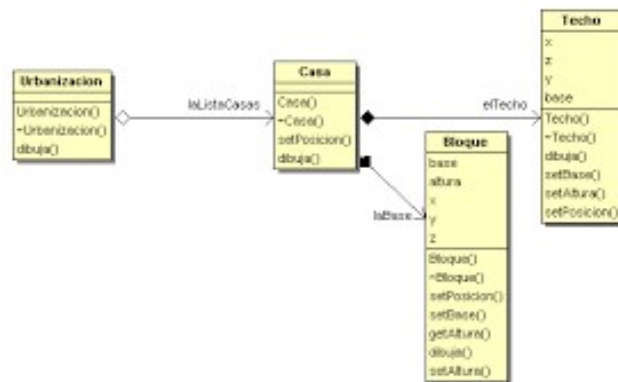


# UD5: UML. Elaboración de Diagramas de clases (I)

Entornos de desarrollo  
M<sup>a</sup> Carmen Safont Richarte

# INDICE

1. UML: Unified Modeling Language
2. Tipos de diagrama UML
3. Elaboración de Diagramas de clases.



# 1. UML: Unified Modeling Language

## CONTENIDOS

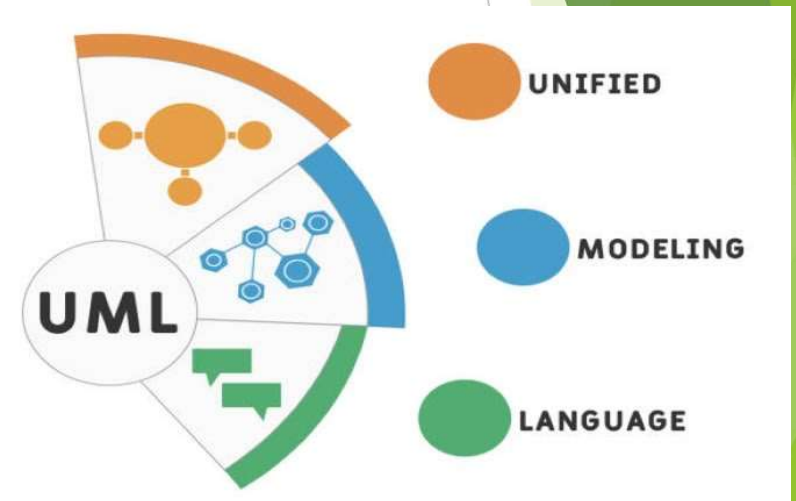
- 1.1 Introducción
- 1.2 ¿Qué es UML?
  - 1.2.1 Modelado de un proyecto
  - 1.2.2 ¿Por qué UML?
- 1.3 Breve historia de UML

# 1. UML: Unified Modeling Language

## 1.1. Introducción.

*Si viajas a un país anglosajón como EEUU o Inglaterra, tendrás que hablar inglés para hacerte entender, si vas a un país hispanohablante como Argentina, tendrás que hablar español, y si vas a una empresa de desarrollo de software, tendrás que manejar UML, que es el lenguaje que se utiliza en los proyectos software orientados a objetos.*

*Por tanto, UML es una herramienta fundamental que necesitas conocer porque luego la vas a utilizar a lo largo de tu vida profesional.*



# 1. UML: Unified Modeling Language

## 1.2. ¿Qué es UML?

### Definición oficial de UML:

*UML es un lenguaje de propósito general que ayuda a especificar, visualizar y documentar modelos de sistemas software, incluido su estructura y diseño, de tal forma que se unifiquen todos sus requerimientos*

# 1. UML: Unified Modeling Language

## 1.2. ¿Qué es UML?

- Es un lenguaje visual estandarizado de modelado.
- Está especialmente desarrollado para ayudar a todos los intervinientes en el desarrollo y modelado de un sistema o un producto software a describir, diseñar, especificar, visualizar, construir y documentar todos los artefactos que lo componen.
- Dispone de numerosos tipos de diagramas.
- Cada tipo de diagrama muestra un aspecto diferente del **modelo** de software.

*El modelado es la principal forma de visualizar el diseño de una aplicación con la finalidad de compararla con los requisitos antes de que el equipo de desarrollo comience a codificar.*

# 1. UML: Unified Modeling Language

## 1.2.1. Modelos y Diagramas

- **Modelar** consiste en diseñar aplicaciones software antes de su implementación. Es una parte esencial en proyectos de tamaño medio o grande.
- Un **modelo** es una abstracción de un elemento del sistema a desarrollar, desde un punto de vista determinado y, por lo tanto, con un nivel de detalle específico.
- Un **diagrama** consiste en una representación gráfica de una colección de modelos para representar una parte específica del sistema.
- Con el modelado del sistema podemos definir todos sus requisitos de forma totalmente a su implementación.
- Esto permite tener una **visión global** del sistema antes de iniciar su desarrollo, lo que implica:
  - Tener unos requisitos bien definidos.
  - Reducir problemas en el proceso de implementación.
  - Tener una visión global del sistema en una primera fase de su desarrollo.
  - Para poder modelar un sistema y conducir el proceso de desarrollo, es necesario el uso de una solución software

# 1. UML: Unified Modeling Language

## 1.2.1. Modelos y Diagramas

- **Los diagramas de UML** son representaciones gráficas que muestran de forma parcial un sistema de información, bien esté siendo desarrollado o ya lo haya sido. Suelen estar acompañados de documentación que les sirve de apoyo, adoptando múltiples formas. Además, UML no excluye la posibilidad de mezclar diagramas, algo que, de hecho, suele ser bastante común. Siempre teniendo en cuenta una de las máximas de UML: **Una imagen vale más que mil palabras.**
- Como **principal desventaja** ampliamente mencionada de UML podemos nombrar el hecho de que se trata de un lenguaje muy amplio, haciendo, en ocasiones, complicado utilizar todas las posibilidades que ofrece. No obstante, los analistas tienden a utilizar los diagramas de forma sencilla, consiguiendo que sean entendidos fácilmente por cualquier persona que accedan a ellos.



# 1. UML: Unified Modeling Language

## 1.2.2. ¿Por qué UML?

- Los modelos o diagramas de UML nos ayudan a trabajar a un mayor nivel de abstracción. Permite modelar cualquier tipo de aplicación corriendo en cualquier combinación de hardware y software, sistema operativo, lenguaje de programación y red, es decir, UML es independiente de la plataforma hardware sobre la que actúa el software. Su flexibilidad permite modelar cualquier tipo de aplicación e, incluso, otros tipos de proyecto que no son puramente software.
- UML ofrece ese modelado utilizando diagramas y se denomina lenguaje por ser una forma común de expresarse por todos los analistas, desarrolladores y usuarios. Está desarrollado para ayudar a todos estos (y más) perfiles a especificar, visualizar, construir y documentar todos los componentes de un proyecto.

# 1. UML: Unified Modeling Language

## 1.2.2. ¿Por qué UML?

A pesar de que cada diagrama UML en particular aporta su visión particular al modelado, el lenguaje en su conjunto tiene algunas características que interesa resaltar:

- Es muy **sencillo**. Pese a que si es usado de forma completa puede llegar a complicarse, lo normal es que se simplifique.
- Es capaz de modelar **todo tipo de sistemas**.
- Es un lenguaje **universal**, haciendo que todos los miembros del equipo se relacionen a través de sus diagramas sean del ámbito que sean.
- Es fácilmente **extensible**. Tiene mecanismos sencillos para especializar los conceptos fundamentales.
- Es **visual** y, por lo tanto, intuitivo.
- Es **independiente** del desarrollo, del lenguaje y de la plataforma.
- Bien ejecutado aporta un conjunto considerable de **buenas prácticas**.
- **No está completo**. Utilizando los distintos diagramas no podemos estar seguros de comprender con totalidad el sistema que va a desarrollarse. Los diagramas, para facilitar su comprensión pueden (y suelen) omitir información, pueden tener partes que se entienden de distintas maneras o, incluso, pueden tener conceptos que no pueden ser representados por ningún diagrama.

# 1. UML: Unified Modeling Language

## 1. 3. Breve historia del UML

- En un inicio, varios autores utilizan lenguajes de modelado (Booch, OMT, etc.).
- En 1995 se unifican en la versión 0.8 de UML.
- En 1997 UML se modifica con la experiencia obtenida en su utilización en empresas, lo que permite la creación de la versión **1.0**.
- En 1998, UML se aprueba por el OMG (Object Management Group), donde pasa a englobarse desde ese momento.
- Actualmente la última versión es la 2.5.1

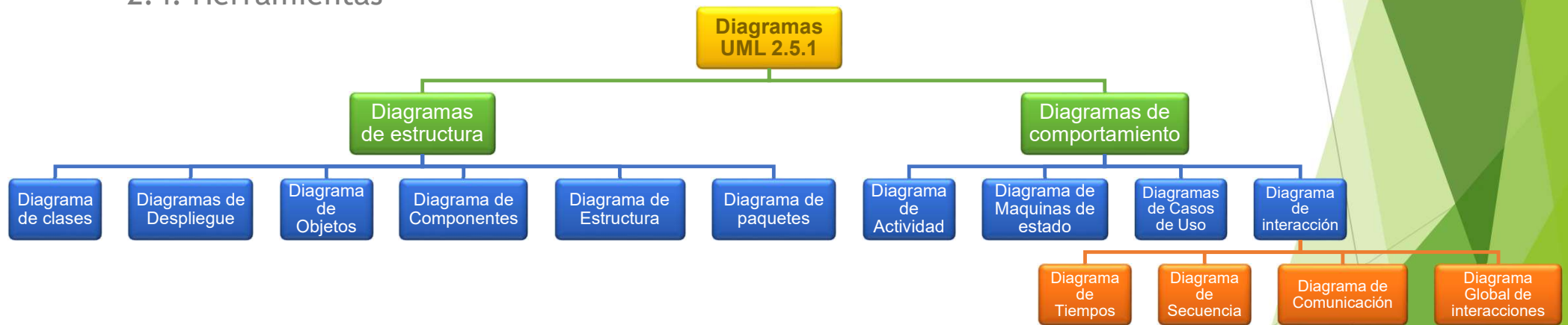
## 2. Tipos de diagramas UML

2.1 Diagramas estructurales

2.2 Diagramas de comportamiento

2.3. Alternativas a UML

2.4. Herramientas



## 2. Tipos de diagramas UML

### 2.1. Diagramas de Estructura

- Los diagramas estructurales muestran la estructura estática del sistema y sus partes en diferentes niveles de abstracción.
- Existen un total de **siete** tipos de diagramas de estructura.
- Nosotros nos centraremos únicamente en el más importante, los diagramas de clases

## 2. Tipos de diagramas UML

### 2.1. Diagramas de Estructura

#### Diagrama de clases

- Muestra la estructura del sistema, subsistema o componente utilizando clases con sus características, restricciones y relaciones: asociaciones, generalizaciones, dependencias, etc.

#### Diagrama de componentes

- Muestra componentes y dependencias entre ellos. Este tipo de diagramas se utiliza para el desarrollo basado en componentes (CDB), para describir sistemas con arquitectura orientada a servicios (SOA).

#### Diagrama de despliegue

- Muestra la arquitectura del sistema como despliegue (distribución) de artefactos de software.

#### Diagrama de objetos

- 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.

#### Diagrama de paquetes

- Muestra los paquetes y las relaciones entre los paquetes.

#### Diagrama de perfiles

- Diagrama UML auxiliar que permite definir estereotipos personalizados, valores etiquetados y restricciones como un mecanismo de extensión ligero al estándar UML. Los perfiles permiten adaptar el metamodelo UML para diferentes plataformas o dominios.

#### Diagrama de estructura compuesta

- Muestra la estructura interna (incluidas las partes y los conectores) de un clasificador estructurado.

## 2. Tipos de diagramas UML

### 2.2. Diagramas de Comportamiento

- A diferencia de los diagramas estructurales, los diagramas de comportamiento muestran cómo se comporta un sistema de información de forma dinámica. Es decir, describe los cambios que sufre un sistema a través del tiempo cuando está en ejecución.
- Hay un total de siete diagramas de comportamiento.
- Nosotros nos centraremos únicamente en los más importantes, los diagramas de caso de uso y los diagramas de secuencia.

## 2. Tipos de diagramas UML

### 2.2. Diagramas de Comportamiento

#### Diagrama de actividades

- Muestra la secuencia y las condiciones para coordinar los comportamientos de nivel inferior, en lugar de los clasificadores que poseen esos comportamientos. Estos son comúnmente llamados modelos de flujo de control y flujo de objetos.

#### Diagrama de casos de uso

- Describe un conjunto de acciones (casos de uso) que algunos sistemas o sistemas (sujetos) deben o pueden realizar en colaboración con uno o más usuarios externos del sistema (actores) para proporcionar algunos resultados observables y valiosos a los actores u otros interesados del sistema(s).

#### Diagrama de máquinas de estado

- Se utiliza para modelar el comportamiento discreto a través de transiciones de estados finitos. Además de expresar el comportamiento de una parte del sistema, las máquinas de estado también se pueden usar para expresar el protocolo de uso de parte de un sistema.

#### Diagrama de interacción

- Es un subconjunto de los diagramas de comportamiento. Comprende los siguientes diagramas:

##### Diagrama de secuencia

- Es el tipo más común de diagramas de interacción y se centra en el intercambio de mensajes entre líneas de vida (objetos).

##### Diagrama tiempos

- Se centran en las condiciones que cambian dentro y entre las líneas de vida a lo largo de un eje de tiempo lineal.

##### Diagrama global de interacciones

- Los diagramas globales de interacciones brindan una descripción general del flujo de control donde los nodos del flujo son interacciones o usos de interacción.



## 2. Tipos de diagramas UML

### 2.3. Alternativas a UML

- Si bien UML es el estándar más utilizado y reconocido existen algunas alternativas que se han desarrollado para abordar diferentes enfoques o necesidades específicas en el modelado de sistemas.
- Estas alternativas pueden ser utilizadas en situaciones donde UML pueda resultar limitado o donde se necesite un enfoque más especializado para el modelado de sistemas. Sin embargo, es importante destacar que UML sigue siendo el estándar predominante y ampliamente aceptado para el modelado de sistemas de software debido a su versatilidad y amplia gama de diagramas para representar diferentes aspectos de un sistema.

SysML (Systems Modeling Language):	BPMN (Business Process Model and Notation):	ERD (Entity-Relationship Diagrams):	Archimate:	Flowchart (Diagrama de Flujo):	DSL (Domain-Specific Languages):
<ul style="list-style-type: none"><li>• Diseñado para el modelado de sistemas de ingeniería y sistemas físicos, es una extensión de UML que se centra en el modelado de sistemas complejos.</li></ul>	<ul style="list-style-type: none"><li>• Especializado en modelar procesos de negocios y flujos de trabajo. Aunque no es un reemplazo directo de UML, se utiliza comúnmente en conjunto con él para representar aspectos de procesos de negocio.</li></ul>	<ul style="list-style-type: none"><li>• Utilizado principalmente en el modelado de datos y bases de datos. Si bien no reemplaza a UML, se enfoca en la representación de relaciones entre entidades y atributos en un contexto de bases de datos. Puedes aprender más sobre este diagrama a través de <a href="#">esta entrada en el blog</a>.</li></ul>	<ul style="list-style-type: none"><li>• Un estándar de modelado de arquitectura empresarial que se enfoca en la representación de la arquitectura y la infraestructura empresarial, incluyendo aspectos como procesos, aplicaciones, estructuras de datos, etc.</li></ul>	<ul style="list-style-type: none"><li>• Aunque más simple en comparación con UML, los diagramas de flujo son útiles para representar algoritmos, flujos de trabajo simples y procesos de toma de decisiones.</li></ul>	<ul style="list-style-type: none"><li>• Estos son lenguajes de modelado diseñados específicamente para un dominio particular o un problema específico. A menudo, se utilizan para representar conceptos y abstracciones en un dominio específico de manera más precisa que UML.</li></ul>

## 2. Tipos de diagramas UML

### 2.4. Herramientas para construir diagramas UML (online)

- La demanda de colaboración en línea combinado con las nuevas increíbles interfaces web que pueden crearse con las nuevas tecnologías (como HTML5) ha creado un nuevo mercado en rápido crecimiento de herramientas para **construir diagramas UML online gratis** y, también, de pago. De hecho, muchas de estas herramientas son más herramientas de dibujo que herramientas de modelado, pero, al menos, nos ofrecen la utilidad de dibujar y compartir nuestros modelos a través de Internet.
- Existen en la actualidad muchas herramientas (y más que siguen entrando al mercado) con una amplia variedad de características diferentes. Esta lista incluye exclusivamente herramientas online, desechando bibliotecas para entornos de desarrollos u otros tipos de herramienta, pero también existen otras herramientas de escritorio

## 2. Tipos de diagramas UML

### 2.4. Herramientas para construir diagramas UML (online)

- Con un fuerte énfasis en los aspectos de control colaborativo y de revisión, Gliffy afirma ser «la aplicación de diagramación en línea más utilizada».
- Gliffy admite todos los diagramas UML junto con una variedad de otros tipos de diagramas, incluido un soporte sólido para los modelos de procesos BPMN.
- Viene con complementos para Confluence y Jira, así que claramente se integra muy bien con estas herramientas. Eso es importante si su equipo los usa y desea integrar sus modelos en el resto de su discusión / proceso de desarrollo.

Gliffy



- GenMyModel comenzó como una herramienta de modelado únicamente para UML, pero desde hace un tiempo se ha expandido para cubrir también el modelado de negocios con soporte de Archimate y BPMN.
- Ofrece un repositorio de modelos centralizado para equipos (con administración de derechos de acceso) que permite una colaboración fácil y simultáneamente de modelos.
- Al contrario de otras herramientas en esta lista, esta es más una herramienta de modelado que una herramienta de dibujo (con sus ventajas y desventajas). Como tal, una distinción clave de GenMyModel es su soporte para exportar modelos como XML (el formato “estándar” para intercambiar modelos) y sus capacidades de generación de código.

GenMyModel



- Destaca por su sencillez. Simplemente accediendo a draw.io en su navegador tienes un lienzo vacío para comenzar a dibujar.
- Viene con formas para modelado UML básico, ER y BPMN.
- Aun así, es un claro ejemplo de una herramienta que no entiende realmente la semántica de lo que estás dibujando, así que básicamente puedes hacer lo que quieras y utilizarlo para construir tus diagramas UML online gratis (aunque no cumplen a rajatabla las normas)
- También falla en el aspecto de colaboración, pero se integra bien con Google Drive, Dropbox, OneDrive y otros para guardar automáticamente los modelos en su ubicación preferida.

Draw.io



- Basado en HTML5.
- Con soporte UML y capacidades de colaboración en tiempo real.
- Puede importar archivos de Visio, lo que es definitivamente una ventaja, ya que una gran base de usuarios para estas herramientas parece provenir de entornos de modelado más similares a los de un escritorio, como esta herramienta.
- Más allá de UML, Lucidchart también viene con plantillas de dibujo para ER, modelado de procesos, diagramas de red y más.

Lucidchart



## 2. Tipos de diagramas UML

### 2.4. Herramientas para construir diagramas UML (online)

- Diagrama de colaboración en tiempo real.
- Con más de 50 tipos de diagramas y miles de ejemplos para comenzar.
- También se puede utilizar para trabajar fuera de línea y sincronizar su trabajo después.
- El único pero que tiene es que, ahora mismo, requiere Flash instalado para funcionar.

Creately



- Su finalidad es crear «un dibujo» y no un modelado.
- Aún así, tiene algunas características interesantes, como la posibilidad de almacenar múltiples versiones de su diagrama para rastrear los cambios y la capacidad de publicar comentarios para enviar comentarios directamente sobre los diagramas.
- Cacao admite los diagramas de casos de uso, secuencia, clase, actividad y máquina de estado.

Cacao



- Una herramienta UML en línea sencilla pero gratuita para crear diagramas UML de forma rápida.
- Se ejecuta en el navegador, y, por lo tanto, no requiere ninguna instalación.
- UMLetino se basa en UMLet (disponible como herramienta independiente o complemento de Eclipse).
- Los diagramas se pueden exportar como archivos XML o de imagen.
- A pesar de sus limitaciones (en comparación con otras herramientas), si estás buscando una herramienta en línea UML gratuita y fácil de usar, esta es definitivamente una opción a considerar.

UMLetino



- Se trata de un software de dibujo en línea HTML5 puro y de código abierto (no se requieren Flash, Java u otros complementos).
- Bajo licencia GPL.
- Fácil de descargar e instalar en tu propio servidor si así lo prefieres.
- El único «pero» es que, como un software centrado en diagramas de flujo, solo cubre un tipo de diagramas UML: máquinas de estado.
- Tampoco dispone de colaboración síncrona.
- Aun así sigue siendo una buena alternativa para construir tus diagramas UML online free.
- Los diagramas que se realizan utilizando esta web solo pueden ser guardados como imágenes.

Diagramo

