Introducción a UML. Casos de Uso

Entornos de desarrollo

ÍNDICE DE CONTENIDOS

- Modelado de Software
- Claves en el desarrollo del Software
- UML
- Tipos de diagramas
- Caso de uso

¿Qué es un modelo?

- Un **Modelo** es una simplificación de la realidad.
- *Un modelo es resultado de un proceso de **abstracción** y ayuda a comprender y razonar sobre una realidad.*
- *Un **modelo software** es una descripción de un aspecto del sistema, expresada en un lenguaje bien definido.*

Buscamos:

- ✓ Modelar la complejidad.
- ✓ Independencia del lenguaje de codificación.
- ✓ Evaluar el modelo antes de implementarlo.

¿Por qué es útil modelar?

- Porque permite utilizar un lenguaje común que facilita la comunicación entre los miembros del equipo de desarrollo.

- Con **UML** podemos documentar todos los artefactos (información que es utilizada o producida mediante un proceso de desarrollo de software, por ejemplo, modelos) de un proceso de desarrollo (requisitos, condiciones,… suelen venir definidos por el **cliente**). Permite definir los **objetivos** que debe cumplir un proyecto software (arquitectura, pruebas, versiones,...) por lo que se dispone de documentación que trasciende al proyecto.

¿Porqué es útil modelar?

- Hay **estructuras** que trascienden lo representable en un lenguaje de programación, como las que hacen referencia a la arquitectura del sistema:
  - ✓ Conjunto de decisiones significativas acerca de la organización de un sistema software.
  - ✓ Selección de los elementos estructurales a partir de los cuales se compone el sistema.
  - ✓ Comportamiento y colaboraciones entre elementos.
  - ✓ Composición de los elementos para satisfacer las necesidades de sistemas mayores.
  - ✓ Elementos estructurales y de comportamiento en subsistemas progresivamente mayores y el estilo arquitectónico(organización, de estos elementos, sus interfaces, colaboraciones y su composición).
  - ✓ Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose modelos precisos, no ambiguos y completos.

¿Porqué es útil modelar?

UML puede conectarse a lenguajes de programación mediante **ingeniería directa** (transformación de un modelo en código a través de su traducción a un determinado lenguaje de programación) e **inversa** (transformación del código en un modelo a través de su traducción desde un determinado lenguaje de programación).

¿Porqué es útil modelar?

- El modelado es realizado en el **análisis y diseño** de aplicaciones software antes de escribir el código.
- Se crean un conjunto de modelos ("*planos del software*") que permiten **especificar** aspectos del sistema como los **requisitos, la estructura y el comportamiento**.
- "Una empresa software con éxito es aquella que **produce de manera consistente software de calidad** que satisface las necesidades de los usuarios".
- "El **modelado es la parte esencial** de todas las actividades que conducen a la producción de software de calidad".

Utilidad de modelos

Proposito de los modelos:

- ✓ Capturar y precisar requerimientos de un dominio de conocimiento, que sea comprensible por todos los partes interesadas (stakeholders) del proyecto.
- ✓ Pensar sobre un diseño de un sistema
- ✓ Capturar decisiones de diseño de un sistema.
- ✓ Explorar posibles soluciones a un problema económicamente.
- ✓ Generar productos de trabajo útiles.
- ✓ Documentar.

Utilidad de modelos

- Hay estructuras que **no son visibles** en los programas.
- Ayuda a **razonar** sobre el **cómo se implementa**.
- Se *facilita la comunicación entre el equipo* al existir un lenguaje común.
- Se dispone de **documentación** que trasciende al proyecto.
- **Generación de código** a partir de modelos.
- Ha surgido un nuevo paradigma de desarrollo de software a partir de modelos.
- Los modelos visualizan *cómo es o queremos* que sea el sistema especifican la estructura y comportamiento del sistema.
- *Guían* la construcción del sistema.
- *Documentan* las decisiones.

¿ Qué es UML ?

- **UML**, por sus siglas en Ingles, Unified Modeling Language.(Lenguaje Unificado de Modelado).

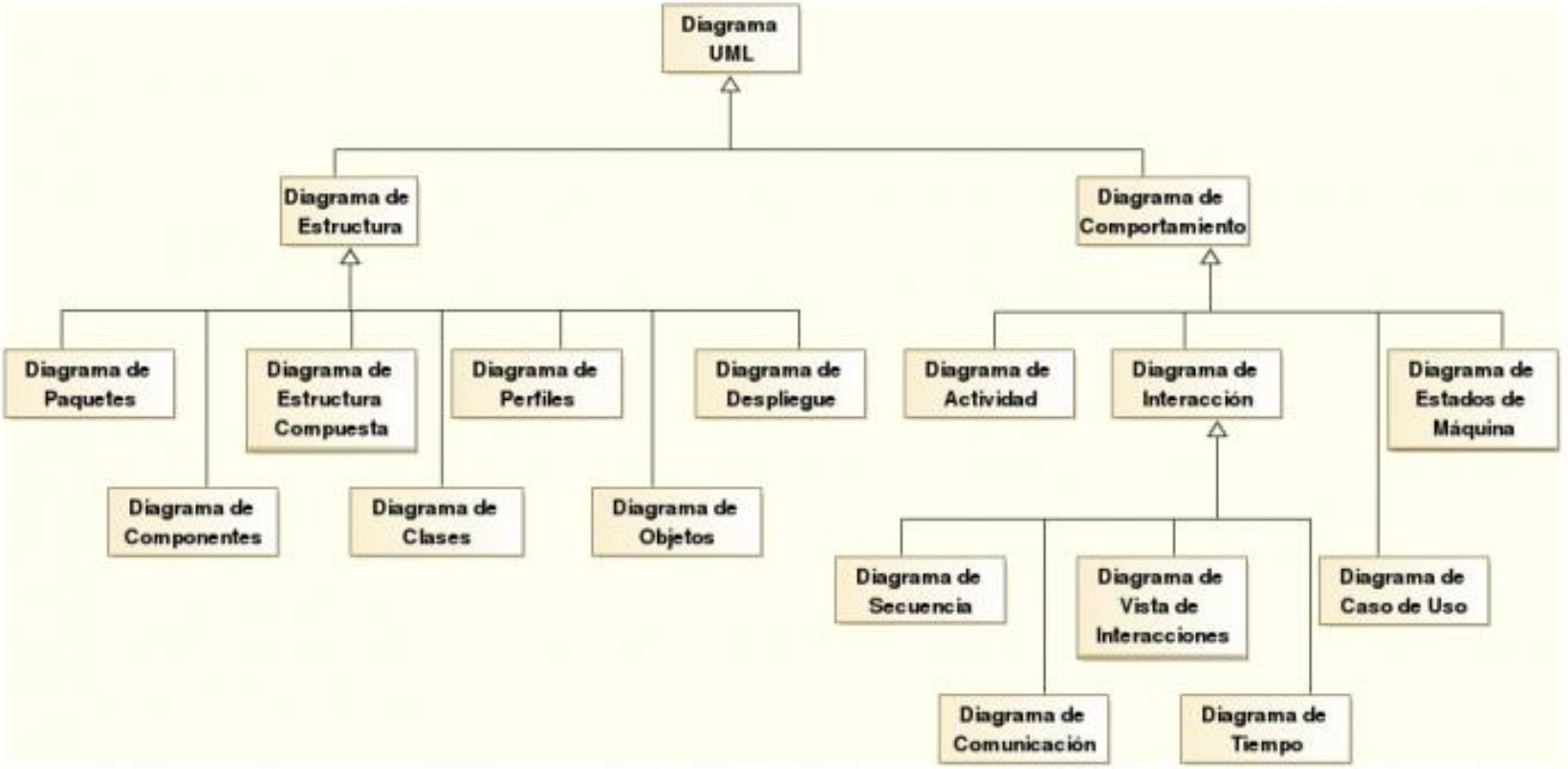
- **UML** es un lenguaje para
  - ✓ Visualizar
  - ✓ Especificar
  - ✓ Construir
  - ✓ Documentar

- Es importante resaltar que UML es un “lenguaje” para **especificar métodos o procesos**. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que esta descrito el modelo.

Lenguajes de modelado, modelos y diagramas.

- Un **lenguaje de modelado** permite expresar los distintos modelos que se producen en el proceso de desarrollo.
- Un **modelo** es una representación abstracta de una especificación, un diseño o un sistema desde un punto de vista particular.
- Un **diagrama** es una representación de (parte de) un modelo de diseño.
- Un **modelo** se representa por uno o más diagramas.

UML: Diagramas



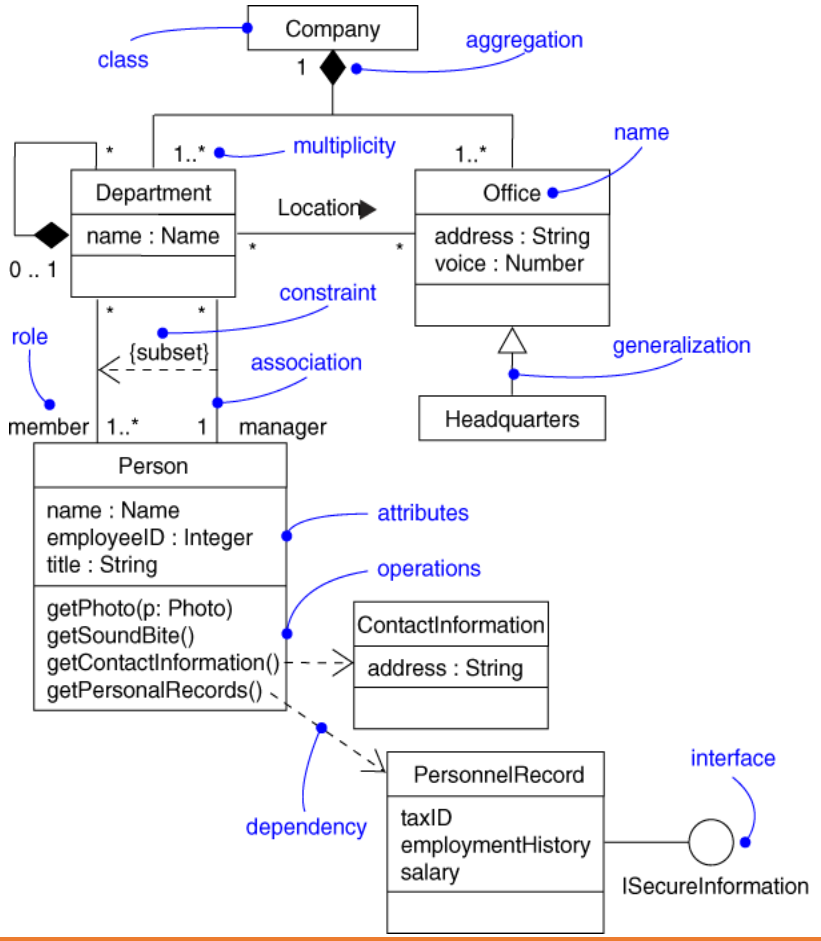
UML: Diagramas

Diagramas de Estructura: Enfatizan en los elementos que deben existir en el sistema modelado.

- ✓ Diagrama de Clases
- ✓ Diagrama de Componentes
- ✓ Diagrama de Objetos
- ✓ Diagrama de Estructura Compuesta
- ✓ Diagrama de Despliegue
- ✓ Diagrama de Paquetes

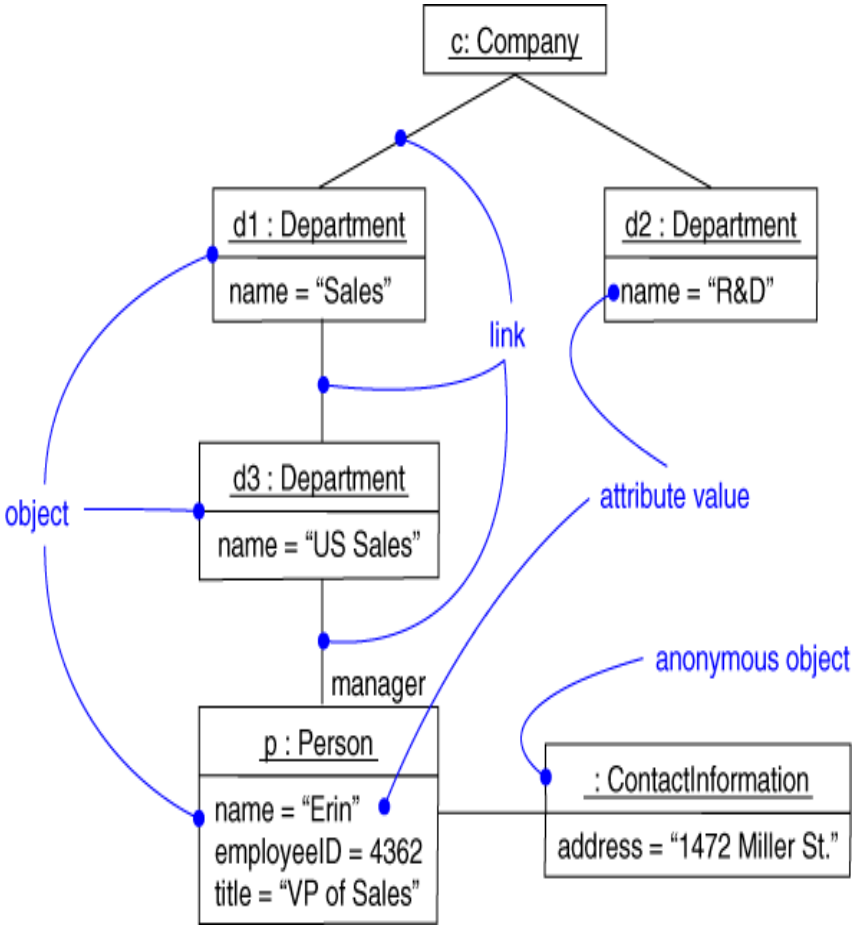
Diagramas de estructura: Diagramas de clase.

Muestran la estructura de un sistema concreto al modelar sus clases, atributos, operaciones y relaciones entre objetos.



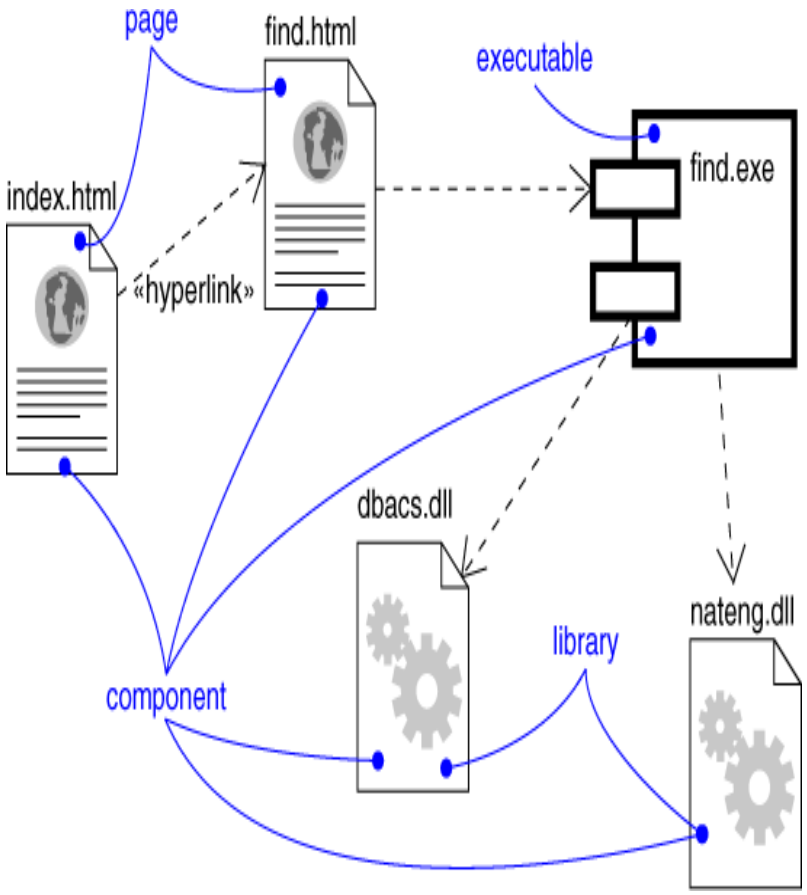
Diagramas de estructura: Diagrama de Objetos

Muestra una instantánea de un conjunto de objetos y sus relaciones



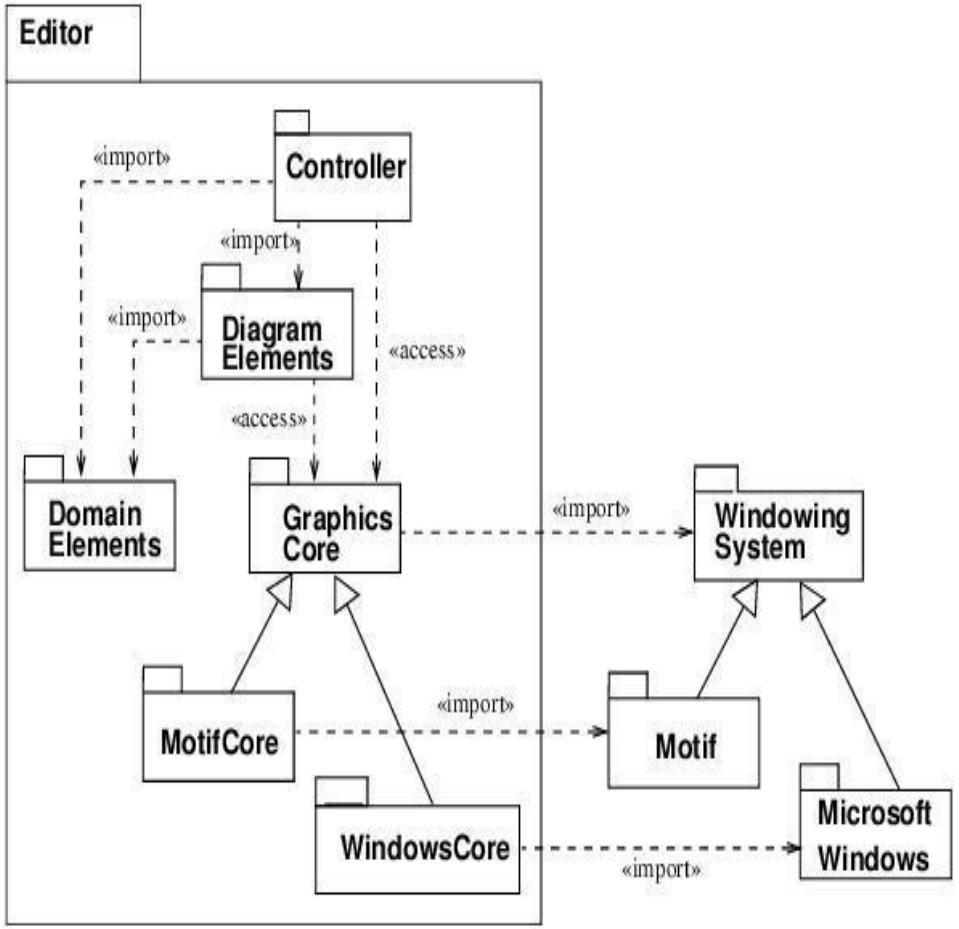
Diagramas de estructura: Diagrama de componentes

- Muestra la organización y dependencias entre un conjunto de componentes, la vista de implementación de un sistema.
- Están relacionados a diagramas de clases en donde un componente se corresponde con una o más clases, interfaces o colaboraciones.



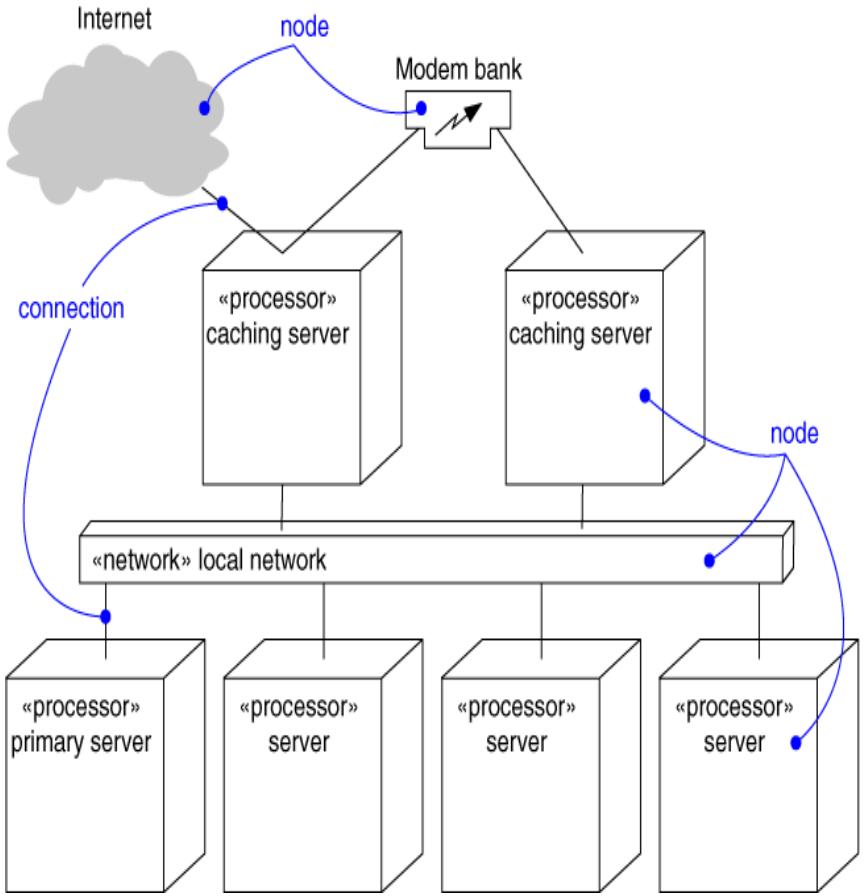
Diagramas de estructura: Diagrama de paquetes

Muestra la descomposición del modelo en unidades de organización y sus dependencias.



Diagramas de estructura: Diagrama de despliegue

Muestra los enlaces de comunicación física entre elementos de hardware y las relaciones entre máquinas físicas y procesos: **qué se ejecuta y dónde**.



Diagramas de comportamiento

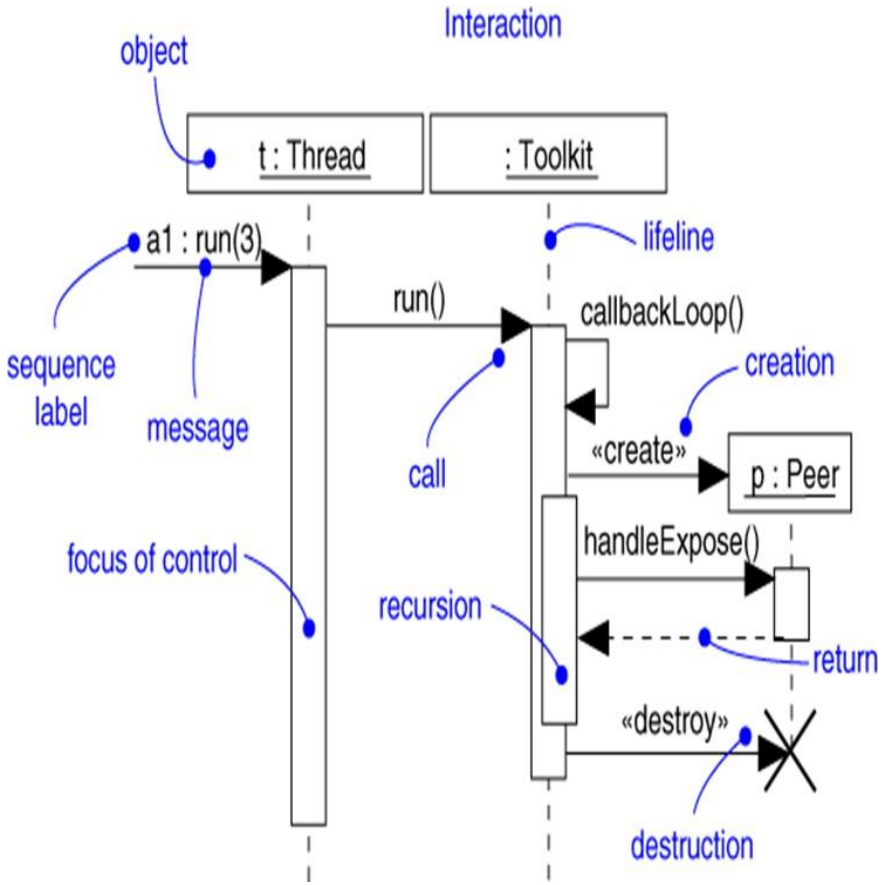
Este tipo de diagramas expresan las secuencias de estados por los que pasa un objeto a lo largo de su vida en respuesta a ciertos eventos “externos”.

Tipos de diagramas:

- ✓ Diagramas de casos de uso
- ✓ Diagrama de secuencia
- ✓ Diagrama de colaboración
- ✓ Diagrama de estados
- ✓ Diagrama de actividades
- ✓ Diagrama cronológico
- ✓ Diagrama general de interacciones

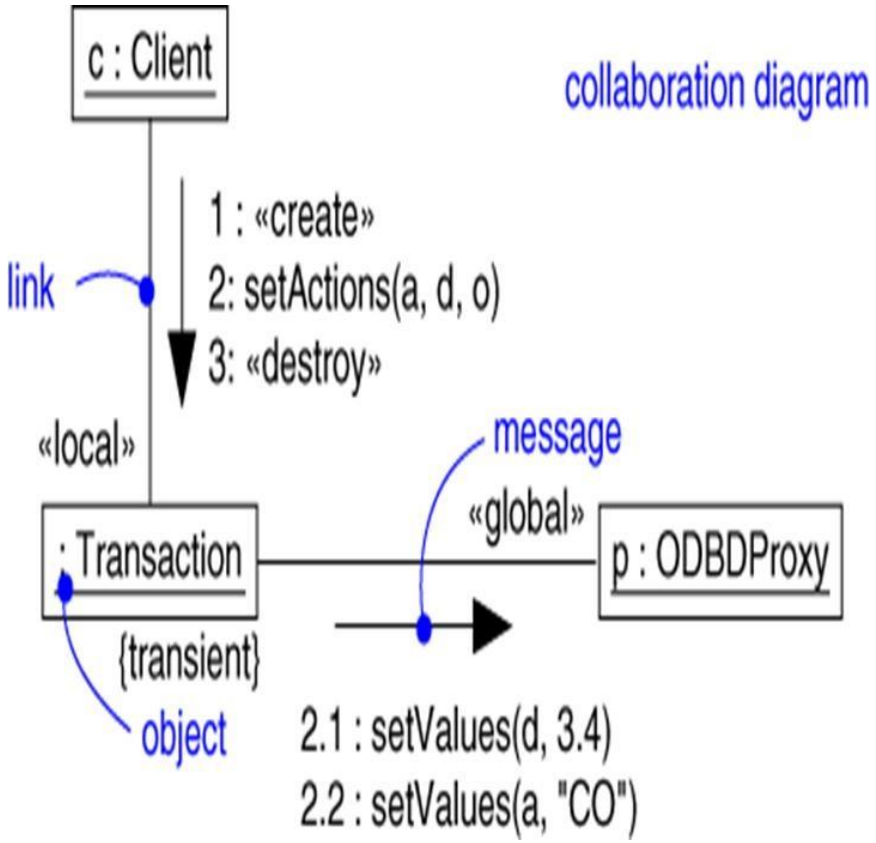
Diagramas de comportamiento: Diagrama de secuencia

Es un diagrama de *interacción que muestra los objetos y actores* que participan en una colaboración poniendo el énfasis en el ordenamiento en el tiempo de los mensajes



Diagramas de comportamiento: Diagrama de colaboración

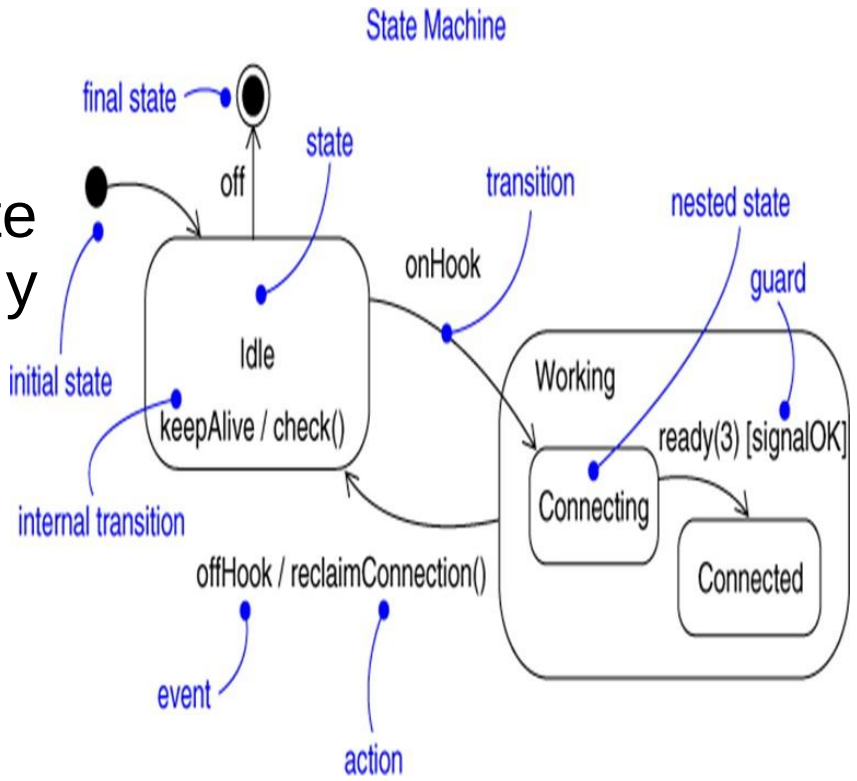
Un diagrama de interacción que pone el énfasis en la organización estructural de los objetos o roles que envían y reciben mensajes.



Diagramas de comportamiento: Diagrama de estados

Muestra un **autómata** que consiste de estados, transiciones, eventos y actividades

¡Máquina de Turing!



Diagramas de comportamiento: Diagrama de actividades

Muestra la estructura de un proceso u otro cálculo como el **flujo de control** y datos paso a paso en el cálculo.