
Introducción a UML.

Casos de Uso

Entornos de desarrollo

A solid orange horizontal bar at the bottom of the slide.

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 de 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

Propósito de los modelos:

- Capturar y precisar requerimientos de un dominio de conocimiento, que sea comprensible por todas las 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 económico.
- 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.

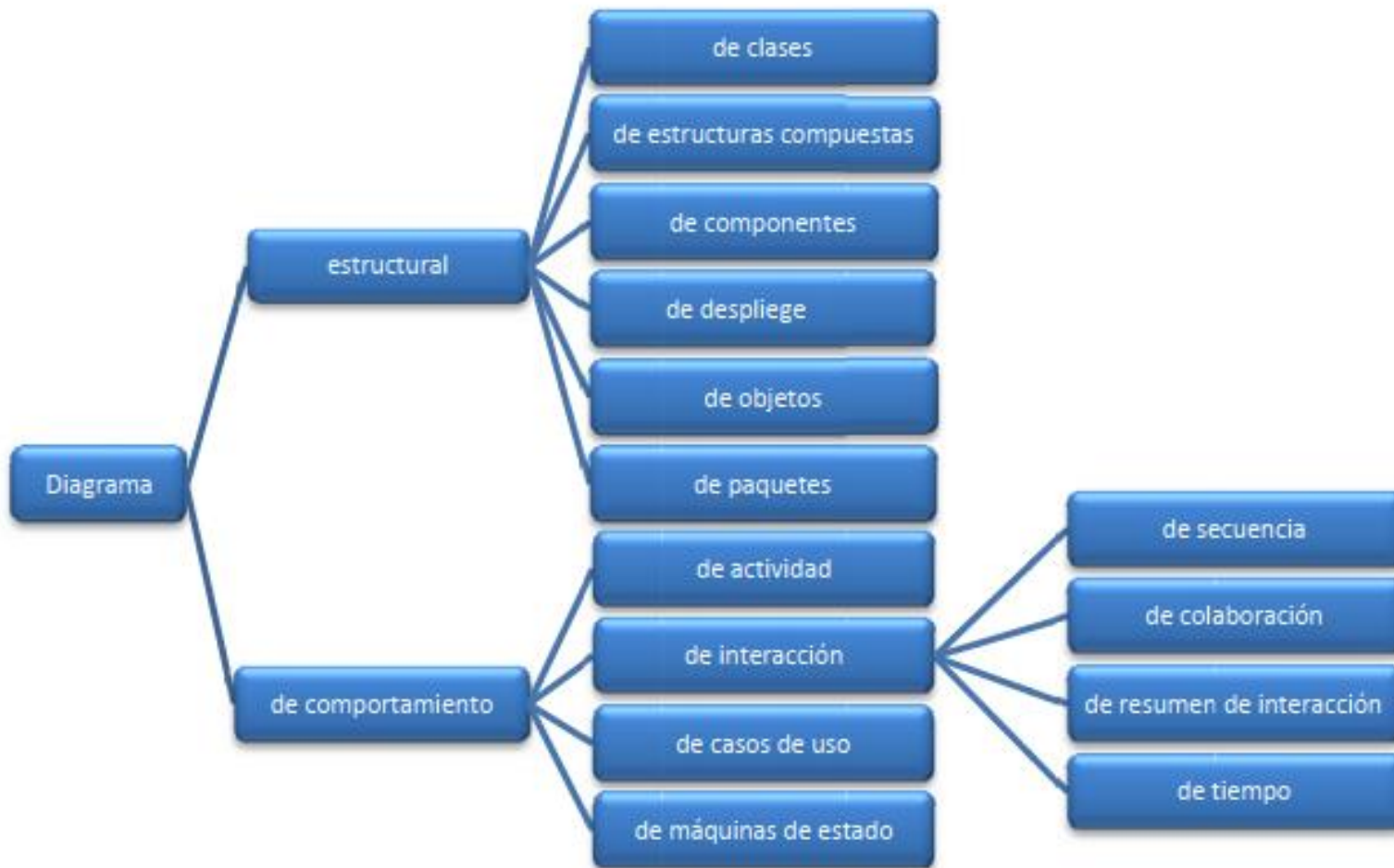
¿ Qué es UML ?

- **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 documentary 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 (una parte de) un modelo de diseño.
- Un **modelo** se representa por uno o más diagramas.

Lenguaje de Modelado UML 2.0

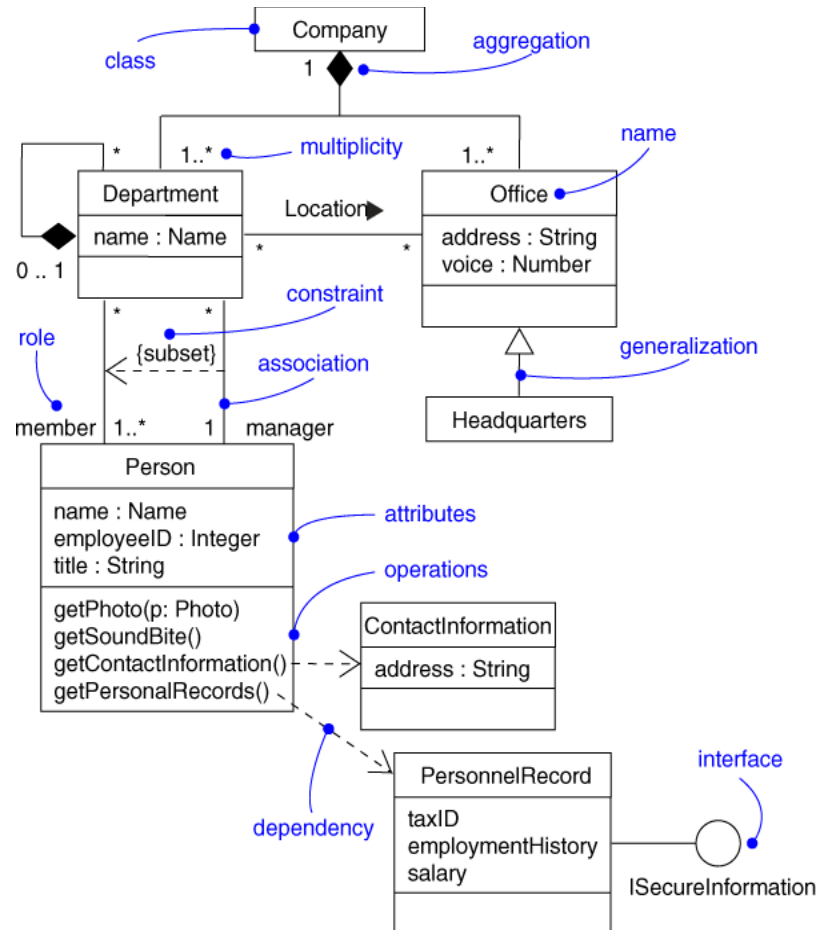


Lenguaje de Modelado UML 2.0

- En UML 2.0 hay 13 tipos diferentes de 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 (UML 2.0)
 - ✓ Diagrama de Despliegue
 - ✓ Diagrama de Paquetes

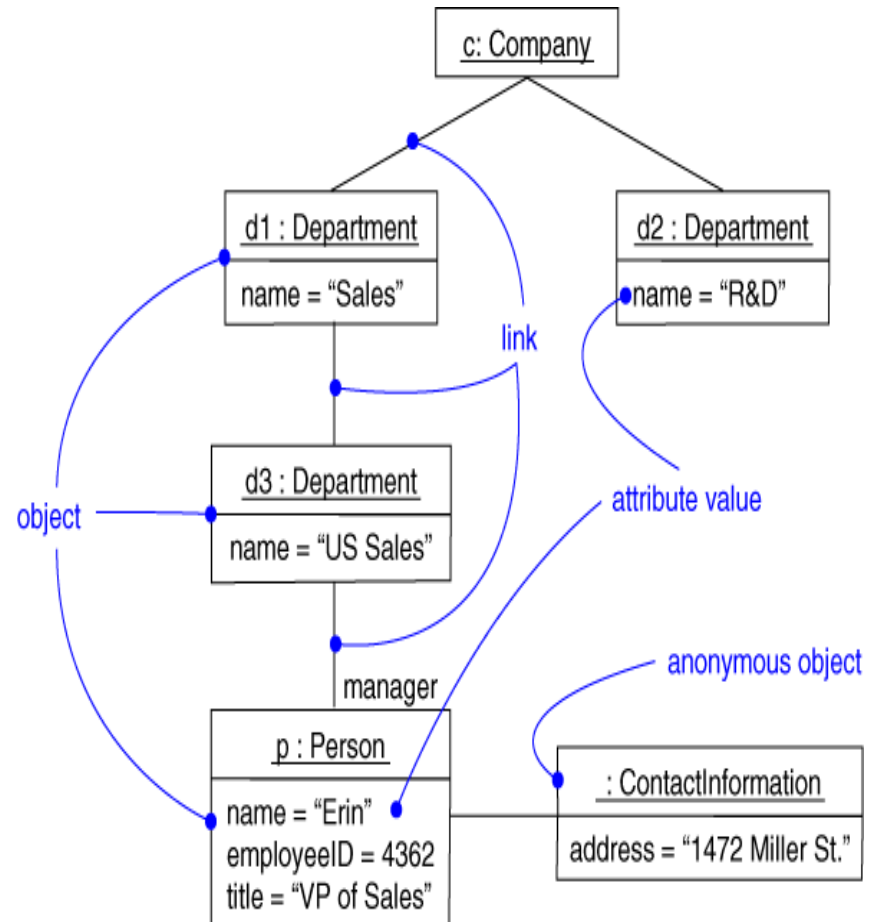
Diagramas de estructura: Diagramas de clase.

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



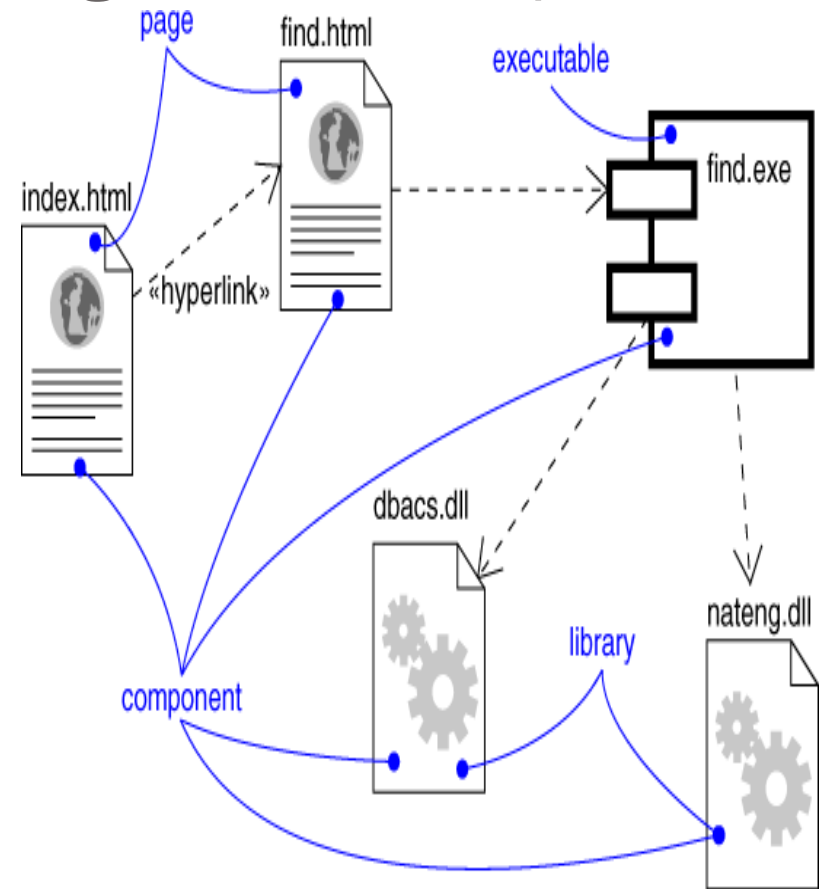
Diagramas de estructura: Diagrama de Objetos

Muestra una instantánea de un conjunto de objetos y sus relaciones.



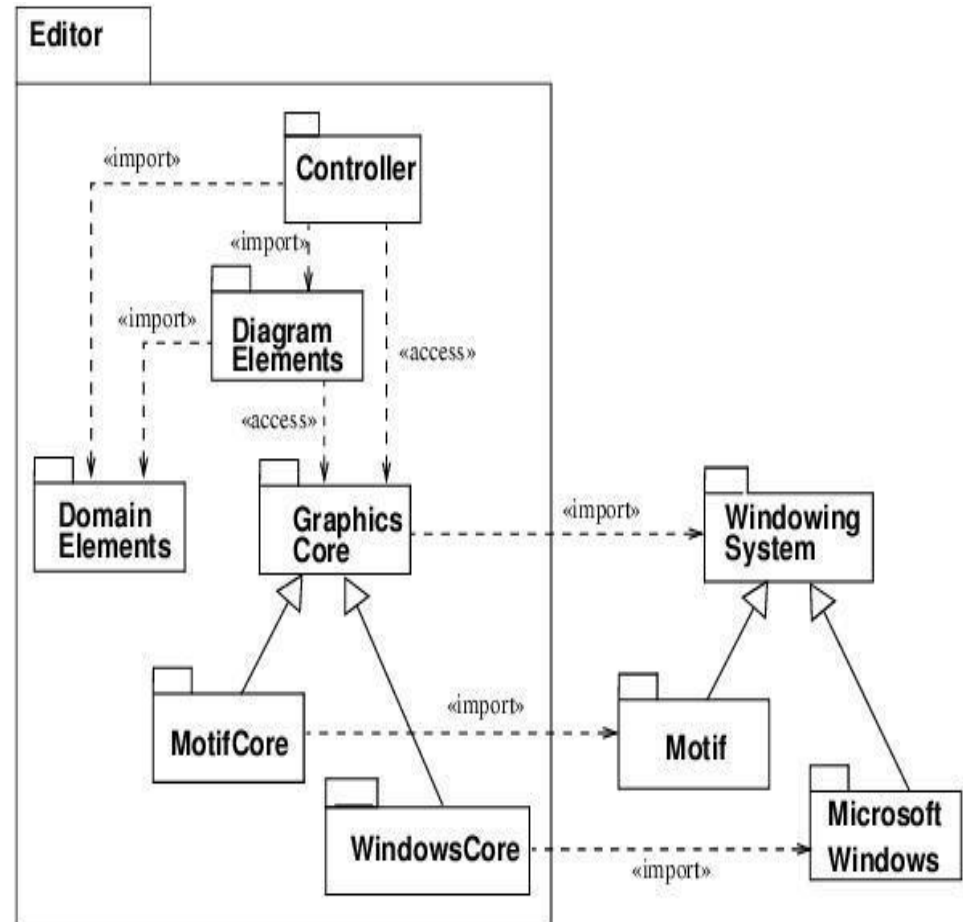
Diagramas de estructura: Diagrama de componentes

- 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.



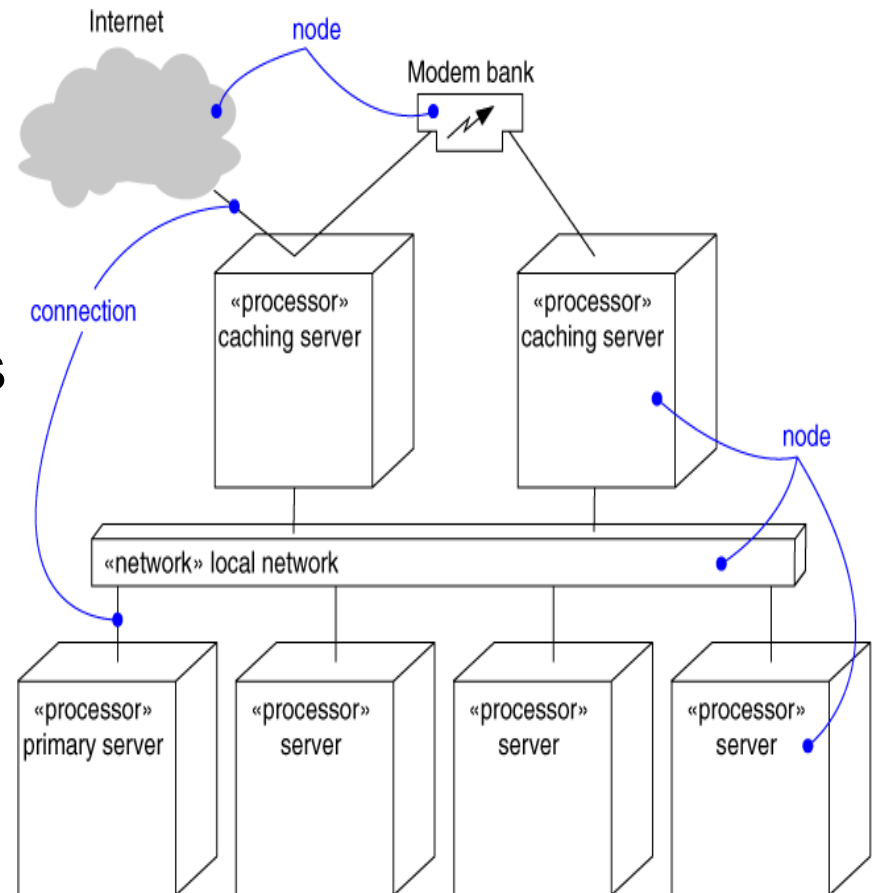
Diagramas de estructura: Diagrama de paquetes

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.**

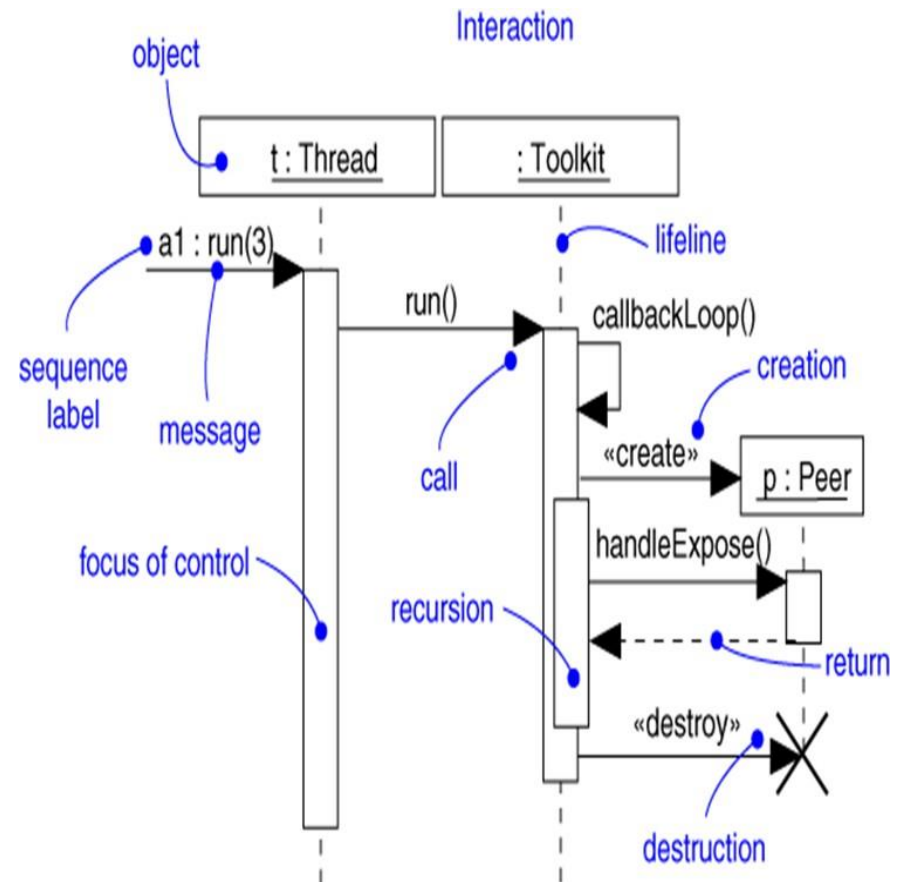


Diagramas de comportamiento

- Diagrama que expresa las secuencias de estados por los que pasa un objeto a lo largo de su vida en respuesta a eventos.
- Diagramas:
 - ✓ Diagramas de casos de uso
 - ✓ Diagrama de secuencia
 - ✓ Diagrama de colaboración
 - ✓ Diagrama de estados
 - ✓ Diagrama de actividades
 - ✓ Diagrama cronológico
 - ✓ Diagrama general de interacciones

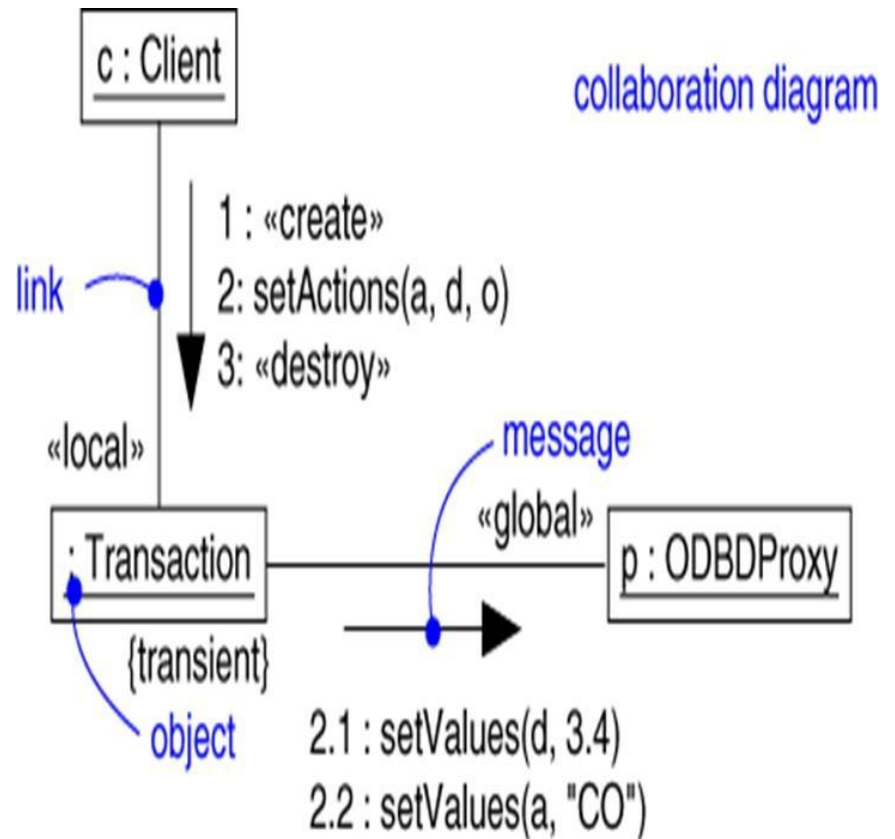
Diagramas de comportamiento: Diagrama de secuencia

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

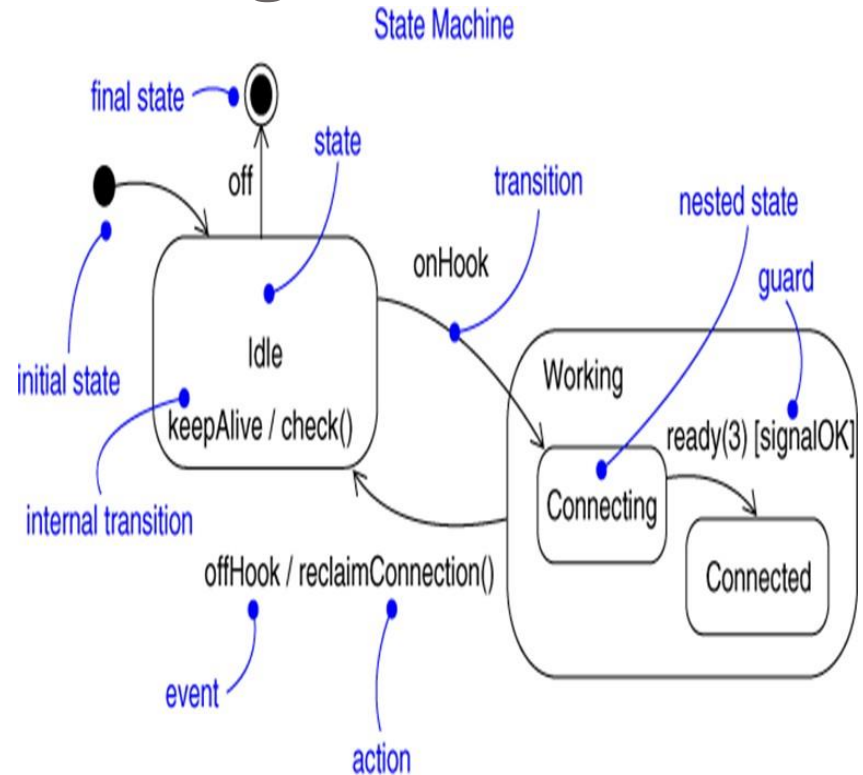
Un diagrama de interacción que pone el énfasis en la organización estructural de los objetos o roles que envían y reciben mensajes.



Diagramas de comportamiento: Diagrama de estados

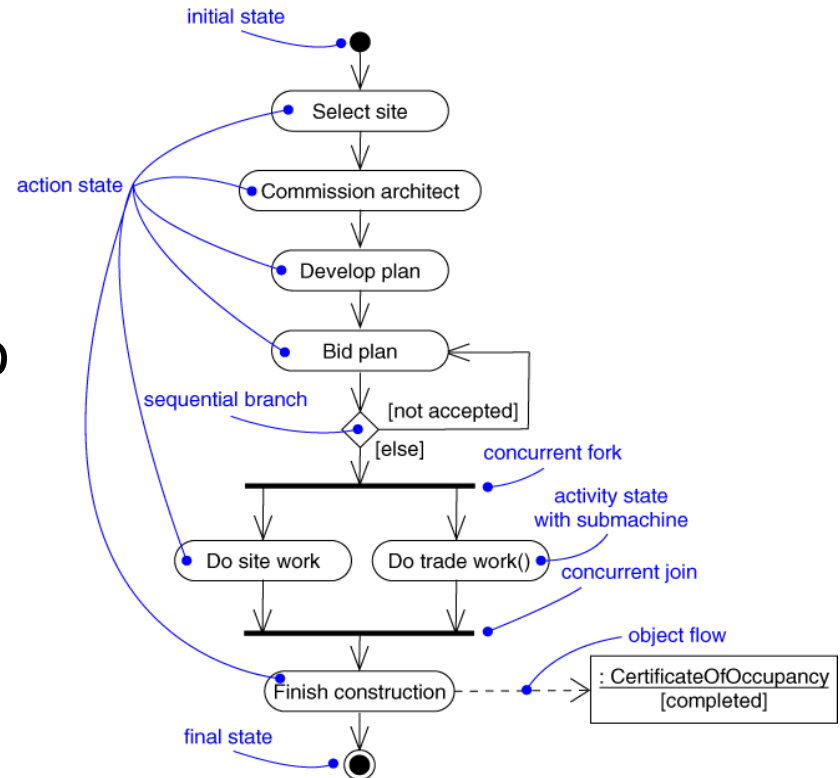
Muestra un **autómata** que consiste de estados, transiciones, eventos y actividades.

¡Máquina de Turing!



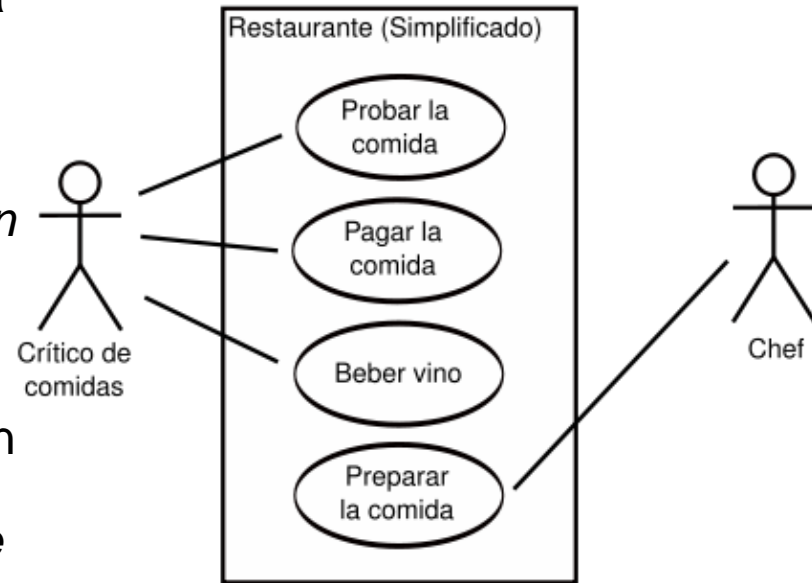
Diagramas de comportamiento: Diagrama de actividades

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



Diagramas de comportamiento: Diagramas de casos de uso

- Un diagrama de caso de uso es **una descripción** de las actividades que deberá realizar alguien o algo para llevar a cabo algún proceso (requisito funcional).
- Los personajes o *entidades que participarán* en un diagrama de caso de uso se denominan **actores**.
- En el contexto de ingeniería del software, un diagrama de caso de uso *representa a un sistema o subsistema como un conjunto de interacciones que se desarrollarán* entre casos de uso y entre estos y sus actores en respuesta a un evento que inicia un actor principal.



Diagramas de comportamiento: Diagramas de casos de uso

- No es orientado a objetos.
- Encapsula el sistema, visión de “caja negra”.
- Un caso de uso es un “uso típico del sistema”:
 - ✓ Secuencia de acciones realizadas por el sistema, que producen un resultado observable y valioso para un usuario en particular.
 - ✓ Tienen lugar bajo ciertas circunstancias:
 - 1) Es iniciado por un actor.
 - 2) Se puede describir como un conjunto de actividades.
 - 3) Produce un resultado de valor observable para algún actor.
- Se utiliza para representar la funcionalidad del sistema desde el punto de vista de los usuarios.
 - ✓ Presentan actores, casos de uso y relaciones

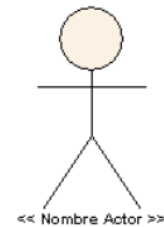
Diagramas de comportamiento: Diagramas de casos de uso

- Un caso de uso ayuda a responder las siguientes preguntas:
 - ✓ ¿Quién hace qué?
 - ✓ ¿Cuándo lo hace?
 - ✓ ¿Qué actividades se realizan?
 - ✓ ¿Qué elementos del sistema se utilizan?

Diagramas de comportamiento: Diagramas de casos de uso

➤ ¿Qué es un actor?

Un Actor es una idealización de una persona externa, de un proceso, o de una cosa que interactúa con un sistema. Pueden existir actores secundarios que intervienen, pero no son esenciales.



➤ ¿Qué es un escenario?

Un escenario es una instancia de un caso de uso.

En un caso de uso, pueden haber varios escenarios, ya que un caso de uso es una unidad de funcionalidad, proporcionada por el sistema.

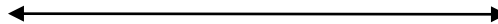


Diagramas de comportamiento: Diagramas de casos de uso

➤ ¿Qué es una relación de comunicación?

Vincula a un actor con un caso de uso. Estas relaciones pueden permitir:

- ✓ Información que un actor puede suministrar al sistema.
- ✓ Cambios en el sistema que son transmitidos al actor.
- ✓ Servicios que el sistema debe suministrar a los actores.



Diagramas de comportamiento: Diagramas de casos de uso

- Los diagramas de casos de uso, se pueden especificar de forma textual, por ejemplo:

Nombre	Nombre caso uso
Actores	Persona que demanda la funcionalidad
Objetivo	Objetivo específico del caso de uso
Precondiciones	Acciones que se deben cumplir para iniciar el caso de uso.
Postcondición	Estado del sistema tras realizar la acción.
Escenario básico	Secuencia de pasos entre el usuario y el sistema.

Diagramas de comportamiento: Diagramas de casos de uso

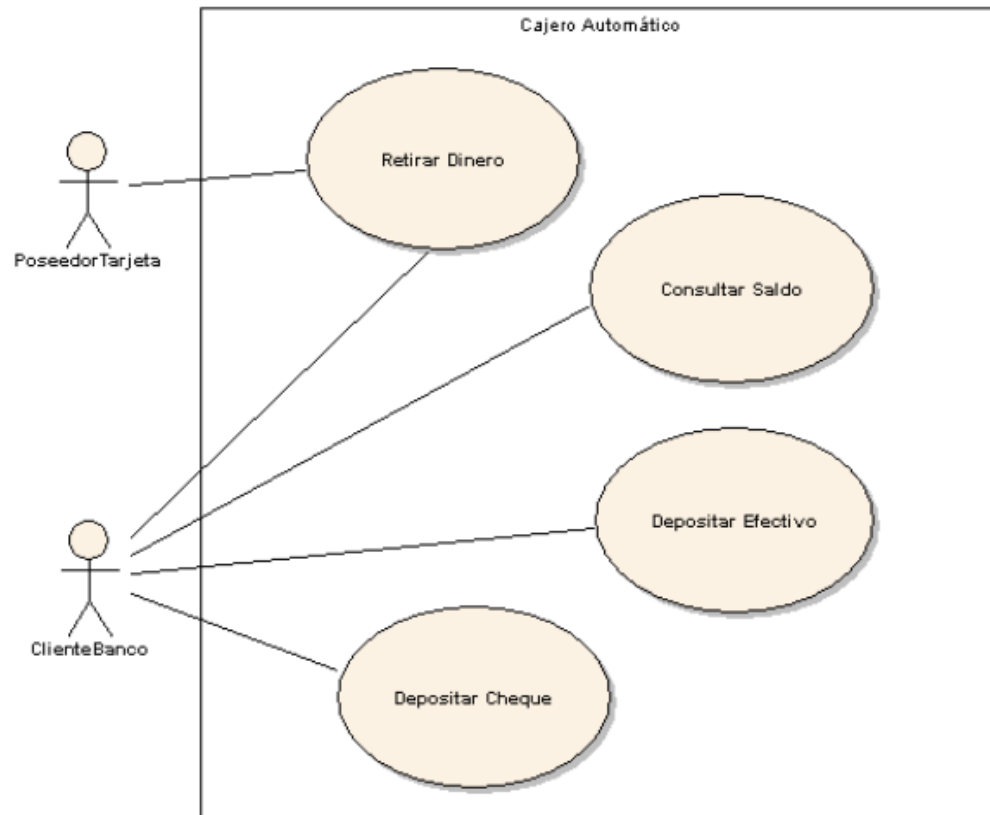
Modela los requerimientos de un sistema simplificado de Cajero Automático, que ofrece los siguientes servicios:

- ✓ Distribución de dinero a cada poseedor de una tarjeta inteligente a través de un lector de tarjetas y un distribuidor de efectivo.
- ✓ Consulta del saldo de cuentas, facilidades para depósito de efectivo y cheques para los clientes del banco poseedor de una tarjeta del mismo.

Paso a realizar para la resolución de un caso de uso:

- 1) Identificamos los actores.
- 2) Identificamos los casos de uso.
- 3) Identificamos relaciones entre ellos.
- 4) Construimos un diagrama de casos de uso.

Diagramas de comportamiento: Diagramas de casos de uso



Nombre

Consultar Saldo

Actores

ClienteBanco

Objetivo

Obtener la cantidad de saldo disponible en su cuenta

Precondiciones

Tiene que tener cuena del banco

Postcondición

Saldo mostrado

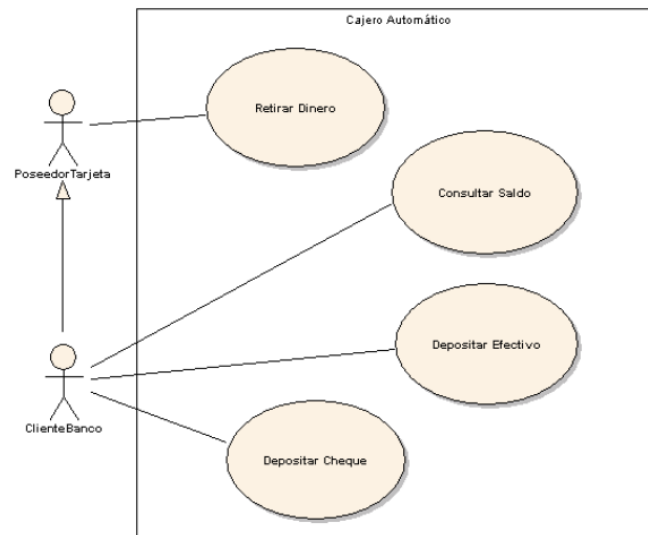
Escenario básico

- Inserta tarjeta
- Introduce contraseña
- Selecciona la opción de consultar saldo
- Retira tarjeta

Diagramas de comportamiento: Diagramas de casos de uso

➤ Generalización-especialización entre actores:

Indicarían que un actor es más general que otro si A es una especialización de B, una instancia de A podrá comunicarse con los mismos casos de uso que B



Diagramas de comportamiento: Diagramas de casos de uso

➤ Relaciones entre casos de uso:

a) Extensión (extend)

- ✓ El caso de uso final se puede extender con el comportamiento del caso de uso inicial en un punto concreto del primero. Es decir, si A <<extend>> B , significa que una instancia del caso de uso B podría incorporar el comportamiento especificado en A (si se cumplen las condiciones especificadas en el punto de extensión).
- ✓ El comportamiento se añadiría en el punto de extensión de B, referenciado por la relación extend.
- ✓ **Un punto de extensión** es una referencia al interior del caso (B), hacia el punto donde se podrán insertar secuencias de acciones de otros casos (A)

Diagramas de comportamiento: Diagramas de casos de uso



Diagramas de comportamiento: Diagramas de casos de uso

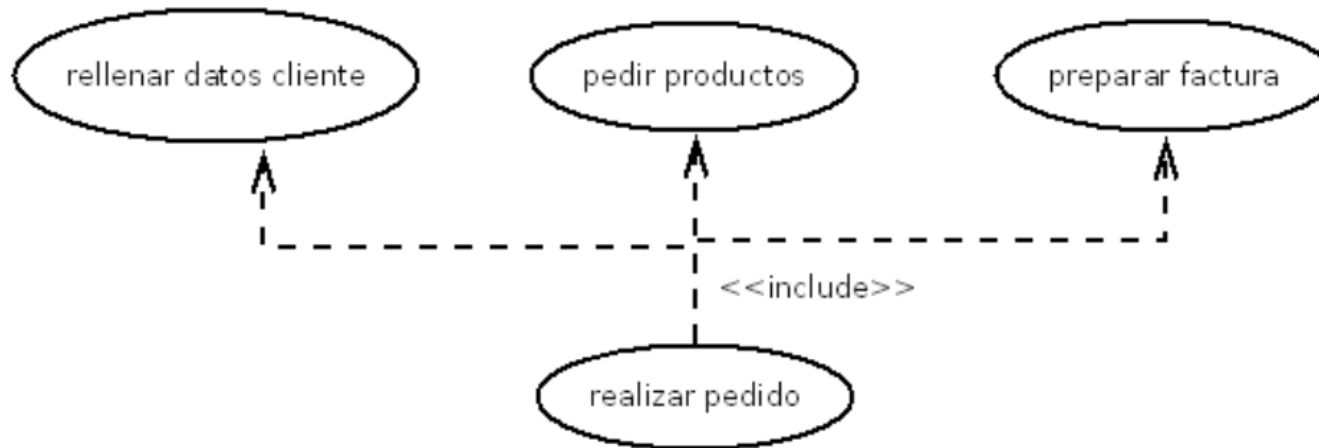
➤ Relaciones entre casos de uso:

a) Extensión (extend)

b) Inclusión (include)

- ✓ El caso de uso inicial incluye el comportamiento del caso de uso final (subcasos).
- ✓ Una relación A << include >> B significa que una instancia de A también incorporaría el comportamiento especificado en B.
- ✓ Se incorporaría en el lugar indicado en A.

Diagramas de comportamiento: Diagramas de casos de uso



Diagramas de comportamiento: Diagramas de casos de uso

➤ Relaciones entre casos de uso:

- a) Extensión (extend)
- b) Inclusión (include)
- c) Generalización
 - ✓ Un caso de uso hereda de otro.
 - ✓ El caso de uso secundario hereda las acciones y significado del primario y además agrega sus propias acciones.



Ejercicios:

- 1) Modelar el diagrama de casos de uso de una máquina expendedora de café.
- 2) Cajero automático 2.0:
 - El cliente, mete la tarjeta e introduce el pin o contraseña. El cajero le suministra el dinero al cliente y un recibo con el movimiento y saldo.
 - Por otro lado, el cajero queda vacío y el empleado agrega dinero al cajero.
- 3) El sistema de una biblioteca el usuario pide prestado un libro a la bibliotecaria, posteriormente, lo devuelve y la bibliotecaria cada día actualiza el catálogo de la biblioteca.
- 4) Agencia de viajes:
 - Un cliente entra en una agencia de viajes y pide al agente que le reserve un vuelo. El agente de viajes toma nota de los detalles relevantes, comprueba los calendarios de vuelo, y entonces propone un itinerario que el cliente acepta.
 - El agente realizar la reserva. El cliente paga el vuelo, y entonces el agente confirma la reserva. El agente emite el billete y se lo entrega.
- 5) Sistema de compras:
 - Se necesita realizar un sistema que permita el acceso al personal del departamento de compras el hacer pedidos, teniendo en cuenta que se quiere contemplar el que haya pedidos urgentes.
 - Una vez hecho el pedido se quiere poder realizar un seguimiento del mismo
 - Tanto para hacer el seguimiento como para poder realizar pedidos hay que estar validado en el sistema, que se puede hacer conociendo la clave o por reconocimiento de la retina.