



Lecturas

UNIDAD 2

Yatziri Amparo Esquivel Cruz

INSTITUTO TECNOLÓGICO DE PABELLÓN DE ARTEAGA

Hora 1

Para poder llevar a cabo un proyecto de software es necesario tener una buena documentación y modelado ya que si se desarrolla un software complejo es indispensable por si en determinado momento surge un error poder detectarlo fácilmente y darle solución para que no se ve afectado todo el avance del mismo. Es importante que el UML sea entendible y describa a la perfección lo que se desea hacer, dar una buena visión del proyecto para que así sea fácil de descifrar por los desarrolladores o miembros del equipo.

El UML está conformado por diferentes diagramas como su nombre lo dice es un modelo y por ende contiene estos diagramas que hacen más sencilla la interpretación de lo que se desea hacer, aunque estos diagramas describen lo que hará el sistema no describen como es que se va a implementar.

El diagrama de clases muestra las acciones y atributos de un objeto, estos diagramas están formados por rectángulos y en cada uno de ellos se clasifican las acciones y atributos, esto resulta más entendible lo que se va a realizar.

También está el diagrama de objetos, de casos de uso, de estados, de secuencias, de actividades, de colaboraciones, de componentes y de distribución, cada uno de estos tiene un enfoque diferente a las partes del sistema por eso es importante agregarlos a la documentación.

Hora 2

La orientación a objetos hace que el sistema sea más fácil de usar y comprender tanto para los desarrolladores como para el cliente. Un objeto es la instancia de una clase a su vez también está conformado por acciones y atributos.

Así mismo tiene diferentes partes que son fundamentales por ejemplo la abstracción que da prioridad solo las propiedades y acciones del objeto que sean necesarias ya que no todos los problemas son iguales de complejos ni necesitan el mismo desarrollo. También otro aspecto importante es la herencia ya que de un objeto se pueden heredar acciones y atributos a otros objetos, esto hace que se ahorre mayor documentación posible y así mismo facilita la implementación del dicho sistema.

El polimorfismo es el que realiza las operaciones dentro de las acciones, deduce cual es el que se va a hacer en cada una, una operación puede ser la misma para diferentes objetos, pero funcionar de manera distinta para todos. En el encapsulamiento se trabaja la ocultación de la información, por ejemplo, en una computadora podemos ver una imagen bonita y diseñada pero detrás de ella se encuentra código, objetos, etc. Puede decirse que el encapsulamiento es una interfaz.

Hora 3

En el uso de la orientación a objetos se establecen diferentes aspectos, concepción de una clase, atributos y operaciones. En la concepción de una clase se utilizan los rectángulos así mismo debe tener un nombre, esto para diferencias cuales son las acciones y atributos.

Los atributos están ligados a la concepción de una clase ya que es una característica o una propiedad y describe que es lo que debe de tener cada una de ellas, una clase puede tener uno o ningún atributo. De ahí se desprende a la identificación de los objetos que van a conformar la clase deben de ser solo los necesarios y saber identificar si es atributo o acción ya que en algunas ocasiones es confuso porque en algún objeto el atributo puede ser el mismo pero la acción totalmente diferente.

También es importante asignar un valor a cada objeto, uno de ellos puede ser el tipo de valor, por ejemplo, "string". Las acciones u operaciones deben especificar qué es lo que hará el objeto exactamente, estos se ubican debajo de la lista de objetos, separados por una línea.

Gracias a esto es mucho más sencillo poder implementar el proyecto y también se podrá dar un buen modelado al cliente para saber si exactamente lo que desea, de lo contrario esto podrá corregirse a tiempo y también si algo falla al momento de la implementación será más fácil identificar en donde ocurrió el problema. Con estos diagramas de clase nos damos cuenta de que tan complejo puede resultar nuestro proyecto.

Hora 4

La asociación es una conexión entre clases, estas asociaciones sirven para saber cuáles clases se van a conectar con otras en ocasiones solo pueden estar conectadas todas a una, la representación de la asociación es por medio de una flecha, estas tienen restricciones, por ejemplo, para poder saber cuál es el orden en que van a funcionar las clases oh cual es el momento en que van a accionar, esto se escribe cerca del diagrama de clase y para identificar si es una restricción se pone entre llaves. Los vínculos son los que conectan a los objetos en lugar de las clases y se subraya el nombre para diferenciarlo.

La multiplicidad es el número de veces que se va a repetir, o la cantidad con que cuenta las clases y objetos, por ejemplo, en la clase triciclo puede tener de uno a tres llantas. También de uno a uno, de uno a muchos, de uno a ninguno o uno, de uno a uno o más.

En la dependencia se relacionas unas clases con otras. En la jerarquía de herencia se divide en niveles, es decir que de cada clase surge un objeto y de ellos mismos surgen más.

Hora 5

La agregación indica que una clase se asocia con otra clase, esta es representada por medio de un rombo en color blanco, en la agregación para saber la restricción es representada mediante una O que indica que puede ser una de las dos clases asociadas. En la composición se dice que una clase debe de coincidir con la otra para poder ser asociadas, y su símbolo de representación es un rombo en color negro que indica que está conectado al "todo".

Los contextos se utilizan para hacer un mapa o agrupación de todas las clases, es decir, que se ordenan en conjunto todos los diagramas para ir dando forma al modelaje. La interfaz es el conjunto de operaciones que realiza una clase, es decir todas las acciones que debe de hacer esa misma.

La visibilidad es una relación más de la interfaz, esta tiene tres niveles el público, privado y nivel protegido. Se aplica a operaciones o atributos y con cada uno de los niveles se identifica de que clases son heredadas otras y así poder deducir cual nivel le corresponde a cada uno.

El ámbito es otro aspecto de los atributos y operaciones, hay dos tipos de ámbitos el de instancia y archivado. En el de instancia es independiente cuenta con su propio atributo y operación y en el archivado solo contara con un atributo para todas las instancias de la clase.

Hora 6

Los casos de uso son estructuras que muestran cómo va a estar diseñado un sistema, debido a estos casos de uso resulta más sencillo a un cliente aprobar el proyecto o no, así mismo a los desarrolladores les muestra una fácil implementación del software. Estos se van construyendo conforme a entrevistas personales con los clientes, se hacen las preguntas más entendiblemente posibles y también se toma en cuenta la formulación de las preguntas ya que un cliente no puede llegar a tener el mismo dialecto que el desarrollador es importante este aspecto ya que de lo contrario podría resultar una pérdida de tiempo y esto retrasaría la implementación del proyecto.

Es importante tener una buena estructura de casos de uso una vez entrevistado el cliente ya que de acuerdo a las necesidades del cliente se podrá deducir entre los desarrolladores si el software que requiere el cliente no es complejo o tiende a llevarse mucho tiempo, si ocurre esto se podrán proponer algunas estructuras al cliente similares a su necesidad y así poder colaborar con él y ayudarlo a solucionar su problema.

La inclusión de los casos de uso menciona que es hacer uso de otros casos, reutilizar acciones por así decirlo, o también para no poner muchos casos de uso unirlos unos con otros siempre y cuando realicen los casos de ambos.

Hora 7

El diagrama de casos de uso es una representación de lo que hará el sistema, por ejemplo, en mi proyecto tengo dos perfiles que son administrador y usuario, así mismo cada uno tiene objetos diferentes e iguales, pero no pueden hacer el mismo uso de ellos. Debido a estos diagramas de casos de uso resulta mucho más sencillo de entender el desarrollo de nuestro proyecto ya que da una visión exacta de que es lo que le corresponde hacer a cada uno de nuestras clases y sus objetos.

La importancia de estos diagramas es que se puede dar un buen modelado al cliente para confirmar si es lo que desea que su software realice, también puede servir como retroalimentación y al momento de que exista un error poder localizarlo y tomar otra solución, o también saber si lo que conforma nuestro diagrama es en verdad necesario o totalmente inusual.

Así como podemos unir los casos también podemos eliminarlos ya que en ocasiones algunos de ellos no tienen ninguna función y también resultan ser innecesarios para nuestro proyecto y de ser así podría hacer el proyecto algo complejo y más extenso sin necesidad alguna.

Gracias a estos diagramas me resulto muchísimo más fácil entender que es lo que estaba haciendo en cuanto a mi proyecto, considero que son muy útiles y prácticos para el desarrollo de software.

Hora 8

El diagrama de estados se usa para organizar el comportamiento de las clases y objetos, representa la secuencia de estados que atraviesa un objeto de la clase a lo largo de su vida. Tiene un inicio y un final y cada evento representa algo que hace que el objeto pueda cambiar.

El rectángulo con vértices redondos representa el estado, el círculo relleno representa el inicio, las flechas son líneas de transición que representan el movimiento que ocurre de un estado a otro, es decir, el evento que origina la transición, y una diana representa el punto final.

Así como en las clases se puede dividir el nombre, atributos y acciones, en estos diagramas de estados también es posible se puede dividir en nombre, variables de estado y actividades. Las variables de estado son el tiempo y las actividades son las acciones y sucesos y utiliza entrada, salida y hacer.

En los sucesos y detalles se pueden agregar detalles y debido a esto hacer que suceda una modificación del estado. A ello están ligadas las condiciones de seguridad ya que dependiendo en lo que se está trabajando y cual estado se modificara se pueden tomar decisiones para poder mejorar esto y a su vez obtener un mejor resultado de ello.

Hora 9

Un diagrama de secuencia representa el intercambio de mensajes entre objetos, es el comportamiento dinámico de información, es utilizado para definir como se realiza un caso de uso por lo que está ligado al diagrama de casos de uso. Un objeto es un componente del software que tiene una función específica dentro de él. Un objeto se representa mediante un rectángulo y dentro contiene el nombre, de él sale una línea vertical hacia abajo, esta línea se conoce como línea de vida, es decir, el tiempo que un objeto está presente.

También contiene una activación que representa la ejecución de una operación que realiza el objeto, se representa mediante un rectángulo superpuesto en la línea de vida.

Un mensaje representa el paso de un mensaje entre dos objetos o de un objeto en sí mismo, se representa utilizando flechas, la flecha con punta de dos líneas representa la simple, la flecha con punta rellena representa la sincrónico y la flecha con una línea en la punta representa el asincrónico.

El diagrama de secuencia tiene dos dimensiones la horizontal y la vertical, la primera representa la disposición de los objetos y la segunda representa el paso del tiempo.

En la recursividad una operación se invoca varias veces a sí misma, si se tiene un problema ese se puede dividir en problemas más pequeños para solucionarlos por partes y así poder dar una solución incluyéndolos a todos.

Hora 10

Los diagramas de colaboraciones también muestran la forma en que los objetos colaboran entre sí así como los mensajes que pasan entre ellos, estos destacan el contexto y organización de los objetos, este diagrama se basa de acuerdo al espacio y por lo general evita la multiplicidad.

Con estos diagramas se puede sincronizar el envío de mensajes de unos objetos a otros para poder tener un orden y respetar los tiempos de envíos. Un objeto activo envía mensajes a los objetos pasivos para poder tener interacción con los objetos activos.

Cuando un objeto envía un mensaje a un conjunto de objetos de la misma clase es representado a través de varios rectángulos para el conjunto de objetos y para saber que el mensaje será enviado a todos los objetos se utiliza una condición entre corchetes y un asterisco.

Con estos diagramas sabremos identificar el espacio que es utilizado también se podrán enviar mensajes de un objeto a diversos objetos esto para la sincronización y organización del envío de mensajes.

Hora 11

El diagrama de actividades resalta las actividades, muestra el flujo de control o el flujo de objetos, son utilizados para representar la forma en que un sistema realiza su implementación. Estos tienen una secuencia de acciones que van desde un punto inicial hasta un punto final.

La representación de las actividades es mediante rectángulos con esquinas redondas, angosto y ovalado, una flecha representa la transición de una actividad a otra, el inicio se representa por un círculo relleno y el final por una diana.

En el diagrama de actividades durante su recorrido se tendrá que llegar a una decisión, hay dos formas de tomar la decisión correcta, la primera es ir directamente hacia la decisión y la segunda es poner una condicional para que se decida qué es lo que se va a hacer se representa mediante un rombo pequeño en color blanco. La segunda opción es la que se utiliza en los diagramas de flujo y considero que sería más utilizable.

Dicho diagrama puede establecer las reglas de la secuencia a seguir, así como elegir el orden en el que se realizaran las cosas. Los marcos de responsabilidad muestran quien tiene responsabilidades dentro de un proceso.

Hora 12

Un componente es una parte física que conforma a un sistema, los componentes son parte de una interfaz, cuando un objeto tiene que mostrarse lo hace a través de una interfaz solo haciendo la acción y no mostrando lo que hay detrás de él. A la relación entre una clase y su interfaz se le conoce como realización, un componente puede hacer uso de los servicios de otro componente.

El reutilizar un componente puede ayudar mucho al desarrollo del software y ahorraría trabajo a los mismos desarrolladores, ya que gracias a la reutilización se evitan muchas líneas de código lo que hace un buen ahorro de trabajo. Existen tres tipos de componentes el de distribución, para trabajar en el producto y de ejecución. El símbolo de representación de un diagrama de componentes es un rectángulo que tiene dos sobrepuestos en el lado izquierdo, los nombres deben ir dentro del diagrama.

Estos diagramas de componentes pueden ser implementados para la realización de páginas web, por ejemplo, un applet que funciona en una página web, así como también en el desarrollo de software. Hay dos maneras de representar las interfaces, en la primera es representada mediante un rectángulo y en la segunda es representada mediante un círculo vacío.

Hora 13

El diagrama de distribución es donde representamos el hardware donde esta nuestro software y a cada componente se le conoce como nodo. Hay dos tipos de nodos el que ejecuta y procesa y el que no lo hace, los que no lo hacen pueden ser los dispositivos de salida como las bocinas.

Para la representación del nodo se utiliza un cubo y dentro de él se escriben sus elementos y el nombre que se escribe de manera de cadena de texto. Si tenemos información extra y necesitamos ponerla en el diagrama de distribución se coloca en seguida dividiendo el cubo con líneas para poder diferenciar, por ejemplo, si es que pertenece a un paquete o es un elemento.

Puede existir conexión entre nodos y se representa a través de una línea, la conexión puede ser de tipo satelital, infrarrojo, no es necesariamente un cable o alambre.

Estos diagramas nos ayudan a saber cómo es que están conectados nuestros sistemas o como deberías de ser instalados en el lugar que desee nuestro cliente.