

Instituto Tecnológico de Pabellón de Arteaga  
Tecnologías De La Información Y Las  
Comunicaciones.

Prof. Eduardo Flores Gallegos

UNIDAD III

Tarea I

Denise Kirstie Martinez Santana

9no Semestre

19 de septiembre de 2018

## Contenido

<b>Resumen 13</b> .....	3
Diagramas de Distribución .....	3
<b>Resumen 14</b> .....	3
Nociones de los fundamentos del UML .....	3
<b>Resumen 15</b> .....	3
Adaptación de UML en un proceso de desarrollo. ....	3
<b>Resumen 16</b> .....	4
Presentación del caso por estudiar. ....	4
<b>Resumen 17</b> .....	4
Elaboración de un análisis de domino. ....	4
<b>Resumen 18</b> .....	5
Recopilaciones de las necesidades del sistema. ....	5
<b>Resumen 19</b> .....	5
Desarrollo de los casos de uso. ....	5
<b>Resumen 20</b> .....	6
Orientaciones a las interacciones y cambios de estado. ....	6
<b>Resumen 21</b> .....	6
Diseño del aspecto sensación y distribución. ....	6
<b>Resumen 22</b> .....	7
Noción de los patrones de diseño. ....	7
<b>Resumen 23</b> .....	7
Modelado de sistemas incrustados .....	7
<b>Resumen 24</b> .....	7

## 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 metamodelado 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 esta 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 domino.

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 a 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 trabajarán con el contexto de la entrevista del proceso del negocio, el objetivo es producir un diagrama de clase inicial. El modelador de los objetos agregará sustancias a este diagrama mediante la derivación de clases abstractas, asociaciones y multiplicidades. La derivación de agregados u objetos compuestos le ayudará 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 comunicaran 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 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 palm tops. 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 subprocesos 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 cual tarea se ejecutará a continuación. Estos conceptos de aplicaciones mediante modelos de dispositivos inteligentes ejercitados que varían 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 don buenos ejemplos de eelo 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.