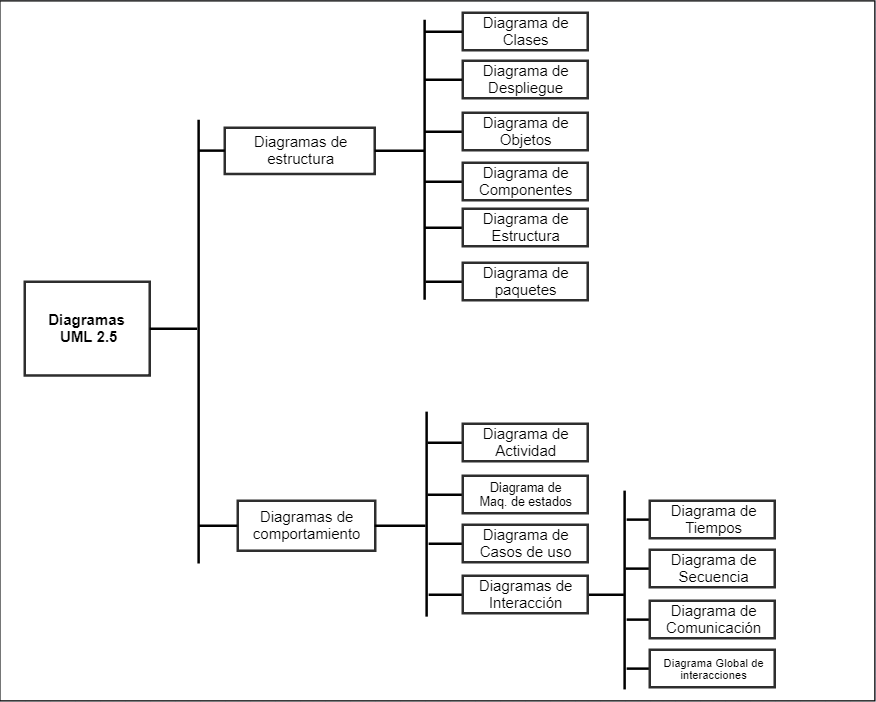
**UML - Lenguaje Unificado de Modelado**

#### Los creadores originales de UML son 3: Jim Rumbaugh, Grady Booch e Ivar Jacobson.

## **versiones de UML**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Description** |
| **1.1** | 11-1997 | UML 1.1 proposal is adopted by the OMG. |
| **1.3** | 03-2000 | Contains a number of changes to the UML metamodel, semantics, and notation, but should be considered a minor upgrade to the original proposal. |
| **1.4** | 09-2001 | Mostly "tuning" release but not completely upward compatible with the UML 1.3. Addition of **[profiles](https://www.uml-diagrams.org/profile.html)** as UML extensions grouped together. Updated **visibility** of features. Stick arrowhead in **interaction diagrams** now denotes **asynchronous call**. Model element may now have multiple **stereotypes**. Clarified collaborations. Refined definitions of components and related concepts. **[Artifact](https://www.uml-diagrams.org/artifact.html)** was added to represent physical representations of components. |
| **1.5** | 03-2003 | Added **[actions](https://www.uml-diagrams.org/activity-diagrams-actions.html)** (see Part 5 of spec) - executable actions and procedures, including their run-time semantics, defined the concept of a data flow to carry data between actions, etc. |
| **1.4.2** | 01-2005 | This version was accepted as ISO specification (standard) ISO/IEC 19501. UML 1.5 was released 2 years before. |
| **2.0** | 08-2005 | New diagrams: object diagrams, **[package diagrams](https://www.uml-diagrams.org/package-diagrams-overview.html)**, [**composite structure diagrams**](https://www.uml-diagrams.org/composite-structure-diagrams.html), interaction overview diagrams, timing diagrams, **[profile diagrams](https://www.uml-diagrams.org/profile-diagrams.html)**. **Collaboration diagrams** were renamed to **[communication diagrams](https://www.uml-diagrams.org/communication-diagrams.html)**.  [**Activity diagrams**](https://www.uml-diagrams.org/activity-diagrams.html) and **[sequence diagrams](https://www.uml-diagrams.org/sequence-diagrams.html)** were enhanced. Activities were redesigned to use a Petri-like semantics. Edges can now be contained in partitions. Partitions can be hierarchical and multidimensional. Explicitly modeled **[object flows](https://www.uml-diagrams.org/activity-diagrams.html" \l "object-flow-edge)** are new.  Classes have been extended with internal structures and **[ports](https://www.uml-diagrams.org/port.html)** (composite structures). Information flows were added. A collaboration now is a kind of classifier, and can have any kind of behavioral descriptions associated. Interactions are now contained within classifiers and not only within collaborations. It is now possible for [**use cases**](https://www.uml-diagrams.org/use-case.html) to be owned by **[classifiers](https://www.uml-diagrams.org/classifier.html)** in general and not just packages.  New notation for concurrency and branching using combined fragments. Notation and/or semantics were updated for components, realization, deployments of artifacts. Components can no longer be directly deployed to **[nodes](https://www.uml-diagrams.org/deployment-diagrams.html" \l "node)**. **[Artifacts](https://www.uml-diagrams.org/artifact.html)** should be deployed instead. Implementation has been replaced by «**[manifest](https://www.uml-diagrams.org/deployment-diagrams.html" \l "manifestation)**». Artifacts can now manifest any packageable element (not just components, as before). It is now possible to deploy to nodes with an internal structure.  New metaclasses were added: connector, collaboration use, connector end, **[device](https://www.uml-diagrams.org/deployment-diagrams.html" \l "device)**, deployment specification, **[execution environment](https://www.uml-diagrams.org/deployment-diagrams.html" \l "execution-environment)**, accept event action, send object action, structural feature action, value pin, activity final, central buffer node, data stores, flow final, interruptible regions, loop nodes, parameter, **[port](https://www.uml-diagrams.org/port.html)**, behavior, behaviored classifier, duration, interval, time constraint, combined fragment, creation event, destruction event, execution event, interaction fragment, interaction use, receive signal event, send signal event, extension, etc.  Many stereotypes were eliminated from the Standard UML Profile, e.g. «destroy», «facade», «friend», «profile», «requirement», «table», «thread».  Integration between structural and behavioral models was improved with better support for executable models. |
| **2.1** | 04-2006 | Minor revision to UML 2.0 - corrections and consistency improvements. |
| **2.1.1** | 02-2007 | Minor revision to the UML 2.1 |
| **2.1.2** | 11-2007 | Minor revision to the UML 2.1.1 |
| **2.2** | 02-2009 | Fixed numerous minor consistency problems and added clarifications to UML 2.1.2 |
| **2.3** | 05-2010 | Minor revision to the UML 2.2, clarified **[associations](https://www.uml-diagrams.org/association.html)** and association classes, added [**final classifier**](https://www.uml-diagrams.org/classifier.html#final-classifier), updated **[component diagrams](https://www.uml-diagrams.org/component-diagrams.html)**, composite structures, actions, etc. |
| **2.4.1** | 08-2011 | UML revision with few fixes and updates to classes, packages - added [**URI package attribute**](https://www.uml-diagrams.org/package-diagrams.html#uri-package); updated actions; removed creation event, execution event, send and receive operation events, send and receive signal events, renamed destruction event to **[destruction occurrence specification](https://www.uml-diagrams.org/sequence-diagrams.html" \l "destruction-occurrence-seq)**; **[profiles](https://www.uml-diagrams.org/profile-diagrams.html)** - changed stereotypes and applied stereotypes to have upper-case first letter - [**«Metaclass»**](https://www.uml-diagrams.org/profile-metaclass.html) and **[stereotype application](https://www.uml-diagrams.org/stereotype.html" \l "stereotype-applied-profile)**. |
| [**2.5**](https://www.uml-diagrams.org/references.html#ref-uml-25) | 06-2015 | UML 2.5 is called a "minor revision" to the UML 2.4.1, while they spent a lot of efforts to simplify and reorganize UML specification document. The UML specification was re-written "*to make it easier to read*". For example, they tried to "*reduce forward references as much as possible*".  There are no longer two separate infrastructure and superstructure documents, the UML 2.5 specification is a single document. **[Package merge](https://www.uml-diagrams.org/package-merge.html)** is no longer used within the specification.  Four UML compliance levels (L0, L1, L2, and L3) are eliminated, as they were not useful in practice. UML 2.5 tools will have to support complete UML specification. **[Information flows](https://www.uml-diagrams.org/information-flow-diagrams.html)**, **[models](https://www.uml-diagrams.org/package-diagrams/model.html)**, and **[templates](https://www.uml-diagrams.org/template.html)** are no longer auxiliary UML constructs. At the same time, [**use cases**](https://www.uml-diagrams.org/use-case-diagrams.html), **[deployments](https://www.uml-diagrams.org/deployment-diagrams-overview.html)**, and the **[information flows](https://www.uml-diagrams.org/information-flow-diagrams.html)** became "**supplementary concepts**" in UML 2.5.  UML 2.5 has a number of fixes, clarifications, and explanations added. They updated description for multiplicity and multiplicity element, clarified definitions of aggregation and composition, and finally fixed wrong «instantiate» dependency example for Car Factory. New notation for **[inherited members](https://www.uml-diagrams.org/inherited-property.html)** with a caret '^' symbol was introduced. UML 2.5 clarified **[feature redefinition](https://www.uml-diagrams.org/redefining-property.html)** and overloading. They also moved and rephrased definition of qualifiers.  Default for **[generalization sets](https://www.uml-diagrams.org/generalization.html" \l "generalization-set)** was changed from **{incomplete, disjoint}** to **{incomplete, overlapping}**.  There are few clarifications and fixes for stereotypes, state machines, and activities. Protocol state machines are now denoted using «protocol» instead of {protocol}. [**Use cases**](https://www.uml-diagrams.org/use-case.html) are no longer required to express some needs of **[actors](https://www.uml-diagrams.org/use-case-actor.html)** and to be initiated by an actor. |
| **2.5.1** | 12-2017 |  |

**Diagramas estructurales y de comportamiento de UML**



Diagramas estructurales

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:

[Diagrama de clases](https://diagramasuml.com/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](https://diagramasuml.com/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](https://diagramasuml.com/despliegue/)

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

[Diagrama de objetos](https://diagramasuml.com/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](https://diagramasuml.com/paquetes/)

Muestra los paquetes y las relaciones entre los paquetes.

[Diagrama de perfiles](https://diagramasuml.com/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](https://diagramasuml.com/estructura-compuesta/)

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

Diagramas de comportamiento

A diferencia de los diagramas estructurales, muestran como 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, clasificados de la siguiente forma:

#### [Diagrama de actividades](https://diagramasuml.com/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](https://diagramasuml.com/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 máquina de estados](https://diagramasuml.com/estados/)

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.

#### ****Diagramas de interacción****.

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

#### [Diagrama de secuencia](https://diagramasuml.com/secuencia/)

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

#### [Diagrama de comunicación](https://diagramasuml.com/comunicacion/)

Se enfoca en la interacción entre líneas de vida donde la arquitectura de la estructura interna y cómo esto se corresponde con el paso del mensaje es fundamental. La secuencia de mensajes se da a través de una numeración.

#### [Diagrama de tiempos](https://diagramasuml.com/diagrama-de-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](https://diagramasuml.com/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.

Diagramas de estructuras: Diagrama de clases

Diagramas de comportamiento: Diagrama de casos de uso, diagramas flowchar y diagrama de actividades.

Diagramas de interacción: Diagrama de secuencia.

Fuente: <https://diagramasuml.com/>