

UT6 DIAGRAMAS DE COMPORTAMIENTO

RESULTADOS DE APRENDIZAJE ASOCIADOS
6. Genera diagramas de comportamiento valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno
CRITERIOS DE EVALUACIÓN
a) Se han identificado los distintos tipos de diagramas de comportamiento.
b) Se ha reconocido el significado de los diagramas de casos de uso.
c) Se han interpretado diagramas de interacción.
d) Se han elaborado diagramas de interacción sencillos.
e) Se ha interpretado el significado de diagramas de secuencia.
f) Se han elaborado diagramas de secuencia sencillos.
g) Se ha interpretado el significado de diagramas de colaboración
h) Se han elaborado diagramas de colaboración sencillos.
i) Se ha interpretado el significado de diagramas de actividades
j) Se han elaborado diagramas de actividades sencillos.
k) Se han interpretado diagramas de estados.
l) Se han planteado diagramas de estados sencillos.

UT6 DIAGRAMAS DE COMPORTAMIENTO

Índice de contenido

1.- Contenidos.....	3
2.- Introducción	3
3.- Diagramas de casos de uso	5
3.1.- Identificar casos de uso	7
3.2.- Relaciones en un diagrama de casos de uso	14
3.2.1- Diferencias entre include y extend	17
4.- Diagramas de interacción.....	18
4.1- Diagramas de secuencia	19
5.- Diagramas de comunicación o colaboración	21
6.- Diagramas de actividad	22
7.- Diagramas de estados	24
8.- Referencias bibliográficas	26

1.- Contenidos

En el capítulo anterior vimos cómo se elabora un diagrama de clases. Esta herramienta UML resulta ser de gran utilidad y esencial para el análisis y diseño en la ingeniería de software orientada a objetos. Es cierto que este tipo de diagrama se considera un elemento clave; sin embargo, **aún requiere de complementos para abarcar íntegramente los principios de la programación orientada a objetos.**

Por ejemplo, los diagramas de clases por sí solos no logran captar cómo el usuario interactuará con el sistema ni contemplan consideraciones esenciales tales como la instanciación y eliminación de objetos, o la transmisión y manejo de mensajes entre estos.

Por tanto, si bien los diagramas de clases solucionan la representación de la estructura fija de un sistema, para la definición de su comportamiento dinámico, UML ofrece un conjunto adicional de instrumentos. Entre estos se incluyen, pero no se limitan a, los siguientes tipos de diagramas comportamentales:

- Diagramas de casos de uso.
- Diagramas de interacción.
- Diagramas de secuencia.
- Diagramas de comunicación o colaboración.
- Diagramas de actividades.
- Diagramas de estados

2.- Introducción

Como ya se ha dicho, los **diagramas de clases** nos dan información sobre la **estructura estática del sistema**, pero no nos aportan información sobre el comportamiento **dinámico** del mismo.

Para modelar este tipo de información, utilizaremos **los diagramas de comportamiento.**

Se incluyen los siguientes tipos:

Tipo de Diagrama	Descripción / Función
Casos de uso	Representan las funcionalidades del sistema desde la perspectiva del usuario, mostrando la interacción entre el usuario y los distintos elementos del sistema.
Interacción	Describen cómo objetos o instancias interactúan para llevar a cabo un proceso o parte de un proceso, enfocándose en el flujo de mensajes y la secuencia de estos.
Secuencia	Específicos del tiempo y el orden en que los objetos interactúan entre sí mediante mensajes, ideal para visualizar secuencias de eventos.
Colaboración	Se centran en la organización y relaciones entre objetos o roles, mostrando cómo cooperan para cumplir una funcionalidad.
Actividades	Muestran el flujo de control o flujo de trabajo paso a paso a través de la funcionalidad del sistema, similares a los diagramas de flujo.
Estados	Detallan los estados por los que puede pasar un objeto durante su vida útil, las transiciones entre estos estados y las actividades durante estos cambios.

A continuación, se muestra una tabla con una descripción algo menos formal, pero para que se entienda mejor:

Tipo de Diagrama	Descripción / Función Formal	Ejemplo Sencillo
Casos de uso	Representan las funcionalidades del sistema desde la perspectiva del usuario, mostrando la interacción entre el usuario y los distintos elementos del sistema.	Como cuando abres una nueva aplicación de teléfono móvil y te aparece una guía rápida que señala: "Aquí tocas para escribir un mensaje", "Desliza aquí para ver notificaciones", "Mantén presionado para ver opciones". Cada indicación es un "caso de uso" mostrando lo que puedes hacer y cómo.
Interacción	Describen cómo objetos o instancias interactúan para llevar a cabo un proceso o parte de un proceso, enfocándose en el flujo de mensajes y la secuencia de estos.	Como en una receta de cocina donde cada ingrediente y utensilio 'interactúan' siguiendo los pasos para preparar un pastel.
Secuencia	Específicos del tiempo y el orden en que los objetos interactúan entre sí mediante mensajes, ideal para visualizar secuencias de eventos.	Como un guion de una obra de teatro que muestra el orden exacto en que los actores entran y hablan.
Colaboración	Se centran en la organización y relaciones entre objetos o roles, mostrando cómo cooperan para cumplir una funcionalidad.	Como un plan de emergencia que muestra quién debe llamar al 911, quién usa el extintor y quién evacua el edificio.
Actividades	Muestran el flujo de control o flujo de trabajo paso a paso a través de la	Como los pasos que sigues para armar un mueble: primero las

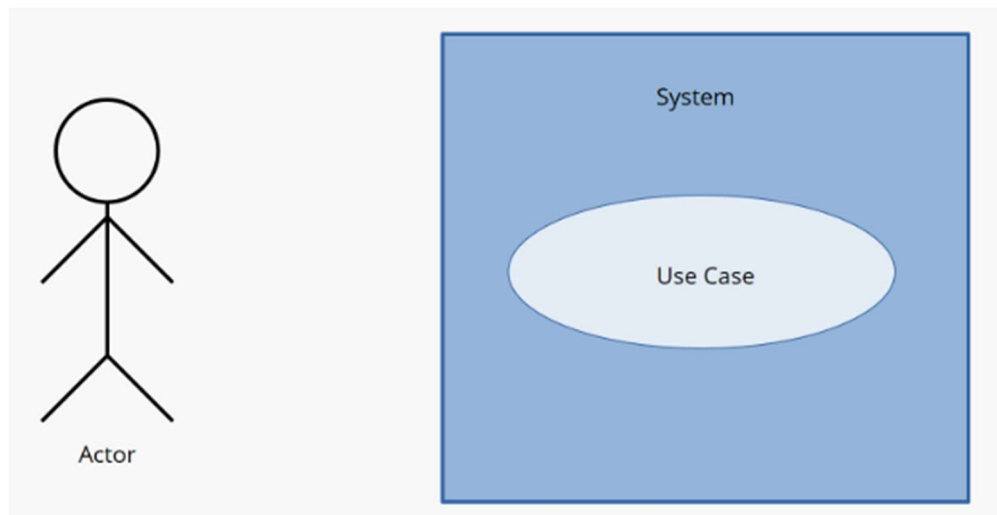
	funcionalidad del sistema, similares a los diagramas de flujo.	patas, luego la base, y así sucesivamente.
Estados	Detallan los estados por los que puede pasar un objeto durante su vida útil, las transiciones entre estos estados y las actividades durante estos cambios.	Como un seguimiento del estado de un paquete: recibido, en tránsito, entregado.

En este tema veremos algunos de estos diagramas a través de ejemplos sencillos.

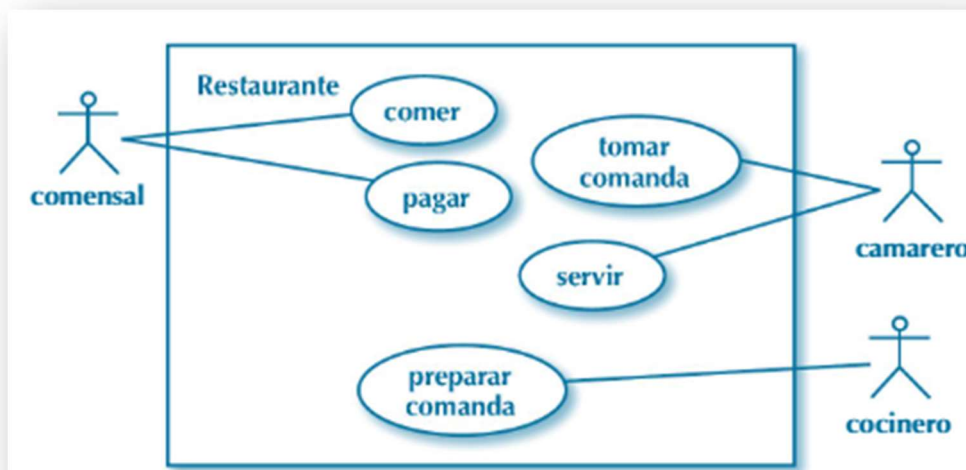
3.- Diagramas de casos de uso

Los diagramas de casos de uso son herramientas clave para los analistas, ya que permiten definir cómo los usuarios interactuarán con un sistema. Estos diagramas ilustran la relación entre el sistema y el mundo exterior:

1. El **sistema** se representa con un rectángulo e incluye todo lo que se programará. Todo lo que está fuera del rectángulo, como los actores, son elementos que pueden interactuar con el sistema, pero no son parte de su programación interna. El sistema tiene un nombre que usualmente se coloca en la parte superior.
2. Los **casos de uso** son las acciones o funcionalidades que el sistema puede realizar. Cada caso de uso está vinculado a métodos de diferentes clases dentro de la programación y se identifica con un nombre específico que indica su función.
3. El **actor** es una persona o rol que utiliza el sistema, interactuando con él a través de los casos de uso. Cada actor está identificado con un nombre o rol y puede tener varios roles o casos de uso asociados a él.
4. Las **relaciones**. Existen varios tipos de relaciones que veremos más adelante. La más típica es la asociación entre actores y casos de uso que se representa mediante una línea continua. Significa la participación del actor en el caso de uso.



Ejemplo de un diagrama de **casos de uso** sencillo:

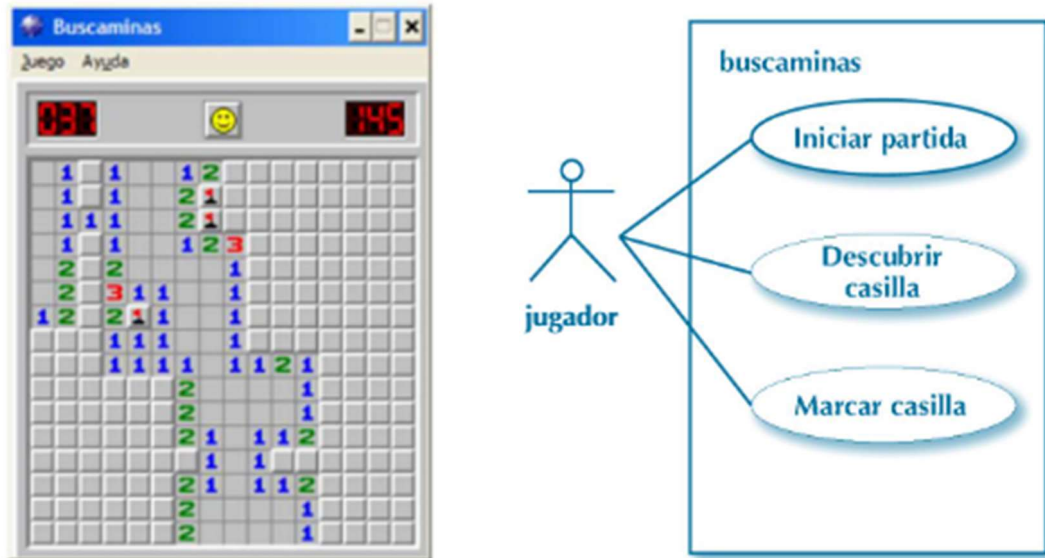


NOTA: Si hay una persona que realiza distintos roles, no agrupes los roles en un único actor, puesto que, en un futuro, dichas tareas podrían ser realizadas por distintas personas o actores.



Más notas: Intenta no realizar el diagrama de casos de uso pensando en el diagrama de clases. Es cierto que están relacionados, pero es mejor estrategia diseñar ambos de forma independiente.

Otro ejemplo de caso de uso:



Debemos preguntarnos lo siguiente:

- ¿Qué acciones se realizan en este sistema? → Identificamos los diferentes **casos de uso**
- ¿Quién ejecuta esas acciones? → Identificamos los **actores**

3.1.- Identificar casos de uso

Los casos de uso **se documentan con texto informal**. Describen tanto lo que hace el acto como lo que hace el sistema cuando interactúa con él.

Una plantilla para una especificación sencilla de casos de uso contiene la siguiente información:

1. **Nombre del caso de uso:** Es el título que se le da al caso de uso y que generalmente refleja la acción principal o el objetivo que se logra con él. Debe ser descriptivo y claro, por ejemplo, "Registrar Usuario" o "Procesar Pago".
2. **ID del caso de uso:** Es un identificador único para cada caso de uso, lo que facilita la referencia y organización dentro del proyecto. Suele ser un número o una combinación de letras y números.

3. **Breve descripción:** Aquí se escribe una o dos frases que resumen qué hace el caso de uso, es decir, cuál es el objetivo del caso de uso y por qué es importante.
4. **Actores implicados** en el caso de uso: Son las entidades (usualmente personas, sistemas o dispositivos) que interactúan directamente con el sistema. En esta sección se listan todos los actores que participan en el caso de uso.
5. **Precondiciones:** Estas son las condiciones que deben cumplirse antes de que el caso de uso pueda empezar. Establecen el contexto necesario para que se pueda llevar a cabo la acción.
6. **Curso normal:** Es la secuencia de pasos que describen la interacción estándar entre los actores y el sistema para lograr el objetivo del caso de uso. Se escribe en forma de lista numerada y refleja el flujo principal de eventos.
7. **Postcondiciones:** Son el estado en el que el sistema debe encontrarse después de que se completa el caso de uso. Es el resultado de ejecutar el caso de uso si todo ha ido según el curso normal.
8. **Alternativas:** Aquí se describen los flujos alternativos o excepciones al curso normal. Se incluyen situaciones como errores, condiciones inesperadas o acciones alternativas que los actores podrían realizar. Es importante para comprender cómo el sistema debe manejar situaciones que se desvían del curso esperado.

Veamos todo esto mejor con un ejemplo:

Nombre: Jugar Partida de Buscaminas
ID: CU-1

Descripción: El usuario inicia una nueva partida de Buscaminas y juega hasta que gana la partida o revela una mina.
Actores: Usuario (jugador)
Precondiciones: <ul style="list-style-type: none"> El usuario tiene que seleccionar la dificultad del juego. El juego está cargado y listo para iniciar una nueva partida.
Curso normal: <ol style="list-style-type: none"> El usuario inicia una nueva partida. El juego despliega un tablero con un número específico de casillas ocultas. El usuario selecciona una casilla para revelar. Si la casilla es una mina, el juego termina. Si la casilla no es una mina, muestra un número con las minas adyacentes o un espacio en blanco si no hay minas cercanas. El usuario continúa revelando casillas hasta que todas las no minadas están descubiertas y gana el juego.
Postcondiciones: <ul style="list-style-type: none"> Si el usuario gana, se muestra un mensaje de victoria. Si el usuario revela una mina, se muestra un mensaje de derrota y se revelan todas las minas.
Alternativas: <ul style="list-style-type: none"> Si el usuario marca una casilla como mina y es incorrecto, puede desmarcarla y seleccionar otra. Si el usuario decide terminar el juego antes de revelar una mina o ganar, puede seleccionar la opción de rendirse o reiniciar.

Vemos otro ejemplo diferente para otro caso de uso distinto:

Nombre: Reservar una Mesa en Restaurante
ID: CU-2
Descripción: El usuario realiza una reserva en línea para asegurar una mesa en un restaurante específico en la fecha y hora deseada.
Actores: Cliente (usuario que realiza la reserva)
Precondiciones:

- El usuario debe tener una cuenta en la plataforma de reservas del restaurante.
- El restaurante debe tener disponibilidad en la fecha y hora seleccionadas por el usuario.

Curso normal:

1. El usuario inicia sesión en la plataforma de reservas del restaurante.
2. El usuario selecciona la fecha y hora para la reserva.
3. El usuario elige el número de comensales.
4. El usuario revisa la disponibilidad y confirma la reserva.
5. El sistema registra la reserva y envía una confirmación al correo electrónico del usuario

Postcondiciones:

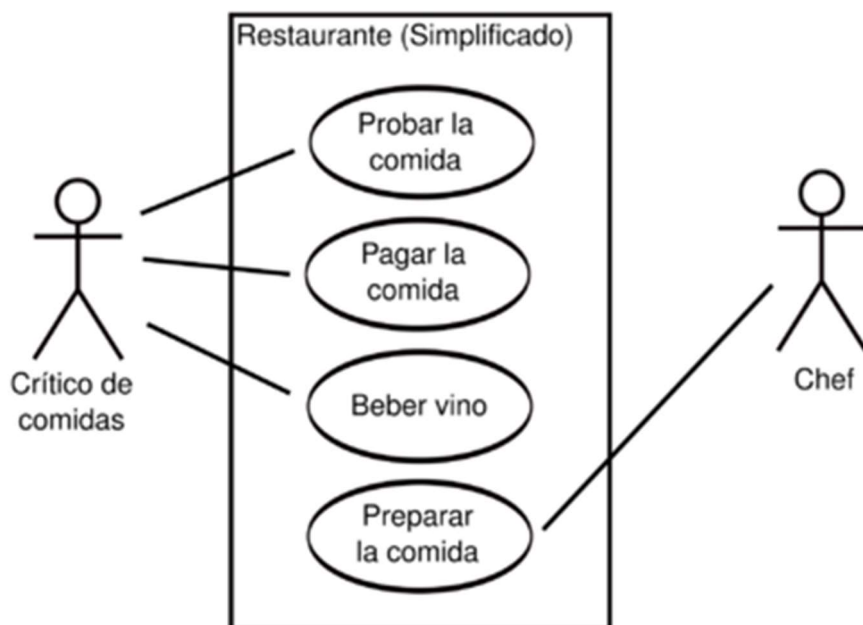
- El usuario recibe una notificación por correo electrónico con los detalles de la reserva.
- El restaurante tiene la mesa reservada para el usuario en la fecha y hora acordadas.

Alternativas:

- Si no hay disponibilidad, el usuario puede elegir otra fecha u hora.
- Si el usuario no recibe la confirmación por correo, puede contactar directamente al restaurante para verificar el estado de la reserva.

Ahora veamos un ejemplo de Caso de uso y sus plantillas:

Caso de uso:



Las correspondientes plantillas de especificación del diagrama de casos de uso:

Nombre: Probar la comida
ID: CG-PROBAR-001
Descripción: El crítico gastronómico evalúa la presentación, el sabor y la calidad de los platos servidos en el restaurante.
Actores: Crítico Gastronómico
Precondiciones: <ul style="list-style-type: none"> El restaurante ha sido notificado de la visita del crítico, pero el personal no sabe su identidad para asegurar una evaluación imparcial. El crítico gastronómico ha sido sentado y le han presentado el menú.
Curso normal: <ol style="list-style-type: none"> El crítico selecciona diversos platillos para probar un amplio espectro de la oferta culinaria. El personal de cocina prepara y sirve los platillos. El crítico evalúa meticulosamente cada plato por su presentación, sabor, textura y creatividad.

Postcondiciones:

- El crítico tiene suficiente información para redactar su reseña culinaria.

Alternativas:

- Si algún plato no cumple con los estándares esperados, el crítico podría solicitar una entrevista con el chef.

Nombre: Pagar la comida

ID: CG-PAGAR-002

Descripción: El crítico gastronómico evalúa la experiencia del servicio al cliente y procedimientos de pago del restaurante.

Actores: Crítico Gastronómico

Precondiciones:

- El crítico ha finalizado su degustación de platos
- La cuenta ha sido presentada al crítico.

Curso normal:

1. El crítico revisa la cuenta para asegurarse de que todos los cargos son correctos.
2. El crítico completa el pago y observa la eficiencia y cortesía del personal.
3. El crítico evalúa la experiencia general de pago.

Postcondiciones:

- El crítico ha completado la transacción y tiene la información necesaria para su reseña sobre el servicio.

Alternativas:

- Si hay discrepancias en la cuenta, el crítico lo discute con el personal para entender el procedimiento de resolución.

Nombre: Beber vino
ID: CG-BEBER-003
Descripción: El crítico gastronómico examina la selección de vinos del restaurante y su maridaje con los platos.
Actores: Crítico Gastronómico
Precondiciones: <ul style="list-style-type: none"> • El crítico tiene acceso a la carta de vinos. • Se han servido los platillos seleccionados para la cata
Curso normal: <ol style="list-style-type: none"> 1. El crítico elige un vino de la carta. 2. El personal sirve el vino. 3. El crítico degusta el vino y evalúa su calidad y cómo complementa la comida.
Postcondiciones: El crítico ha experimentado el maridaje de vinos y tiene detalles para su crítica.
Alternativas: Si el vino no está a la altura, el crítico solicita otro y compara las selecciones.

Y la última plantilla de especificación de casos de uso:

Nombre: Preparar la comida
ID: CG-PREPARAR-004
Descripción: El chef realiza la preparación de los platillos que serán evaluados por el crítico gastronómico, mostrando sus habilidades culinarias y técnicas de cocina.
Actores: Chef
Precondiciones: <ul style="list-style-type: none"> • El chef tiene la lista de platos que serán evaluados por el crítico. • Todos los ingredientes necesarios están disponibles en la cocina. • El crítico está presente en el restaurante y listo para observar.

Curso normal:

1. El chef organiza su lugar de trabajo con los ingredientes y utensilios necesarios.
2. Siguiendo las recetas y estándares del restaurante, el chef comienza la preparación de los platos.
3. Durante la preparación, el chef demuestra técnicas de cocina y manejo de ingredientes.
4. Una vez terminados, los platos son presentados al crítico para su evaluación.

Postcondiciones:

- El chef ha completado la preparación de los platos de manera artística y técnica.
- Los platos están listos para ser servidos y evaluados por el crítico gastronómico.


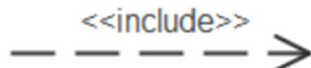
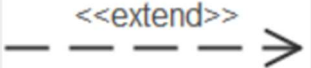
Alternativas:


- Si el crítico solicita un plato no previsto, el chef debe adaptarse y preparar la solicitud especial.
- Si el crítico desea entender una técnica específica, el chef puede hacer una demostración detallada.

Lo siguiente que deberemos hacer es describir las diferentes relaciones existentes en un diagrama de casos de uso.

3.2.- Relaciones en un diagrama de casos de uso

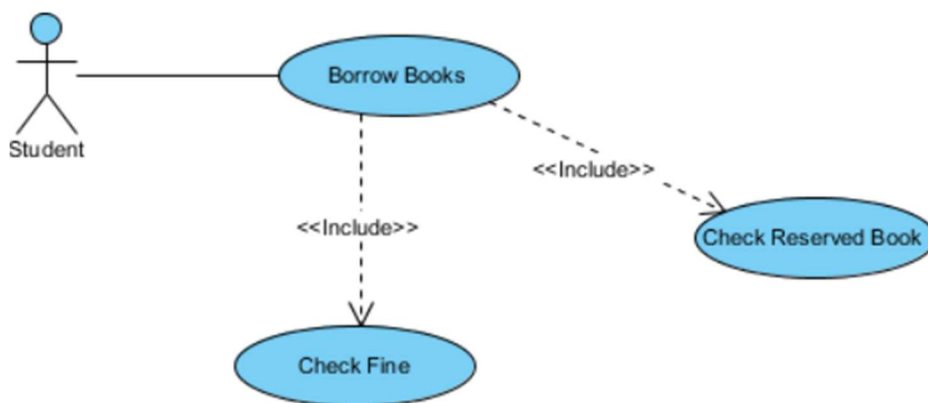
En un diagrama de casos de uso nos podemos encontrar con los siguientes tipos de relaciones:

Relación	Función	Notación
Asociación	Indica que un actor está involucrado en un caso de uso.	
Inclusión (<i>Include</i>)	Representa una situación donde un caso de uso incluye la funcionalidad de otro.	
Extensión (<i>Extend</i>)	Muestra que, bajo ciertas condiciones, un caso de uso puede	

	extender el comportamiento de otro.	
Generalización	Indica que un caso de uso hereda de otro, similar a la herencia de clases en la programación orientada a objetos.	

Ejemplos:

- Inclusión:**

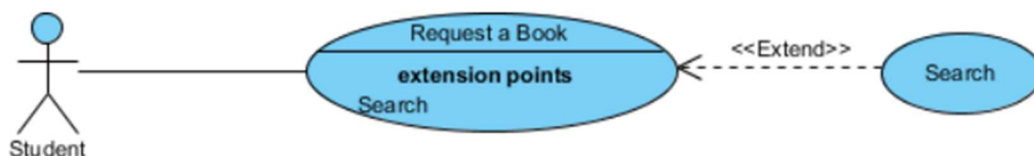


Explicación: Relación de inclusión (include): un caso de uso incluye el comportamiento completo de un caso de uso general.

- Permite la composición jerárquica de casos de uso, así como la reutilización entre casos de uso.

- Extensión:**

Una relación de **extensión** para especificar que un caso de uso (extensión) amplía el comportamiento de otro caso de uso (base). Este tipo de relación revela detalles sobre el sistema o la aplicación que normalmente están ocultos en el caso de uso



Explicación: Relación de extensión (*extend*): un caso de uso añade acciones, que pueden ser opcionales, al comportamiento de un caso de uso general.

- El caso de uso extendido puede incluir comportamiento del caso de uso que se extiende, aunque no tiene que incluir todo el comportamiento.

- **Generalización:**

En cuanto a actores se refiere:

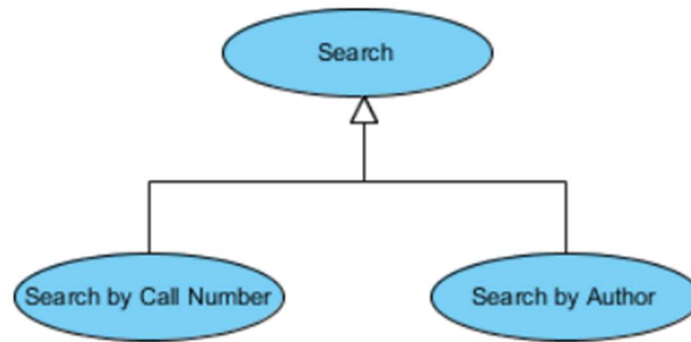


Generalización entre casos de uso:

Una relación de **generalización** es una relación en la que un elemento del modelo (hijo) se basa en otro elemento del modelo (padre).

Las relaciones de generalización se utilizan en diagramas de clases, diagramas de componentes, diagramas de implementación y diagramas de casos de uso para indicar que el elemento secundario acepta todos los atributos, operaciones y relaciones definidas en el elemento principal

Ejemplo:



3.2.1- Diferencias entre include y extend

En el caso de "Inclusión", el caso de uso incluido es un paso obligatorio y siempre se ejecuta como parte del caso de uso principal. En "Extensión", el caso de uso extendido solo se ejecuta bajo ciertas condiciones, añadiendo comportamiento adicional al caso de uso principal.

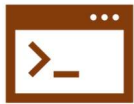
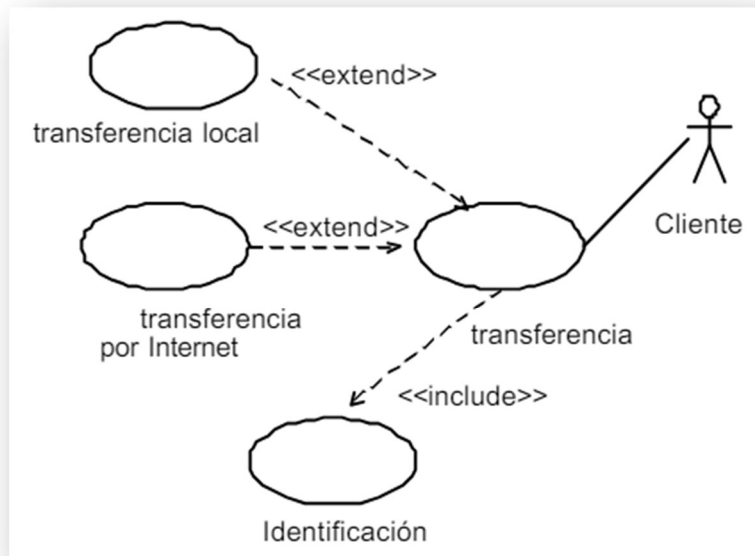
Más ejemplos:

- **Inclusión (*Include*):**

Ejemplo: Imagina que vas a un banco para abrir una cuenta nueva. El proceso "Abrir Cuenta" siempre incluirá "Verificar Identidad" como un paso necesario. No puedes abrir una cuenta sin verificar tu identidad, por lo que "Verificar Identidad" es un caso de uso que siempre se incluye dentro de "Abrir Cuenta".

- **Extensión (*Extend*):**

Ejemplo: Considera que estás utilizando una aplicación para ver el clima. El caso de uso principal es "Ver Pronóstico del Tiempo", que te da la información básica del clima actual y las previsiones. Ocasionalmente, si hay una alerta meteorológica, se extiende el caso de uso básico con "Mostrar Alerta de Clima Severo". Esta extensión solo ocurre bajo ciertas condiciones (por ejemplo, si hay una tormenta inminente), no cada vez que miras el pronóstico.



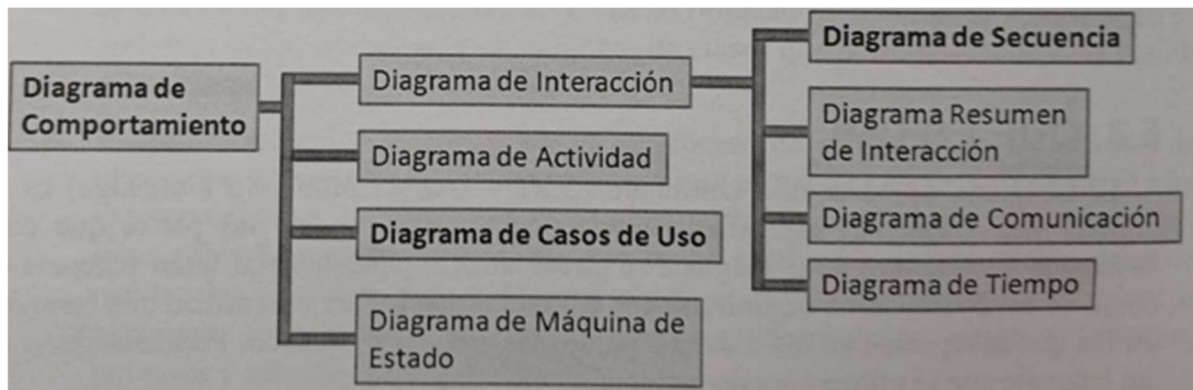
Enlace interesante:

<https://blog.visual-paradigm.com/es/the-four-types-of-relationship-in-use-case-diagram/>

4.- Diagramas de interacción

Los diagramas de interacción en UML 2.0 son herramientas que permiten visualizar el comportamiento dinámico de los sistemas, mostrando las interacciones entre diferentes elementos del modelo. Este conjunto de diagramas incluye los:

- Diagramas de Secuencia
- Diagramas de Comunicación
- Diagramas de Descripción General de la Interacción
- Diagramas de Tiempo.

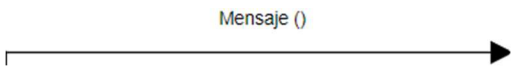

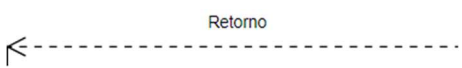



4.1- Diagramas de secuencia

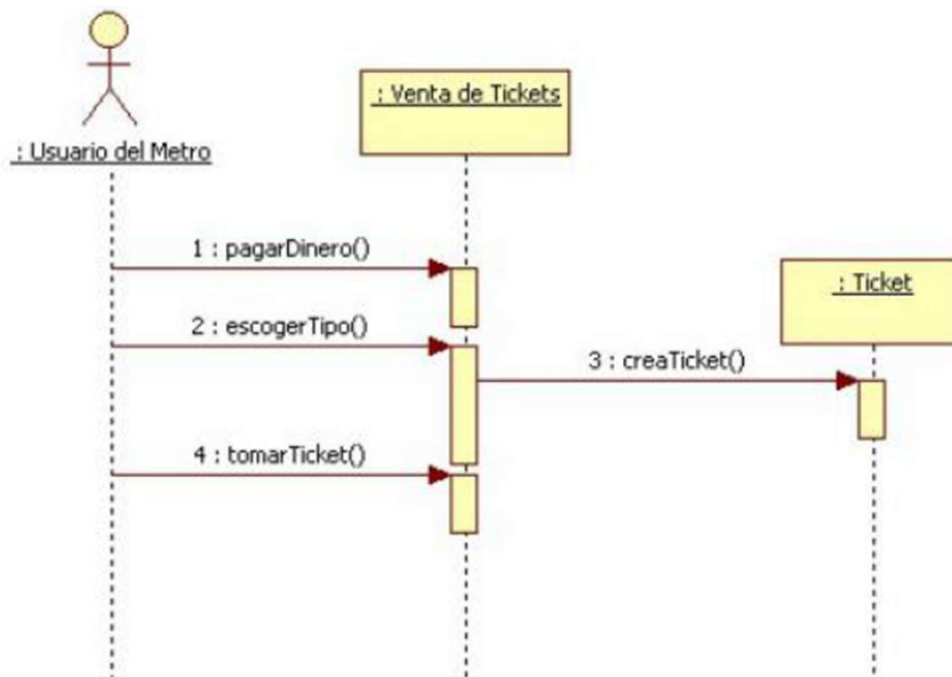
Los **diagramas de secuencia** permiten añadir una característica fundamental al análisis como es el **tiempo**. Es muy importante conocer las interacciones entre los distintos objetos del sistema, pero, si se conoce la secuencia y la temporización de estos, el desarrollador software tendrá una visión mucho más completa del trabajo que tiene que realizar en el sistema.

Los principales elementos en un diagrama de secuencia son los siguientes:

Símbolo	Función	Notación
Marco	Define el contexto o límite del diagrama de secuencia.	
Línea de vida	Representa la existencia de un objeto durante un período de tiempo en un diagrama.	
Actor	Representa un participante externo que interactúa con el sistema (puede ser un usuario humano o un sistema externo).	

Mensaje síncrono	Indica una llamada de operación donde el emisor espera una respuesta antes de continuar.	
Mensaje asíncrono	Muestra una llamada de operación donde el emisor no necesita esperar una respuesta para continuar.	
Mensaje de retorno	Indica que se ha completado una llamada de operación y se está devolviendo el control.	
Activación	Representa el período durante el cual un objeto está ejecutando una operación.	

Ejemplo de diagrama de secuencia:

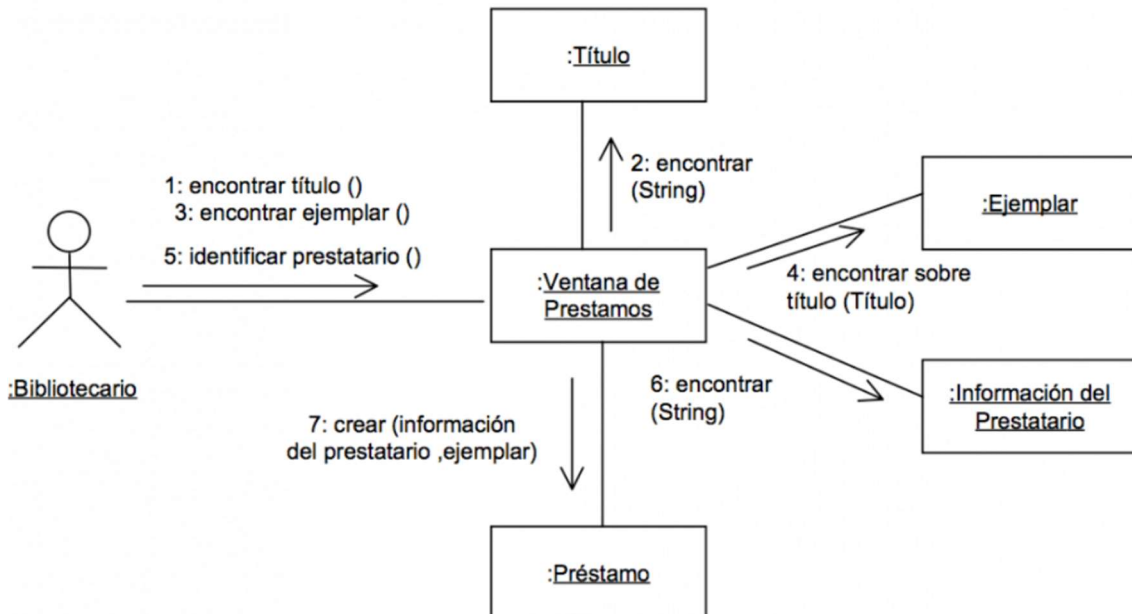


Explicación: En el diagrama, un usuario del metro realiza un proceso para obtener un ticket. El proceso comienza con el usuario pagando dinero (1), luego seleccionando el tipo de ticket que desea (2). Después, el sistema de venta de tickets procesa la información y crea el ticket (3). Finalmente, el usuario toma el ticket (4). Cada paso está numerado y muestra la secuencia de interacciones entre el usuario y el sistema.

5.- Diagramas de comunicación o colaboración

Un diagrama de colaboración muestra la misma información que un diagrama de secuencia, pero de forma diferente. En los diagramas de colaboración no existe una secuencia temporal en el eje vertical; es decir, la colocación de los mensajes en el diagrama no indica cuál es el orden en el que se suceden. Además, la colocación de los objetos es más flexible y permite mostrar de forma más clara cuáles son las colaboraciones entre ellos.

Ejemplo:



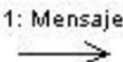


Explicación:

1. **Bibliotecario:** Encuentra el título del libro.
2. **Título:** Retorna una cadena de texto (probablemente el nombre del libro encontrado).
3. **Bibliotecario:** Encuentra un ejemplar del libro.
4. **Ejemplar:** Realiza una búsqueda basada en el título del libro.
5. **Bibliotecario:** Identifica al prestatario (la persona que tomará prestado el libro).

6. **Ventana de Préstamos:** Encuentra la información del prestatario en el sistema, que se representa como una cadena de texto (posiblemente el nombre o ID del prestatario).
7. **Préstamo:** Crea el préstamo con la información del ejemplar y del prestatario.




Los elementos que se utilizan en este tipo de diagramas son:

Símbolo	Función	Notación
Objetos o roles	Representan entidades o conceptos que interactúan.	
Enlaces	Conexiones entre objetos que pueden comunicarse.	
Mensajes	Llamadas y respuestas enviadas entre objetos.	
Número de secuencia	Orden en que se envían/reciben los mensajes.	Números anteponiendo los nombres de los mensajes, indicando la secuencia temporal.
Iteración	Mensajes que se repiten varias veces.	
Alternativa	Mensajes que se envían en diferentes ramas condicionales.	
Anidamiento	Mensajes que se encuentran dentro de otros mensajes.	

6.- Diagramas de actividad

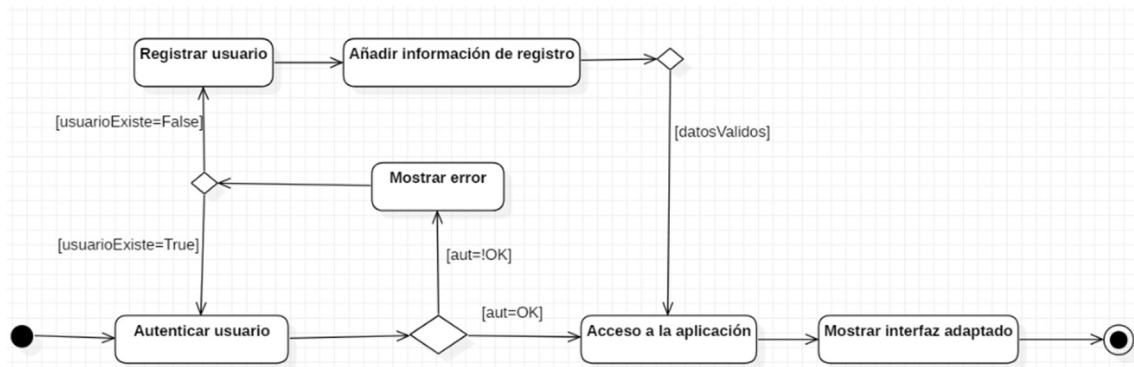
El diagrama de actividad UML es una herramienta visual que representa el flujo de actividades, acciones y decisiones dentro de un proceso o sistema. Se distingue por su capacidad para capturar la secuencia y el paralelismo de actividades, así como los puntos de decisión que pueden derivar en diferentes caminos.

Los elementos básicos de un diagrama de actividad son los siguientes:

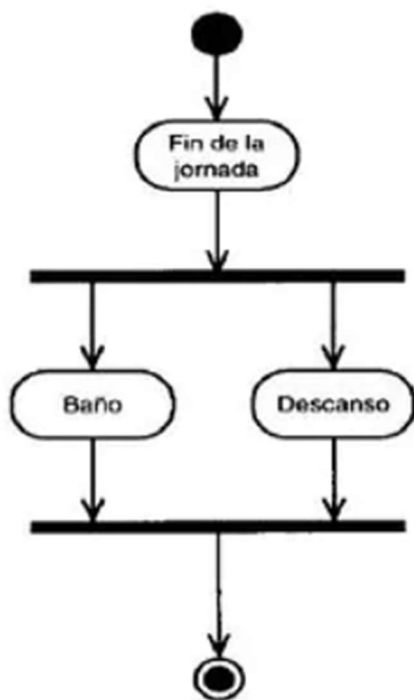
Símbolo	Función	Notación
Nodo inicial	Indica el inicio de un flujo de actividad.	
Nodo final de actividad	Marca el fin de un flujo de actividad.	
Flujo de control	Representa la transición de una actividad a otra.	

Actividad o acción	Una tarea o paso en el proceso.	
Bifurcación o fusión	Divide el flujo en varios flujos concurrentes o los reúne.	
División o Unión	Representa el comienzo o fin de actividades concurrentes.	

Ejemplo de diagrama de actividad:



Otros ejemplos:



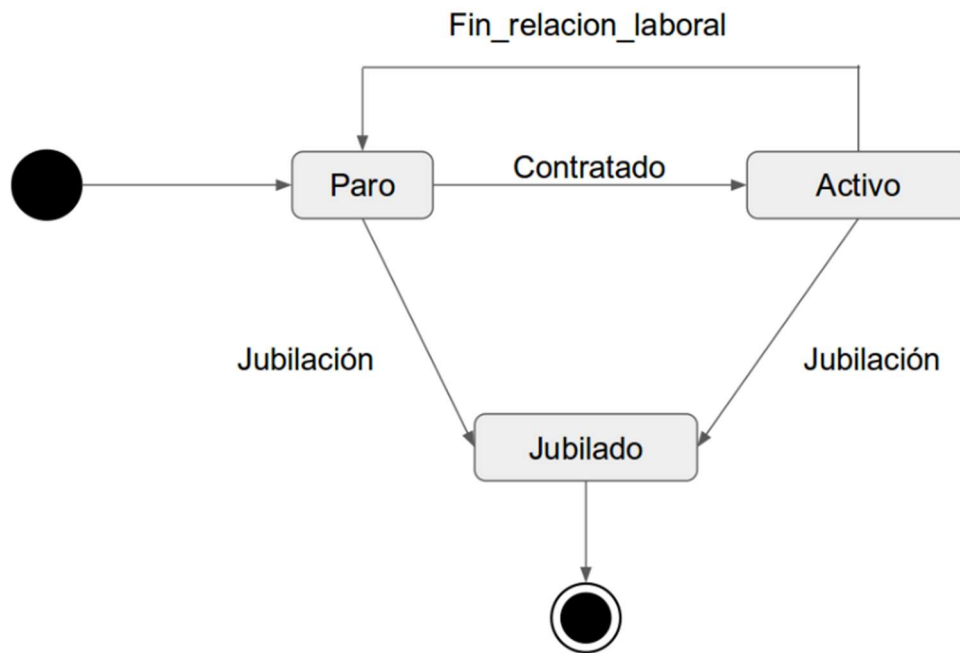
7.- Diagramas de estados

Los diagramas de estados en UML son herramientas que se utilizan para representar los diferentes estados de un objeto a lo largo de su vida útil. Capturan cómo un objeto cambia de un estado a otro, respondiendo a eventos o condiciones internas o externas

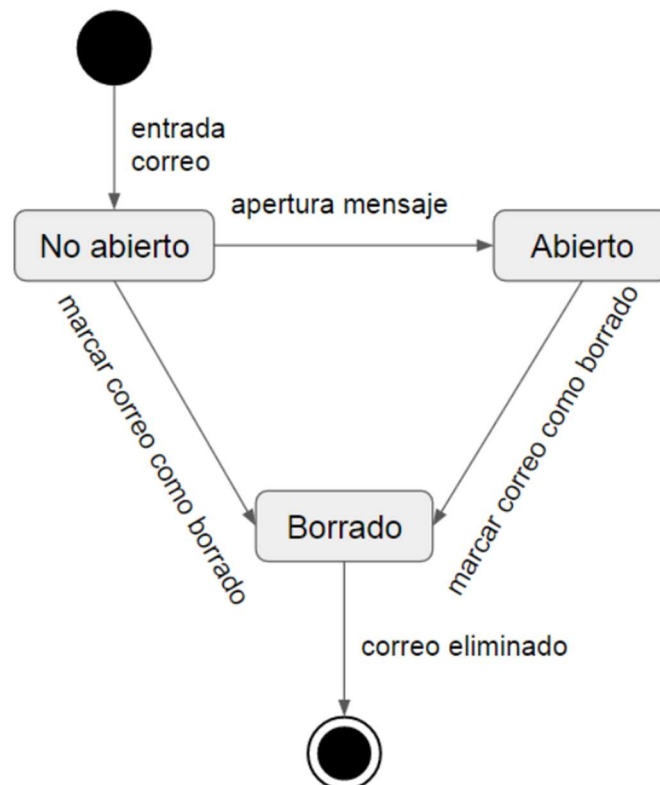
Los elementos básicos de este diagrama son:

Símbolo	Función	Notación
Estado inicial del objeto	Marca el comienzo del ciclo de vida del objeto.	
Estado final del objeto	Indica el fin del ciclo de vida del objeto.	
Transición	Muestra el cambio de un estado a otro.	
Estado	Representa una condición durante la vida del objeto.	

Ejemplo de diagrama de estados:



Otro ejemplo de diagrama de estados:



8.- Referencias bibliográficas

- ❖ Moreno Pérez, J.C. *Entornos de desarrollo*. Editorial Síntesis.
- ❖ Ramos Martín, A. & Ramos Martín, M.J. *Entornos de desarrollo*. Grupo editorial Garceta.