede hacer una gran contribución al desarrollo del software y les ahorraría a los desarrolladores el trabajo evitando la reutilización de muchas líneas de código, lo que ahorra mucho trabajo. Hay tres tipos de componentes: distribución, procesamiento del producto y ejecución. El icono de representación del diagrama de componentes es un rectángulo con dos superposiciones a la izquierda. Los nombres deben insertarse en el diagrama. Estos diagramas de componentes se pueden implementar para la implementación de sitios web, por ejemplo, para un applet que funciona en un sitio web, así como para el desarrollo de software. Hay dos formas de representar las interfaces: la primera está representada por un rectángulo y la segunda por un círculo vacío.

Hora 13 – Diagramas de distribución

Un diagrama de distribución IJML es más que nada para ilustrar cómo se ve un sistema cuando se conjuga. Los diagramas de distribución muestran la disposición física de los distintos nodos que componen un sistema y la distribución de componentes en esos nodos. Un nodo es un elemento físico que existe en tiempo de ejecución y es un recurso informático que generalmente tiene algo de memoria y se procesa con frecuencia. Los Power_Nodes se utilizan para modelar la topología del hardware en el que se ejecuta el sistema. Por lo general, es un procesador o dispositivo en el que se pueden implementar los componentes.