

Análisis y diseño orientado a objetos.

Unidad I: Síntesis de temas y conceptos.

1. Modelado orientado a objetos y la UML

El paradigma de la orientación a objetos concibe los sistemas como un conjunto de objetos. Cada objeto es una representación de alguna cosa o evento real.

Objetos

Los objetos son personas, lugares o cosas relevantes para el sistema a analizar. Los sistemas orientados a objetos describen las entidades como objetos. Algunos objetos comunes son clientes, artículos, pedidos, etcétera. Los objetos también pueden ser pantallas de GUI o áreas de texto en la pantalla.

Todo objeto tiene un nombre (se le puede identificar), un estado (generalmente hay algunos datos asociados a él) y un comportamiento (se le pueden hacer cosas a objeto y él puede hacer cosas a otros objetos).

Clase: Es una descripción de un conjunto de objetos similares. Una clase contiene los atributos y las operaciones sobre esos atributos que hacen que una clase tenga la entidad que se desea. Una clase define el conjunto de atributos compartidos y comportamientos que se encuentran en cada objeto de la clase.

Atributo: Es una característica concreta de una clase.

Método: Es una operación concreta de una determinada clase.

Instancia: Es una manifestación concreta de una clase (un objeto con valores concretos).

1.1 ¿Qué es UML?

El Lenguaje Unificado de Modelado es un lenguaje visual para modelado de sistemas de propósito general. Está asociado frecuentemente al modelado para software de sistemas con orientación a objetos. Los diagramas de UML son fácilmente entendidos por las personas y son reconocidos por las computadoras para la generación de software.

El UML NO nos da una clase de metodología de modelado, sino que provee una sintaxis que puede ser usada para construir modelos. El Proceso Unificado (UP) es una metodología que nos dice que empleados, actividades, y artefactos necesitamos para crear un modelo de software del sistema.

UML no está ligado a una metodología de desarrollo o un ciclo de vida, es capaz de ser utilizado por todas las metodologías existentes. UML y UP trabajando juntos unifican toda la experiencia de la ingeniería de software en la actualidad en cuanto a lenguajes de modelado visual.

2 Objetos y UML

La premisa básica de UML es que “nosotros podemos modelar software y otros sistemas como colecciones de objetos que interactúan”.

Existen dos aspectos a considerar en un modelo:

- Estructura estática: Describe que objetos son importantes en el modelado y como se relacionan.
- Comportamiento dinámico: describe el ciclo de vida de esos objetos y como interactúa con cada uno de los otros para enviar los requerimientos de funcionalidad al sistema.

2.1 Vista general de UML

El lenguaje UML se compone de tres elementos básicos, los bloques de construcción, las reglas y algunos mecanismos comunes. Estos elementos interaccionan entre sí para dar a UML el carácter de completitud y no-ambigüedad que antes comentábamos.

2.2 Bloques de construcción

Los bloques de construcción se dividen en tres partes: **Elementos o cosas**, que son las abstracciones de primer nivel, **Relaciones**, que unen a los elementos entre sí, y los **Diagramas**, que son agrupaciones interesantes de elementos.

2.2.1 Elementos o cosas

2.2.1.1 Estructurales

Los elementos estructurales en UML, es su mayoría, son las partes estáticas del modelo y representan cosas que son conceptuales o materiales. Son **sustantivos**, como:

- **Clase:** Una clase es una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica. Una clase implementa una o más interfaces.
- **Interfaz:** Una interfaz describe un conjunto de especificaciones de operaciones (o sea su signatura) pero nunca su implementación.
- **Colaboración:** Define una interacción y es una sociedad de roles y otros elementos que colaboran para proporcionar un comportamiento cooperativo mayor que la suma de los comportamientos de sus elementos. Las colaboraciones tienen una dimensión tanto estructural como de comportamiento. Una misma clase puede participar en diferentes colaboraciones.
- **caso de uso:** Un caso de uso es la descripción de un conjunto de acciones que un sistema ejecuta y que produce un determinado resultado que es de interés para un actor particular. Un caso de uso se utiliza para organizar los aspectos del comportamiento en un modelo. Un caso de uso es realizado por una colaboración.
- **clase activa:** Es una clase cuyos objetos tienen uno o más procesos o hilos de ejecución por lo y tanto pueden dar lugar a actividades de control. Una clase activa es igual que una clase, excepto que sus objetos representan elementos cuyo comportamiento es concurrente con otros elementos.

- **Componente:** Un componente es una parte física y reemplazable de un sistema que conforma con un conjunto de interfaces y proporciona la implementación de dicho conjunto. Un componente representa típicamente el empaquetamiento físico de diferentes elementos lógicos, como clases, interfaces y colaboraciones.
- **Nodo:** Un nodo es un elemento físico que existe en tiempo de ejecución y representa un recurso computacional que, por lo general, dispone de algo de memoria y, con frecuencia, de capacidad de procesamiento.

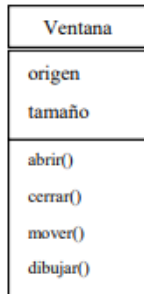


Figura 3. Clases

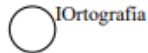


Figura 4. Interfaz

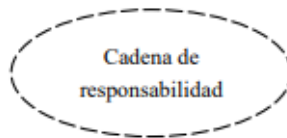


Figura 5. Colaboración



Figura 6. Casos de Uso

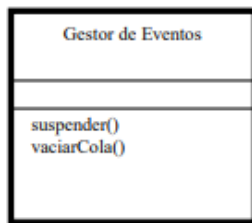


Figura 7. Clases Activas

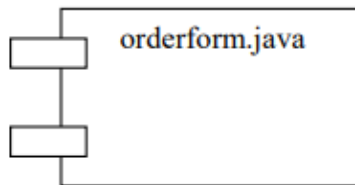


Figura 8. Componentes

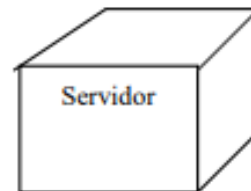


Figura 9. Nodos

2.2.1.2 Comportamiento

Los elementos de comportamiento son las partes dinámicas de un modelo. Se podría decir que son los verbos de un modelo y representan el comportamiento en el tiempo y en el espacio. Los principales elementos son los dos que siguen:

- **Interacción:** Es un comportamiento que comprende un conjunto de mensajes intercambiados entre un conjunto de objetos, dentro de un contexto particular para conseguir un propósito específico. Una interacción involucra otros muchos elementos, incluyendo mensajes, secuencias de acción (comportamiento invocado por un objeto) y enlaces (conexiones entre objetos).

- **Máquinas de estados:** Es un comportamiento que especifica las secuencias de estados por las que van pasando los objetos o las interacciones durante su vida en respuesta a eventos, junto con las respuestas a esos eventos. Una máquina de estados involucra otros elementos como son estados, transiciones (flujo de un estado a otro), eventos (que disparan una transición) y actividades (respuesta de una transición).

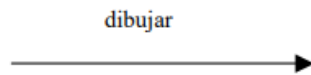


Figura 10. Mensajes

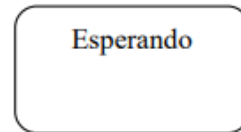


Figura 11. Estados

2.2.1.3 Agrupación

El principal elemento de agrupación es el paquete, que es un mecanismo de propósito general para organizar elementos en grupos. Los elementos estructurales, los elementos de comportamiento, incluso los propios elementos de agrupación se pueden incluir en un paquete.

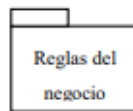


Figura 12. Paquetes

2.2.1.4 Anotación o documentación

Los elementos de anotación son las partes explicativas de los modelos UML. Son comentarios que se pueden aplicar para describir, clasificar y hacer observaciones sobre cualquier elemento de un modelo. El tipo principal de anotación es la nota que simplemente es un símbolo para mostrar restricciones y comentarios junto a un elemento o un conjunto de elementos.

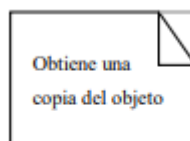


Figura 13. Notas

2.2.2 Relaciones

- **Dependencia:** Es una relación semántica entre dos elementos en la cual un cambio a un elemento (el elemento independiente) puede afectar a la semántica del otro elemento (elemento dependiente). Se representa como una línea discontinua, posiblemente dirigida, que a veces incluye una etiqueta.



- **Asociación:** Es una relación estructural que describe un conjunto de enlaces, los cuales son conexiones entre objetos. La asociación se representa con una línea continua, posiblemente dirigida, que a veces incluye una etiqueta. A menudo se incluyen otros adornos para indicar la multiplicidad y roles de los objetos involucrados.



- **Generalización:** Es una relación de especialización / generalización en la cual los objetos del elemento especializado (el hijo) pueden sustituir a los objetos del elemento general (el padre). De esta forma, el hijo comparte la estructura y el comportamiento del padre.



- **Realización:** Es una relación semántica entre clasificadores, donde un clasificador especifica un contrato que otro clasificador garantiza que cumplirá. Se pueden encontrar relaciones de realización en dos sitios: entre interfaces y las clases y componentes que las realizan, y entre los casos de uso y las colaboraciones que los realizan.



- **Agregación:** es un tipo especial de asociación y representa una relación estructural entre un todo y sus partes.



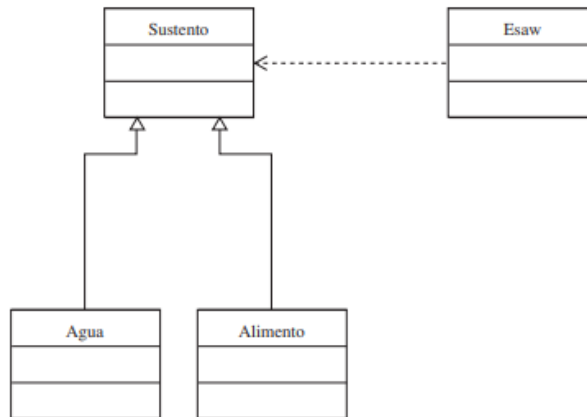
- **Composición:** Es una variación de la agregación simple que añade una semántica importante. La composición es una forma de agregación, con una fuerte relación de pertenencia y vidas coincidentes de la parte del todo.



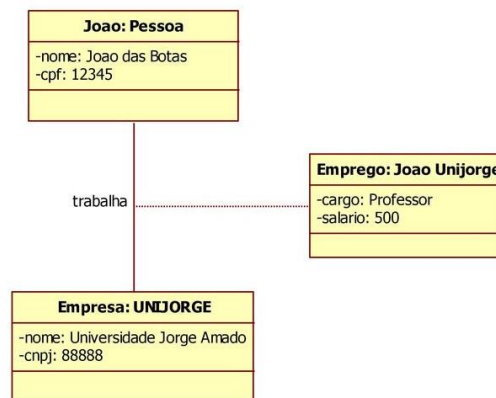
2.2.3 Diagramas

Los diagramas se utilizan para representar diferentes perspectivas de un sistema de forma que un diagrama es una proyección del mismo. UML proporciona un amplio conjunto de diagramas que normalmente se usan en pequeños subconjuntos para poder representar las cinco vistas principales de la arquitectura de un sistema.

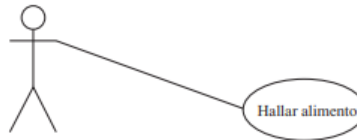
- **Diagramas de Clases:** Muestran un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Estos diagramas son los más comunes en el modelado de sistemas orientados a objetos y cubren la vista de diseño estática o la vista de procesos estática (sí incluyen clases activas).



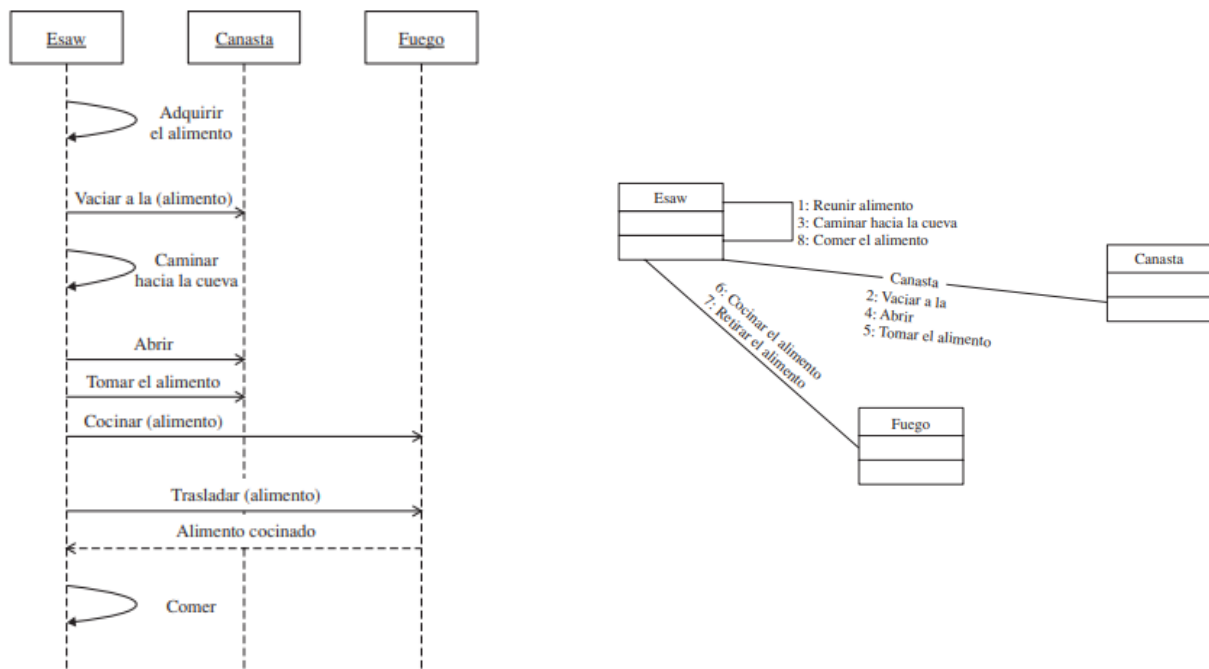
- **Diagramas de Objetos:** Muestran un conjunto de objetos y sus relaciones, son como fotos instantáneas de los diagramas de clases y cubren la vista de diseño estática o la vista de procesos estática desde la perspectiva de casos reales o prototípicos.



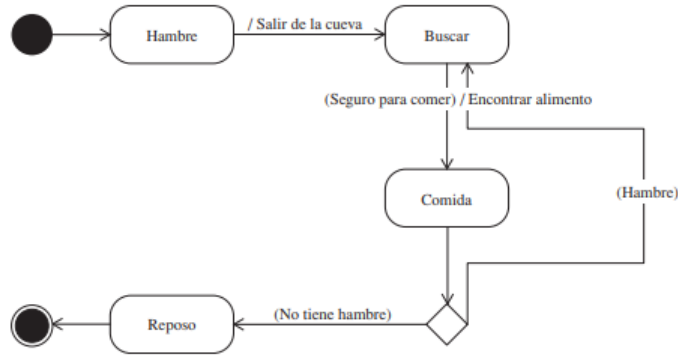
- **Diagramas de Casos de Usos:** Muestran un conjunto de casos de uso y actores (tipo especial de clases) y sus relaciones. Cubren la vista estática de los casos de uso y son especialmente importantes para el modelado y organización del comportamiento.



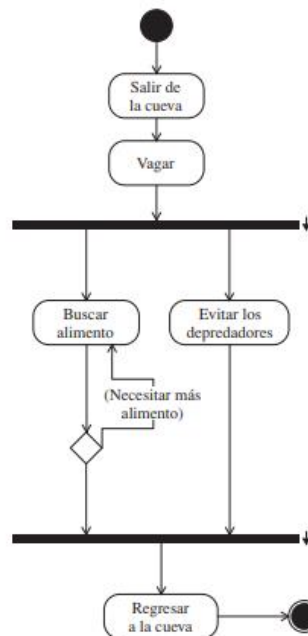
- **Diagramas de Secuencia y de Colaboración:** Tanto los diagramas de secuencia como los diagramas de colaboración son un tipo de diagramas de interacción. Constan de un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar unos objetos a otros. Cubren la vista dinámica del sistema. Los diagramas de secuencia enfatizan el ordenamiento temporal de los mensajes mientras que los diagramas de colaboración muestran la organización estructural de los objetos que envían y reciben mensajes. Los diagramas de secuencia se pueden convertir en diagramas de colaboración sin pérdida de información, lo mismo ocurren en sentido opuesto.



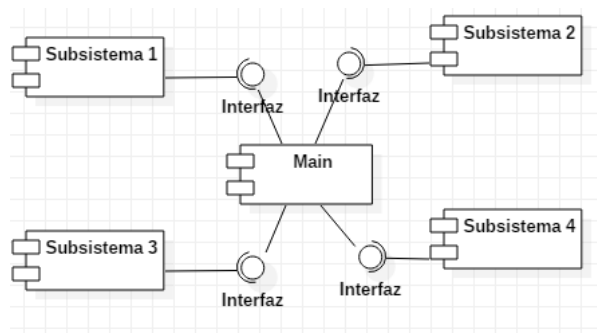
- **Diagramas de Estados:** Muestran una máquina de estados compuesta por estados, transiciones, eventos y actividades. Estos diagramas cubren la vista dinámica de un sistema y son muy importantes a la hora de modelar el comportamiento de una interfaz, clase o colaboración.



- **Diagramas de Actividades:** Son un tipo especial de diagramas de estados que se centra en mostrar el flujo de actividades dentro de un sistema. Los diagramas de actividades cubren la parte dinámica de un sistema y se utilizan para modelar el funcionamiento de un sistema resaltando el flujo de control entre objetos.

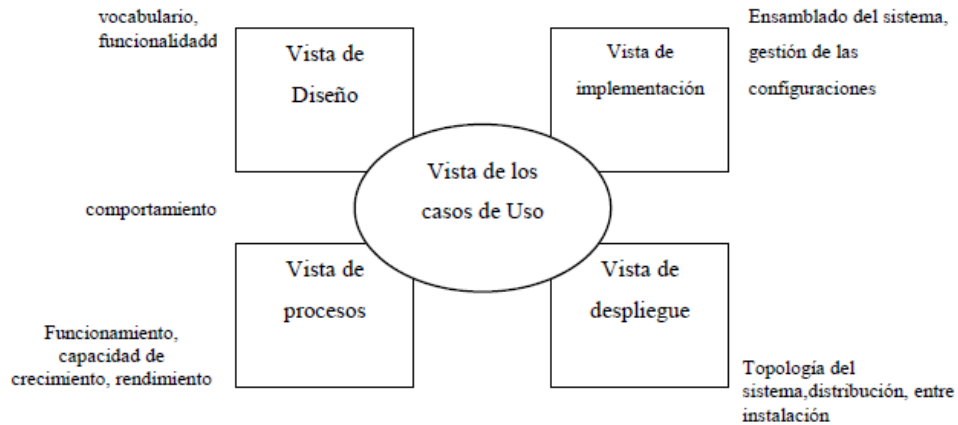


- **Diagramas de Componentes:** Muestra la organización y las dependencias entre un conjunto de componentes. Cubren la vista de la implementación estática y se relacionan con los diagramas de clases ya que en un componente suele tener una o más clases, interfaces o colaboraciones



2.3 Arquitectura

El desarrollo de un sistema con gran cantidad de software requiere que este sea visto desde diferentes perspectivas.



La **vista de casos de uso** comprende la descripción del comportamiento del sistema tal y como es percibido por los usuarios finales, analistas y encargados de las pruebas y se utilizan los diagramas de casos de uso para capturar los aspectos estáticos mientras que los dinámicos son representados por diagramas de interacción, estados y actividades.

La **vista de diseño** comprende las clases, interfaces y colaboraciones que forman el vocabulario del problema y de la solución. Esta vista soporta principalmente los requisitos funcionales del sistema, o sea, los servicios que el sistema debe proporcionar. Los aspectos estáticos se representan mediante diagramas de clases y objetos y los aspectos dinámicos con diagramas de interacción, estados y actividades.

La **vista de procesos** comprende los hilos y procesos que forman mecanismos de sincronización y concurrencia del sistema cubriendo el funcionamiento, capacidad de crecimiento y el rendimiento del sistema. Con UML, los aspectos estáticos y dinámicos se representan igual que en la vista de diseño, pero con el énfasis que aportan las clases activas, las cuales representan los procesos y los hilos.

La **Vista de implementación** comprende los componentes y los archivos que un sistema utiliza para ensamblar y hacer disponible el sistema físico. Se ocupa principalmente de la gestión de configuraciones de las distintas versiones del sistema. Los aspectos estáticos se capturan con los diagramas de componentes y los aspectos dinámicos con los diagramas de interacción, estados y actividades.

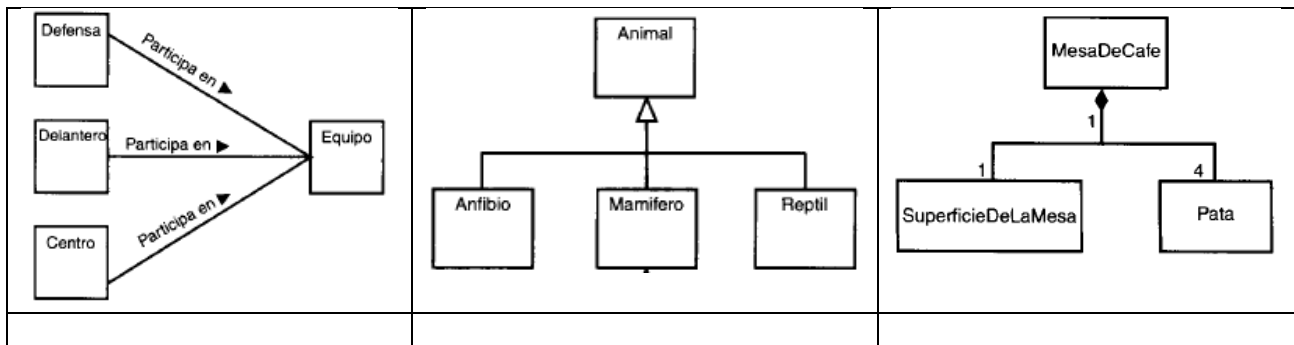
La **vista de despliegue** de un sistema contiene los nodos que forman la topología hardware sobre la que se ejecuta el sistema. Se preocupa principalmente de la distribución, entrega e instalación de las partes que constituyen el sistema. Los aspectos estáticos de esta vista se representan mediante los diagramas de despliegue y los aspectos dinámicos con diagramas de interacción, estados y actividades.

3. Guía de estudio

3.1 Respuesta breve: Conteste las siguientes preguntas

1. Defina el concepto del Lenguaje Unificado de Modelado.
2. ¿Qué es el paradigma de la orientación a objetos?
3. ¿Cuál es la diferencia entre una clase y un objeto?
4. ¿Cuáles son los elementos básicos de la estructura de UML?
5. ¿Qué son los bloques de construcción?
6. ¿Cuáles son los tipos de elementos o cosas de los bloques de construcción de UML?
7. Enumere las cosas o elementos estructurales de los bloques de construcción de UML
8. Explique la diferencia entre una interfaz u una clase.
9. ¿Cuál es el propósito de los elementos estructurales y los elementos de comportamiento respectivamente de los bloques de construcción de UML?
10. ¿Cuáles son los tipos de relaciones que existen?
11. ¿Qué tipo de relación utilizaría para representar que un objeto hereda propiedades de otro objeto?
12. Enumere los tipos de diagramas UML.

3.2 Completación: A continuación se muestra ejemplos de diagramas de la UML, identifique cual es el tipo de relación representada en cada uno.



Bibliografía

- Alarcon, R. (2000). *Diseño Orientado a objetos con UML*. Madrid, España: Grupo EIDOS.
- Kendall, K. (2011). *Analisis y Diseño de sistemas*. Mexico: Prentice Hall.
- Schmuller, J. (s.f.). *Aprendiendo UML en 24 horas*. Prentice Hall.