

# Diagramas del UML

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos.

La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como *modelo*. Recordemos que un *modelo* es una representación simplificada de la realidad; el modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema.

A continuación se describirán los diagramas más comunes del UML y los conceptos que representan:

- [Diagrama de Clases](#)
- [Diagrama de Objetos](#)
- [Diagrama de Casos de Uso](#)
- [Diagrama de Estados](#)
- [Diagrama de Secuencias](#)
- [Diagrama de Actividades](#)
- [Diagrama de Colaboraciones](#)
- [Diagrama de Componentes](#)
- [Diagrama de Distribución](#)
- [Otras características](#)
  - [Paquetes](#)
  - [Notas](#)
  - [Estereotipos](#)

## Diagrama de Clases

[Volver](#)

Los *diagramas de clases* describen la estructura estática de un sistema.

Las cosas que existen y que nos rodean se agrupan naturalmente en categorías. Una *clase* es una categoría o grupo de cosas que tienen *atributos* (propiedades) y *acciones* similares. Un ejemplo puede ser la *clase* "Aviones" que tiene *atributos* como el "modelo de avión", "la cantidad de motores", "la velocidad de crucero" y "la capacidad de carga útil". Entre las acciones de las cosas de esta clase se encuentran: "acelerar", "elevarse", "girar", "descender", "desacelerar".

Un rectángulo es el símbolo que representa a la *clase*, y se divide en tres áreas. Un *diagrama de clases* está formado por varios rectángulos de este tipo conectados por líneas que representan las *asociaciones* o maneras en que las clases se relacionan entre sí.

Nombre de Clase
atributo: Tipo / atributo Derivado
operación( )

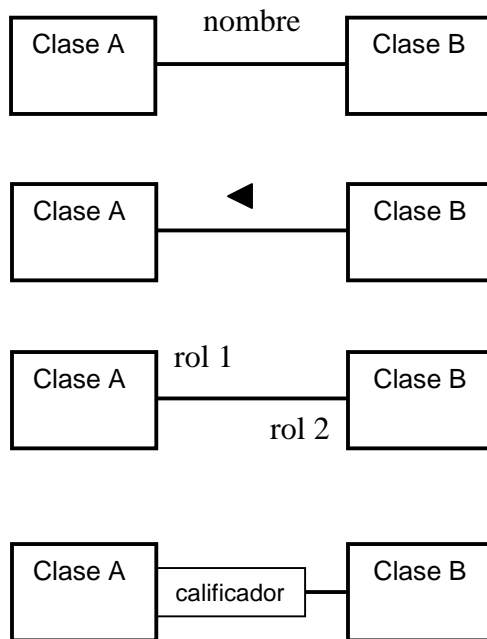
### Clase Abstracta

Las *clases* se representan con rectángulos divididos en tres áreas: la superior contiene el nombre de la clase, la central contiene los *atributos* y la inferior las *acciones*.

Aviones
modelo de avión cantidad de motores velocidad de crucero carga útil
acelerar ( ) elevarse ( ) girar ( ) descender ( ) desacelerar ( )

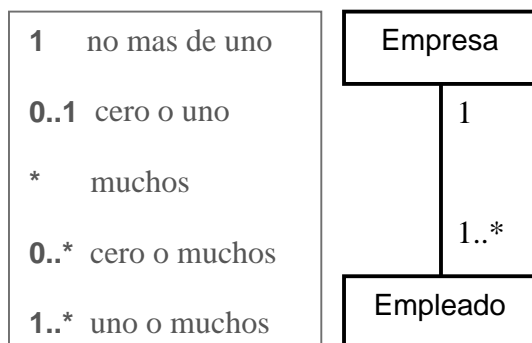
### Clase Aviones

En el área superior figura el nombre de la clase que utilizamos como ejemplo, en la central están sus atributos y en la inferior las acciones que ella realiza. Note que las acciones llevan paréntesis al final del nombre dado que las mismas son funciones y por lo tanto devuelven un valor.



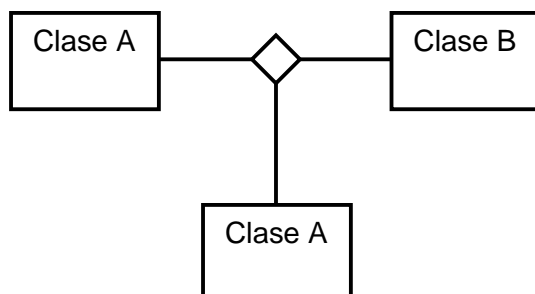
### Asociaciones

Las *asociaciones* son las que representan a las relaciones estáticas entre las clases. El nombre de la *asociación* va por sobre o por debajo de la línea que la representa. Una flecha rellena indica la dirección de la relación. Los *roles* se ubican cerca del final de una *asociación*. Los *roles* representan la manera en que dos *clases* se ven entre ellas. No es común el colocar ambos nombres, el de la asociación y el de los roles a la vez. Cuando una asociación es *calificada*, el símbolo correspondiente se coloca al final de la asociación, contra la clase que hace de calificador.

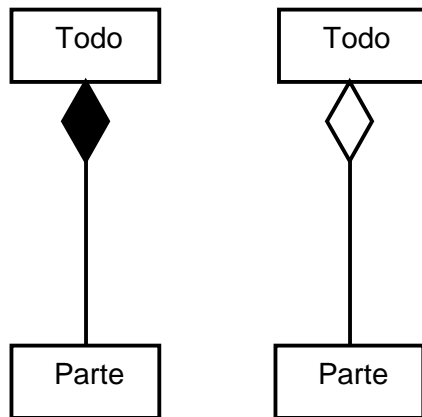


### Multiplicidad

Las notaciones utilizadas para señalar la *multiplicidad* se colocan cerca del final de una *asociación*. Estos símbolos indican el número de instancias de una clase vinculadas a una de las instancias de la otra clase. Por ejemplo, una empresa puede tener uno o más empleados, pero cada empleado trabaja para una sola empresa solamente.



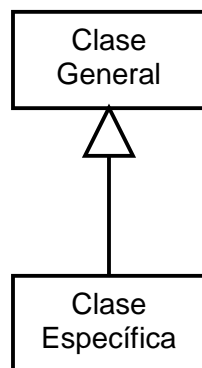
### Asociación Tripartita



### Composición y Agregación

*Composición* es un tipo especial de *agregación* que denota una fuerte posesión de la Clase "Todo", a la Clase "Parte". Se grafica con un rombo diamante relleno contra la clase que representa el todo.

La *agregación* es una relación en la que la Clase "Todo" juega un *rol* más importante que la Clase "Parte", pero las dos clases no son dependientes una de otra. Se grafica con un rombo diamante vacío contra la Clase "Todo".



### Generalización

*Generalización* es otro nombre para *herencia*. Se refiere a una relación entre dos clases en donde una Clase "Específica" es una versión especializada de la otra, o Clase "General". Por ejemplo, Honda es un tipo de auto, por lo que la Clase "Honda" va a tener una relación de *generalización* con la Clase "Auto".

## Diagrama de Objetos

[Volver](#)

Los *Diagramas de Objetos* están vinculados con los *Diagramas de Clases*. Un objeto es una instancia de una clase, por lo que un *diagrama de objetos* puede ser visto como una instancia de un *diagrama de clases*. Los *diagramas de objetos* describen la estructura estática de un sistema en un momento particular y son usados para probar la precisión de los *diagramas de clases*.

Nombre Objeto : Clase

### Nombre de los objetos

Cada *objeto* es representado como un rectángulo, que contiene el nombre del *objeto* y su *clase* subrayadas y separadas por dos puntos.

Nombre Objeto : Clase

Atributo tipo = 'Valor'  
Atributo tipo = 'Valor'  
Atributo tipo = 'Valor'  
Atributo tipo = 'Valor'

### Atributos

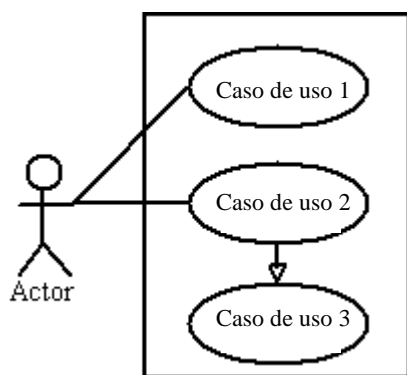
Como con las *clases*, los atributos se listan en un área inferior. Sin embargo, los atributos de los *objetos* deben tener un valor asignado.

## Diagrama de Casos de Uso

[Volver](#)

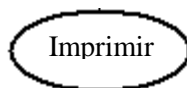
Un *caso de uso* es una descripción de las acciones de un sistema desde el punto de vista del usuario. Es una herramienta valiosa dado que es una técnica de aciertos y errores para obtener los requerimientos del sistema, justamente desde el punto de vista del usuario.

Los *diagramas de caso de uso* modelan la funcionalidad del sistema usando *actores* y *casos de uso*. Los *casos de uso* son servicios o funciones provistas por el sistema para sus usuarios.



### Sistema

El rectángulo representa los límites del sistema que contiene los *casos de uso*. Los *actores* se ubican fuera de los límites del sistema.



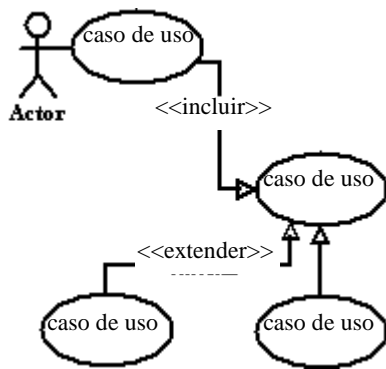
### Casos de Uso

Se representan con óvalos. La etiqueta en el óvalo indica la función del sistema.



### Actores

Los *actores* son los usuarios de un sistema.



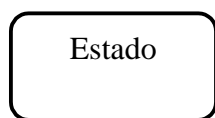
## Relaciones

Las relaciones entre un *actor* y un *caso de uso*, se dibujan con una línea simple. Para relaciones entre *casos de uso*, se utilizan flechas etiquetadas "incluir" o "extender." Una relación "incluir" indica que un *caso de uso* es necesitado por otro para poder cumplir una tarea. Una relación "extender" indica opciones alternativas para un cierto *caso de uso*.

## Diagrama de Estados

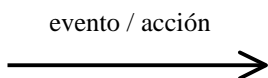
[Volver](#)

En cualquier momento, un *objeto* se encuentra en un *estado* particular, la luz está encendida o apagada, el auto en movimiento o detenido, la persona leyendo o cantando, etc. . El *diagrama de estados* UML captura esa pequeña realidad.



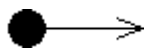
### Estado

El *estado* representa situaciones durante la vida de un *objeto*. Se representa con un rectángulo que tiene sus esquinas redondeadas.



### Transición

Una flecha representa el pasaje entre diferentes *estados* de un *objeto*. Se etiqueta con el evento que lo provoca y con la acción resultante.



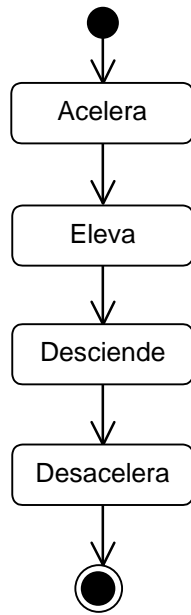
### Estado Inicial



### Estado Final



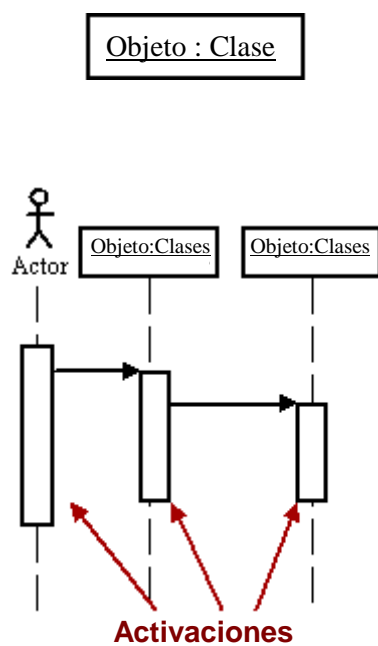
## Ejemplo de Diagrama de Estado



## Diagrama de Secuencias

[Volver](#)

Los *diagramas de clases* y los de *objetos* representan información estática. No obstante, en un sistema funcional, los *objetos* interactúan entre sí, y tales interacciones suceden con el tiempo. El *diagrama de secuencias* UML muestra la mecánica de la interacción con base en tiempos.

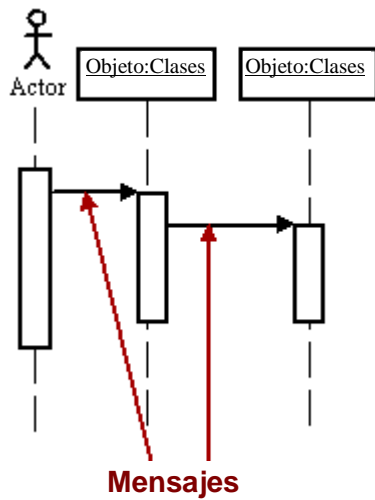


### Rol de la Clase

El *rol* de la *clase* describe la manera en que un *objeto* se va a comportar en el contexto. No se listan los atributos del *objeto*.

### Activación

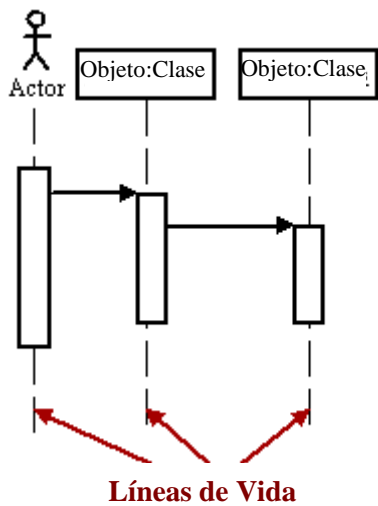
Los cuadros de *activación* representan el tiempo que un *objeto* necesita para completar una tarea.



### Mensajes

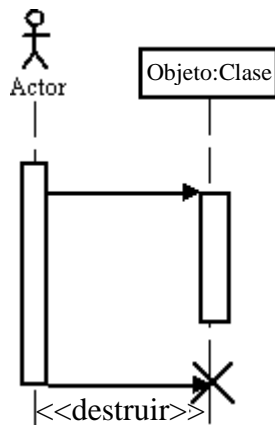
Los *mensajes* son flechas que representan comunicaciones entre *objetos*. Las medias flechas representan *mensajes* asincrónicos. Los *mensajes* asincrónicos son enviados desde un *objeto* que no va a esperar una respuesta del receptor para continuar con sus tareas.

Flecha	Tipo de mensaje
	Simple
	Sincrónico
	Asincrónico
	Rechazado
	Time out



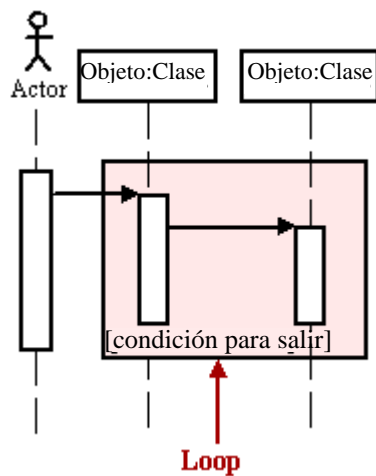
### Líneas de Vida

Las *líneas de vida* son verticales y en línea de puntos, ellas indican la presencia del *objeto* durante el tiempo.



### Destrucción de Objetos

Los *objetos* pueden ser eliminados tempranamente usando una flecha etiquetada "<<destruir>>" que apunta a una X.



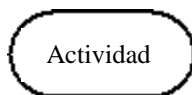
### Loops

Una repetición o *loop* en un *diagrama de secuencias*, es representado como un rectángulo. La condición para abandonar el *loop* se coloca en la parte inferior entre corchetes [ ].

## Diagrama de Actividades

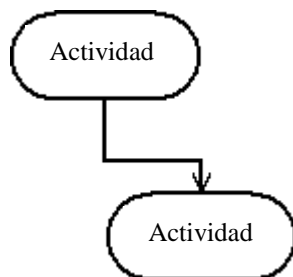
[Volver](#)

Un *diagrama de actividades* ilustra la naturaleza dinámica de un sistema mediante el modelado del flujo ocurrente de *actividad* en *actividad*. Una *actividad* representa una operación en alguna *clase* del sistema y que resulta en un cambio en el *estado* del sistema. Típicamente, los *diagramas de actividad* son utilizados para modelar el flujo de trabajo interno de una operación.



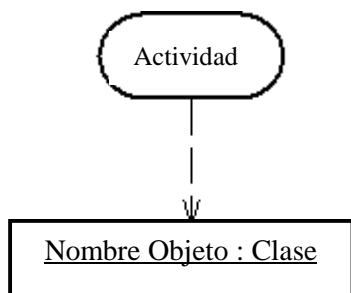
### Estados de Acción

Los *estados de acción* representan las acciones no interrumpidas de los *objetos*.



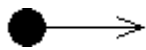
### Flujo de la Acción

Los *flujos de acción*, representados con flechas, ilustran las relaciones entre los *estados de acción*.



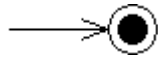
### Flujo de Objetos

El *flujo de objetos* se refiere a la creación y modificación de *objetos* por parte de *actividades*. Una flecha de *flujo de objeto*, desde una *acción* a un *objeto*, significa que la *acción* está creando o influyendo sobre dicho *objeto*. Una flecha de *flujo de objeto*, desde un *objeto* a una *acción*, indica que el estado de *acción* utiliza dicho *objeto*.



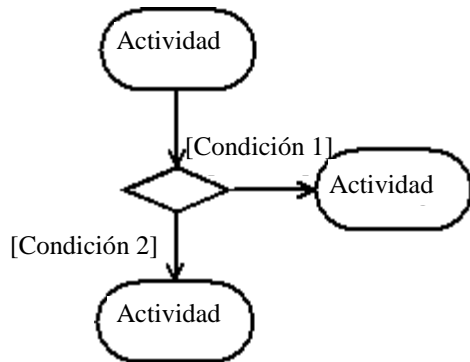
### Estado Inicial

Estado inicial de un estado de *acción*.



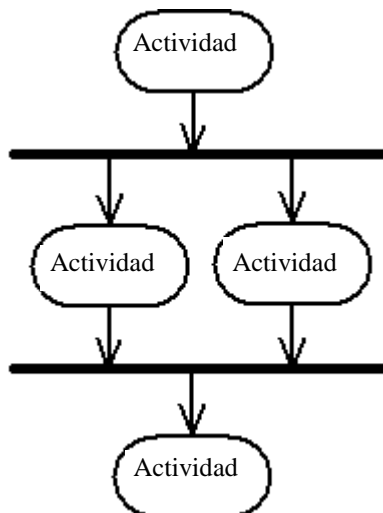
### Final State

Estado final de un estado de *acción*.



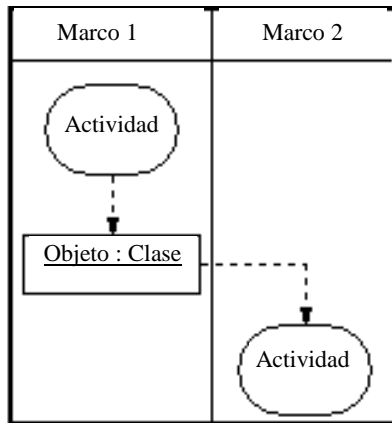
### Ramificación

Un rombo representa una decisión con caminos alternativos. Las salidas alternativas deben estar etiquetadas con una condición.



### Sincronización

Una barra de *sincronización* ayuda a ilustrar la ocurrencia de transiciones paralelas, así quedan representadas las *acciones* concurrentes.



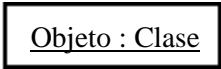
### Marcos de Responsabilidad

Los *marcos de responsabilidad* agrupan a las *actividades* relacionadas en una misma columna.

## Diagrama de Colaboraciones

[Volver](#)

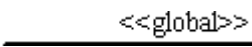
El *diagrama de colaboraciones* describe las interacciones entre los *objetos* en términos de mensajes secuenciados. Los *diagramas de colaboración* representan una combinación de información tomada de los *diagramas de clases*, de *secuencias* y de *casos de uso*, describiendo el comportamiento, tanto de la estructura estática, como de la estructura dinámica de un sistema.



Objeto : Clase

### Rol de la Clase

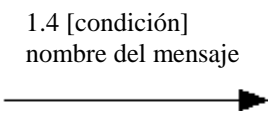
El *rol* de la *clase* describe cómo se comporta un *objeto*. Los atributos del *objeto* no se listan.



<<global>>

### Rol de las Asociaciones

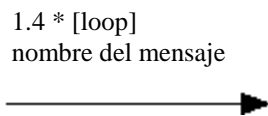
Los *roles de asociación* describen cómo se va a comportar una *asociación* en una situación particular. Se usan líneas simple etiquetadas con un estereotipo\*. (ver al final del documento)



1.4 [condición]  
nombre del mensaje

### Mensajes

Contrariamente a los *diagramas de secuencias*, los *diagramas de colaboración* no tienen una manera explícita para denotar el tiempo, por lo que entonces numeran a los mensajes en orden de ejecución. La numeración puede anidarse; por ejemplo, para mensajes anidados al mensaje número 1: 1.1, 1.2, 1.3, etc. . La condición para un mensaje se suele colocar entre corchetes. Para indicar un loop se usa \* después de la numeración.

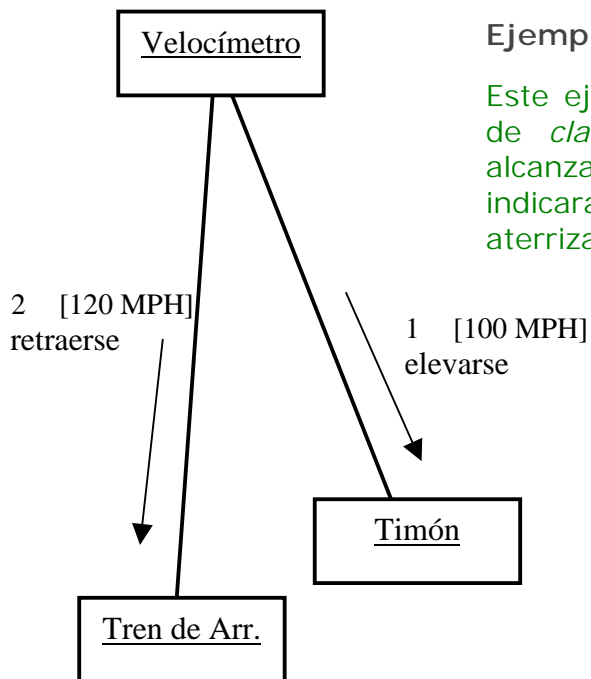


1.4 \* [loop]  
nombre del mensaje



### Ejemplo de Diagrama de Colaboración

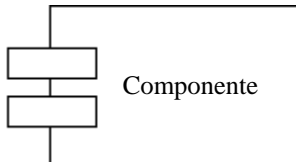
Este ejemplo agrega un velocímetro al conjunto de *clases* que constituyen a un "Avión". Al alcanzar una cierta velocidad el velocímetro indicará al timón que debe elevarse y al tren de aterrizaje que debe retraerse.



## Diagrama de Componentes

[Volver](#)

Un *diagrama de componentes* describe la organización de los componentes físicos de un sistema.



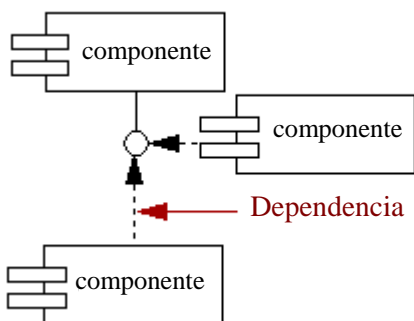
### Componente

Un *componente* es un bloque de construcción física del sistema.



### Interfase

Una *interfase* describe a un grupo de operaciones usada o creada por *componentes*.



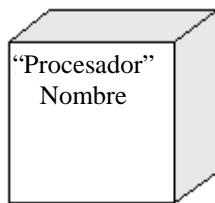
### Dependencias

Las *dependencias* entre *componentes* se grafican usando flechas de puntos.

## Diagrama de Distribución

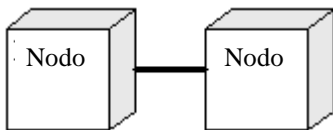
[Volver](#)

El *diagrama de distribución* UML muestra la arquitectura física de un sistema informático. Puede representar a los equipos y a los dispositivos, y también mostrar sus interconexiones y el software que se encontrará en cada máquina.



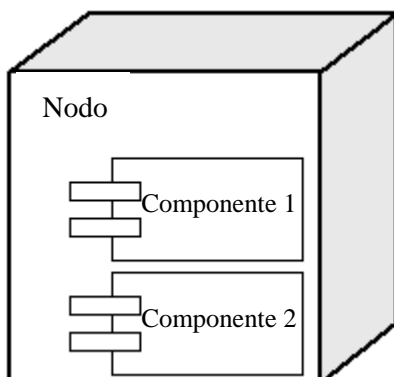
### Nodo

Un *nodo* es un recurso físico capaz de ejecutar *componentes* de código. (Procesador)



### Asociación

La *asociación* se refiere a la conexión física entre los nodos, como por ejemplo Ethernet.



### Componentes y Nodos

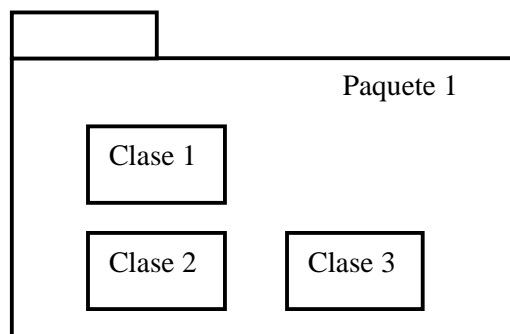
## Otras características

[Volver](#)

### Paquetes

[Volver](#)

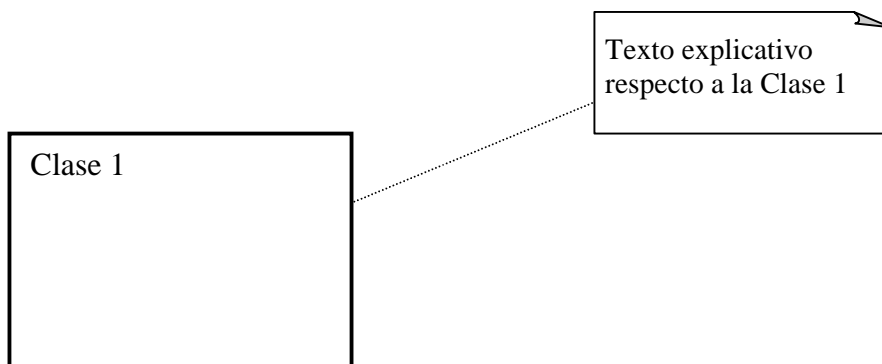
En algunas ocasiones se encontrará con la necesidad de organizar los elementos de un diagrama en un grupo. Tal vez quiera mostrar que ciertas *clases* o *componentes* son parte de un subsistema en particular. Para ello, se pueden agrupar en un *paquete*, que se representa por una carpeta tabular.



### Notas

[Volver](#)

Es frecuente que alguna parte del diagrama no presente una clara explicación del porqué está allí o la manera en que trabaja. Cuando éste sea el caso, la nota UML será útil. La nota tiene una esquina doblada y se adjunta al elemento del diagrama conectándolo mediante una línea punteada.



**Estereotipos\***[Volver](#)

Algunos sistemas requieren de elementos hechos a medida que no se encuentran en el UML. Para ello, los *estereotipos* o *clisés* le permiten tomar elementos propios del UML y convertirlos en otros que se ajusten a las necesidades. Se representan como un nombre entre dos pares de paréntesis angulares.

<<nombre>>