

BASES CONCEPTUALES ACERCA DEL LENGUAJE UNIFICADO DE MODELADO  
(UML) Y PATRONES DE DISEÑO

GA4-220501095-AA2-EV03

Presentado por:

ANDREA SAMIRA GARZÓN RENDÓN

COMPETENCIA:

DISEÑAR SOLUCIÓN

Centro de la Innovación, la Agroindustria y la Aviación

Sena Regional Antioquia

Tecnólogo Análisis y Desarrollo de Software

Ficha 2627079

2023

## **Introducción**

Para la solución de una necesidad de automatizar un proceso, es fundamental planear sobre diseño con UML, esto para cualquier proyecto de software, por ello en esta evidencia de aprendizaje se documenta las bases conceptuales de este lenguaje gráfico, se resume sus características, términos usados en este lenguaje y los procesos planteados por este sistema de modelado.

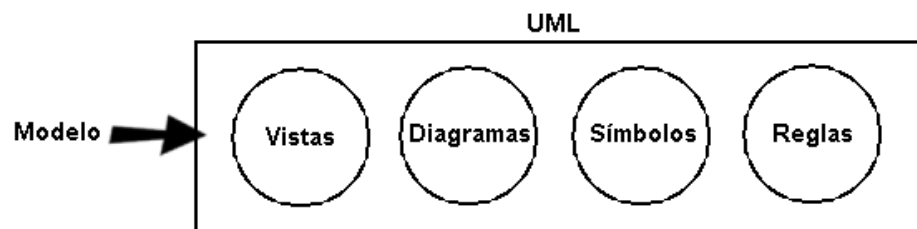
## LENGUAJE UNIFICADO DE MODELADO (UML)

Como su nombre y sus siglas lo indican (Unified Modeling Language) es conocido como un lenguaje de modelado o gráfico visual, es diferente de los métodos de análisis y diseño más conocidos, pero más que un lenguaje es un estándar de normas y gráficos en los que se basa el cómo se deben representar los esquemas de desarrollo de software. Su función principal es visualizar los estados y las interacciones entre objetos dentro de un sistema.

### Características de una herramienta UML:

- Capacidad de diagramación, y los diferentes tipos de diagramas que soporta la herramienta.
- Sus esquemas de apoyo de diseño, documentación, construcción e implantación de sistema.
- Flexibilidad para admitir cambios no previstos durante el diseño o el rediseño.

Un modelo es expresado en un lenguaje de modelado. Este consiste de vistas, diagramas, elementos de modelo y un conjunto de mecanismos generales o reglas que indican cómo utilizar los elementos, las reglas son sintácticas, semánticas y pragmáticas.



- **Vistas:** Muestran varios aspectos del sistema modelado. No es una gráfica sino una abstracción que consiste en un número de diagramas que muestran un panorama completo del sistema. Las vistas también ligan el lenguaje de modelado a los métodos o procesos elegidos para el desarrollo. Las diferentes vistas de UML son:
  - ✓ **Vista Use-Case:** Muestra la funcionalidad del sistema según como la perciben los actores externos.
  - ✓ **Vista Lógica:** Muestra cómo se diseña la funcionalidad dentro del sistema, en contexto de la estructura estática y conducta dinámica del sistema.
  - ✓ **Vista de Componentes:** Muestra la organización de los componentes de código.
  - ✓ **Vista Concurrente:** Muestra la concurrencia en el sistema, dirigiendo los problemas con la comunicación y sincronización que están presentes en un sistema concurrente.
  - ✓ **Vista de Distribución:** Muestra la distribución del sistema en la arquitectura física con computadoras y dispositivos nodos.
- **Diagramas:** Gráficas que describen el contenido de una vista. UML tiene nueve tipos de diagramas que proveen en conjunto todas las vistas de un sistema como diagramas de caso de uso, de clases, de objetos, de estados, de secuencia, de colaboración, de actividad, de componentes y de distribución.

- **Símbolos o Elementos de modelo:** Representan conceptos comunes orientados a objetos, tales como clases, objetos y mensajes, y las relaciones entre estos como asociación, dependencia y generalización. Un elemento de modelo es utilizado en varios diagramas diferentes, pero siempre tiene el mismo significado y simbología.
- **Reglas o Mecanismos generales:** Proveen comentarios extras, información o semántica acerca del elemento de modelo; además proveen mecanismos de extensión para adaptar o extender UML a un método o proceso específico, organización o usuario.

### **Tipos de diagramas en UML**

En este estándar de modelado se pueden crear bastantes diagramas, entre esos se definirán algunos:

- ✓ Diagramas de casos de uso: representan a los actores y casos de uso que intervienen en un desarrollo de software.
- ✓ Diagramas de clases: para UML una clase es una entidad, no una clase software, así que puede ser un diagrama del dominio o representación de conceptos que intervienen en un problema, o también un diagrama de clases software. El sentido de un diagrama UML se lo da quien lo construye.
- ✓ Diagramas de secuencia: Se usan para representar objetos software y el intercambio de mensajes entre ellos.

- ✓ Diagramas de colaboración: Se usan para representar objetos o clases y la forma en que se transmiten mensajes o se colaboran entre ellos para cumplir un objetivo.
- ✓ Diagramas de estados: Se usan para representar cómo un sistema cambia de estado a medida que se producen determinados eventos.
- ✓ Diagrama de componentes: Se crean cuando un sistema es complejo y consta de demasiadas clases.
- ✓ Diagrama de despliegue: Esto proporciona la representación real del hardware del sistema y sigue un enfoque más real sobre cómo se desplegaría el sistema.
- ✓ Diagrama de objetos: Es como un desglose de un diagrama de clases, se basan principalmente en entidades del mundo real.
- ✓ Diagrama de paquetes: Esto proporciona una representación de un sistema con sus módulos y subsistemas de más alto nivel. Aquí se incluyen varias estructuras de diagrama de despliegue.
- ✓ Diagrama de perfil: Se introdujo en UML 2. Aunque estos tipos de diagramas UML no son tan populares, pueden utilizarse para representar la meta estructura del sistema.
- ✓ Diagrama de la estructura compuesta: Se usan para representar la estructura interna de una clase. Representa cómo se relacionan las diferentes entidades de una clase entre sí y cómo se asocia la propia clase con la entidad o sistema exterior.

- ✓ Diagramas de comportamiento: Cubren las especificaciones restantes bajo UML. A diferencia de los diagramas estructurales, no son estáticos, sino que representan procesos y situaciones dinámicas.
- ✓ Diagramas de interacción: Son un subtipo de los diagramas de comportamiento, representan situaciones dinámicas. En particular, son adecuados para modelar el comportamiento en el que los elementos intercambian información.

Los diagramas UML se utilizan para representar los siguientes componentes del sistema:

- Objetos individuales - elementos básicos.
- Clases - combina elementos con las mismas propiedades.
- Relaciones entre objetos - jerarquía y comportamiento o comunicación entre objetos.
- Actividad - combinación compleja de acciones/módulos de comportamiento.
- Interacciones entre objetos e interfaces.

Categoría	Tipo de diagrama	Aplicación
Estructura	Diagrama de clases	Visualizar clases
	Diagrama de objetos	Estado del sistema en un momento dado
	Diagrama de componentes	Estructurar componentes y mostrar relaciones
	Diagrama de estructura compositiva	Divide los componentes o clases en sus componentes y aclara sus relaciones.
	Diagrama de paquete	Agrupar las clases en paquetes, muestra la jerarquía y la estructura de los paquetes.
	Diagrama de distribución	Distribución de componentes a los nodos informáticos
	Gráfica de perfil	Ilustra contextos de uso a través de estereotipos, condiciones límite, etc.
Comportamiento	Diagrama de casos de uso	Representa varias aplicaciones
	Diagrama de actividades	Describe el comportamiento de diferentes procesos (paralelos) en un sistema.
	Diagrama de máquina de estados	Documenta cómo un objeto es movido de un estado a otro por un evento.
Comportamiento: interacción	Diagrama secuencial	Secuencia temporal de las interacciones entre objetos
	Diagrama de comunicación	Distribución de roles de los objetos dentro de una interacción
	Diagrama de tiempos	Limitación de tiempo para los acontecimientos que conducen a un cambio de estado
	Diagrama de interacción	Secuencias y actividades interactivas



## **Glosario:**

- Objetos: Representan una entidad y el componente básico.
- Clase: Plano de un objeto.
- Abstracción: Comportamiento de una entidad del mundo real
- Encapsulación: Mecanismo para enlazar los datos y ocultarlos del mundo exterior.
- Herencia: Mecanismo para crear nuevas clases a partir de una existente.
- Polimorfismo: Define el mecanismo para salidas en diferentes formas.
- Actividad: Llevar a cabo el comportamiento en un diagrama de máquina de estados.
- Actor: un rol que asume un usuario cuando invoca un caso de uso.
- Agregación: Tipo de asociación que se utiliza para representar una relación más fuerte entre dos clases.
- Asociación: Una relación con 2 o más extremos, donde cada extremo está en una clase.
- Atributo: un dato significativo propiedad de una clase, que a menudo contiene valores que describen cada instancia de la clase.
- Bloque: el bloque es un lugar donde todos los agregados se recolectan en un solo lugar.
- Diagrama de clases: Tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos y las relaciones entre las clases.
- Componente: Representa un módulo de con una interfaz bien definida.

- Restricción: Condición booleana del lenguaje de restricción de objetos que puede no ser falsa si una clase se considera válida.
- Dependencia: existe una dependencia entre dos elementos definidos si un cambio en la definición de uno da como resultado un cambio en el otro.
- Evento: cuando ocurre en un objeto, puede causar una transición en un diagrama de máquina de estado.
- Herencia: donde una nueva clase más específica deriva parte de su definición de una clase más general existente.
- Paquete: colección o agrupación de clases relacionadas o de clases con funcionalidad relacionada.
- Caso de uso: un caso de uso se puede definir como una secuencia de acciones, incluidas las variaciones, que el sistema puede ejecutar y que produce un resultado observable que tiene algún valor para un actor que interactúa con el sistema.

## **Bibliografía**

- <https://www.edrawsoft.com/es/uml/>
- <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/uml-lenguaje-unificado-de-modelado-orientado-a-objetos/>
- <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/uml-lenguaje-unificado-de-modelado-orientado-a-objetos/>
- [https://hmong.es/wiki/Glossary\\_of\\_Unified\\_Modeling\\_Language\\_terms](https://hmong.es/wiki/Glossary_of_Unified_Modeling_Language_terms)
- <https://www.studocu.com/co/document/servicio-nacional-de-aprendizaje/identificar-los-requisitos-del-software/glosario-uml-software/31639421>
- <https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml>