



**INSTITUTO TECNOLÓGICO  
de Pabellón de Arteaga**

**ITEC**

## *Tareas Cuarta Unidad*

*Materia:*

*Ingeniería de Software*

*Docente:*

*Eduardo Flores Gallegos*

*Alumno:*

*José Alejandro Chávez Rendón*

*Carrera:*

*Ingeniería en Tecnologías de la Información y Comunicaciones*

Índice:	No. Página
Portada general	1
Índice	2
Tarea 1. Reseña.	3
Tarea 2. Reseña.	6



**INSTITUTO TECNOLÓGICO  
de Pabellón de Arteaga**

**ITEC**

*Tarea 1. Reseña:*

*“24 horas de UML” Primera Parte*

*Materia:*

*Ingeniería de Software*

*Docente:*

*Eduardo Flores Gallegos*

*Alumno:*

*José Alejandro Chávez Rendón*

*Carrera:*

*Ingeniería en Tecnologías de la Información y Comunicaciones*

**26/05/2019**

## 24 Horas de UML. Primera parte

Ficha:

Schmuller, J. (2001). *Aprendiendo UML en 24 horas*. Prentice Hall, Pearson Educación. Recuperado el 20 de mayo de 2019, de: <https://drive.google.com/drive/folders/1dTUmJgifu70DBecrWkPgU7GTDikMKRB3?usp=sharing>

Un veterano de más de veinte años en Tecnología de la Información, Joseph Schmuller es un arquitecto técnico en Blue Cross-Blue Shield of Florida. Es autor de varios libros sobre computación, incluidas las tres ediciones de "Teach Yourself UML in 24 Hours" (SAMS), y ha escrito numerosos artículos sobre tecnología avanzada. Desde 1991 hasta 1997, fue Editor en Jefe de la revista "PC AI".

Él es un ex miembro de la Asociación Americana de Estadística, y ha enseñado estadística en niveles de pregrado y posgrado. Tiene un B.S. de Brooklyn College, un Máster en Administración de Empresas de la Universidad de Missouri-Kansas City, y un Ph.D. de la Universidad de Wisconsin. Vive con su familia en Jacksonville, Florida, donde es profesor adjunto en la Universidad del Norte de Florida.

En su libro “Aprendiendo UML en 24 horas”, Joseph Schmuller propone un objetivo, con solo 24 horas, aprender todo lo necesario relacionado con una herramienta muy útil e importante en el desarrollo de software como lo es el Lenguaje Unificado de Modelado (UML por sus siglas en inglés) una herramienta muy útil empleada aún hoy en día, de ahí la importancia de aprender de ella, para lograrlo eso, Joseph divide por horas el contenido del libro. En cada hora se abordan diferentes conceptos y características, diagramas e imágenes que sirven de ejemplo, partes importantes del UML y al final de cada una de las horas, un resumen de lo visto en el capítulo, una sección de preguntas y respuestas y una de sección de taller, conformada por un cuestionario y ejercicios para el lector.

El libro incluye una con una sección de reconocimientos, en la que Schmuller agradece el apoyo personal, de su familia y amigos y el apoyo laboral para la realización del libro, da crédito a las personas detrás de él y a todo el equipo que ha hecho posible que el lector tenga este recurso tan valioso a su alcance.

A lo largo del libro se emplean diversas herramientas para cumplir el objetivo, entre ellas se incluyen algunas digitales, como un programa en el que se puede ejecutar un diagrama UML, se incluyen capturas de cómo funciona dicho programa y también algunos modelos y diagramas. Por si fuera poco, va más allá con algunos diagramas y se aborda algo de historia del lenguaje.

Además de las horas, el libro está dividido en partes, en la primera de ellas titulada “Para iniciar”, se emplea para dar atrapar al lector, una introducción al lenguaje

UML y el porqué es necesario, se aborda también la orientación a objetos, su utilización, el uso de interfaces, casos de uso, los diagramas respectivos, las diferencias entre cada diagrama, su estructura y ejemplos de cómo emplearlos, además de un diagrama que sirva de ejemplo.

Todo ello siempre explicando que es cada diagrama, profundizando en el tema y fundamentando el texto con sucesos o hechos en la historia y con la sección para autoevaluarnos, todo esto en el lapso de una hora. El formato en el que se divide el libro nos sirve de ayuda a los lectores, pues nos permite administrar nuestro tiempo de forma más sencilla, así sabemos cuántas horas llevamos invertidas, podemos ver nuestro avance de forma más clara, prestar atención en las que nos falta por ver y volver de forma fácil a alguna en la que tengamos duda o donde queramos aclarar algo, además anexa materiales al final como las respuestas correctas a la sección de Taller, con la cual podemos ver si nuestras respuestas están acertadas o ver si es necesario dedicarle una hora extra a ese capítulo.

Luego de los agradecimientos tenemos la Parte 1. Donde vemos las horas necesarias para completarla y lo que se podrá ver en cada una de ellas. Esto empieza con lo que se dijo antes, una introducción al tema para que el lector entienda la importancia del lenguaje UML, explica lo que es y porque es una de las herramientas más importantes en el desarrollo de sistemas actualmente. Además nos explica que antes de la existencia del UML, el desarrollar sistemas era con mucha frecuencia algo al azar, se generaba una documentación que no era entendida por los programadores o que no era entendible para el cliente, lo que no permitía saber si el cliente estaba satisfecho con la propuesta dada y al final el

resultado podía ser muy complicado y dar más problemas al cliente en vez de solucionárselos, esto provocaba que los programadores no hicieran análisis profundos y hacer las cosas con algo de espíritu aventurero, algo inadecuado para los sistemas actuales y los negocios de alto riesgo.

Algo en lo que el UML ayuda, es reducir el tiempo de desarrollo necesario para crear el software o el sistema, así obtienes un software de calidad, pero invirtiendo menos tiempo y ese tiempo puede ser invertido en otros aspectos.

El texto continúa con creación del UML, que es el resultado del trabajo de Grady Booch, James Rumbaugh e Ivar Jacobson, quienes se dieron cuenta del problema antes mencionado, hacía falta una metodología que facilitara el trabajo, cada quien tenía su diseño propio pero al final lo integraron en uno solo. Continuamos con los diagramas, el texto nos enseña cuales son, en que se aplican y que se logra con todos ellos. Son una herramienta importante para el analista de sistemas y son de importancia para el cliente, facilitan las cosas para los desarrolladores y para cualquiera relacionado con los proyectos, pues cada diagrama se dirige a cada tipo de persona implicada en el proyecto

El texto establece algo importante, un modelo UML indica qué es lo que se supone que hará el sistema, más no indica como lo hará.

Hora 2:

Continuamos con la orientación a objetos, empieza definiendo algunos conceptos relacionados con dicho tema, nos pone en contexto y demuestra que es un tema que podemos identificar en todos lados. Vemos la herencia que es la agrupación

de varios objetos con las mismas acciones en la que vemos que un objeto puede tomar características de la clase a la que pertenece. El texto continúa con el polimorfismo, un aspecto importante pues especifica que la acción puede tener un nombre similar en diferentes objetos. Esta hora termina con el concepto de agregación que es la relación que tiene un objeto con otros para completar sus acciones.

### Hora 3

Aquí se aplica lo visto en la hora anterior, ahora relacionaremos las características del UML con los conceptos de orientación a objetos y definiremos algunos conceptos más. Vemos que una clase es representada como un rectángulo, dentro se coloca el nombre del objeto y abajo van los atributos que son sus características y las operaciones que realizará, esto nos permite tener un diagrama con menor complejidad. El texto nos dice que estas clases representan el vocabulario de un área del conocimiento, permiten entender al cliente y al darse a entender, podemos utilizar un diagrama de clases para animar al cliente a dar más detalles sobre lo que quiere

### Hora 4

En este punto se emplean las clases anteriores, y les añadimos ese “algo” faltante, una relación entre ellas que permite definir mejor el diagrama, pues hasta este punto, las clases parecen más una lista de elementos en lugar de la representación que queremos lograr. Tenemos muchas imágenes que sirven de guía para esto, relacionar nuestras clases y ahora sí, realizar el diagrama que nos



permita transmitir nuestra idea. La asociación es la conexión fundamental entre las clases, cada una juega un papel. Luego definimos la multiplicidad, que puede ser en muchos tipos y que explica cuantos objetos de una clase se relacionan con un objeto de una clase asociada. Vemos que esas asociaciones pueden contener atributos por sí mismas, luego en herencia vemos que una clase secundaria puede heredar los atributos y operaciones de otra clase principal, Por lo cual la clase principal tiene datos más genéricos que la secundaria. Lo último en esta hora es la dependencia, su uso más común es mostrar que la firma en la operación de una clase se utiliza a otra clase. Esta dependencia se representa como otra figura y en ella se reúne a las dos clases.

## Hora 5

Lo siguiente es la Agregación, composición, interfaces y realización, temas que se abordan en esta hora con conceptos nuevos y la continuación de las relaciones de clases. El texto nos dice que la agregación establece la asociación que conforma un todo, dicha clase se genera con el resto de componentes, en la composición cada componente puede pertenecer tan solo a un todo, está muy ligada con la agregación.

Lo siguiente que vemos es un diagrama de contexto que enfoca la atención en una clase específica dentro de un sistema y uno de composición es un mapa detallado de un mayor.

## Hora 6.

Lo siguiente en el libro son los casos de uso, un área principal del UML y en esta sección veremos que son, su importancia, como incluirlos y como se extienden. Con ellos se establecen diagramas dinámicos y se permite crear diagramas en los que el sistema implicado y sus clases van cambiando con el paso del tiempo. Nos permite mostrar como lucirá el sistema para nuestros usuarios potenciales. Después el texto nos dice que para aplicar el caso de uso es importante saber cuáles son los requerimientos y analizarlos, estos análisis nos permitirán un desarrollo adecuado de nuestro sistema.

## Hora 7

En esta sección el autor nos devuelve a los diagramas, en esta ocasión los de casos de uso, una herramienta poderosa que permite entender cómo debe comportarse el sistema, facilitan la comunicación entre analistas y usuarios y para ello, El autor nos dice que es importante analizar a los actores iniciales del programa. Lo siguiente es que se aborda es la estructura de un diagrama de este tipo, la representación de un caso de uso es una elipse en estos diagramas y los actores se representan como personas, el nombre de cada actor aparece debajo del mismo. De la elipse hay una línea que conecta con cada uno de los actores involucrados. La inclusión y la extensión también son parte de este tipo de diagramas y se obtienen gracias a los diagramas de clases, ahí su relación y la necesidad de irlos realizando todos.

## Hora 8

En la hora 8 continuamos con los diagramas de estados que se caracterizan por el cambio de sistema y hacer dichos cambios con el paso del tiempo, los objetos que lo componen modifican su estado y subestados que se pueden dividir en diferentes tipos, todo esto como respuesta a los sucesos y al tiempo. Cuenta con punto inicial y punto final. En la estructura de los diagramas, en los recuadros irán las acciones del objeto guiadas por líneas con dirección. También se nos dice que este tipo de diagramas integran nuevos tipos: de condiciones, de subestados secuenciales y estado histórico.

## Hora 9

En la hora 9 tenemos los diagrama de secuencias que constan de objetos que se representan del mundo usual; con rectángulos subrayados, mensajes representados por línea continua, con dirección y el tiempo representado como una progresión vertical. Con ellos se envían mensajes entro objetos los cuales pueden ser simples, síncronos y asíncronos. El libro nos dice que en estos diagramas el tiempo es representado verticalmente, se inicia de la parte superior y avanzan hacia la parte inferior

## Hora 10

Continuamos con los diagramas y pasamos a los de colaboración. En estos diagramas se muestra al objeto como tal y su relación con otros objetos, un diagrama de colaboraciones sirve como una extensión de un objeto. Se nos dice que en este tipo, para representar un mensaje se dibuja una flecha cercana a la

línea de asociación entre dos objetos, esta flecha apunta a uno de ellos y se termina el mensaje poniéndolo entre paréntesis

## Hora 11

En la hora 11 vemos los diagramas de actividad. Se nos muestra que el propósito de estos diagramas es mostrar una visión simplificada de lo que ocurre mientras se realiza alguna operación o proceso y en ellos se da realce a las actividades a realizar en el proceso de desarrollo del software. Nos da también su estructura, la cual se representa por rectángulos con esquinas redondeadas, si dentro de una actividad se lleva a cabo un procedimiento, se muestra una flecha de transición y al igual que un diagrama de actividad cuenta con punto de inicio y un fin. Una decisión se representa con un rombo y también emplea indicaciones, que sirven para que se ejecute una actividad.

Chávez Rendón José Alejandro