

INSTITUTO TECNOLÓGICO DE PABELLON DE ARTEAGA

ING. Tecnologías De La Información Y Las
Comunicaciones.

UNIDAD III

ALUMNO:

Karla Leticia Romo Castorena

DOCENTE:

Eduardo Flores

SEMESTRE:

9°

FECHA: 14 de septiembre de 2018

Contenido

SECCIÓN 2	4
Resumen 1	4
Introducción al UML	4
Resumen 2	4
Orientación a objetos	4
Resumen 3	4
Uso de la orientación a objetos	4
Resumen 4	5
Uso de relaciones	5
Resumen 5	5
Agregación, composición, interfaces y realización.	5
Resumen 6	5
Introducción a los casos de uso	5
Resumen 7	6
Diagramas de casos de usos.	6
Resumen 8	6
Diagramas de estados	6
Resumen 9	6
Diagramas de secuencias	6
Resumen 10	7
Diagramas de colaboraciones	7
Resumen 11	7
Diagramas de actividades	7
Resumen 12	8
Diagramas de componentes	8
Resumen 13	8
Diagramas de Distribución	8
Resumen 14	9
Nociones de los fundamentos del UML	9
Resumen 15	9

Adaptación de UML en un proceso de desarrollo.	9
Resumen 16	9
Presentación del caso por estudiar.	9
Resumen 17	10
Elaboración de un análisis de domino.	10
Resumen 18	10
Recopilaciones de las necesidades del sistema.	10
Resumen 19	10
Desarrollo de los casos de uso.	10
Resumen 20	11
Orientaciones a las interacciones y cambios de estado.	11
Resumen 21	11
Diseño del aspecto sensación y distribución.	11
Resumen 22	11
Noción de los patrones de diseño.	11
Resumen 23	12
Modelado de sistemas incrustados	12
Resumen 24	12
El futuro del UML	12

SECCIÓN 2

Resumen 1

Introducción al UML

El desarrollo de los sistemas es una actividad humana, un sistema de notación fácil de comprender, el procesos de desarrollo tiene una gran cantidad de errores permite a los creadores del sistema generar diseños que capturen sus ideas en forma convencional y fácil de comprender para comunicarlás a otras personas. Los diagramas UML están compuestos por diversos elementos gráficos que se combinan para conformar diagramas, cuenta con reglas para para combinar tales elementos para hacer el análisis del sistema y así generar anteproyectos de varias facetas que sean comprensibles para los clientes, desarrolladores y todos aquellos que estén involucrados en el proceso. El UML indica que es lo que supuestamente ara el sistema, mas no como lo hará.

Resumen 2

Orientación a objetos.

La orientación a objetos en un paradigma que depende de algunos principios fundamentales, la orientación a objetos ha tomado por asalto y en forma legítima al mundo de software. Como medio para la generación de programas y fomentar una metodología basada en componentes para el desarrollo y ver qué es lo que hace funcionar a los objetos y como utilizarlos en análisis y diseño. El mundo de la orientación a objetos tiene el propósito de la categorización en plantillas para fabricar objetos.

La herencia es un aspecto importante de la orientación a objetos ya que este hereda los atributos y las operaciones de su clase.

El polimorfismo es especificar una acción que pueda tener el mismo nombre en diferentes clases, cada clase ejecutara la operación de forma distinta.

Resumen 3

Uso de la orientación a objetos.

Por encima y debajo de los atributos, operaciones, responsabilidades y restricciones puede agregar mayor información a una clase. Los clientes prestan atención a los sustantivos que se utilizan para describir las entidades de su negocio ya que dichos sustantivos se convertirán en las clases de su modelo. Se puede utilizar un estereotipo para organizar las listas de atributos y operaciones y además abreviar una clase al mostrar solo un subconjunto de atributos y operaciones, esto hace un diagrama de clases menos complejo.

Resumen 4

Uso de relaciones.

Cuando las clases se conectan entre sí de forma conceptual se le conoce como una asociación, el modelo inicial es una línea que conecta ambas clases con el nombre de la asociación, cuando una clase se asocia con otra cada una de ellas juega un papel dentro. Puede representarse estos papeles en el diagrama escribiéndolos cerca de la línea que se encuentra junto a la clase que juega el papel correspondiente.

La asociación puede funcionar en direcciones inversas, para mostrar varias asociaciones en un mismo diagrama y multiplicidades en cada uno de los extremos, la clase heredada es secundaria de la clase principal su modelo inicial tiene atributos y operaciones en común.

Resumen 5

Agregación, composición, interfaces y realización.

La meta final es crear una idea estática de un sistema, con todas las conexiones entre las clases que lo conforman, en muchas ocasiones las clases constan de otras clases, este es un tipo especial de relación conocida como agregación o acumulación. Los componentes y la clase que contribuyen son una asociación que conforma un todo. Se puede representar como una jerarquía dentro de la clase completa. Una línea conectará el todo con un componente mediante un rombo sin relleno que se colocará en la línea más cercana al todo.

Un diagrama de contexto enfoca la atención en una clase específica dentro de un sistema. Un diagrama de contexto es un mapa detallado de un mapa mayor. Muestra un diagrama de clases anidado dentro de un gran símbolo rectangular de clase.

Una realización es una asociación entre una clase y una interfaz, una colección de operaciones que cierta cantidad de clases podría utilizar. Una interfaz se representa como una clase sin atributos. La realización se representa en el UML mediante una línea discontinua con una punta de flecha en forma de triángulo sin relleno que conecta a la clase con la interfaz.

Resumen 6

Introducción a los casos de uso

Los diagramas que establecen una idea dinámica mostrarán la forma en que el sistema y sus clases cambian con el tiempo. Las ideas estáticas ayudan a que un analista se comunique con el cliente. La idea dinámica ayudará a comunicarse con un grupo de desarrolladores y ayudará a estos últimos a crear programas.

Es una colección de escenarios iniciados por una entidad llamada actor un caso de uso debería dar por resultado algo de valor ya sea para el actor que lo inicie o para el otro. Cuando se deriva un caso de uso es importante destacar las condiciones para iniciar el caso de uso y los resultados obtenidos como consecuencia del mismo esto dará fundamentos a la terminología que se utilizará para hablar con los usuarios.

Resumen 7

Diagramas de casos de usos.

El caso de uso es un poderoso concepto que ayudara al analista a comprender la forma en que un sistema deberá comportarse. Le ayuda a obtener los requerimientos desde el punto de vista del usuario. Es necesario aprender a visualizar los conceptos del caso de uso.

Esta visualización permite mostrar los casos de uso a los usuarios para que a ellos se les pueda dar mayor información. A su vez, una representación visual le ayuda a combinar los diagramas de caso de uso con otro tipo de diagramas. Una de las finalidades del proceso de análisis de un sistema es generar una colección de casos de uso. Los beneficios del caso de uso es que le muestran los confines entre el mundo exterior. Generalmente los actores están fuera del sistema mientras que los casos de uso están dentro de él. Los diagramas de caso figuran con fuerza en el proceso de análisis. Se empieza por entrevistas con los clientes para obtener diagramas de clases. Estos proporcionan una base para entrevistar a los usuarios, para crear los modelos de caso de uso y profundice en cada caso de uso de alto nivel. Los diagramas resultantes de caso de uso darán los fundamentos para el diseño y el desarrollo.

Resumen 8

Diagramas de estados

Una manera para caracterizar un cambio en un sistema es decir que los objetos que lo componen modificaron su estado como respuesta a los sucesos y al tiempo. El diagrama de estado captura este tipo de cambios, presenta los estados en los que puede encontrarse un objeto con las transacciones entre los estados y muestra los puntos inicial y final de una secuencia de cambios de estados. Una transacción también puede ocurrir por la actividad en un estado: una transacción ocurre de forma que se conoce transacción no desencadenada. Finalmente un transacción ocurre cuando se cumple una condición particular o condición de seguridad.

Los subestados pueden ser secuenciales o concurrentes un estado se consta de subestados se conoce como estado compuesto.

Resumen 9

Diagramas de secuencias

El diagrama de secuencia establece el siguiente paso y muestra la forma en que los objetivos se comunican entre sí al transcurrir el tiempo, permite expandir su campo de visión y le muestra la forma en que un objeto interacciona con otro. En este campo de visión expandió incluirá una importante dimensión: el tiempo. La idea primordial es que las interacciones entre los objetos se realizan en una secuencia establecida y que la secuencia se toma su tiempo en ir del principio al final. Al momento de crear un sistema tendrá que especificar las secuencias y para ello utilizara al diagrama correspondiente.

El diagrama de secuencia consta de objetos que se representan del modo usual: rectángulos con nombre, mensajes representados por líneas continuas con una punta de flecha y el tiempo representado como una progresión vertical.

Un diagrama de secuencia puede mostrar ya sea una instancia de un caso de uso o puede ser genérico e incorporar todos los escenarios de un caso de uso. Los diagramas de secuencia genéricos con frecuencia dan la oportunidad de representar instrucciones condicionales. En ciertos 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 pequeño rectángulo sobrepuesto a la actividad.

Resumen 10

Diagramas de colaboraciones

Los diagramas de colaboración muestran la forma que los objetos colaboran entre sí como sucede con un diagrama de secuencias. Muestran los objetos junto con los mensajes que se envían entre ellos. Ambos tipos de diagramas son similares. De hecho son semánticamente equivalentes, esto significa que representan la misma información y podrá convertir un diagrama de secuencias en un diagrama de colaboración equivalente y viceversa.

Un diagrama de objetos muestra los objetos como tales y sus relaciones entre sí. Un diagrama de colaboración es una extensión de uno de objetos. A demás de la realización entre los objetos. Los diagramas colaboración muestran mensajes que se envían los objetos entre sí.

Por lo general la multiplicidad dado que podría ser una fuerte confusión. Aprovechamos la equivalencia de ambos tipos de diagramas. Para desarrollar los conceptos de los diagramas de colaboración.

Las condiciones se presentan como antes, mediante la colocación de la instrucción condicional entre corchetes. El esquema de numeración de las etiquetas representa esto de forma muy similar a los manuales técnicos que muestran encabezados.

Resumen 11

Diagramas de actividades

Un diagrama de actividades del UML es muy parecido a los viejos diagramas de flujo. Les muestra los pasos así como puntos de decisión y bifurcaciones. Es útil para mostrar lo que ocurre en un proceso de negocio u operaciones. Para empezar un diagrama de actividades ha sido diseñado para mostrar una visión simplificada de lo que ocurre durante una operación o proceso. Es una extensión de diagrama de estados.

Cunado una ruta se divide en dos o más, tal dispersión se presenta con una línea gruesa perpendicular a las rutas, misma que se reúne en una línea similar. Dentro de un diagrama de secuencias puede mostrar una señal, cuya transmisión se representa con un pentágono convexo, y la recepción con un cóncavo. Un diagrama de actividades se puede representar a las actividades de acuerdo con la responsabilidad asignada.

Es posibles combinar al diagrama de actividades con símbolos de otros diagramas con lo que se producirán diagramas híbridos.

Resumen 12

Diagramas de componentes

Un componente de software es una parte física de un sistema y se encuentra en la computadora, no en la mente del analista. Cuando trates con los componentes tendrá que tratar con sus interfaces, un objeto oculta al mundo exterior lo que hace, el objeto tiene que presentar un “rostro” al mundo exterior lo que hace, este conjunto de operaciones de clase en clase. La interfaz es la construcción UML que le permite hacer esto. Una interfaz es un conjunto de operaciones que especifica algo respecto al comportamiento de una clase. La interfaz es un conjunto de operaciones que presenta una clase a otra. En si una interfaz puede ser física o conceptual. La interfaz que utiliza una clase es la misma que la utiliza su implementación de software.

Solo se pueden ejecutar las operaciones de un componente a través de su interfaz. De la misma manera que en el caso de una clase y su interfaz, la relación entre un componente y su interfaz se conoce como realización.

Un componente puede hacer disponible su interfaz para que otros componentes puedan utilizar las operaciones que contienen. Es decir un componente puede acceder a los servicios de otro componente. El componente que proporciona los servicios se dice que provee una interfaz de exportación al que accede a los servicios se dice que utiliza una interfaz de importación.

La representación de un componente es un rectángulo con otros dos rectángulos pequeños sobrepuestos en su lado izquierdo. Puede representar una interfaz de dos formas la primera es un rectángulo que contiene información de la interfaz y se conecta con el componente mediante una línea discontinua con una punta de flecha representada por triangulo sin relleno. La otra es un pequeño círculo conectado al componente con una línea continua. Ambos tipos de conexión pretenden mostrar una relación una relación de realización.

Resumen 13

Diagramas de Distribución

El elemento primordial del hardware es un nodo, que es un nombre genérico para todo tipo de recursos de cómputo. Es posible usar 2 tipos de nodos: un procesador el cual puede ejecutar un componente y un dispositivo que no lo ejecuta. En el UML un cubo representa un nodo. Deberá asignar un nombre para el nodo, y podrá utilizar un estereotipo para indicar el tipo de recursos que sea.

Otra forma de indicar los componentes distribuidos es la de mostrarlos en relaciones de dependencia con un nodo. Una línea que asocie a dos cubos representara una conexión entre ellos. Podrá utilizar un estereotipo para dar información respecto a la conexión. Aunque la conexión es el tipo común de asociaciones entre dos nodos. Los diagramas de distribución son útiles para modelar redes. Los modelos presentados en esta hora incluyeron a redes token-ring, thin Ethernet y la red inalámbrica.

Resumen 14

Nociones de los fundamentos del UML

Su panorama del UML le muestra las categorías de los diagramas y a estos en cada categoría. Debido a que hay varias personas a las que les interesa el sistema por distintas razones, debería tener la capacidad de comunicar una visión consistente del sistema en diversas formas. El UML cuenta con arquitectura de cuatro capas: objetos del usuario, modelado, metamodelado y metametamodelado cuando analicé un sistema. Tales capas se distinguen por la generalidad de los elementos que ellas residen. La cuarta capa está orientada a los teóricos y diseñadores del lenguaje, en lugar de los usuarios del lenguaje y analistas del sistema.

El UML incluye varias extensiones propias. Cada uno de estos estereotipos, restricciones y valores etiquetados se orientan a ser usados por uno o dos de los símbolos UML.

Resumen 15

Adaptación de UML en un proceso de desarrollo.

Ahora que ha comprendido los diagramas de UML es hora de ponerlo a funcionar. El UML tiene la intención de impulsar el desarrollo de software, por ello es importante ofrecer los procesos y metodologías de desarrollo como un vehículo para comprender el uso del UML en un contexto.

Este tipo de metodologías secuencial podría fragmentar el desarrollo, de modo que un equipo de desarrollo podría aprovechar la mejor asimilación que se obtiene durante la vida de un proyecto.

Un patrón para el proceso de desarrollo GRAPPLE consta de 5 segmentos: recopilación, análisis, diseño, desarrollo y distribución. Cada segmento consta de diversas acciones y cada una de ellas da por resultados un proceso del trabajo, los diagramas UML constituyen productos del trabajo para varias de las acciones.

Resumen 16

Presentación del caso por estudiar.

El análisis del proceso del negocio empieza con la entrevista de un analista a un restaurantero, durante la entrevista alguien tomara nota en una computadora portátil. Al mismo tiempo, un modelador plasmara en una pizarra blanca un diagrama de actividades que el analista, quien está tomando nota y el restaurantero podrá ver.

El contenido de esta hora se ha orientado al dialogo en una entrevista y la forma en que podría fluir, se ha intercalado notas que le servirán de guía para saber cómo realizar la entrevista. El objetivo fue el de mostrar cómo convertir los resultados de la entrevista en un modelo UML.

Resumen 17

Elaboración de un análisis de dominio.

El primer par de acciones en GRAPPLE se orienta al dominio y no al sistema, ni la hora anterior ni esta se orientara al sistema propuesto. Ciertamente, en lo que hemos visto hasta ahora no se ha propuesto sistema alguno. Solo tenemos una asignación vaga de La Hudra, Nar y GONIFF} para valernos de la tecnología y mejorar el acto de comer en restaurantes. Al intentar descubrir los procesos del negocio, hemos empezado a alimentar el conocimiento del equipo de desarrollo en nuestro escenario.

El equipo tendrá entrevistas con los expertos restauranteros, pero antes de ellos trabajaran con el contexto de la entrevista del proceso del negocio, el objetivo es producir un diagrama de clase inicial. El modelador de los objetos agregara sustancias a este diagrama mediante la derivación de clases abstractas, asociaciones y multiplicidades. La derivación de agregados u objetos compuestos le ayudara a organizar al modelo, será necesario otra entrevista y sesiones para dar cuerpo completamente en el modelo, aunque es posible empezar a agregar atributos y operaciones en este punto.

Resumen 18

Recopilaciones de las necesidades del sistema.

Ahora trabajaremos en la parte modular, es decir en la parte técnica para el restaurante del futuro. Ya cuenta con los procesos del negocio y los diagramas de clases. Ya pueden empezar a codificar, antes que nada, tienen que desarrollar una visión del sistema. Debemos analizar nuestros diagramas de procesos del negocio, pienso que podemos ver varios lugares donde podría ser útil la tecnología basada en computadoras. Los miembros del equipo decidieron que la agilización en la transmisión de la información es la clave para el éxito del sistema y trae a colación tipos de tecnologías para lograrlo. El resultado es un diagrama de paquetes en donde cada paquete representa una sección principal de funcionalidad. Los casos de uso dentro de un paquete se basan en tal funcionalidad.

Resumen 19

Desarrollo de los casos de uso.

Los casos de uso del diagrama de paquetes nos dio una buena idea de los sistemas, el equipo tendrá que llevar a cabo cada uno de ellos, poco a poco han pasado de comprender el dominio a comprender el sistema. Los casos de uso establecieron el puente. Si crees que el desarrollo del sistema se orienta a los casos de uso, ya habrá comprendido todo el proceso.

Un análisis de casos de uso involucran la especificación de una descripción del propio caso de uso, derivar las condiciones previas y resultantes, y especificar los pasos. Un aspecto importante del análisis de los casos de uso es que los componentes del sistema empiezan a ser evidentes.

Resumen 20

Orientaciones a las interacciones y cambios de estado.

Analizar los casos de uso ha ayudado a visualizar las partes funcionales del sistema. Aunque ahora sabemos muchos de ellos, aún tenemos que modelar la forma en que las partes funcionales se comunican entre sí la forma en que cambiara de estado. El objetivo de esto es dar información a los programadores que les facilite su trabajo. Los resultados de este análisis deberán facilitar a los programadores la codificación de los objetos del sistema y la forma en que se comunica entre sí.

Luego de modelar la cooperación entre componentes, el sistema ya está cercano a convertirse en realidad. Conforme modele la cooperación entre componentes podría encontrarse con que será adecuado modificar los casos de uso.

Resumen 21

Diseño del aspecto sensación y distribución.

Se ha cumplido con éxito la gran parte del análisis del sistema conducido por casos de uso. Ahora veremos dos aspectos importantes del diseño del sistema. Ambos pueden, al fin de cuentas orientarse a casos de uso y son muy importantes para el producto final. Las interfaces gráficas del usuario determinan que tan práctico es un sistema. La distribución convierte a la arquitectura física planeada del sistema en una realidad.

El diseño de la interfaz del usuario depende de una visión artística y de una investigación científica, varios de los principios del diseño de la interfaz han salido a la luz. Los casos de uso conducen el diseño a la interfaz del usuario. El sistema tiene que permitir al usuario completar cada caso de uso, y la interfaz es la puerta de acceso hacia cada uno de ellos.

El ingeniero de sistemas del equipo se orientará a la arquitectura física, la arquitectura está conducida por los casos de uso dado que el uso del sistema finalmente determina la naturaleza física y la disposición del mismo.

Resumen 22

Noción de los patrones de diseño.

Esta aplicación, la representación de patrones de diseño, capturan la esencia de las soluciones que han funcionado una y otra vez en proyectos y situaciones reales. El propio UML le permite ir un paso más allá. Le da un mecanismo para crear una clase de una forma similar a la creación de objetos. Podrá establecer una clase de manera que cuando asigne valores a un subconjunto de atributos habrá creado una clase en lugar de un objeto, a este tipo de clase se le conoce como clase parametrizada.

Un patrón de diseño es básicamente una solución que surge de la experiencia práctica con varios proyectos, y los equipos de desarrollo han encontrado que se pueden aplicar en diversos contextos. Cada patrón de diseño describe un conjunto de objetos y clases comunicadas. Podemos crear un patrón de diseño para la calculadora que determine el enésimo valor de una serie aritmética

Los patrones de diseño ofrecen varias ventajas. Permiten que los diseñadores vuelvan a utilizar con facilidad las soluciones ya probadas, incorporar componentes sólidos en los diseños y documentar claramente los sistemas que genere.

Resumen 23

Modelado de sistemas incrustados

Los sistemas de cómputo que no se basan en sistemas de escritorio, laptops o en palmtops. En vez de ello estarán incrustados dentro de aparatos como aviones, trenes y automóviles. Los sistemas incrustados no tienen teclados y monitores que interactúen con nosotros en vez de ello, cada uno es un chip que se encuentra dentro de un dispositivo y ese dispositivo no se parece nada a una computadora. El sistema incrustado decidirá qué debe hacer el dispositivo.

Si utiliza un sistema de este tipo, no tendrá la sensación de trabajar con una computadora. En vez de ello estará interactuando con un dispositivo. Un sistema operativo en tiempo real dirige entre subprocesadores e interrupciones. El núcleo es la parte que administra el tiempo que ocupa la CPU en subprocesos individuales. El programador de tiempo del núcleo determina cuál tarea se ejecutará a continuación. Estos conceptos de aplicaciones mediante modelos de un inteligente dispositivos ejercitado que varía su resistencia.

Resumen 24

El futuro del UML

Como lo indicamos e entonces el UML mediante la adición de estereotipos que permitan hacer su propio modelo de su dominio. También podría agregar estereotipos gráficos que aclaren la información trasladada a su modelo. Los diagramas de distribución dan buenos ejemplos de ello porque las figuras con frecuencias sustituyen a los iconos de UML.

En un sistema experto, las reglas de condición son el bloqueo de construcción de una base de datos de conocimientos, el componente que contiene el conocimiento de un experto en algún dominio humano. Sugerimos un diagrama que visualice las reglas y sus relaciones internas. Este diagrama un cuadro dividido en compartimientos modela a la regla. Los vínculos adyacentes aparecen como líneas de conexión entre las partes adecuadas de las reglas.