

A faded, grayscale image of a man's face, likely a portrait, serves as the background for the slide. The man has dark hair and is looking directly at the camera.

UML

Contenidos

- ▶ Introducción a UML

Introducción a UML

El Lenguaje de Modelado Unificado (UML) es un lenguaje de diseño para modelar software mediante diferentes tipos de diagramas.

Es un lenguaje de modelado visual que permite mediante representaciones, para los proyectos de software:

- **Visualizar:** permite mediante representaciones gráficas construir modelos que representan el dominio del problema y la solución.
- **Especificar:** permite especificar mediante el modelo estático la estructura y mediante el modelo dinámico el comportamiento.
- **Construir:** permite mediante el uso de herramientas CASE la generación de código en base a los modelos planteados.
- **Documentar:** permite gestionar desde los requerimientos, todos los conjuntos de diagramas, códigos, manuales, etc.

Introducción a UML

Los principales diagramas son:

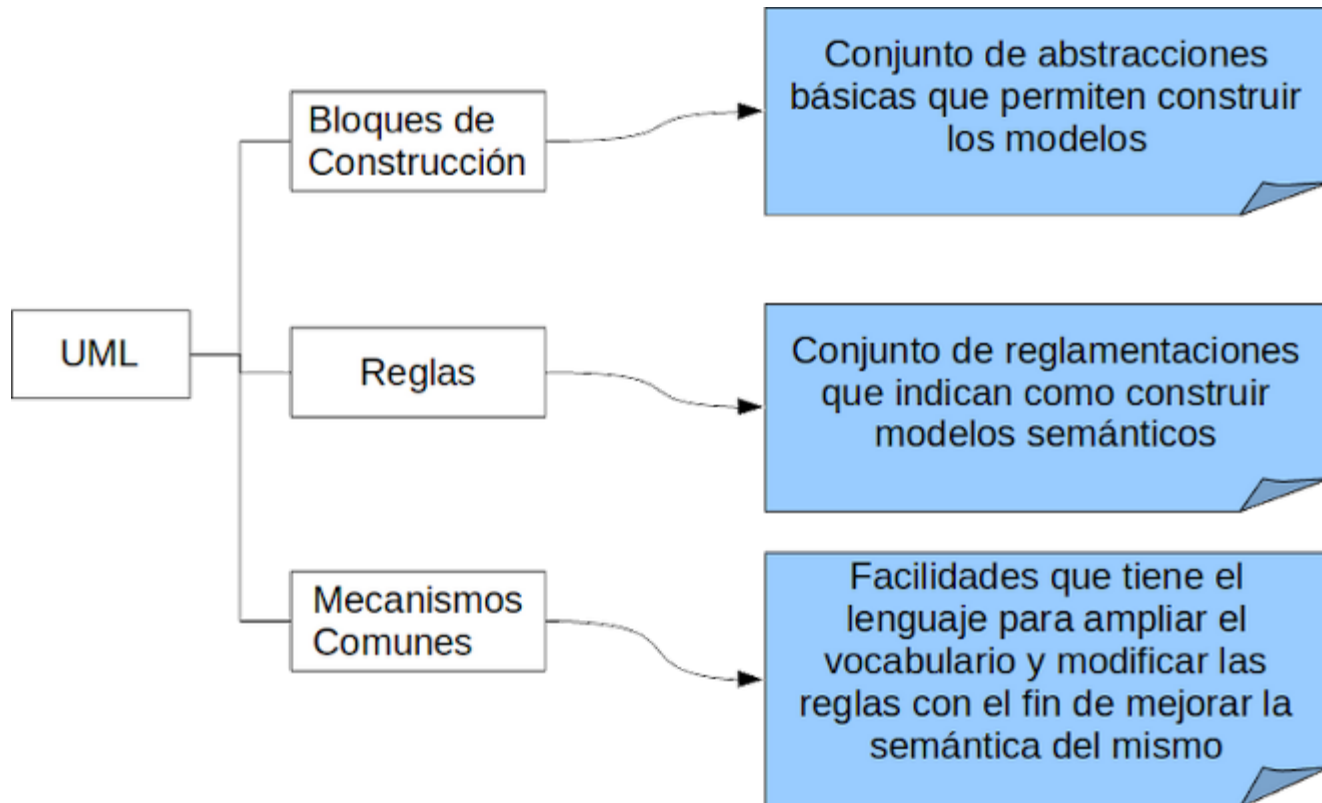
- **Diagrama de clases:** proporciona elementos para modelar las clases (con sus atributos y métodos) y las relaciones y asociaciones entre ellas.
- **Diagrama de implementación:** permite mostrar la distribución física de un sistema software en las plataformas de hardware y entornos de ejecución.
- **Diagramas de casos de uso:** representan la interacción del usuario con las diferentes funciones del sistema.
- **Diagramas de secuencia:** muestra la comunicación dinámica entre los diferentes objetos para llevar a cabo una tarea, describiendo el orden de ejecución.

Introducción a UML

- **Diagramas de comunicación:** representan como los diagramas de secuencia la comunicación entre los objetos para realizar una tarea, enfatizando en las relaciones entre las clases y objetos.
- **Diagramas de actividad:** muestran el comportamiento dinámico del sistema mediante el flujo de control de las acciones que debe realizar.
- **Diagramas de estado:** modelan los estados por los que pasa un objeto y las acciones que desencadenan la transición de estados.

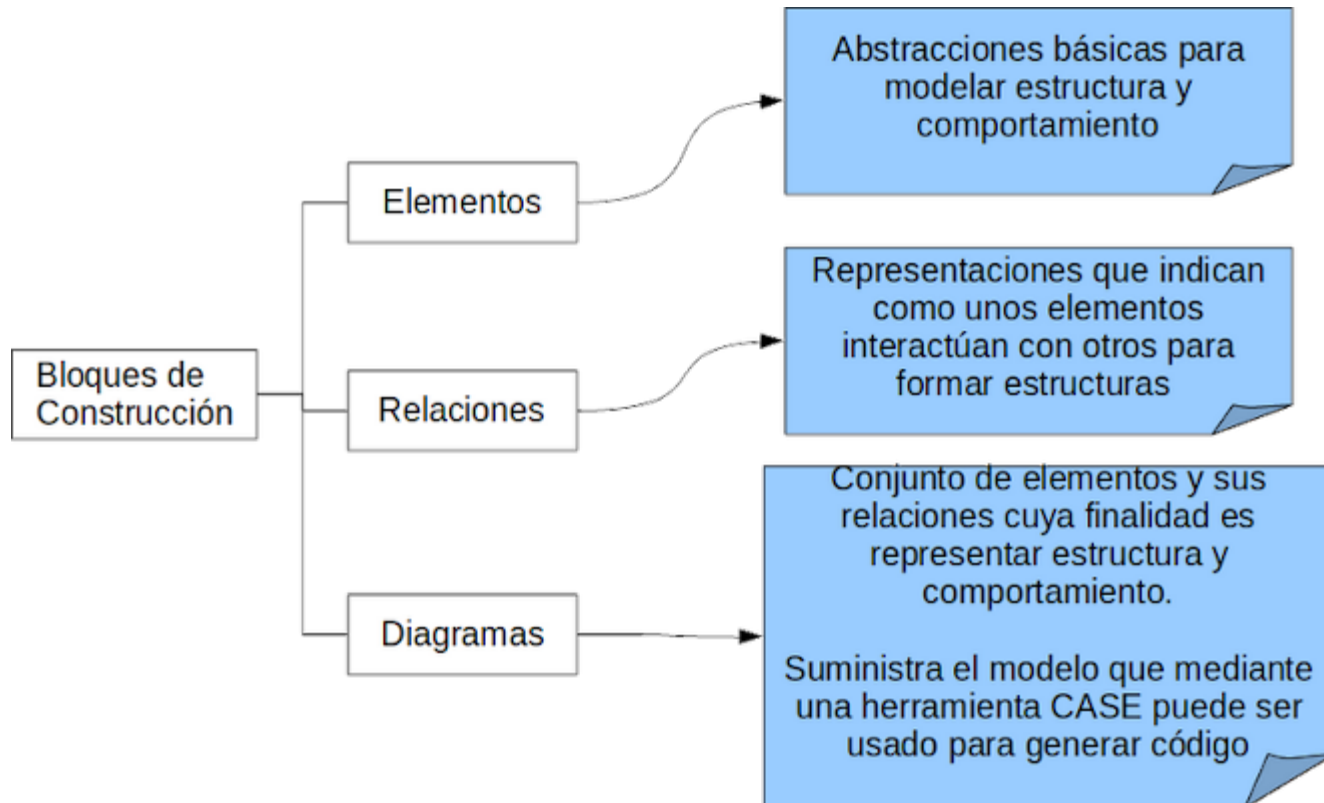
Introducción a UML

Modelo conceptual:



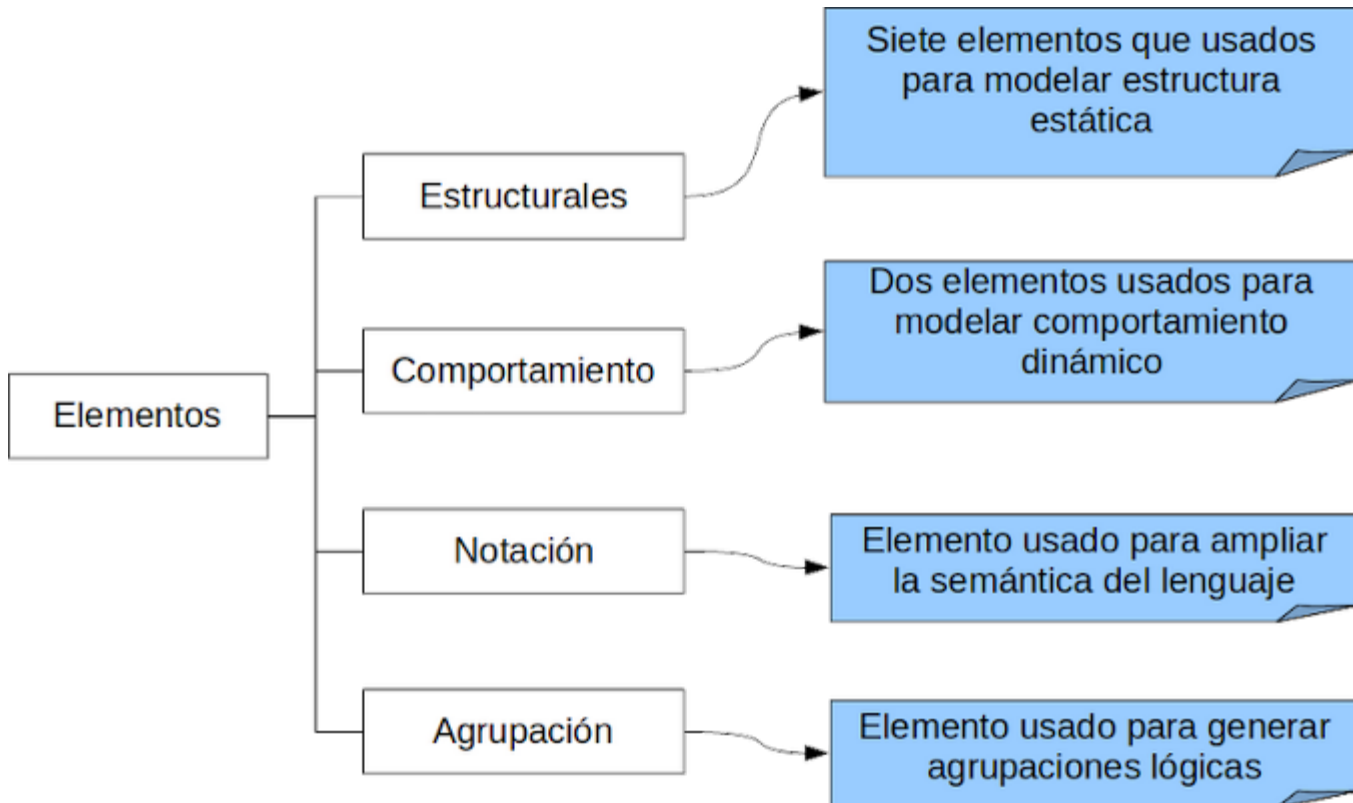
Introducción a UML

Modelo conceptual:



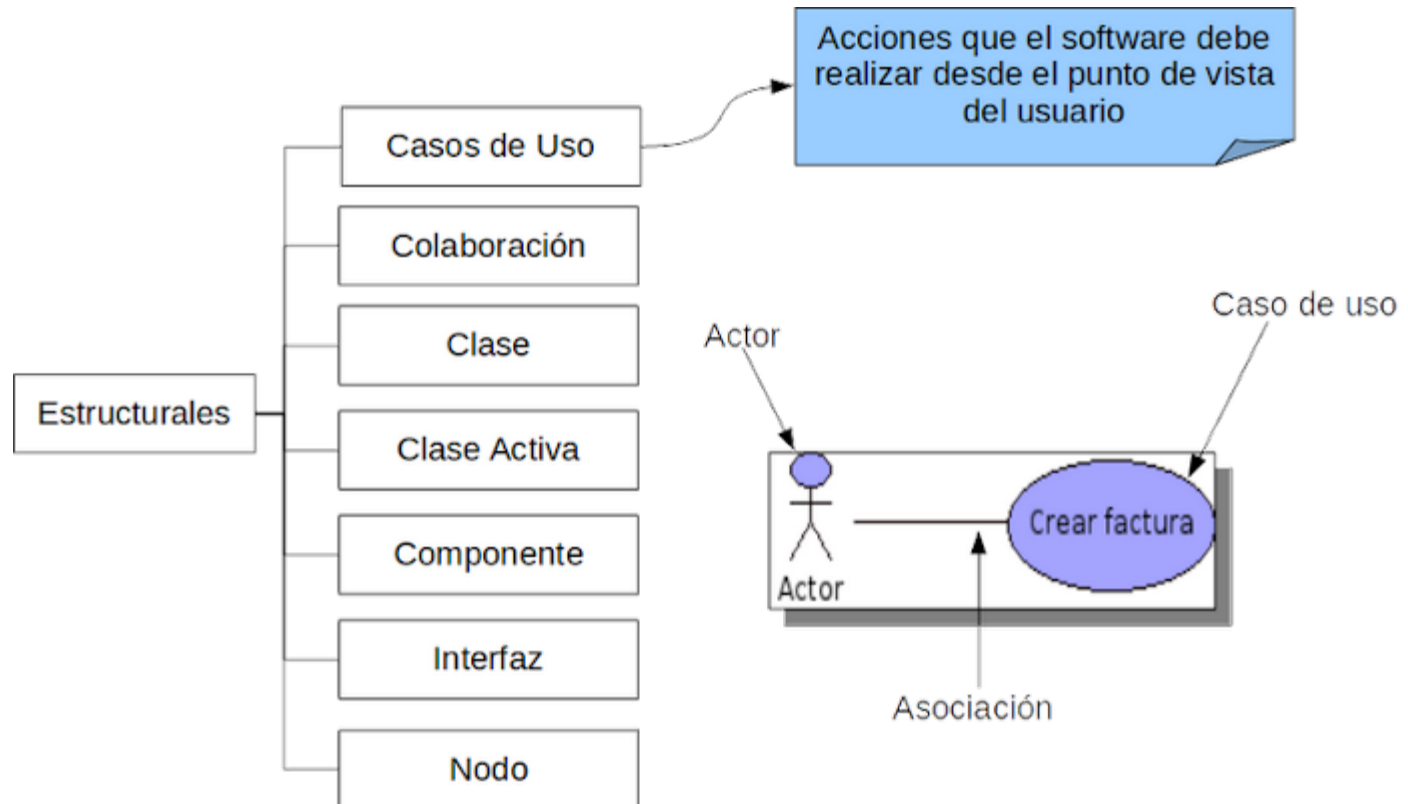
Introducción a UML

Modelo conceptual:



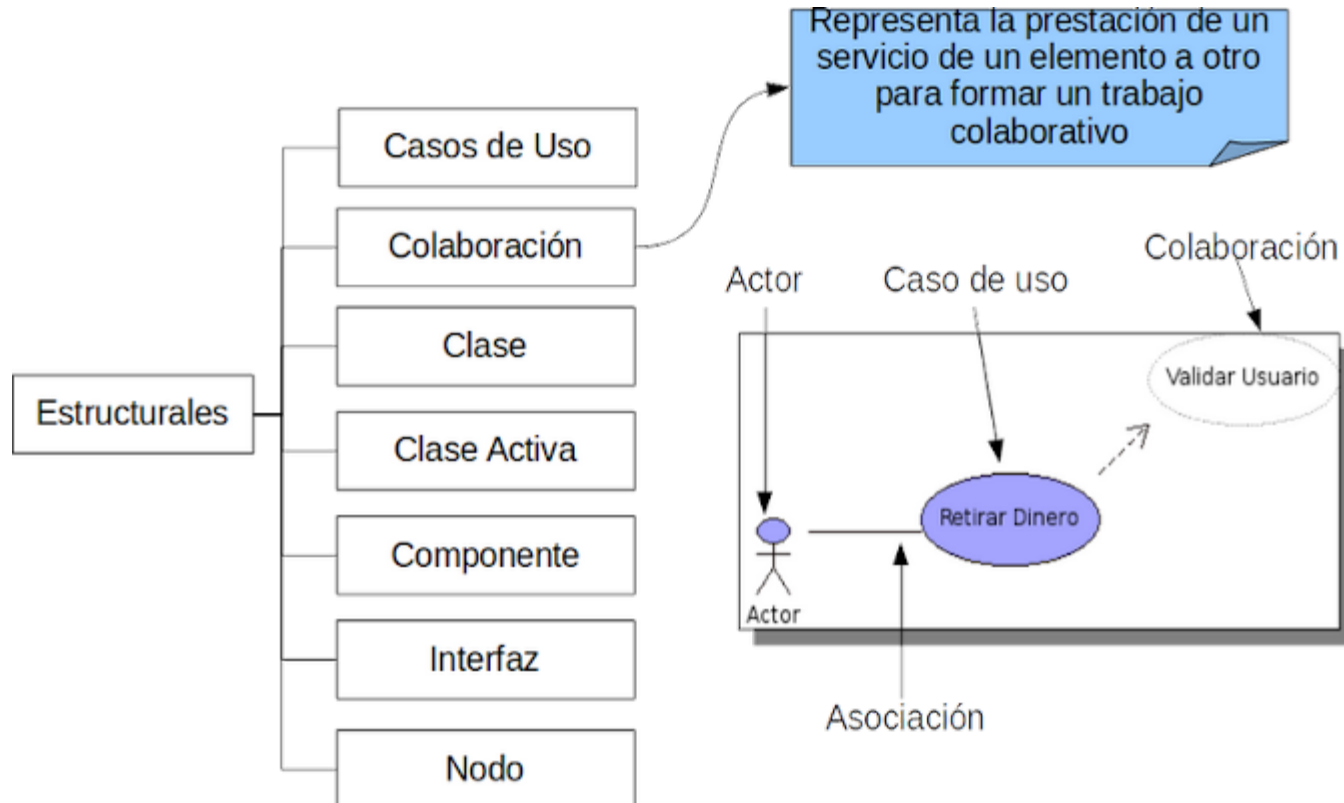
Introducción a UML

Modelo conceptual:



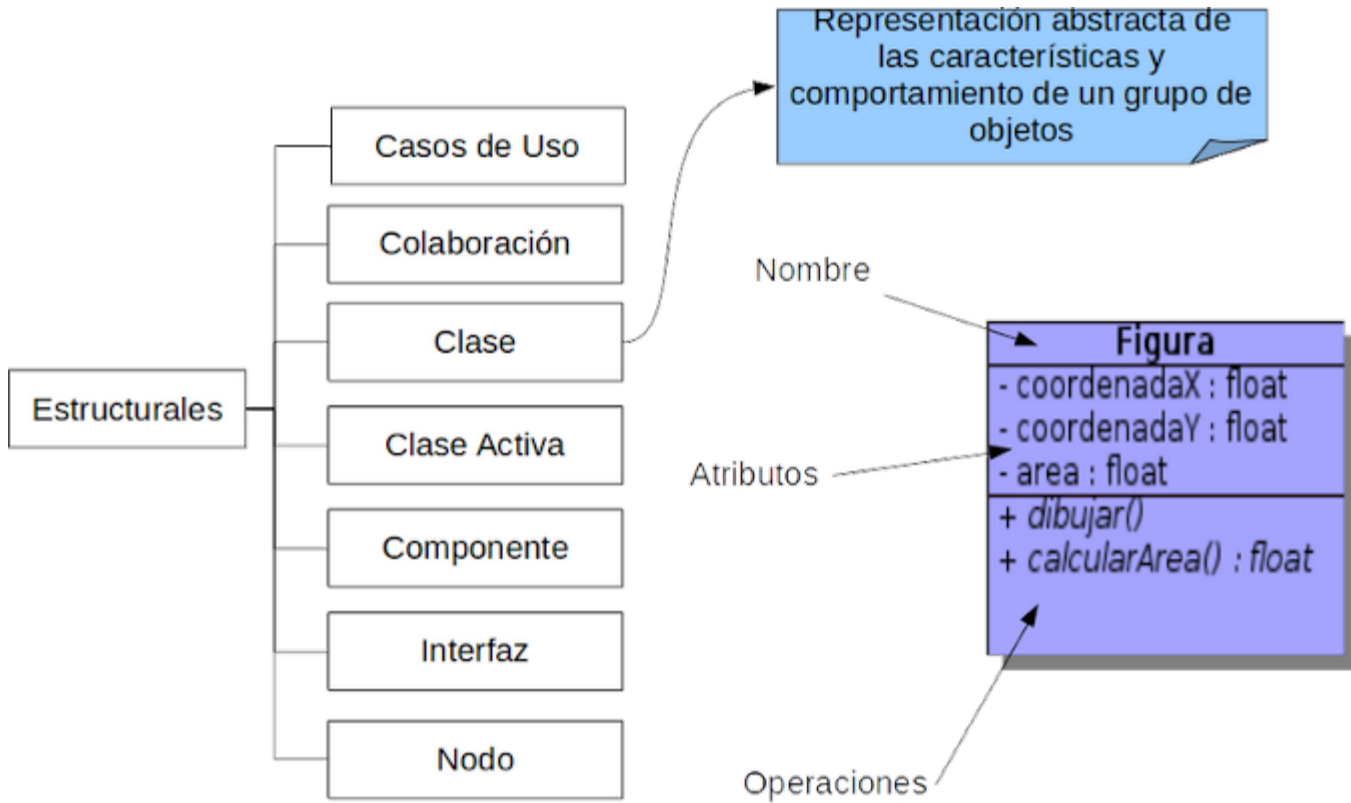
Introducción a UML

Modelo conceptual:



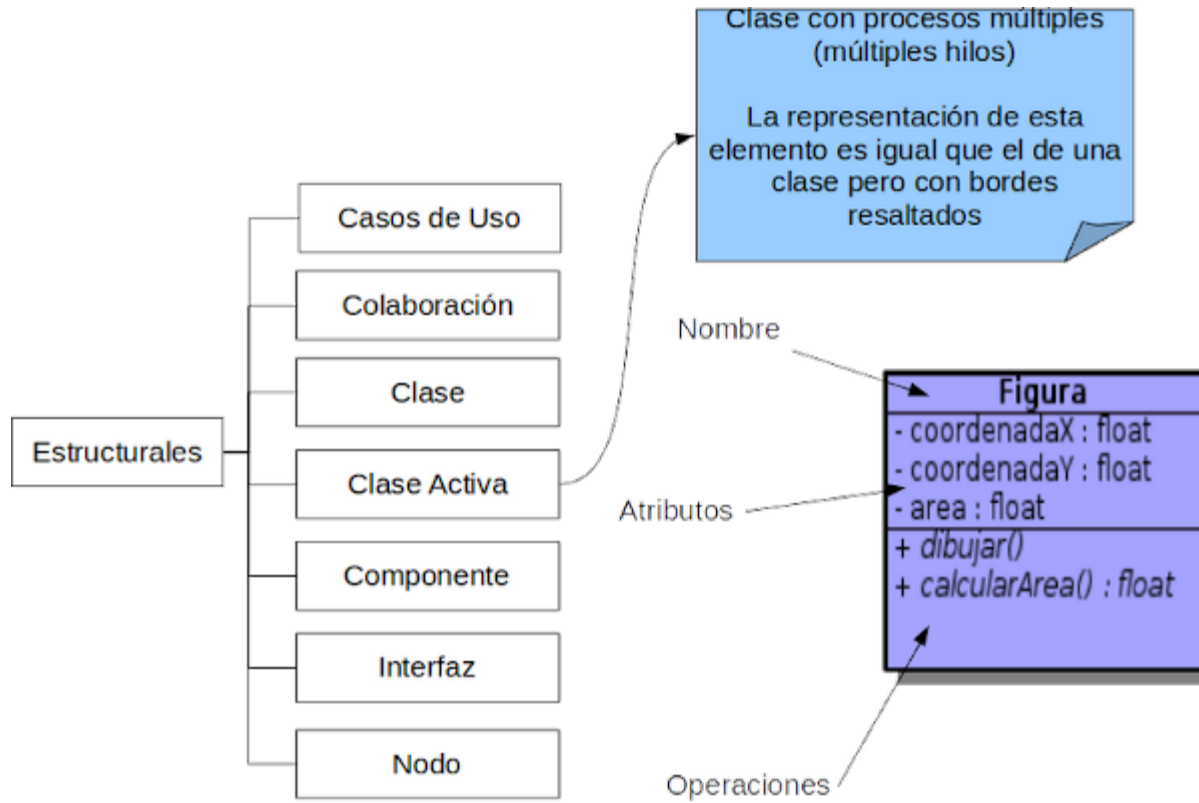
Introducción a UML

Modelo conceptual:



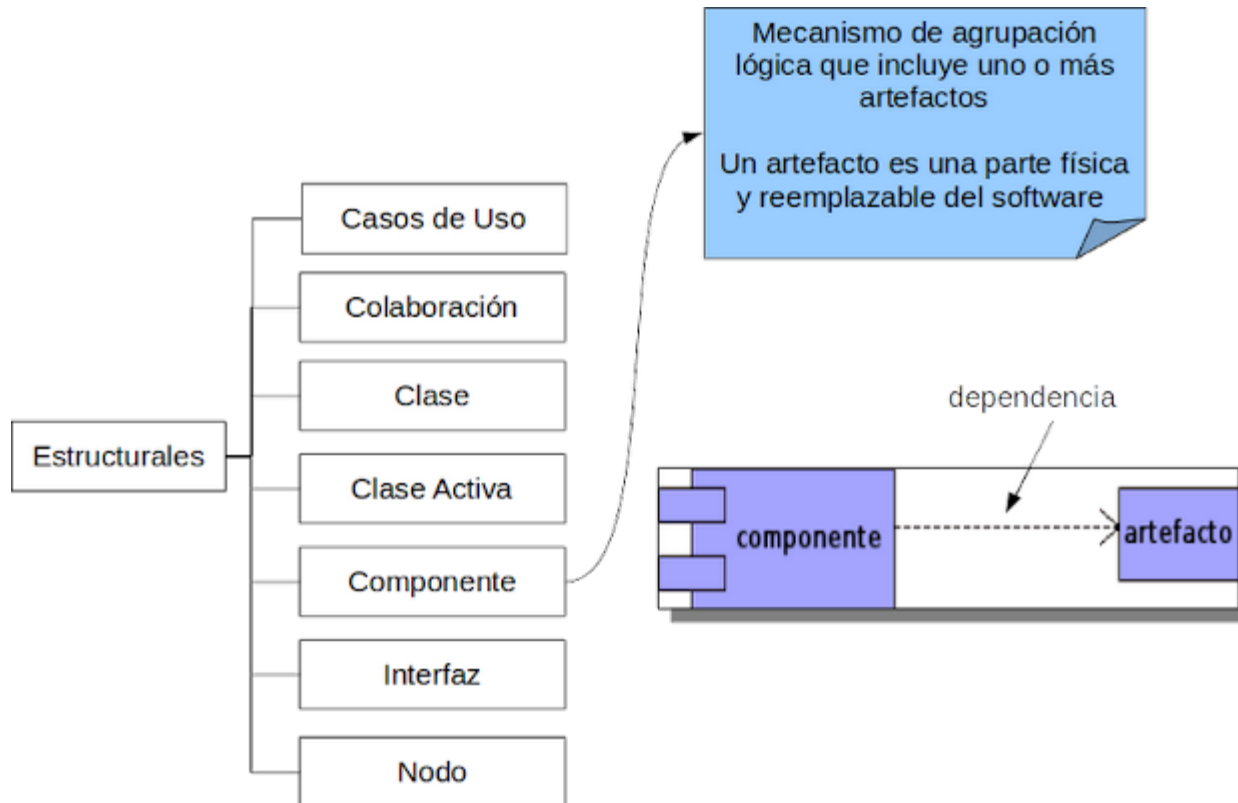
Introducción a UML

Modelo conceptual:



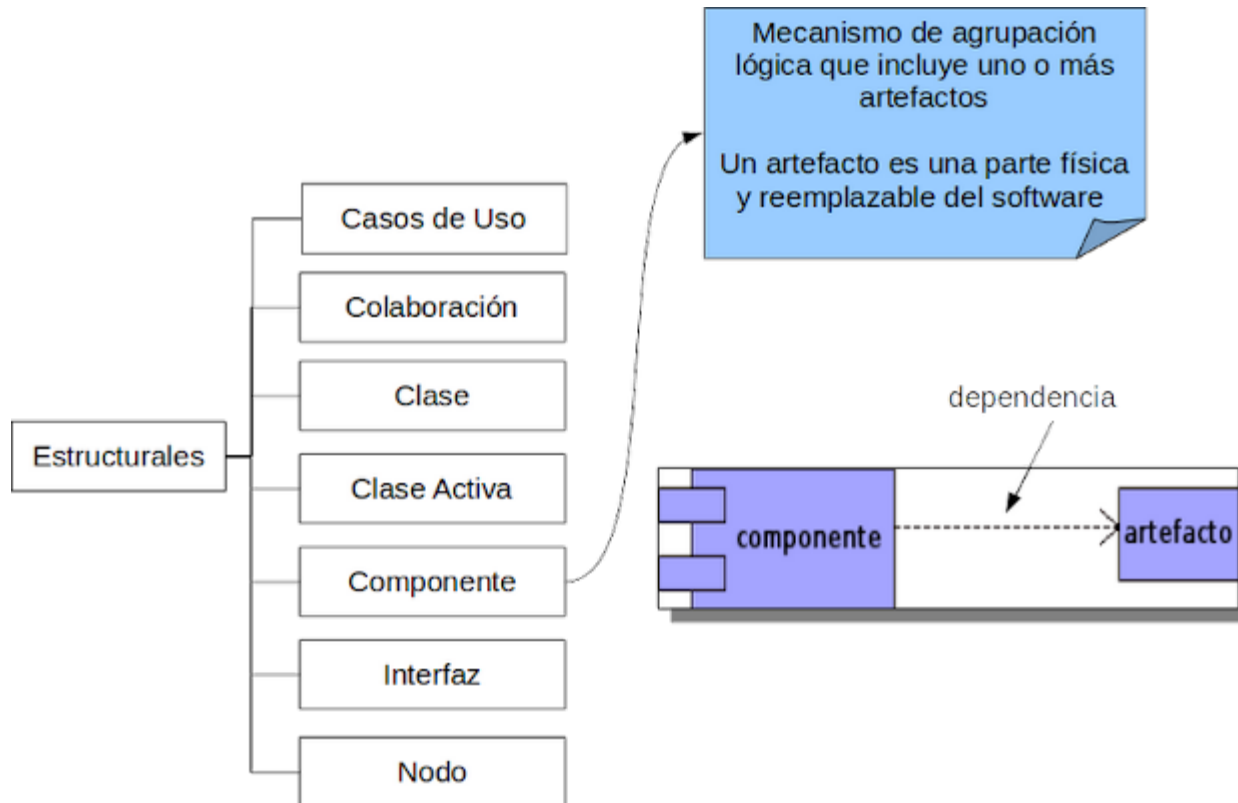
Introducción a UML

Modelo conceptual:



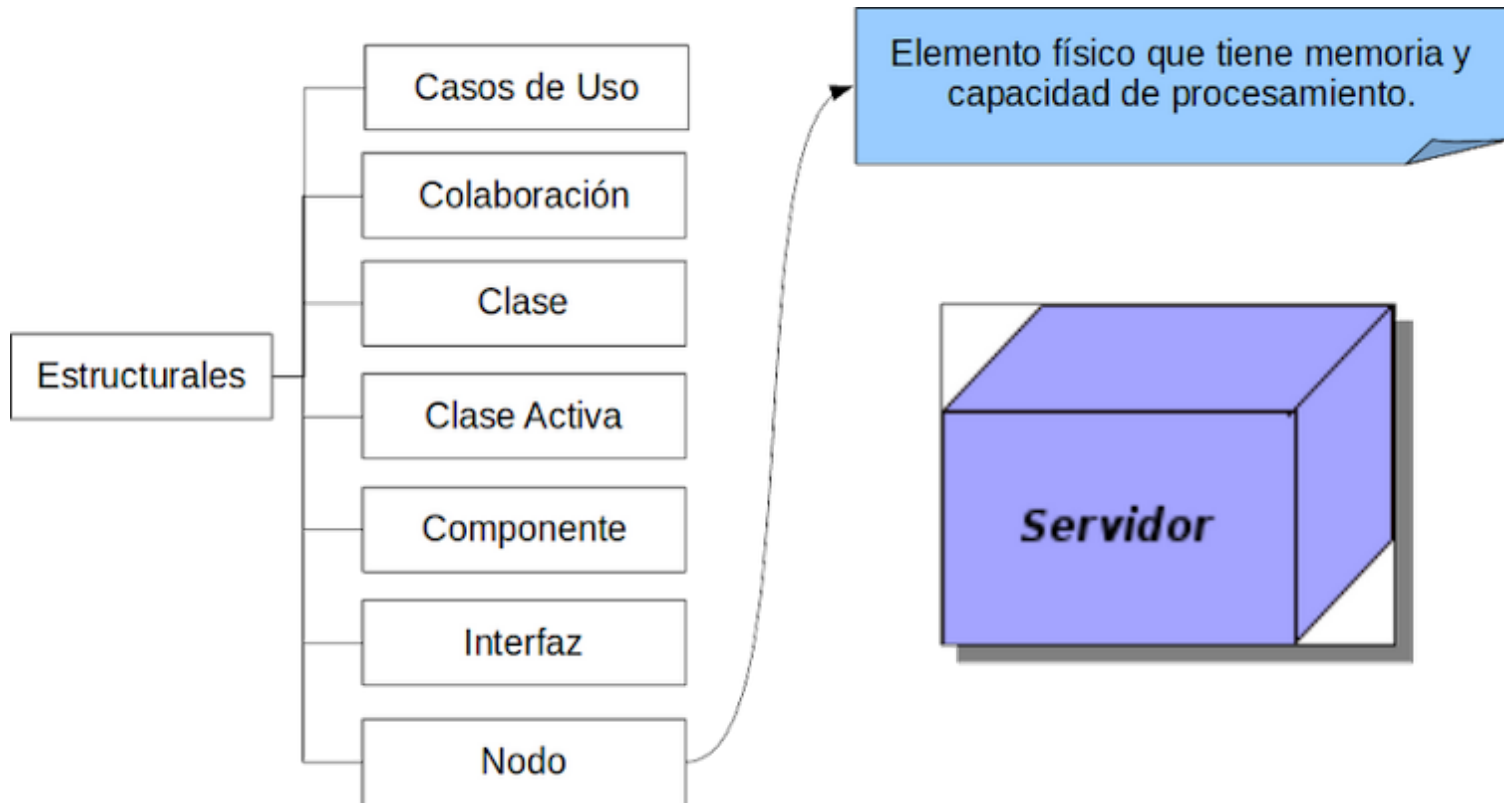
Introducción a UML

Modelo conceptual:



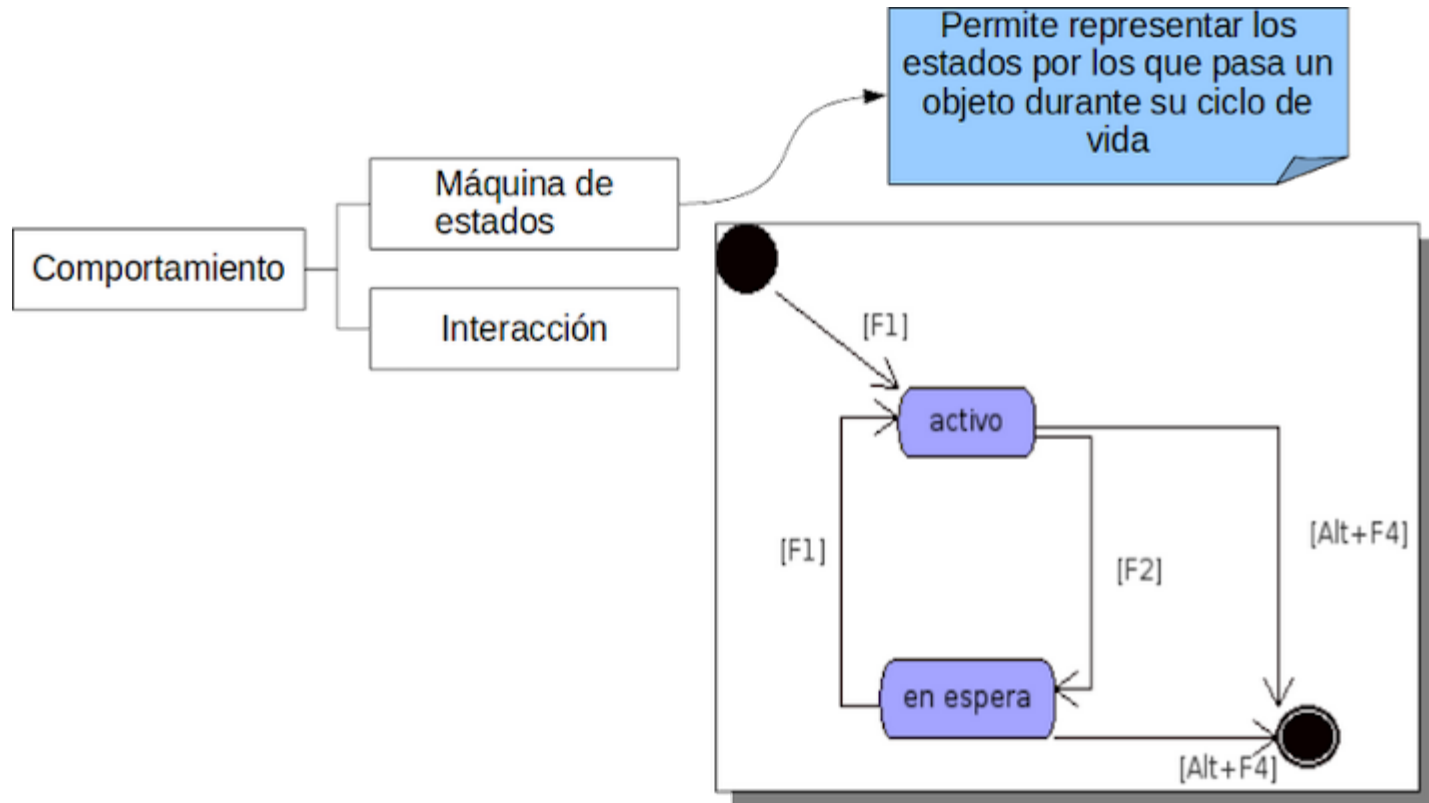
Introducción a UML

Modelo conceptual:



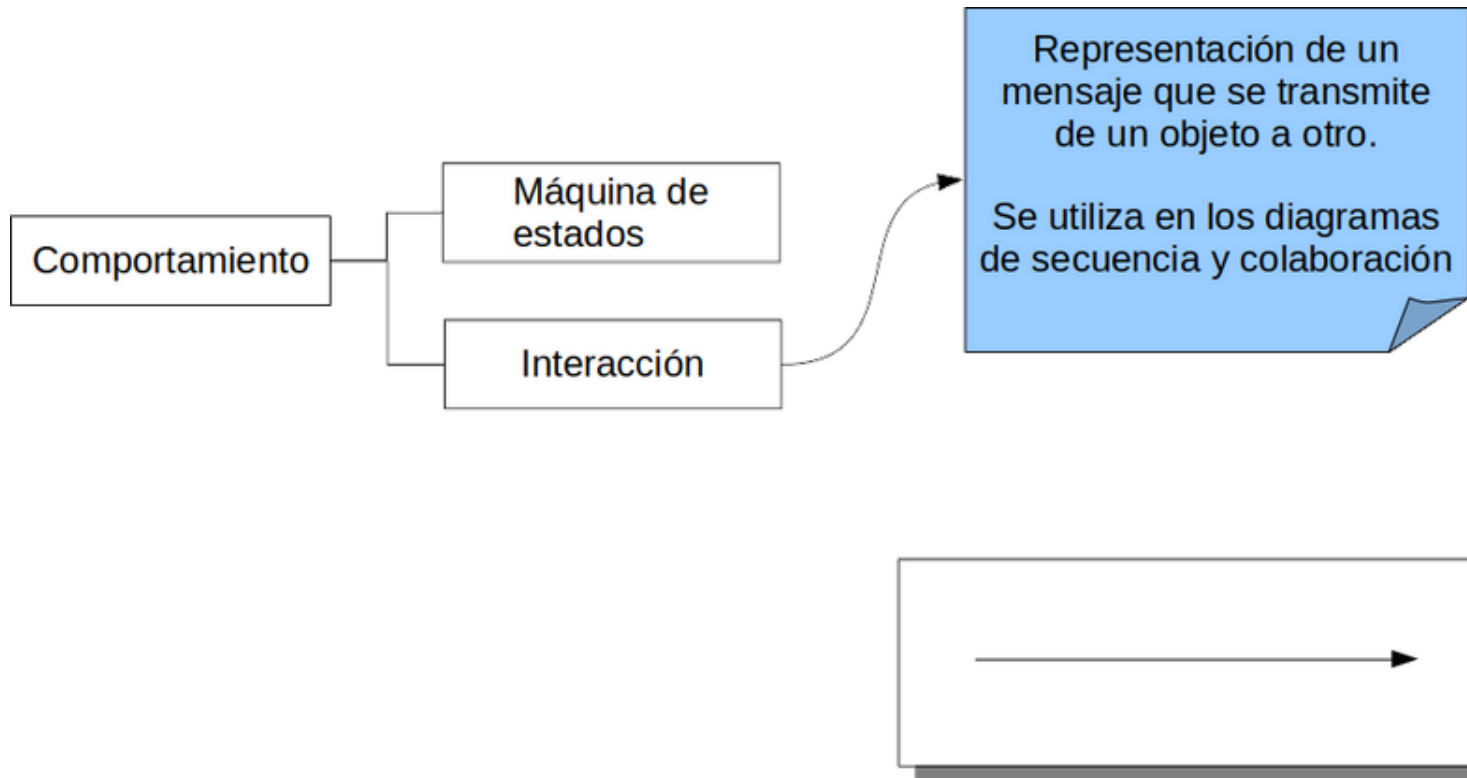
Introducción a UML

Modelo conceptual:



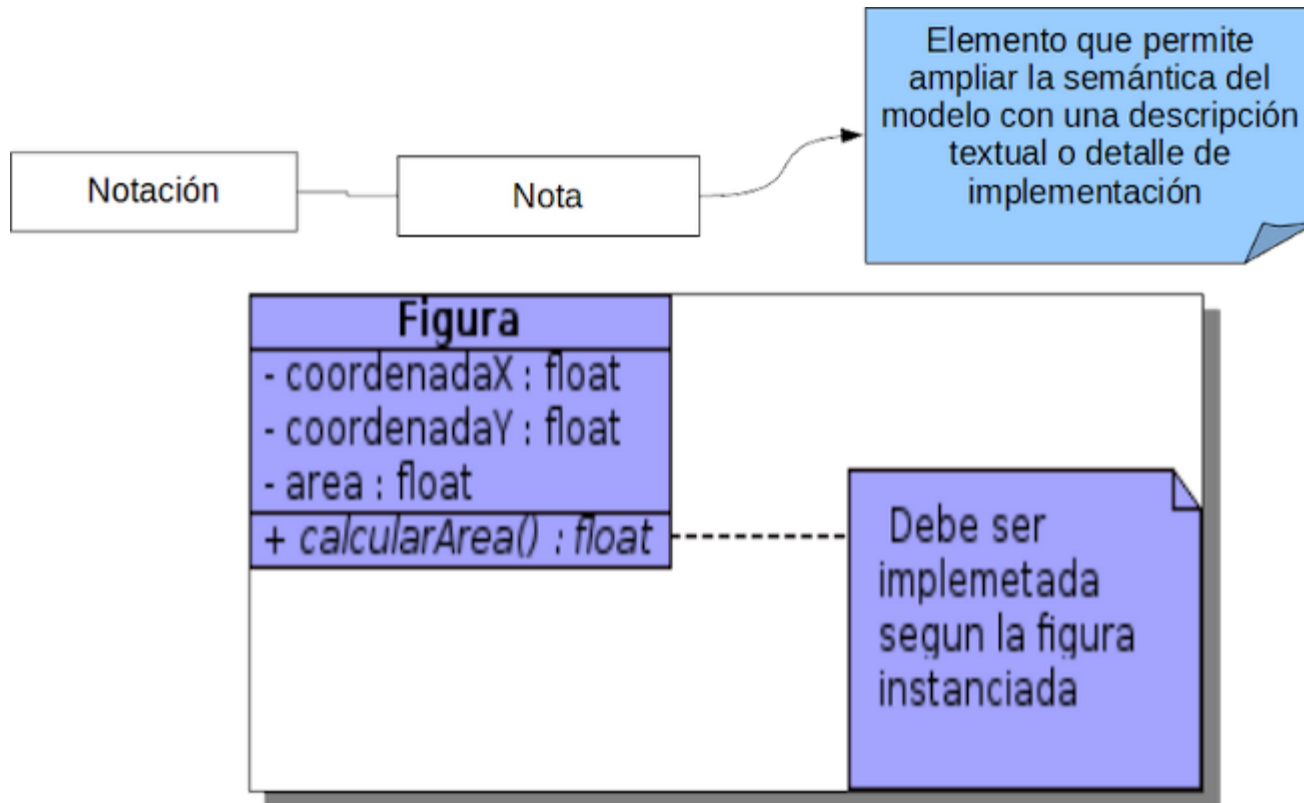
Introducción a UML

Modelo conceptual:



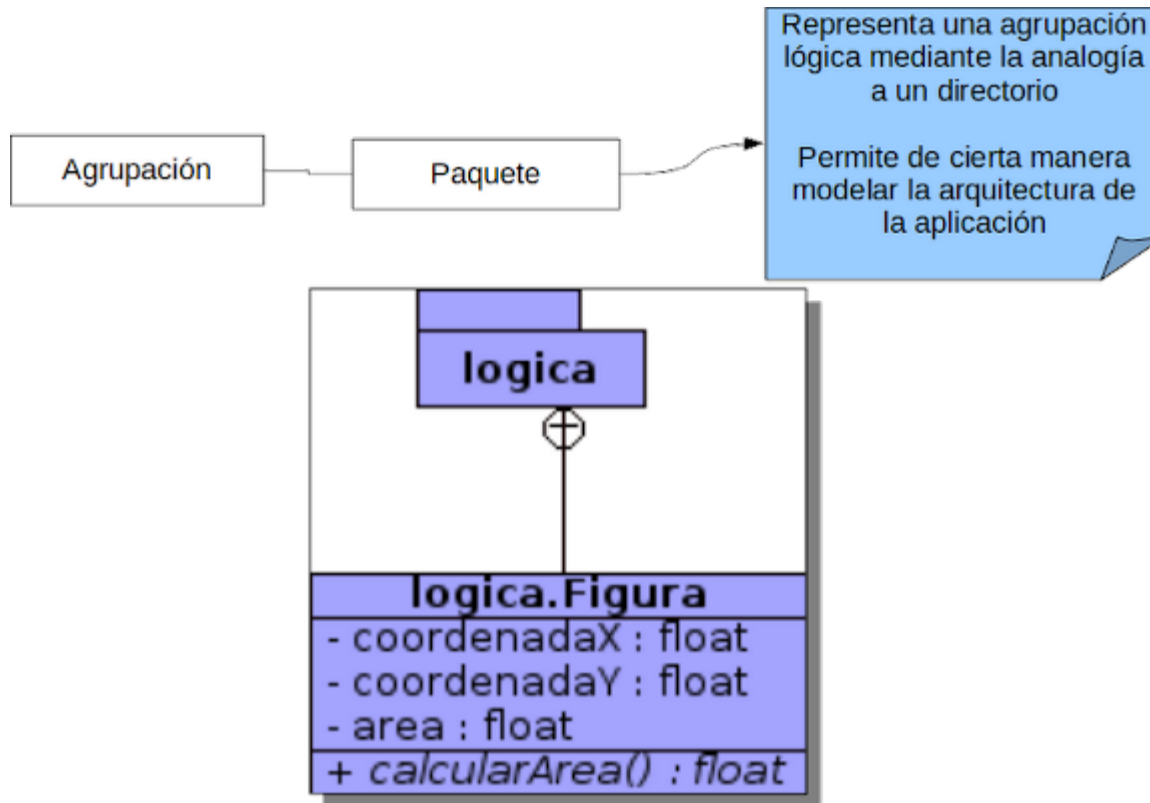
Introducción a UML

Modelo conceptual:



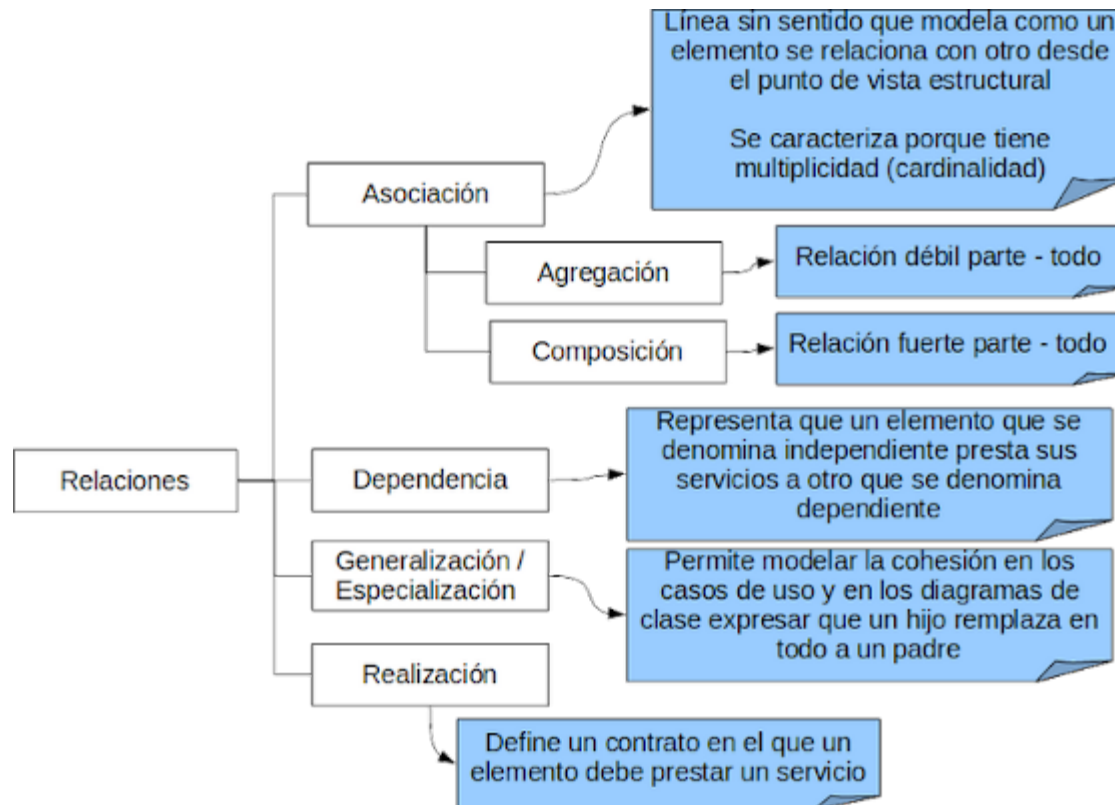
Introducción a UML

Modelo conceptual:



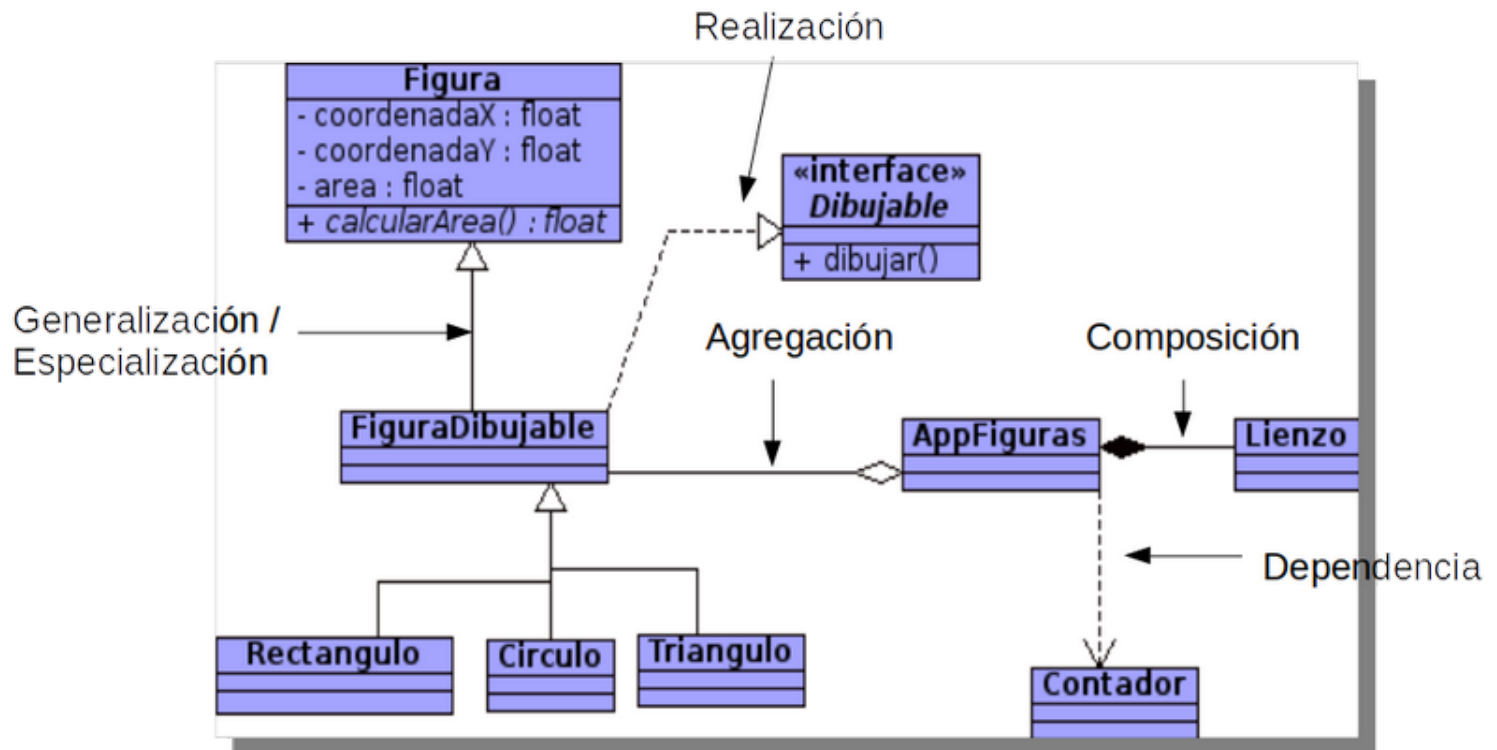
Introducción a UML

Modelo conceptual:



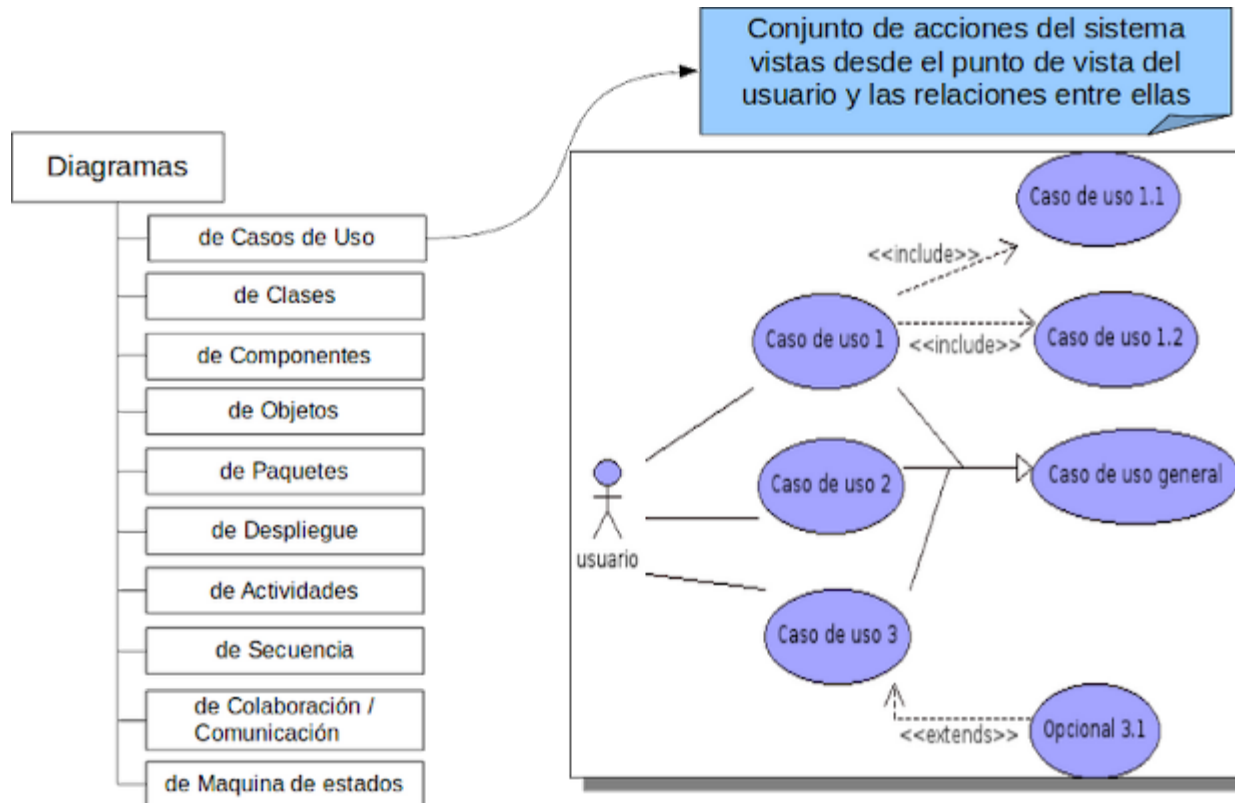
Introducción a UML

Modelo conceptual:



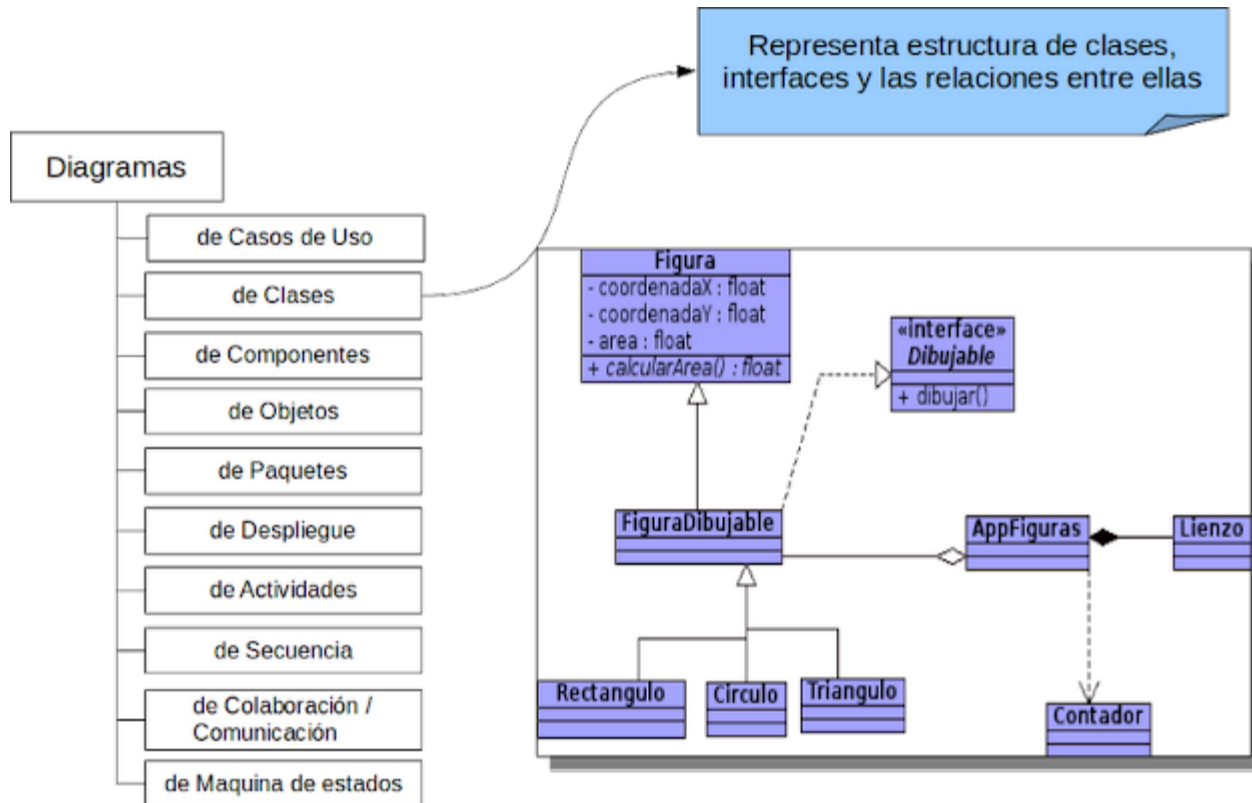
Introducción a UML

Modelo conceptual:



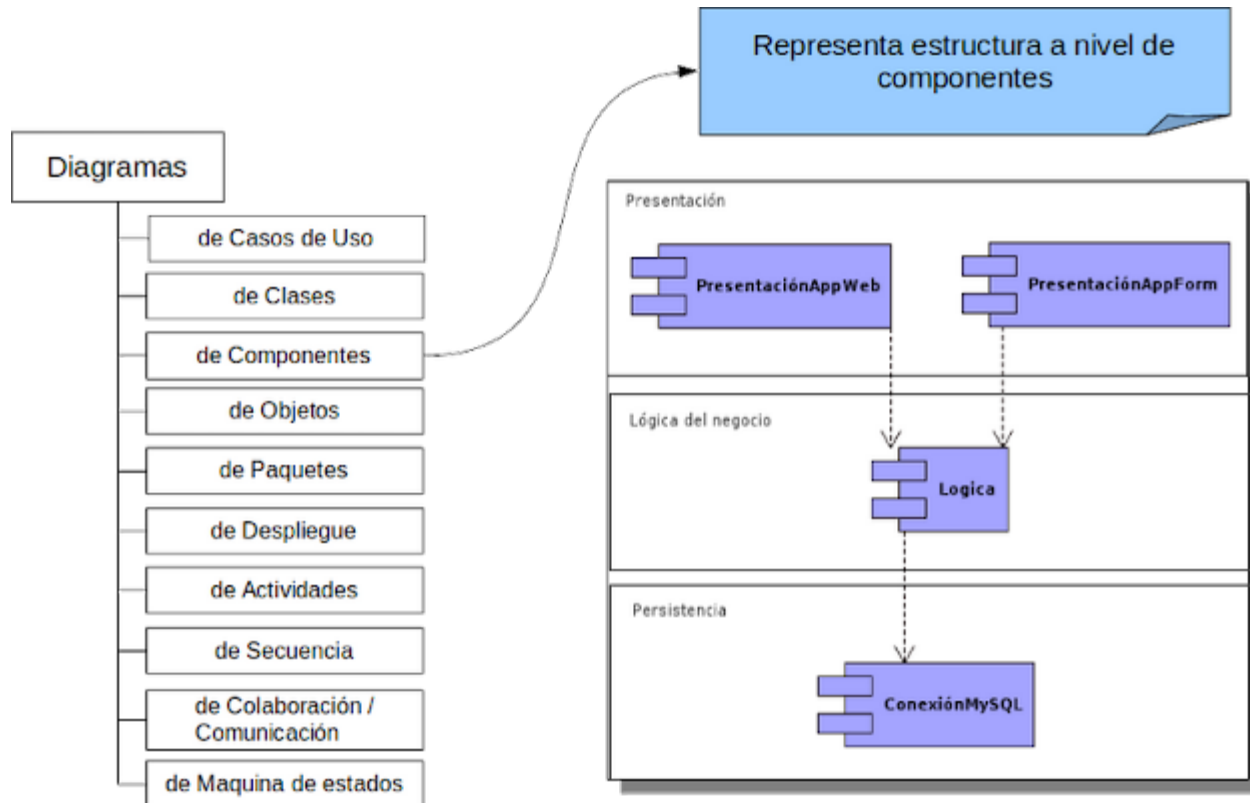
Introducción a UML

Modelo conceptual:



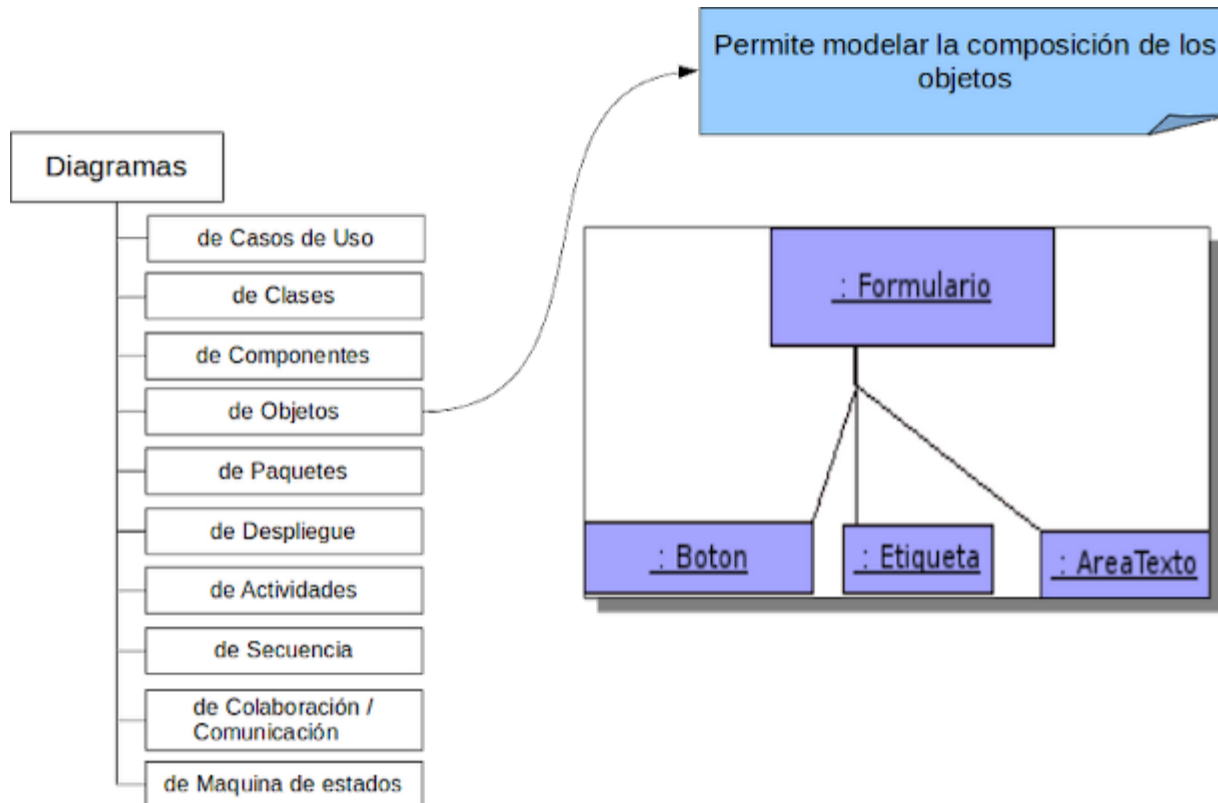
Introducción a UML

Modelo conceptual:



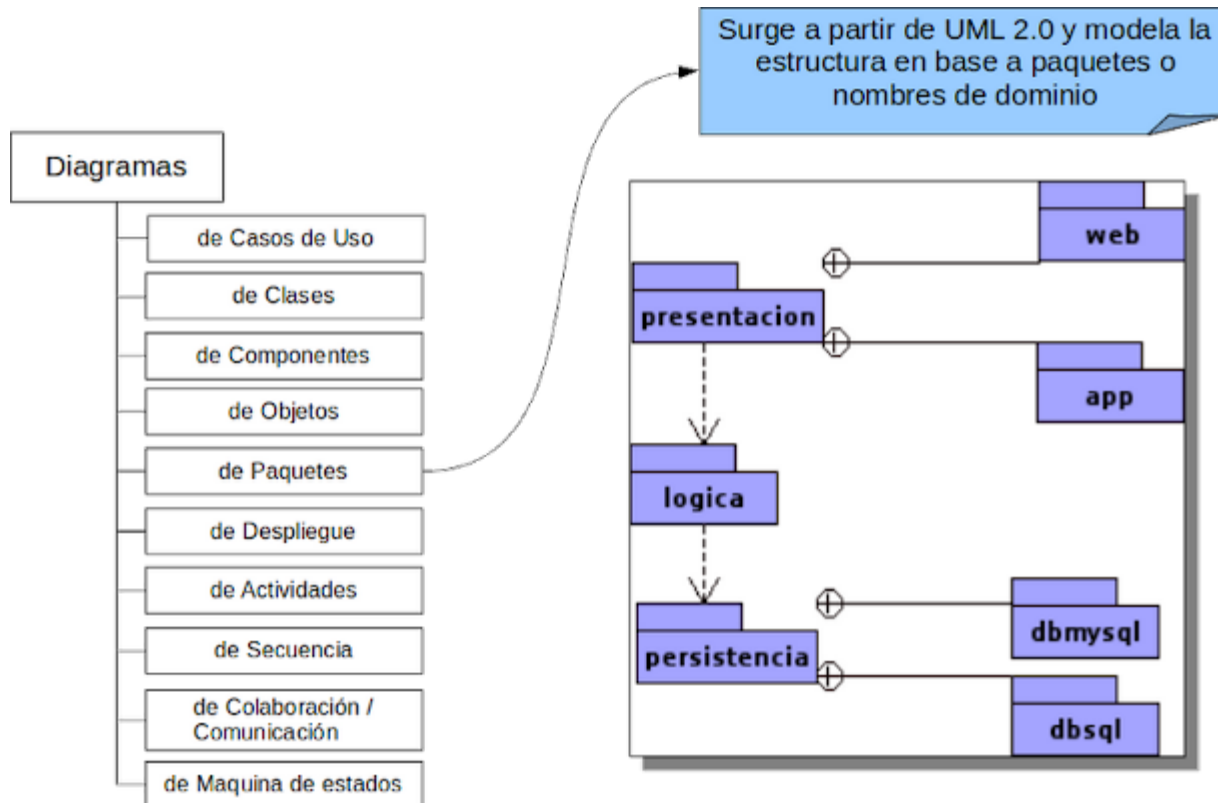
Introducción a UML

Modelo conceptual:



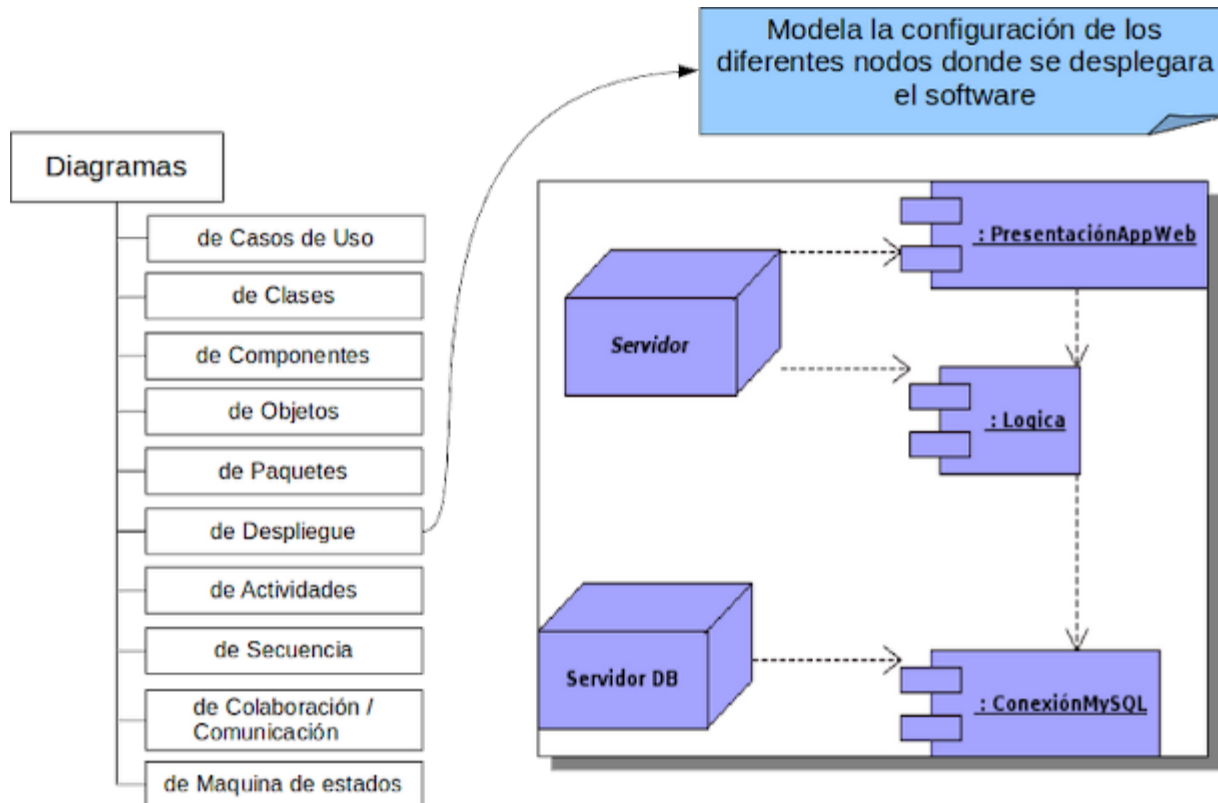
Introducción a UML

Modelo conceptual:



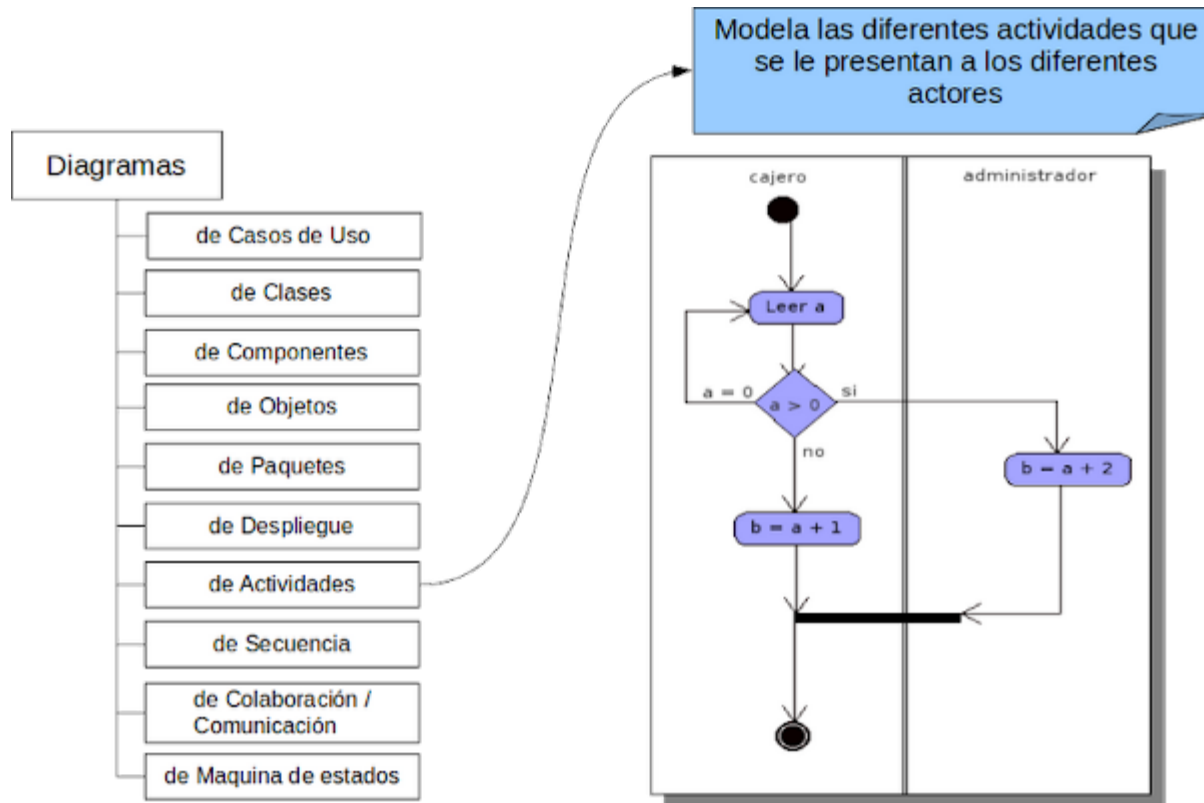
Introducción a UML

Modelo conceptual:



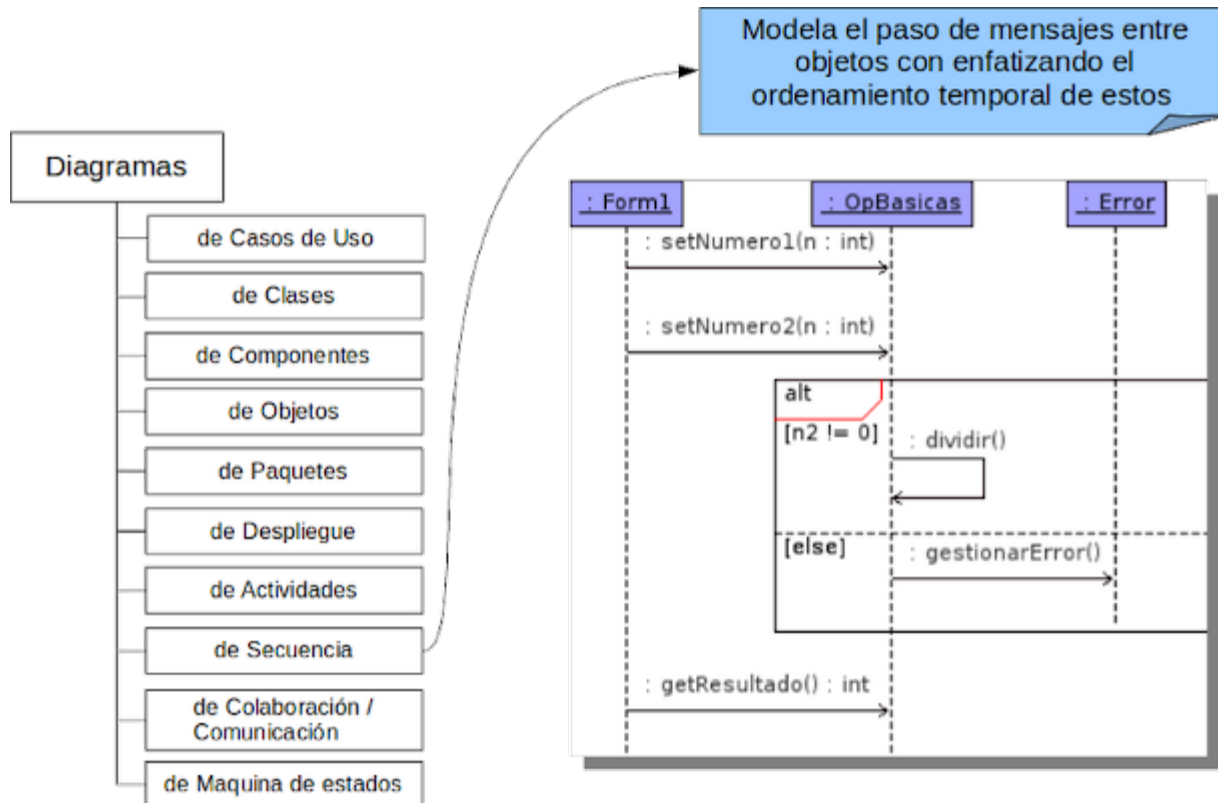
Introducción a UML

Modelo conceptual:



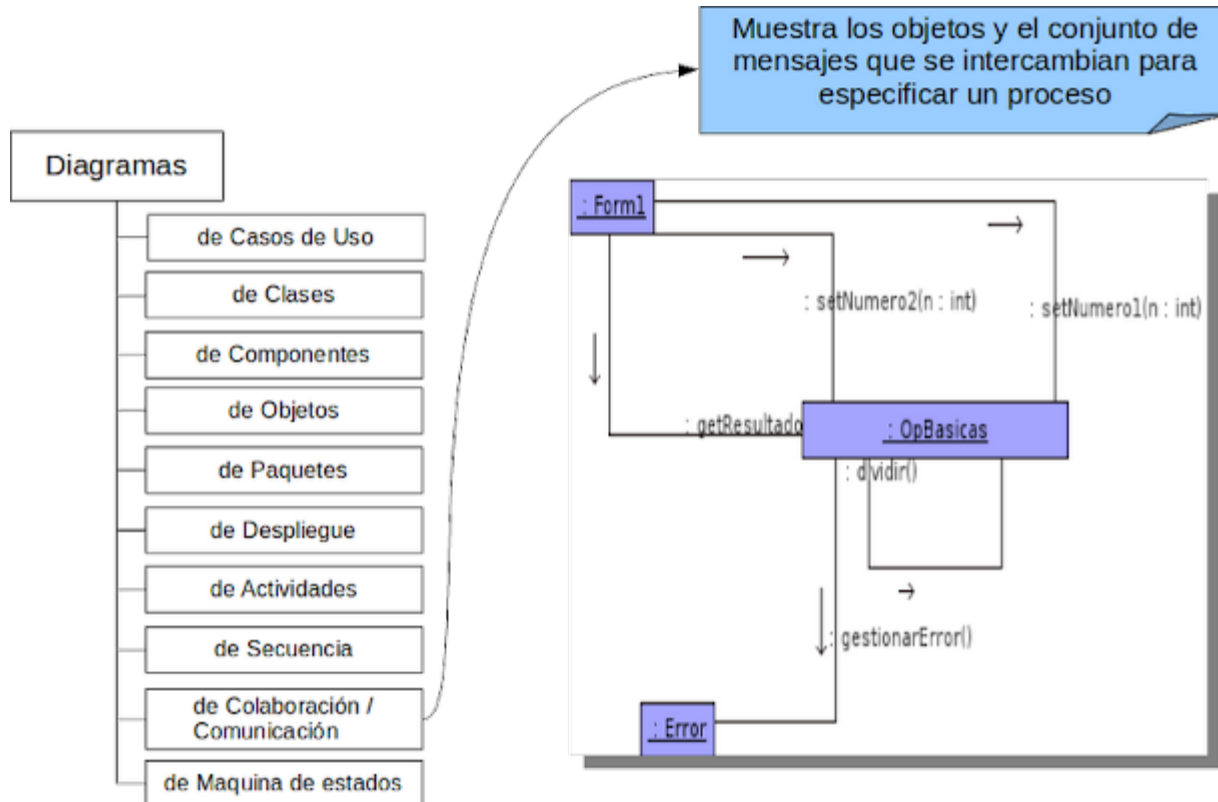
Introducción a UML

Modelo conceptual:



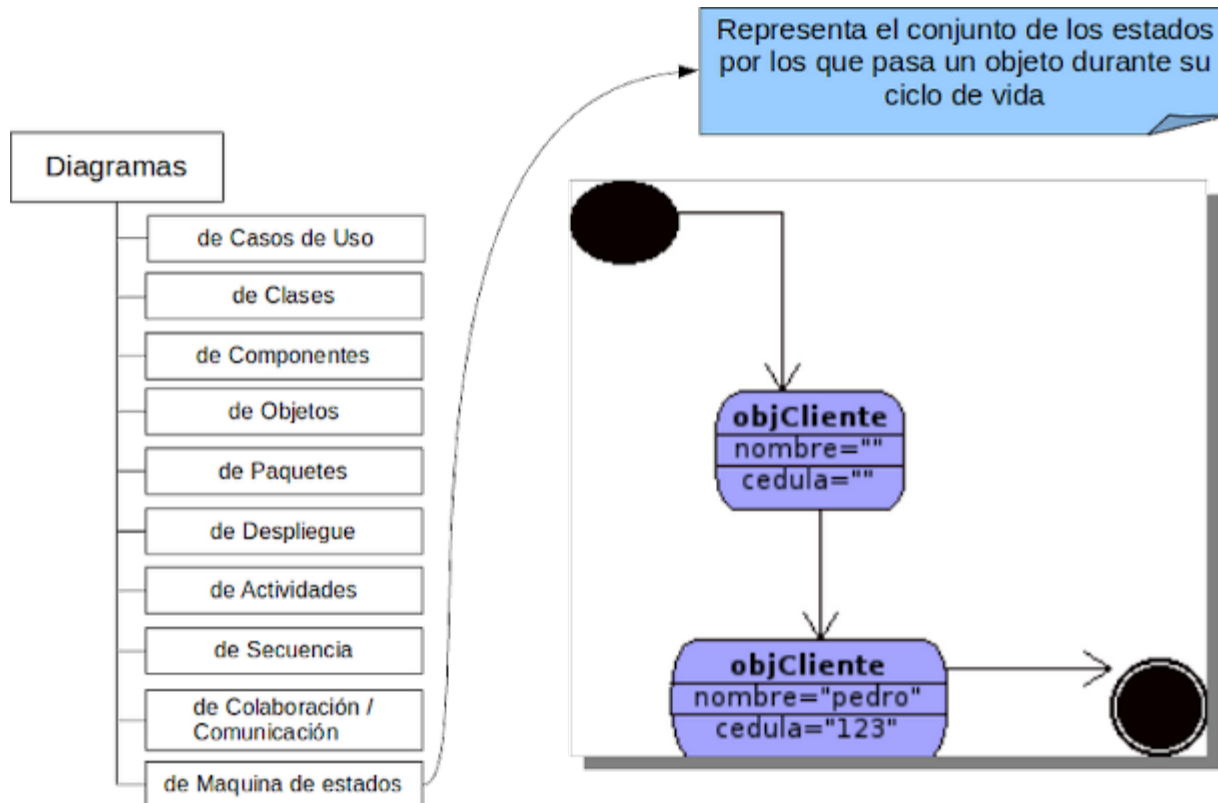
Introducción a UML

Modelo conceptual:



Introducción a UML

Modelo conceptual:



Introducción a UML

Ejemplo:

Cronómetro:

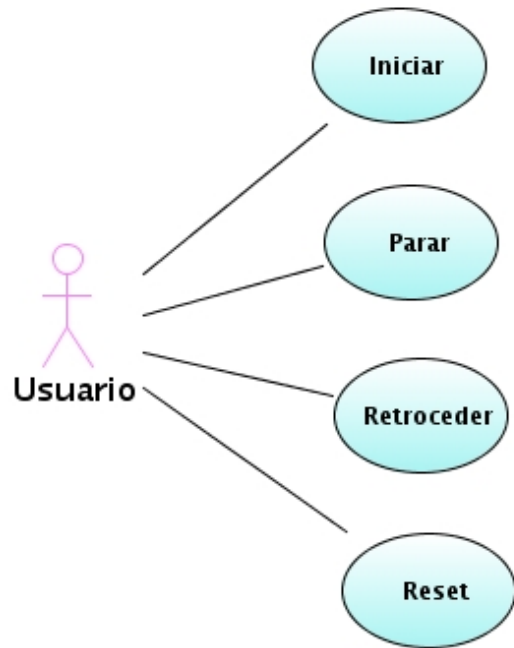
El caso práctico elegido para la elaboración de los diagramas de UML es un aplicativo de escritorio que permita al usuario manejar un cronómetro con la implementación normal de las funciones básicas de un elemento como este: avanzar, parar, retroceder, reset.



Introducción a UML

Ejemplo:

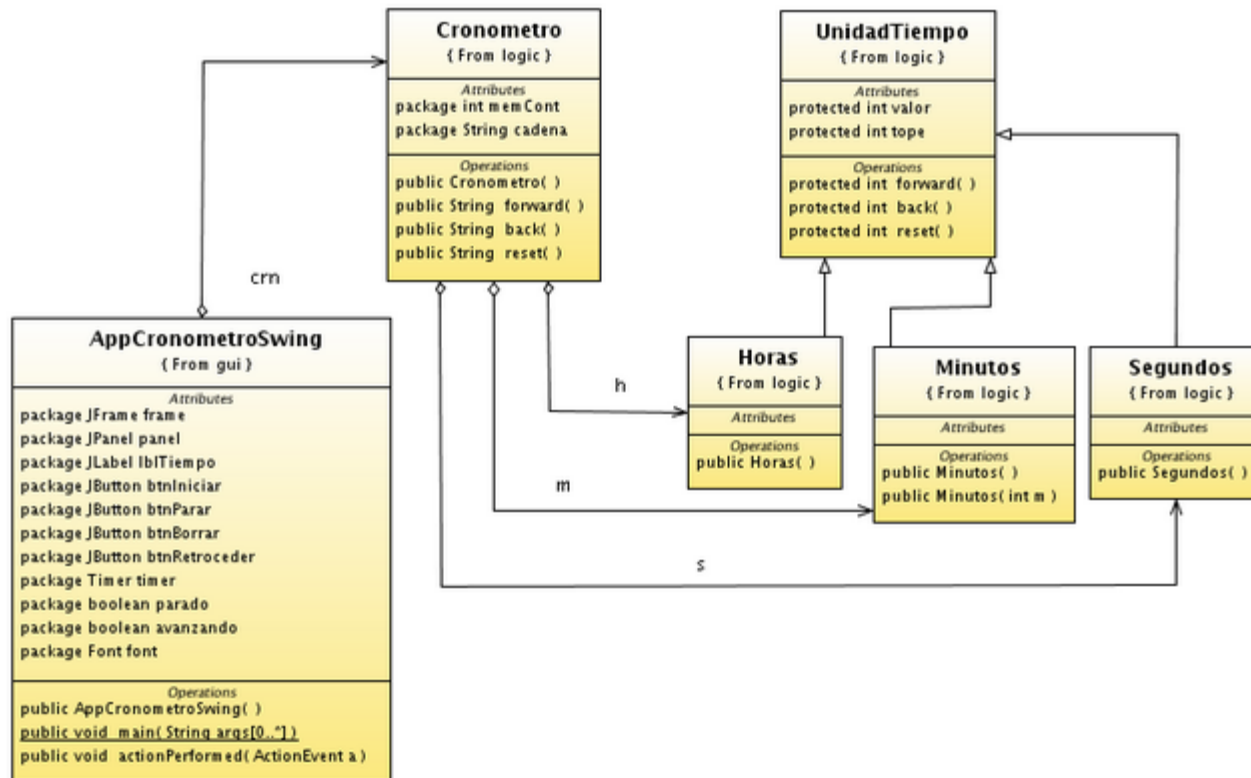
Diagrama de casos de uso:



Introducción a UML

Ejemplo:

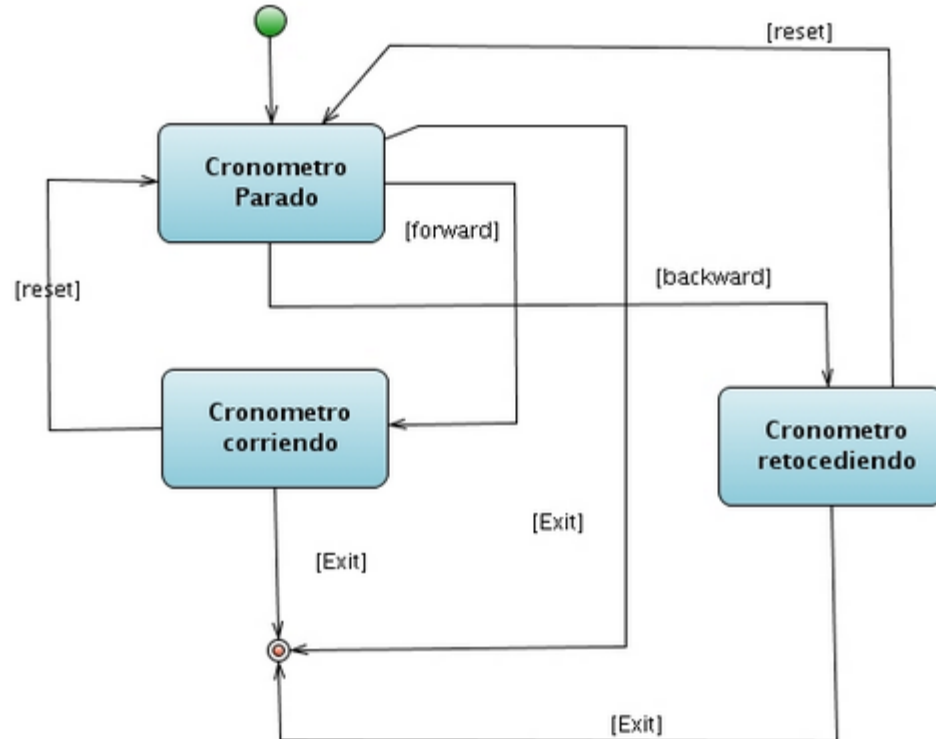
Diagrama de clases:



Introducción a UML

Ejemplo:

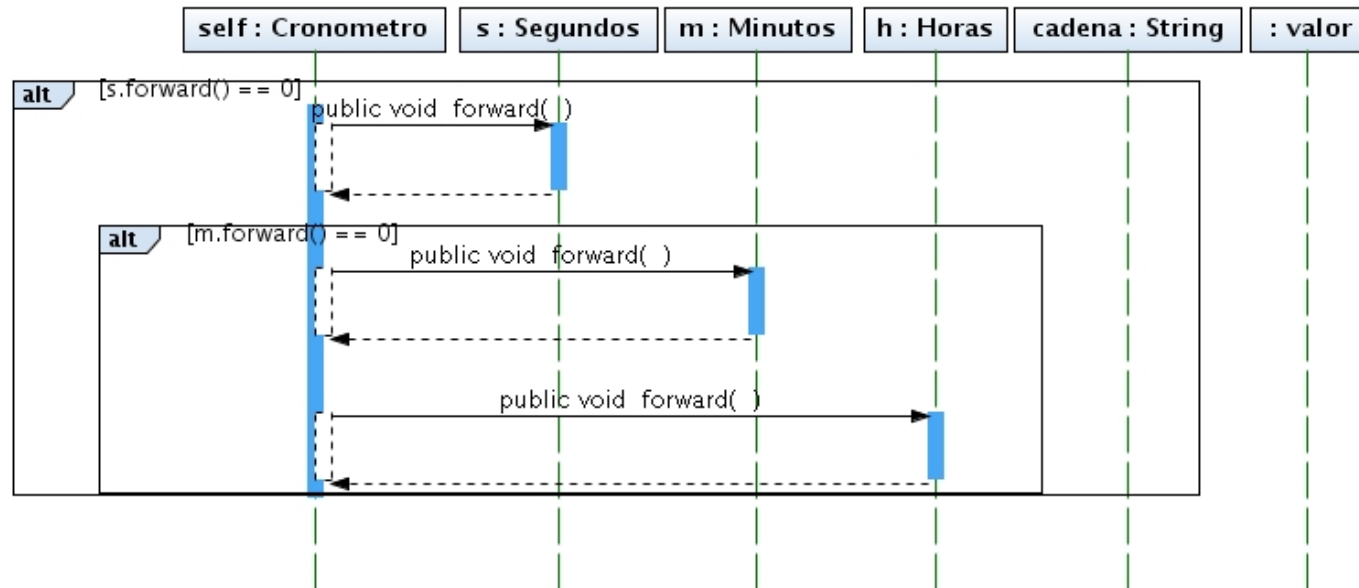
Diagrama de estados:



Introducción a UML

Ejemplo:

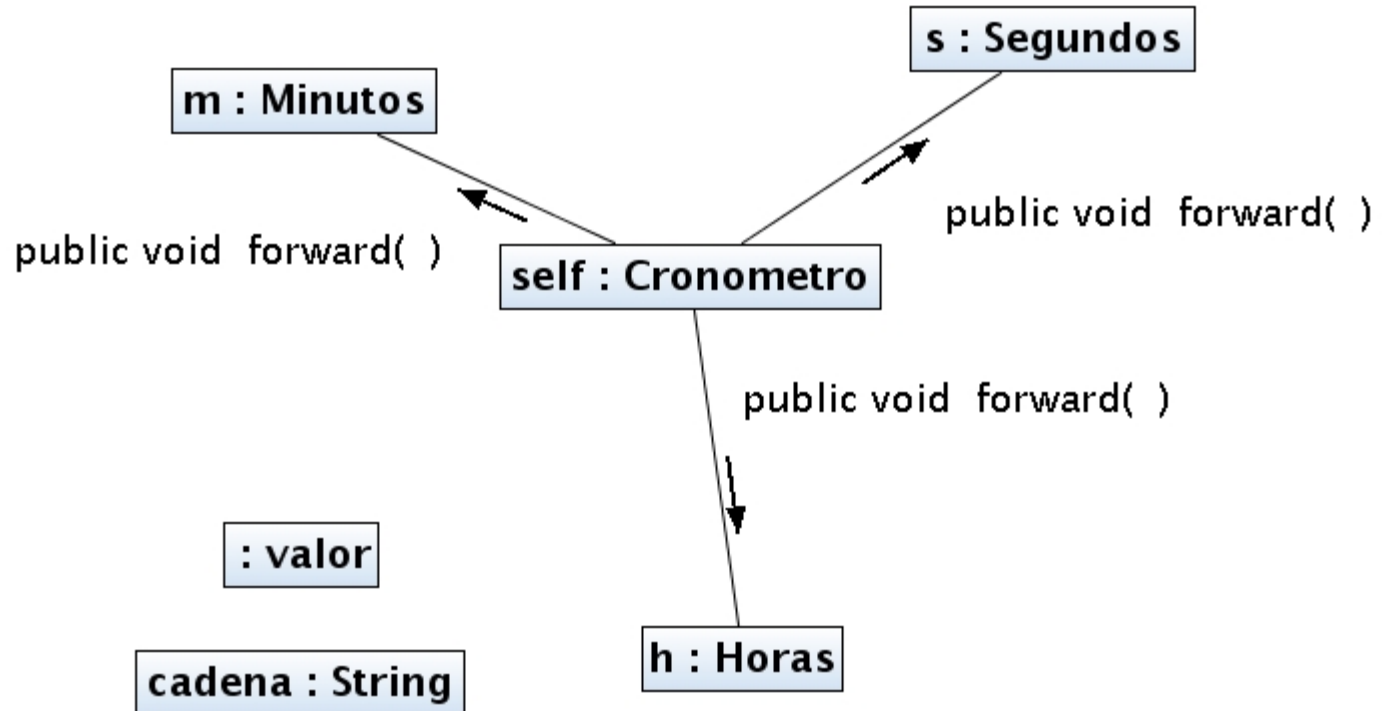
Diagrama de secuencia:



Introducción a UML

Ejemplo:

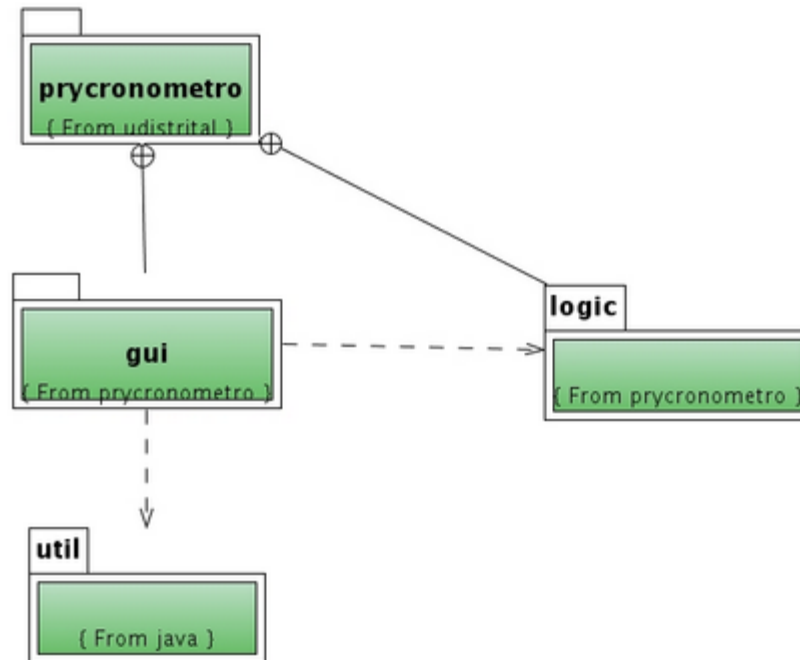
Diagrama de colaboración:



Introducción a UML

Ejemplo:

Diagrama de paquetes:



Diagramas y Vistas

UML define varios modelos para la representación de los sistemas que pueden verse y manipularse mediante un conjunto de diagramas diferentes

Diagramas de estructura:

- Diagrama de clases
- Diagrama de estructuras compuestas
- Diagrama de componentes
- Diagrama de despliegue
- Diagrama de objetos
- Diagrama de paquetes

Diagramas y Vistas

Diagramas de comportamiento:

- Diagrama de casos de uso
- Diagrama de actividad
- Diagramas de interacción
 - *Diagrama de secuencia*
 - *Diagrama de comunicación o colaboración*
 - *Diagrama de visión global de la interacción*
 - *Diagrama de tiempo*
- Diagrama de maquina de estados

Diagramas y Vistas

- Una vista es un subconjunto de las construcciones de modelado de UML que representa un aspecto del sistema
- Las vistas se pueden agrupar en áreas conceptuales

Área	Vistas
Estructural	Vista estática
	Vista de diseño
	Vista de casos de uso
Dinámica	Vista de máquina de estados
	Vista de actividad
	Vista de interacción
Física	Vista de despliegue
Gestión	Vista de gestión del modelo
	Perfiles

Diagramas y Vistas

- Los diagramas UML se pueden organizar en las siguientes vistas:

Vista	Diagrama
Vista estática	Diagramas de clases
Vista de casos de uso	Diagramas de casos de uso
Vista de interacción	Diagramas de secuencia
	Diagramas de comunicación
Vista de actividad	Diagramas de actividad
Vista de la máquina de estados	Diagramas de máquina de estados
Vista de diseño	Diagramas de estructuras compuestas
	Diagramas de colaboración
	Diagramas de componentes
Vista de despliegue	Diagramas de despliegue
Vista de gestión del Modelo	Diagramas de paquetes
Perfiles	Diagramas de paquetes

Vista estática

- Modela conceptos del dominio de la aplicación sus propiedades internas y sus relaciones
- Se denomina vista estática porque no modela comportamiento del sistema dependiente del tiempo

Componentes principales:

- Clases: describen conceptos del dominio de la aplicación o de la solución
- Relaciones:
 - *Asociación*
 - *Generalización*
 - *Dependencia: realización y uso*

Diagramas utilizados:

- Diagrama de clases

Vista de gestión del modelo

- Modela la organización del modelo mismo mediante un conjunto de paquetes
- Un modelo abarca un conjunto de paquetes que contienen los elementos del modelo tales como clases, máquinas de estados y casos de uso
- Un elemento de modelado puede pertenecer a más de un paquete
- La agrupación de elementos de modelado en un paquete no es necesario que coincida con la agrupación física de elementos del sistema

Vista de casos de uso

- Los casos de uso son una técnica para la especificación de requisitos funcionales.
- Modela la funcionalidad del sistema tal como la perciben los agentes externos, denominados actores, que interactúan con el sistema desde un punto de vista particular

Componentes principales:

- Sujeto: sistema que se modela
- Casos de uso: unidades funcionales completas
- Actores: entidades externas que interactúan con el sistema

Diagramas utilizados:

- Diagramas de casos de uso

Vista de interacción

Interacción: unidad de comportamiento que se centra en el intercambio de información observable entre elementos que pueden conectarse

- La comunicación se realiza mediante mensajes
- Viene dada por un par de conjuntos de trazas (secuencias de eventos): trazas válidas e inválidas. La unión de esos conjuntos no cubre necesariamente el universo entero de trazas
- Las interacciones se pueden especializar añadiendo más trazas a la interacción original

Diagramas utilizados:

- Diagrama de secuencia
- Diagrama de comunicación (colaboración)
- Diagrama de visión global de la interacción (variante del diagrama de actividad)
- Diagrama de tiempo

Vista de actividad

- Representación del comportamiento dinámico del sistema mediante actividades

Componentes:

- Actividad: especificación de un comportamiento parametrizado que se expresa como un flujo de ejecución por medio de una secuencia de unidades subordinadas
- Acción: especificación de una unidad fundamental de comportamiento que representa una transformación o procesamiento

Diagramas utilizados:

- Diagrama de actividad

Vista de máquina de estados

- Las máquinas de estados describen los estados que un objeto puede tener durante su ciclo de vida, el comportamiento en esos estados y los eventos que causan los cambios de estado

Componentes:

- Máquina de estados de comportamiento: capturan los ciclos de vida de los objetos, subsistemas y sistemas
- Máquina de estados de protocolo: se usan para especificar las transformaciones legales que pueden ocurrir en un clasificador abstracto como una interfaz o un puerto (protocolos de uso)

Vista de diseño

- La vista de diseño modela la estructura de diseño de la aplicación mediante:
 - *Expansión en clasificadores estructurados*
 - *Colaboraciones que proporcionan la funcionalidad*
 - *Ensamblado a partir de componentes con interfaces bien definidas*
- **Diagramas utilizados:**
 - *Estructuras compuestas*
 - *Colaboración*
 - *Componentes*

Vista de despliegue

- La vista de despliegue representa el despliegue de artefactos de tiempo de ejecución sobre nodos
- Los diagramas de despliegue, junto con los diagramas de componentes, forman parte de la arquitectura física
- La arquitectura física es una descripción detallada del sistema que muestra la asignación de artefactos de software a nodos físicos

Estructura física del software: distribución y dependencias, en tiempo de ejecución, de los artefactos software definidos en la arquitectura lógica

- Componentes: empaquetamiento físico de una colección de elementos de modelado
- Artefactos: porción física de información que usa o produce el proceso de desarrollo de software (manifestaciones físicas de los componentes)
- Especificadores de despliegue: conjunto de propiedades que determinan los parámetros de ejecución de un componente desplegado en un nodo

Vista de despliegue

Estructura del hardware: nodos y forma de conexión entre ellos

- Dispositivos: recursos computacionales (nodos)
- Caminos de comunicación: mecanismos de comunicación entre nodos
- Entornos de ejecución: subnodos de los dispositivos

Perfiles

- El mecanismo de los perfiles permite cambios limitados sobre UML sin modificar el metamodelo subyacente
- Los perfiles y las restricciones permiten que UML sea adaptado a dominios o plataformas específicas manteniendo la interoperabilidad entre herramientas
- Los perfiles se construyen haciendo uso de los principales mecanismos de extensibilidad de UML: restricciones, estereotipos y valores etiquetados