

# **LENGUAJE UNIFICADO DE MODELADO (UML)**

**AUTOR: HELLEN CUBERO**

**NOVIEMBRE: 2020**



**San Marcos**

## INTRODUCCIÓN

El Lenguaje Unificado de Modelado es un lenguaje de modelado de sistemas de software muy conocido y su utilización ha se incrementado en los últimos años. El lenguaje UML es respaldado por el Object Management Group (OMG, 2011).

El UML ha tenido diversas versiones y su enfoque principal radica en visualizar, especificar, construir y documentar sistemas de información. En la presente lectura encontrará los usos, componentes y diagramas basados en la última actualización UML, la versión 2.5.



## Contenidos

Lenguaje Unificado de Modelado (UML).....	3
Objetivos y características UML .....	3
Usos del UML .....	3
UML un lenguaje para visualizar .....	3
UML es un lenguaje para especificar .....	4
UML es un lenguaje para construir .....	4
UML es un lenguaje para documentar .....	4
Tipos de diagramas UML.....	5
1. Diagramas estructurales.....	5
2. Diagramas de comportamiento.....	10
Conclusiones y recomendaciones .....	15
Referencias bibliográficas.....	16

## Lenguaje Unificado de Modelado (UML)

El lenguaje unificado de modelado es abreviado y reconocido como UML. Es la notación gráfica utilizada para visualizar, diseñar e implementar un sistema de software, incluyendo los procesos e interacciones que forman el sistema.

### Objetivos y características UML

La finalidad de UML según el Object Management Group (OMG), es brindar a los arquitectos de sistemas, ingenieros y desarrolladores de software las herramientas para el análisis, el diseño y la implementación de sistemas basados en software, así como el modelado de procesos de negocios y además, hacer progresar el estado de la industria permitiendo la interoperabilidad de herramientas de modelado visual de objetos. (OMG, 2011).

### Usos del UML

Según Jacobson y Rumbaugh (2006), el UML es un lenguaje estándar para escribir planos de software. Puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucre una gran cantidad de software.

### UML un lenguaje para visualizar

Algunos programadores imaginan el sistema y lo van codificando, crean el modelado mentalmente, sin embargo, esto algunas veces genera problemas, por ejemplo la comunicación de los modelos conceptuales a otros está sujeta a errores, los proyectos y organizaciones crean sus propios lenguajes y es difícil comprender cuando alguien se incorpora al grupo, además el desarrollador no deja información escrita sobre los modelos mentales generados y en otras ocasiones los detalles del sistema no se pueden comprender de manera textual, por lo tanto, escribir modelos UML facilita la comunicación y la interpretación (de manera gráfica se comprende mejor), además de que detrás de la

notación UML hay una semántica bien definida, un desarrollador puede escribir el modelo y se interpretado por otro desarrollador sin ambigüedades.

#### UML es un lenguaje para especificar

Permite construir modelos precisos, no ambiguos y completos.

#### UML es un lenguaje para construir

No es un lenguaje de programación visual, pero sus modelos pueden conectarse de forma directa con diferentes lenguajes de programación, es decir, puede establecer correspondencias de un modelo UML a un lenguaje de programación como Java, C++, entre otros.

#### UML es un lenguaje para documentar

Cubre la documentación de la arquitectura de un sistema y todos sus detalles, proporciona un lenguaje para expresar requisitos, pruebas, modelado de actividades de planificación de proyectos y gestión de versiones.

## Tipos de diagramas UML

El UML ha tenido diversas versiones y mejoras, la versión más actual es la 2.5 y según DiagramasUML (2020) los diagramas contenidos son los siguientes:

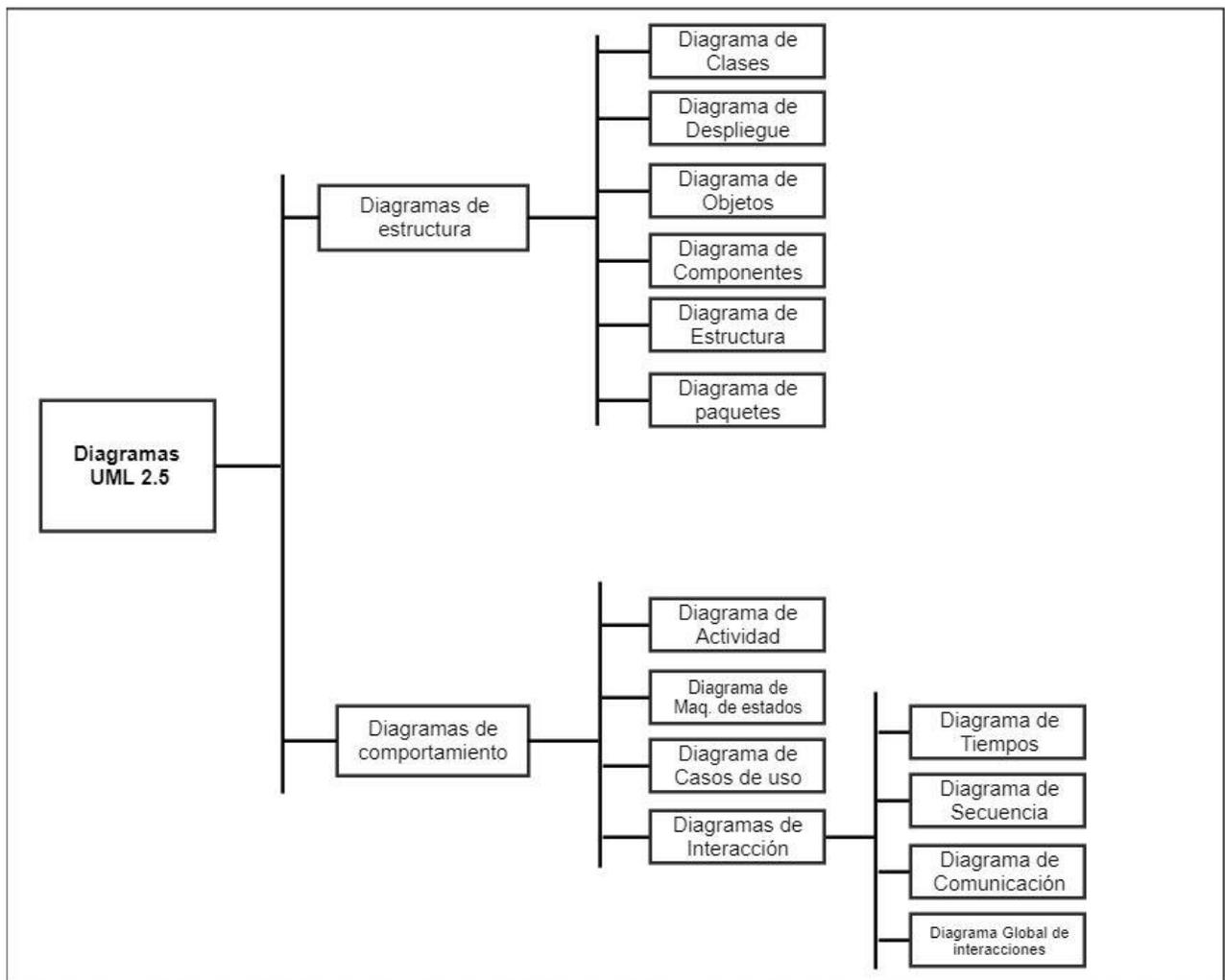


Ilustración 1: Clasificación de diagramas UML

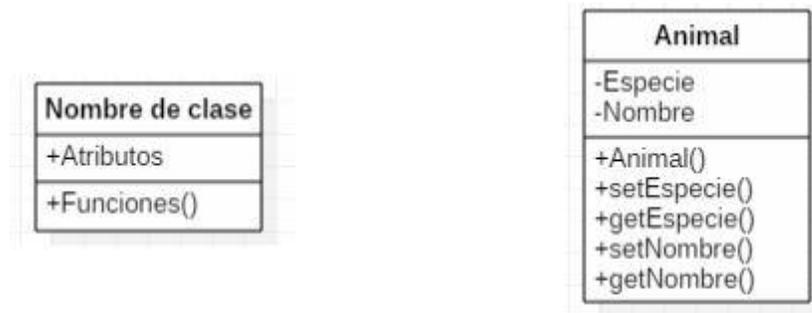
Fuente: DiagramasUML

### 1. Diagramas estructurales

Muestran la estructura estática del sistema y sus partes en diferentes niveles de abstracción. Los tipos de diagramas pertenecientes a esta

categoría son los siguientes:

**1.1 Diagrama de clases:** muestra la estructura del sistema, con sus características, restricciones y relaciones. Se grafica de la siguiente manera:



*Ilustración 2: Notación de una clase*

Fuente: DiagramasUML

Tanto los atributos como las funciones incluyen al inicio de su descripción la visibilidad que tendrán. La visibilidad se identifica con un símbolo:  
 (+) Pública: Significa que puede acceder al atributo o función desde cualquier lugar de la aplicación.

(-) Privada: Significa que puede acceder al atributo o función únicamente desde la misma clase.

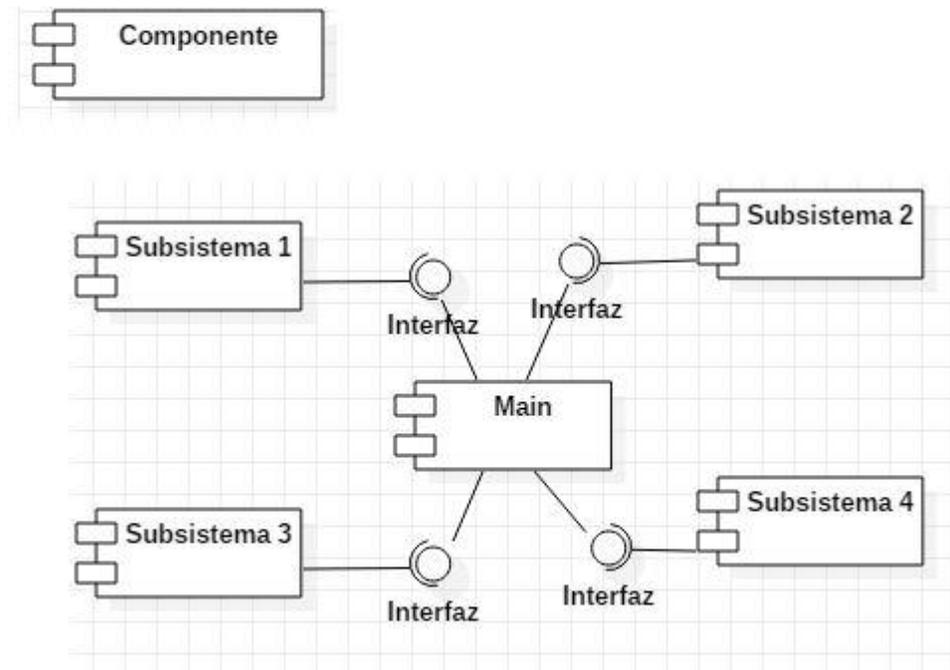
(#) Protegida: Significa que puede acceder al atributo o función desde la misma clase o desde las clases que hereden de ella.

Relaciones: Una relación identifica una dependencia, que puede darse entre dos o más clases. Los tipos de relaciones son los siguientes: asociación, agregación, composición, dependencia y herencia.

Nota: Para obtener más información visite la página: <https://diagramasuml.com/diagrama-de-clases/>

**1.2 Diagrama de componentes:** Proporciona una vista de alto nivel de los componentes dentro del sistema. Está formado por tres elementos: componente, interfaz y relación de dependencia. Un componente puede

representar dos tipos de elementos: componentes lógicos (por ejemplo componentes de negocio o procesos) o componentes físicos (como .net).



*Ilustración 3: Diagrama de componentes*

Fuente: DiagramaUML

Para obtener más información visite la página:  
<https://diagramasuml.com/componentes/>

**1.3 Diagrama de despliegue:** Es utilizado para representar la distribución física de los componentes de software en los distintos nodos físicos de la red.

Un nodo es un elemento físico que interactúa de alguna manera con el sistema o bien forma parte del mismo.

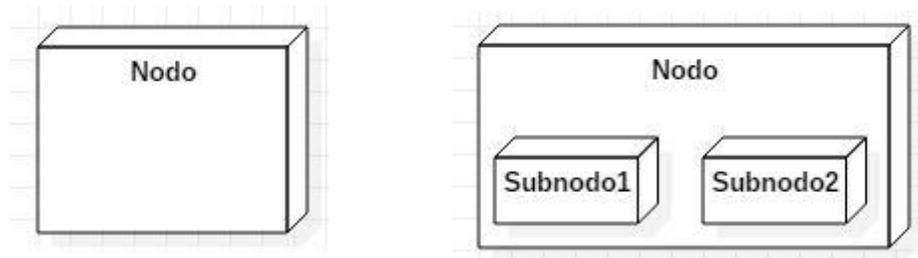


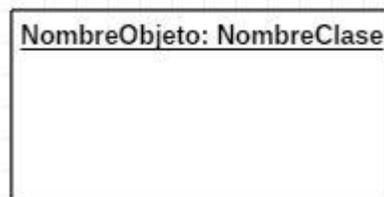
Ilustración 4: Notación de nodos y subnodos

Fuente: DiagramasUML

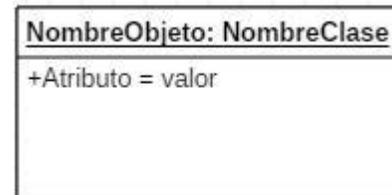
Para obtener más información visite la página:  
<https://diagramasuml.com/despliegue/>

**1.4 Diagrama de objetos:** es un gráfico de instancias, incluyendo objetos y valores de datos. Un diagrama de objeto estático es una instancia de un diagrama de clase; muestra una instantánea del estado detallado de un sistema en un punto en el tiempo.

Los elementos del diagrama de objetos son los siguientes: objetos, atributos y vínculos



Representación de un objeto



Representación de un atributo



**Asociación entre dos objetos**

*Ilustración 5: Diagrama de objetos*

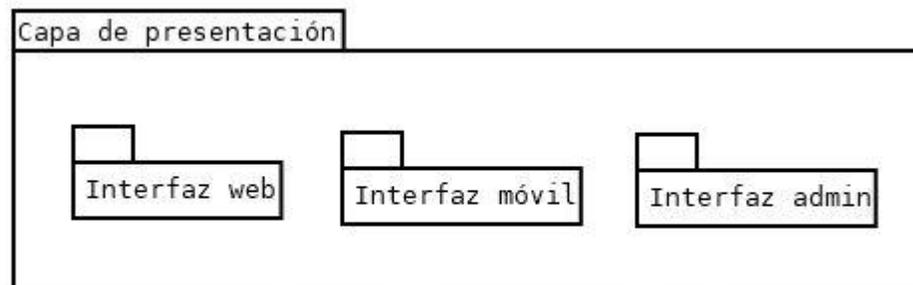
Fuente: DiagramasUML

Para obtener más información, visite la página:  
<https://diagramasuml.com/objetos/>

**1.5 Diagrama de paquetes:** muestra los paquetes y las relaciones entre los paquetes.



**Notación de un paquete**



**Notación de un paquete que contiene otros paquetes**

*Ilustración 6: Diagrama de paquetes*

Fuente: DiagramasUML

Para obtener más información, visite la página:  
<https://diagramasuml.com/paquetes/>

**1.6 Diagrama de perfiles:** permite extender UML para su uso con una plataforma de programación particular (como el Framework .NET de Microsoft o la plataforma Java Enterprise Edition), o modelar sistemas destinados a ser usados en un dominio en particular.

Los nodos y elementos gráficos utilizados en el diagrama de perfiles son los siguientes: perfil, metaclassa, estereotipo, extensión, referencia y aplicación de perfil.

Para obtener información adicional, visite la página:  
<https://diagramasuml.com/perfiles/>

**1.7 Diagrama de estructura compuesta:** desempeña una función similar a un diagrama de clases, pero permite entrar en más detalles al describir la estructura interna de varias clases y mostrar las interacciones entre ellas. Los elementos del diagrama de estructura compuesta son los siguientes: parte, puerto, conector, colaboración, clasificador estructurado y clasificador encapsulado.

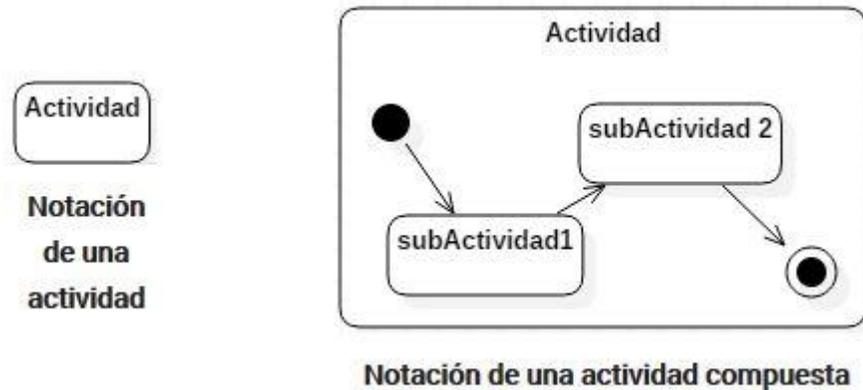
Para obtener información adicional, visite la página:  
<https://diagramasuml.com/estructura-compuesta/>

## 2. Diagramas de comportamiento

Muestra como se comporta un sistema de información en forma dinámica, es decir, describe los cambios que muestra un sistema a través del tiempo cuando está en ejecución.

Los diagramas de comportamiento son clasificados de la siguiente manera:

**2.1 Diagrama de actividades:** muestra una secuencia de acciones, un flujo de trabajo que va desde un punto inicial a un punto final.



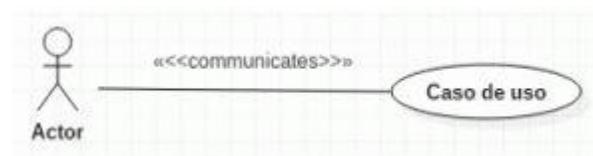
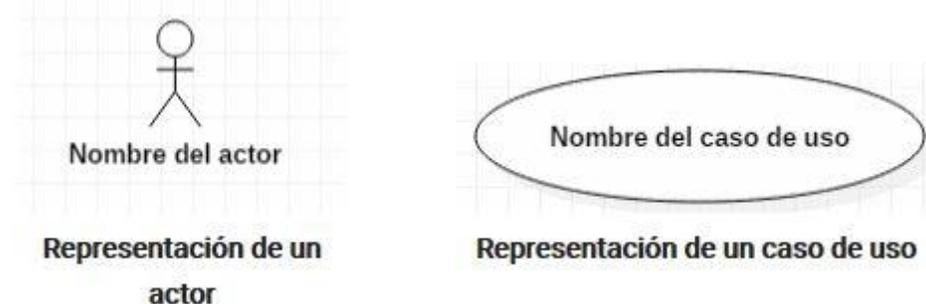
*Ilustración 7: Diagrama de actividad*

Fuente: DiagramasUML

Para obtener más información, visite la página:  
<https://diagramasuml.com/actividades/>

**2.2 Diagrama de caso de uso:** muestra el conjunto de acciones que algunos sistemas deben o pueden realizar en colaboración con algunos actores externos para proporcionar resultados valiosos a otros interesados del sistema.

Los elementos de casos de uso son: actores, casos de uso y relaciones.



*Ilustración 8: Elementos de casos de uso*

Fuente: DiagramasUML

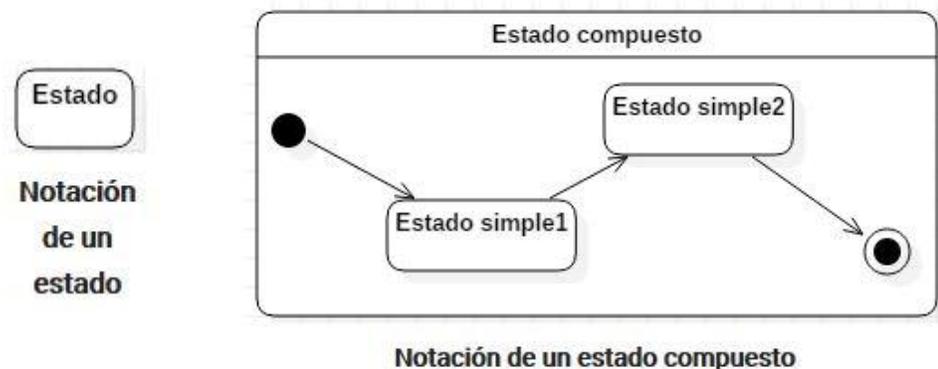


Las relaciones conectan los casos de uso con los actores o los casos de uso entre sí, las relaciones podrían ser <<comunicates>>, <<include>> o <<extend>>.

Para obtener más información visite la página: <https://diagramasuml.com/casos-de-uso/>

**2.3 Diagrama de máquina de estados:** muestra los estados por los que pasa un componente de un sistema de información.

Los diagramas de estados están formados por tres componentes: estados, pseudoestados y transiciones.



*Ilustración 9: Diagrama de estados*

Fuente: DiagramasUML

Para obtener información adicional visite la página: <https://diagramasuml.com/estados/>

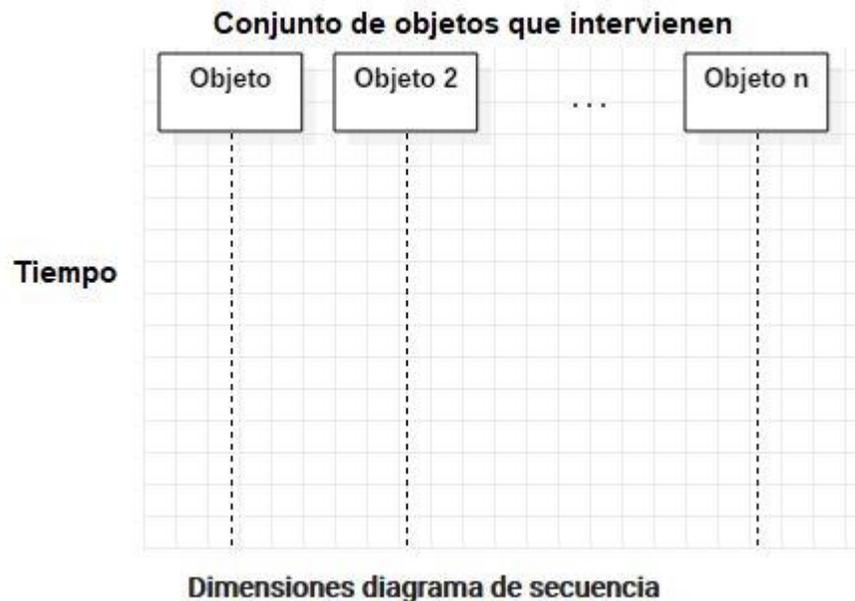
### **2.4 Diagramas de interacción**

Es un subconjunto de los diagramas de comportamiento, posee los siguientes diagramas:

**2.4.1 Diagramas de secuencia:** muestra el intercambio de mensajes entre los distintos objetos del sistema para cumplir con una funcionalidad. Está construido a partir de dos direcciones: horizontal que representan los objetos que participan en la secuencia y vertical representan la línea de tiempo sobre la que los elementos actúan.

El diagrama de secuencia está compuesto por dos elementos: Objetos y

mensajes.



*Ilustración 10: Diagrama de secuencias*

Fuente: DiagramasUML

Para obtener información adicional, visite la página:  
<https://diagramasuml.com/secuencia/>

**2.4.2 Diagrama de comunicación:** muestra las interacciones entre objetos y/o partes, representadas como líneas de vida.

Los elementos de comunicación son los siguientes: Marco o frame, línea de vida y mensaje.

Para obtener información adicional, visite la página:  
<https://diagramasuml.com/comunicacion/>

**2.4.3 Diagrama de tiempo:** describen el comportamiento de los clasificadores individuales y las interacciones de los clasificadores, enfocando la atención en los tiempos de los eventos que causan cambios en las condiciones de las líneas de vida.

El diagrama de tiempos incluye los siguientes elementos: línea de vida,

estados, restricciones de duración y restricciones de tiempo.

Para obtener información adicional, visite la página:  
<https://diagramasuml.com/diagrama-de-tiempos/>

**2.4.4 Diagrama global de interacciones:** se utiliza para vincular los diagramas y lograr un alto grado de navegabilidad entre los diagramas.

Para obtener información adicional visite la página:  
<https://diagramasuml.com/>

Según Jacobson y Rumbaugh (2006), el 80% de la mayoría de los problemas pueden modelarse con el 20% de UML. Los modelos estructurales básicos, tales como clases, atributos, operaciones, casos de uso y paquetes, junto a las relaciones estructurales básicas como la dependencia, la generalización y la asociación, son suficientes para crear modelos estáticos para muchos tipos de dominios de problemas. Si a esta lista se suman los elementos de comportamiento, las características avanzadas de UML se utilizarían para modelar las cosas que aparecen en situaciones más complejas.

**PARA CONOCER EJEMPLOS Y  
DISEÑOS DE LOS DIFERENTES  
DIAGRAMAS INGRESE A LA PÁGINA:**

**[HTTPS://DIAGRAMASUML.COM/](https://diagramasuml.com/)**

## Conclusiones y recomendaciones

A modo de conclusión, el Lenguaje Unificado de Modelado (UML), es un estándar que facilita la interpretación de un sistema de información, debido a que otorga un lenguaje universal que permite a todos los involucrados interpretar de manera correcta los componentes, comunicaciones, entidades, relaciones y diagramas graficados según los objetivos planteados.

Por otra parte, aunque en la lectura se dan a conocer diferentes diagramas UML, es importante mencionar que no todos deben ser utilizados, según un estudio realizado en 2007 (Sommerville, 2011) mostró que la mayoría de los usuarios del UML consideraban que cinco tipos de diagrama podrían representar lo esencial de un sistema (diagramas de actividad, casos de uso, diagramas de secuencias, diagramas de clase y diagramas de estado).

Como recomendación, para ampliar los conocimientos mostrados en la lectura, se recomienda visualizar los ejemplos, diagramas, componentes y demás datos informativos que se encuentran en la página: <https://diagramasuml.com/>.

## Referencias bibliográficas

Object Management Group. (2011). OMG Unified Modeling Language (OMG UML), Infraestructure. Recuperado de <https://www.omg.org/spec/UML/2.4.1/Infrastructure/PDF/>

Jacobson, I., Rumbaugh, J. (2006). El lenguaje unificado de modelado: guía del usuario (2A.ED.) Recuperado de <https://elibro.net/es/lc/usanmarcos/titulos/52531>

DiagramasUML. (2020). Diagramas UML. [Entradas de blog] Recuperado de <https://diagramasuml.com/>

Sommerville, I. (2011). Ingeniería de software (9a.ed.). Pearson Educación. <https://elibro.net/es/ereader/usanmarcos/37857>

## BANCO DE PREGUNTAS

**Nota:** Coloque las respuestas correctas en las primeras líneas. Por ejemplo si la pregunta solo tiene una respuesta correcta esta debe estar en la línea A, si tiene más respuestas correctas estas van en la opción B, C.

### Ejemplo

RM	De acuerdo con la distribución territorial de Costa Rica, cual es la cantidad de cantones que posee
@	La respuesta se localiza en el módulo 1, lectura 1 página 2.
*A.	90
B.	87
*C.	82 (esta se coloca un (*) para indicar que es la respuesta correcta dentro de todas las alternativas)
D.	81
*E.	79

1.	La siguiente definición: “Es la notación gráfica utilizada para visualizar, diseñar e implementar un sistema de software. Incluyendo las interacciones”, hace referencia al concepto de
@	Lectura M1_ Lenguaje unificado de modelado, página 4
*A.	Lenguaje Unificado de Modelado
B.	Diseño de sistemas de software
C.	Estructura de software
D.	Lenguaje de programación de software

2.	La siguiente descripción: “Es utilizado para representar la distribución física de los componentes de software en los distintos nodos físicos de la red”, hace referencia al concepto de
@	Lectura M1_ Lenguaje unificado de modelado, página 9
*A.	Diagramas de despliegue
B.	Diagramas de componentes
C.	Diagramas de clases
D.	Diagrama de objetos

3.	La siguiente descripción: “Proporciona una vista de alto nivel de componentes dentro del sistema. Está formado por tres elementos: componente, interfaz y relación de dependencia”, hace referencia al concepto de
@	Lectura M1_ Lenguaje unificado de modelado, página 8
*A.	Diagrama de componentes
B.	Diagramas de despliegue
C.	Diagramas de objetos
D.	Diagramas de clase

4.	La siguiente descripción: “Muestra una secuencia de acciones, un flujo de trabajo que va desde un punto inicial a un punto final”, hace referencia al concepto de
@	Lectura M1_ Lenguaje unificado de modelado, página 12



*A.	Diagrama de actividades
B.	Diagrama de paquetes
C.	Diagrama de casos de uso
D.	Diagrama de secuencia

5.	La siguiente descripción: “muestra el conjunto de acciones que algunos sistemas deben o pueden realizar en colaboración con algunos actores externos para proporcionar resultados valiosos a otros interesados del sistema”, hace referencia al concepto de
@	Lectura M1_ Lenguaje unificado de modelado, página 12
*A.	Diagrama de casos de uso
B.	Diagrama de paquetes
C.	Diagrama de actividades
D.	Diagrama de secuencia

6.	La siguiente descripción: “Muestra como se comporta un sistema de información en forma dinámica”, hace referencia al concepto de
@	Lectura M1_ Lenguaje unificado de modelado, página 11
*A.	Diagrama de comportamiento
B.	Diagramas de estructura compuesta
C.	Diagrama de perfiles
D.	Diagrama de secuencia

7.	Dos diagramas que pertenecen a la categoría de Diagramas de comportamiento, son los siguientes:
@	Lectura M1_ Lenguaje unificado de modelado, página 6
*A.	Diagrama de actividad y diagrama de comunicación
B.	Diagrama de componentes y Diagrama de casos de uso
C.	Diagrama de paquetes y Diagrama de clases
D.	Diagrama de componentes y Diagramas de objetos

8.	La siguiente definición: “se utiliza para vincular los diagramas y lograr un alto grado de navegabilidad entre los diagramas”
@	Lectura M1_ Lenguaje unificado de modelado, página 15
*A.	Diagrama global de interacciones
B.	Diagrama de tiempo
C.	Diagrama de paquetes
D.	Diagrama de componentes



[www.usanmarcos.ac.cr](http://www.usanmarcos.ac.cr)

San José, Costa Rica