

UNIDAD DIDÁCTICA 6

# ELABORACIÓN DE DIAGRAMAS DE COMPORTAMIENTO

**MÓDULO PROFESIONAL:  
ENTORNOS DE DESARROLLO**



**CESUR**  
Tu Centro Oficial de FP

## ÍNDICE

RESUMEN INTRODUCTORIO .....	2
INTRODUCCIÓN .....	2
CASO INTRODUCTORIO .....	2
1. TIPOS. CAMPOS DE APLICACIÓN .....	4
2. DIAGRAMAS DE CASOS DE USO.....	5
3. DIAGRAMAS DE INTERACCIÓN .....	8
3.1 Diagrama de secuencia .....	8
3.1.1 Elementos de un diagrama de secuencia.....	9
3.1.2 Línea de vida de un objeto .....	10
3.1.3 Elementos de un diagrama de secuencia.....	12
3.2 Diagrama de colaboración .....	13
4. DIAGRAMAS DE ESTADOS.....	15
5. DIAGRAMA DE ACTIVIDAD.....	19
RESUMEN FINAL .....	23

## RESUMEN INTRODUCTORIO

A lo largo de la unidad se conocerán los conceptos relacionados con los diagramas UML, más concretamente, con los diagramas de casos de uso, diagramas de secuencia, diagramas de colaboración, diagramas de estados y diagramas de actividad. Todos estos diagramas permitirán definir la funcionalidad del sistema desde diferentes perspectivas. Los diagramas de casos de uso definen los actores o elementos físicos que intervienen en la aplicación, así como la funcionalidad de forma general. Los diagramas de interacción permiten analizar el código en profundidad a nivel de secuencia temporal (diagramas de secuencia) y de mensajes de los objetos entre sí (diagramas de colaboración). Los diagramas de estados y actividad presentan nodo inicial y final, con la diferencia de que el primero de ellos valora los estados por los que puede pasar la aplicación y el segundo la actividad que aporta al usuario, todo ello a través de una serie de transiciones entre sus diferentes elementos.

## INTRODUCCIÓN

En el mundo de desarrollo del software existen una serie de diagramas que se utilizan para trasladar a todas las personas que integran el equipo como realizar el software y que es lo que debe hacer. En el caso de esta unidad se estudiarán distintos diagramas UML, como, por ejemplo, una aplicación para una cadena de tiendas de ropa.

Estos diagramas se clasifican en diagramas de casos de uso, secuencia, colaboración, estado, actividad, clases, objetos, componentes e implementación. Las factorías de software, a través de las herramientas CASE correspondientes, podrán realizar los diseños adecuados en las diferentes fases del ciclo de vida.

## CASO INTRODUCTORIO

Tras analizar el nuevo caso propuesto por el cliente, tú, junto a tu equipo de trabajo, habéis tomado las riendas del proyecto realizando los diferentes diagramas de clases en función de las actualizaciones fijadas. Sin embargo, las modificaciones en los diagramas de clases debido a estos nuevos requisitos, conlleva revisar todo el aplicativo, ya que van a aparecer mayor número de interacciones, tipos de usuarios y acciones entre las diferentes funcionalidades. Para ello, vais a retomar el diseño de todos los diagramas realizados anteriormente y volver a estudiar uno por uno el cumplimiento de los requerimientos.

Al finalizar la unidad serás capaz de conocer los distintos diagramas UML más usados para las fases de desarrollo de software (diagramas de casos de uso, secuencia, colaboración, estado, actividad, clases, objetos, componentes e implementación).

## 1. TIPOS. CAMPOS DE APLICACIÓN

*Para retomar los diagramas que se utilizaron para su desarrollo, con el fin de conocer en profundidad el desarrollo del proyecto y realizar modificaciones a los mismos para añadir las nuevas implementaciones, es imprescindible que tu equipo se forme y tenga conocimiento de cómo interpretar y realizar estos diagramas.*

**UML** (Unified Modeling Language) o Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos y describe la semántica esencial de estos diagramas y los símbolos en ellos utilizados. A lo largo de la historia han existido numerosos métodos, notaciones y modelos para el diseño orientado a objetos. Con el lenguaje UML, los diseñadores sólo tienen que aprender una única notación válida para los diferentes aspectos del diseño y construcción de aplicaciones. Se puede emplear también para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, organizaciones del mundo real, etc. UML ofrece 9 tipos de diagramas con los cuales se pueden modelar sistemas:

- **Diagrama de Casos para Uso:** modelar los procesos business.
- **Diagrama de Secuencia:** modelar el paso de mensajes entre objetos.
- **Diagrama de Colaboración:** modelar interacciones entre objetos.
- **Diagrama de Estado:** modelar el comportamiento de los objetos en el sistema.
- **Diagramas de Actividad:** modelar el comportamiento de los casos de uso, objetos u operaciones.
- **Diagrama de Clases:** modelar la estructura estática de las clases en el sistema.
- **Diagrama de Objetos:** modelar la estructura estática de los objetos en el sistema.
- **Diagramas de Componentes:** modelar componentes.
- **Diagrama de Implementación:** modelar la distribución del sistema.

UML no es, por tanto, un método, sino varios. Se trata de una estandarización o consolidación de varias notaciones y modelos usados anteriormente. Se debe a los trabajos de Grade Booch, James Rumbaugh e Ivar Jacobson, que han sido los creadores de otras tres metodologías orientadas a objetos.

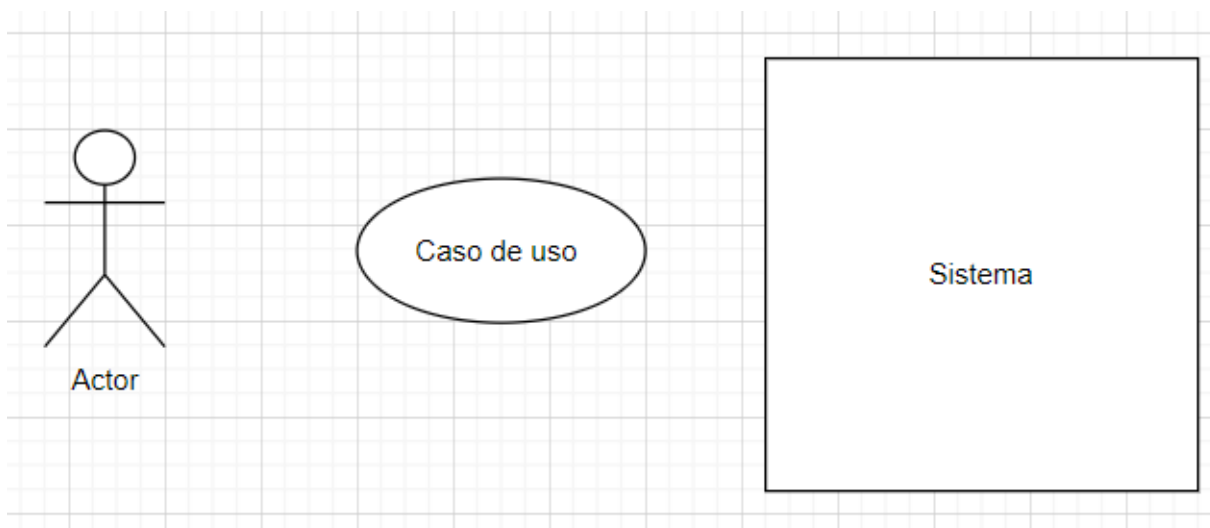


## 2. DIAGRAMAS DE CASOS DE USO

*En la aplicación se ha detectado la necesidad de crear distintos perfiles para distintos tipos de clientes, estos diferentes tipos de clientes interactuarán de forma distinta con la aplicación y tendrán que verse reflejados en los distintos diagramas de casos de uso. Te encargarás de crear esos perfiles con ayuda de tu equipo de trabajo.*

El **diagrama de casos de uso** representa la forma en cómo un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso). Un diagrama de casos de uso consta de los siguientes elementos:

- Actor.
- Casos de Uso.
- Relaciones de Uso, Herencia y Comunicación.



Ejemplo de elementos de diagrama de casos de uso

Fuente: draw.io

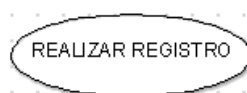
- **El sistema:** Es todo lo que engloba el rectángulo. Es la aplicación que se desea codificar. Todo lo que quede fuera del sistema(rectángulo), son elementos ajenos a esa aplicación, como, por ejemplo, los actores, aunque estos elementos que están fuera, pueden interactuar con el sistema. En la parte superior izquierda o derecha del rectángulo que representa el sistema, contendrá el nombre del mismo (restaurante, gestión de clientes, gestión de nóminas, ...). En proyectos grandes, pueden tenerse sistemas y subsistemas.

- **Actor:** Es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra rol, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema.



Como ejemplo a la definición anterior, se tiene el caso de un sistema de ventas en que el rol de Vendedor, con respecto al sistema, puede ser realizado por un Vendedor o bien por el jefe de Local.

- **Caso de Uso:** Es una operación/tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.



- **Relaciones:** Pueden ser de cuatro tipos:
  - **Asociación:** También conocida como relación de comunicación. Es el tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra operación (caso de uso). Dicha relación se denota con una flecha simple.



- **Inclusión:** Cuando un caso de uso va seguido al comportamiento de otro caso de uso de forma secuencial, enriqueciéndose entre sí. Puede ser vista con la notación <<uses>> o <<include>> (esta última a partir de la versión 1.3 de UML). Se representa con flecha punteada.



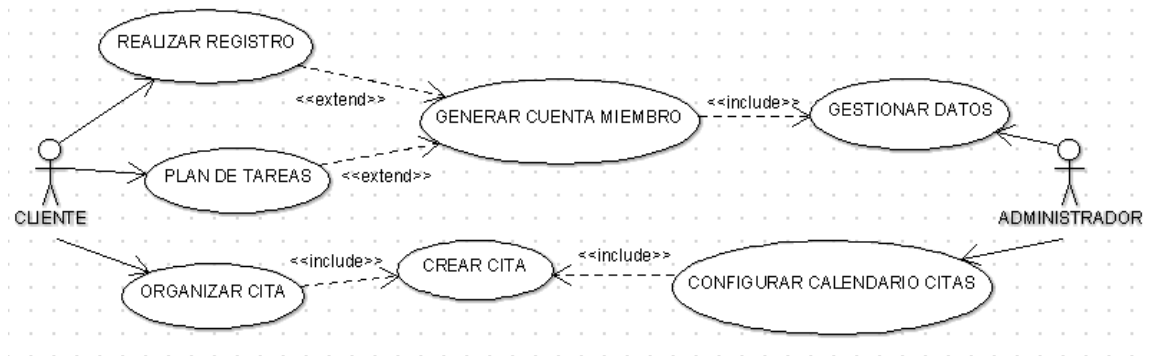
- **Extensión:** Cuando un caso de uso depende implícitamente de otro para completar su función. Lleva incorporado la notación <<extends>> y, al igual que el anterior, se representa con flecha punteada.



- **Generalización** (herencia): Un caso de uso hereda el comportamiento de otro. También es aplicable a los actores, creando relaciones padres e hijos. Se representa mediante una flecha hueca.



El siguiente diagrama de casos de uso muestra un ejemplo con 2 actores (cliente y administrador), 7 casos de usos y 10 relaciones.



Ejemplo de diagrama de casos de uso

Generalmente, los casos de uso se documentan durante la fase de análisis. A diferencia de los diagramas de clases, en los casos de uso, no se adjuntan notas, porque le restan claridad. Normalmente, el documento de los casos de uso se consensua con el cliente y suele incluir, entre otra información:

- El actor que inicia el caso de uso y el actor que se beneficia de él (si existen).
- Condiciones previas del sistema o pre condiciones.
- Qué pasos (casos de uso) van a ir ejecutándose para realizar la tarea determinada. Se establecerá el flujo básico de ejecución y flujos alternativos.
- Se describirán las relaciones del caso de uso.
- Condiciones posteriores a la tarea ejecutada.
- Otra información que se estime relevante para ejecutar esa tarea o escenario concreto.





### VÍDEO DE INTERÉS

Visualiza con detalle la creación de un diagrama de casos de uso y las relaciones:



## 3. DIAGRAMAS DE INTERACCIÓN

*Siguiendo con el desarrollo de la aplicación de tienda de ropa, se ha cambiado la interacción entre distintos tipos de objetos diseñados en los diagramas de clase, para ello tienes que modificar y crear nuevos diagramas de secuencia sobre la aplicación. También tendrás que modificar o añadir diagramas de colaboración, para tener en cuenta que objetos interactúan entre sí.*

En el contexto de UML (Unified Modeling Language), se utilizan dos tipos de diagramas para representar la interacción entre objetos: los diagramas de secuencia y los diagramas de colaboración (también llamados diagramas de comunicación). Ambos tipos de diagramas están destinados a modelar la comunicación y la interacción entre los elementos de un sistema, pero tienen enfoques ligeramente diferentes.

### 3.1 Diagrama de secuencia

Un **diagrama de secuencia** muestra la interacción de un conjunto de objetos de una aplicación a través del tiempo. Esta descripción es importante porque puede dar detalle a los casos de uso, aclarándolos al nivel de mensajes de los objetos existentes. El diagrama de secuencia es más adecuado para observar la perspectiva cronológica de las interacciones, mostrando la secuencia explícita de mensajes y son más apropiados para especificaciones de tiempo real y para escenarios complejos. Un diagrama de secuencia puede mostrar un escenario, es decir, una historia individual de transacción.

Una de las utilidades de los diagramas de secuencia es mostrar la secuencia del comportamiento de un caso de uso. A su vez, muestra gráficamente los eventos que

originan los actores y el impacto que tienen sobre el sistema. La creación de los diagramas de secuencia forma parte de la investigación para conocer el sistema, por lo que es parte del análisis del mismo. Durante la operación del sistema, los actores generan eventos, solicitando alguna operación a cambio. Un evento es una acción externa de entrada que un actor produce en el sistema. Cada evento da origen a una operación del sistema como respuesta. Es conveniente que los nombres de los eventos comiencen con un verbo, pues están orientados a comandos del sistema.



### VÍDEO DE INTERÉS

Conoce más sobre los diagramas de secuencia en este vídeo:

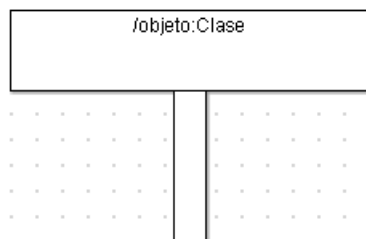


### 3.1.1 Elementos de un diagrama de secuencia

Los elementos de un diagrama de secuencia son:

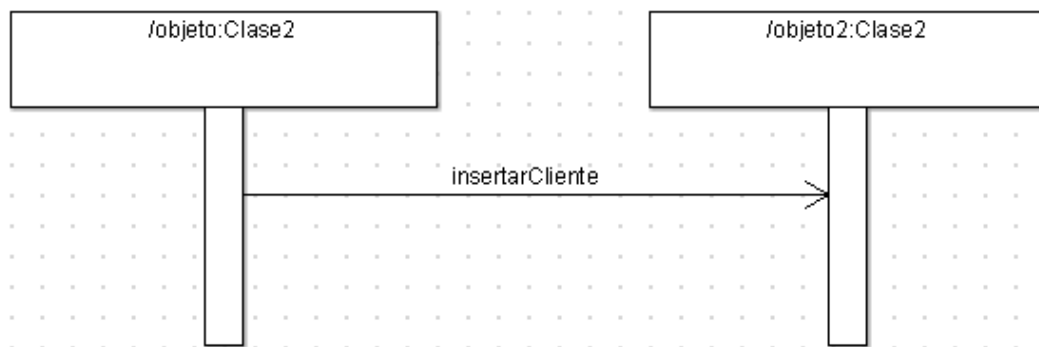
**Objetos/Actor:** Aparecen dentro de un cuadro en la parte superior del diagrama, con la forma *objeto:clase*. Un objeto se representa como una línea vertical (línea de vida), con un rectángulo de encabezado que denotan la activación, es decir, el período de tiempo en el cual el objeto se encuentra desarrollando alguna operación.

El rectángulo de encabezado contiene el nombre del objeto y el de su clase, en un formato *nombreObjeto: nombreClase*.



Representación de un objeto

**Mensajes:** Es una comunicación entre objetos que transmite información con la expectativa de desatar una acción. La recepción de un mensaje es, normalmente, considerada un evento. Se representan mediante una flecha horizontal que va desde la línea de vida del objeto que envió el mensaje, hasta la línea de vida del objeto que ha recibido el mensaje. Si un mensaje requiere un cierto tiempo para llegar a su destino, entonces la flecha del mensaje se dibuja diagonalmente hacia abajo. El envío de mensajes entre objetos se denota mediante una línea sólida dirigida, desde el objeto que emite el mensaje hacia el objeto que lo ejecuta.



Ejemplo sencillo de diagrama de secuencia

Este mensaje puede enviarse al mismo objeto del que sale el mensaje, a este tipo de mensaje se le conoce como mensaje recursivo.



Ejemplo de paso de mensaje al mismo objeto

Fuente: Umbrello

### 3.1.2 Línea de vida de un objeto

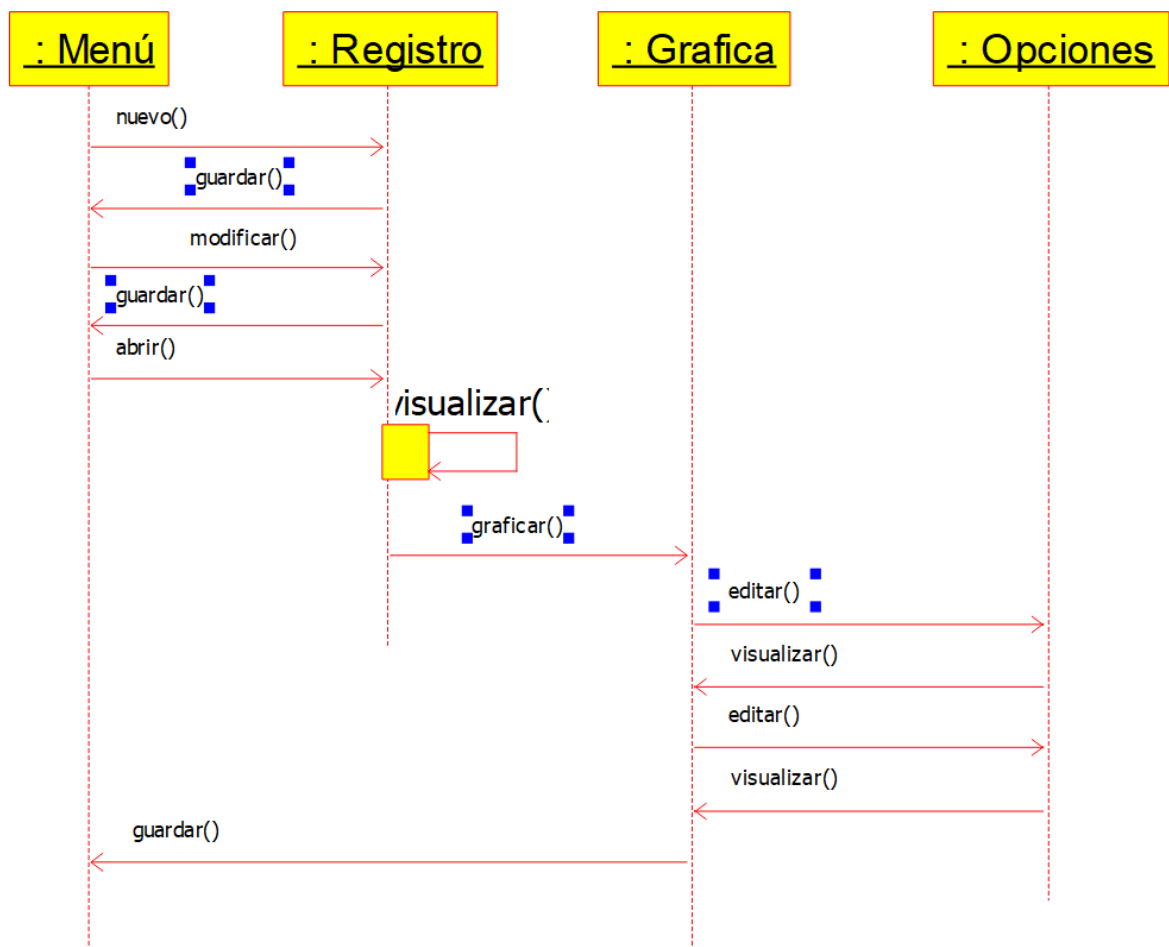
Puede observarse en ella cómo el objeto está representado con un rectángulo y su nombre subrayado y, debajo de él, aparece una línea discontinua que representa el tiempo. En la zona inferior, aparece la actividad más reciente, mientras que la superior representará la más antigua, luego el tiempo discurre de arriba abajo.

: Coche

Ejemplo de Línea de vida de un objeto

Fuente: Umbrello

El objeto en cuestión, pasará de un estado a otro a lo largo de su tiempo vital e interactuará con otros objetos enviando y recibiendo mensajes.



Ejemplo de diagrama de secuencia

Fuente: Propia. Realizado con Umbrello

### 3.1.3 Elementos de un diagrama de secuencia

Los mensajes que se intercambian los objetos son de tres tipos:

1. **Simples.** Pasan el hilo de ejecución de un objeto a otro. Esto quiere decir que, tras el envío del mensaje, el código del objeto no volverá a ejecutarse, a menos que no sea mediante la recepción de otro mensaje.
2. **Sincrónicos.** Esperan a que el método del otro objeto invocado termine su tarea para poder seguir la ejecución del método original.
2. **Asincrónicos.** No impiden que el método origen siga su ejecución normal. Una vez que termina el mensaje invocado, el objeto llamador recibirá una notificación de que la llamada ha terminado y ejecutará el código que esté programado al respecto.

Para distinguir estos tipos de mensajes suelen representarse de distintas formas dependiendo del programa que se use, una de ellas es la que aparece en la siguiente imagen.



Ejemplo de paso de mensaje al mismo objeto

Fuente: Moreno, J.C. (2018). *Entornos de desarrollo*. Madrid: Síntesis

Los diagramas de secuencia se extraen del diagrama de clases y diagrama de casos de uso.



### VÍDEO DE INTERÉS

Conoce cómo crear un diagrama de secuencia y colaboración:



## 3.2 Diagrama de colaboración

El **diagrama de colaboración** es un tipo de diagrama de interacción cuyo objetivo es describir el comportamiento dinámico del sistema de información mostrando cómo interactúan los objetos entre sí, es decir, con qué otros objetos tiene vínculos o intercambia mensajes un determinado objeto. 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. En estos diagramas la comunicación entre objetos se denomina vínculo o enlace (link) y estará particularizada mediante los mensajes que intercambian.

- **Objeto:** Un objeto se representa con un rectángulo dentro del que se incluye el nombre del objeto y, si se desea, el nombre de la clase, separando ambos por dos puntos.
- **Vínculo:** En el diagrama, un vínculo se representa como una línea continua que une ambos objetos, pudiendo tener uno o varios mensajes asociados en ambas direcciones. Como un vínculo instancia una relación de asociación entre clases, también se puede indicar la navegabilidad del mismo mediante una flecha.
- **Mensaje:** Un mensaje se representa con una pequeña flecha colocada junto a la línea del vínculo al que está asociado. La dirección de la flecha va del objeto emisor del mensaje al receptor del mismo. Junto a ella, se coloca el nombre del mensaje y sus argumentos.

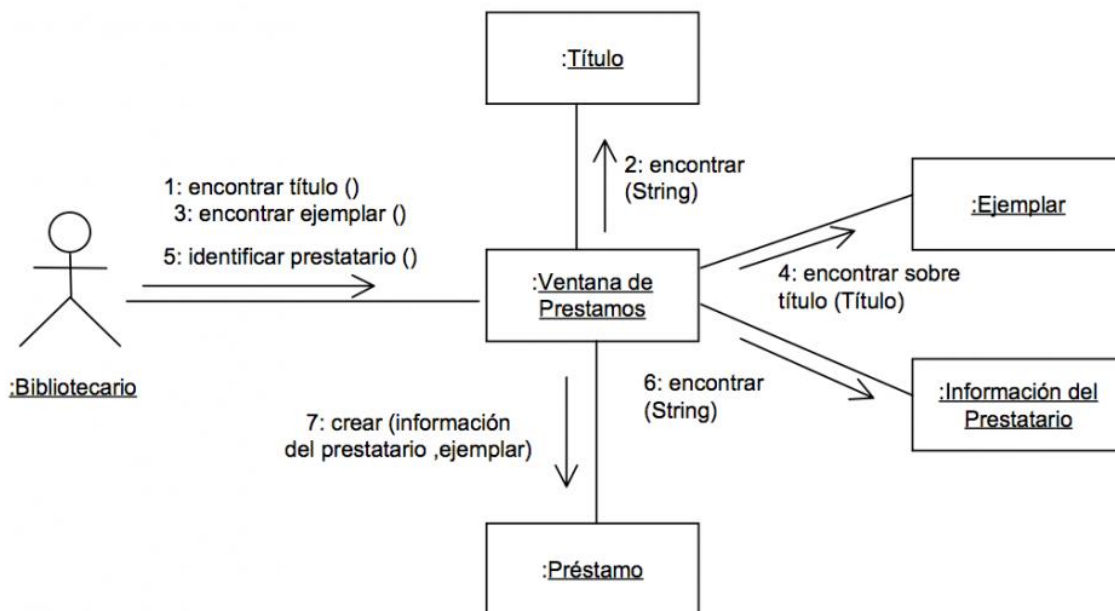


### PARA SABER MÁS

Amplia información sobre los diagramas de colaboración en este enlace:



A diferencia de los diagramas de secuencia, en los diagramas de colaboración siempre se muestra el número de secuencia del mensaje delante de su nombre, ya que no hay otra forma de conocer la secuencia de los mismos. Además, los mensajes pueden tener asociadas condiciones e iteraciones que se representarán como en los diagramas de secuencia. En el siguiente ejemplo se muestra un diagrama de colaboración para el caso de uso “Prestar un ejemplar de una aplicación encargada de los préstamos y reservas de una biblioteca”:



Ejemplo de diagrama de colaboración

## 4. DIAGRAMAS DE ESTADOS

*Con las modificaciones que se han ido haciendo de la aplicación sobre la tienda de ropa, tienes que crear diagramas de estados nuevos, y actualizar algunos de los que ya existen. Los diagramas de estados representaran los componentes y el cambio de estado de los objetos.*

Un **diagrama de estados** muestra el comportamiento dependiente del tiempo de un sistema de información. Representa los estados que puede tomar un componente o un sistema y muestra los eventos que implican el cambio de un estado a otro. Los dos elementos principales en estos diagramas son los estados y las posibles transiciones entre ellos.

- El **estado** de un componente o sistema representa algún comportamiento que es observable externamente y que perdura durante un periodo de tiempo finito. Viene dado por el valor de uno o varios atributos que lo caracterizan en un momento dado.
- Una **transición** es un cambio de estado producido por un evento y refleja los posibles caminos para llegar a un estado final desde un estado inicial.

Desde un estado pueden surgir varias transiciones en función del evento que desencadena el cambio de estado, teniendo en cuenta que, las transiciones que provienen del mismo estado no pueden tener el mismo evento, salvo que exista alguna condición que se aplique al evento. Un sistema sólo puede tener un estado inicial, que se representa mediante una transición sin etiquetar al primer estado normal del diagrama. Pueden existir varias transiciones desde el estado inicial, pero deben tener asociadas condiciones, de manera que sólo una de ellas sea la responsable de iniciar el flujo. En ningún caso puede haber una transición dirigida al estado inicial.

El estado final representa que un componente ha dejado de tener cualquier interacción o actividad. No se permiten transiciones que partan del estado final. Puede haber varios estados finales en un diagrama, ya que es posible concluir el ciclo de vida de un componente desde distintos estados y mediante diferentes eventos, pero dichos estados son mutuamente excluyentes, es decir, sólo uno de ellos puede ocurrir durante una ejecución del sistema. Los diagramas de transición de estados comprenden además otros dos elementos que ayudan a clarificar el significado de los distintos estados por los que pasa un componente o sistema. Estos elementos se conocen como acciones y actividades.



- Una **acción** es una operación instantánea asociada a un evento, cuya duración se considera no significativa y que se puede ejecutar: dentro de un estado, al entrar en un estado o al salir del mismo.
- Una **actividad** es una operación asociada a un estado que se ejecuta durante un intervalo de tiempo hasta que se produce el cambio a otro estado.



#### PARA SABER MÁS

Para observar el funcionamiento de un diagrama de estados, existe una web donde aparece una explicación más detallada con ejemplos:



En este diagrama, los elementos se representan:

**Estado:** Un estado se representa como un rectángulo con las esquinas redondeadas. El nombre del estado se coloca dentro del rectángulo y debe ser único en el diagrama. Si se repite algún nombre, se asume que simboliza el mismo estado. Las acciones y actividades descritas como respuesta a eventos que no producen un cambio de estado, se representan dentro del rectángulo con el formato: nombre-evento (parámetros) [condición] /acción. El estado inicial se representa con un pequeño círculo relleno, y el estado final como un pequeño círculo relleno con una circunferencia que lo rodea.



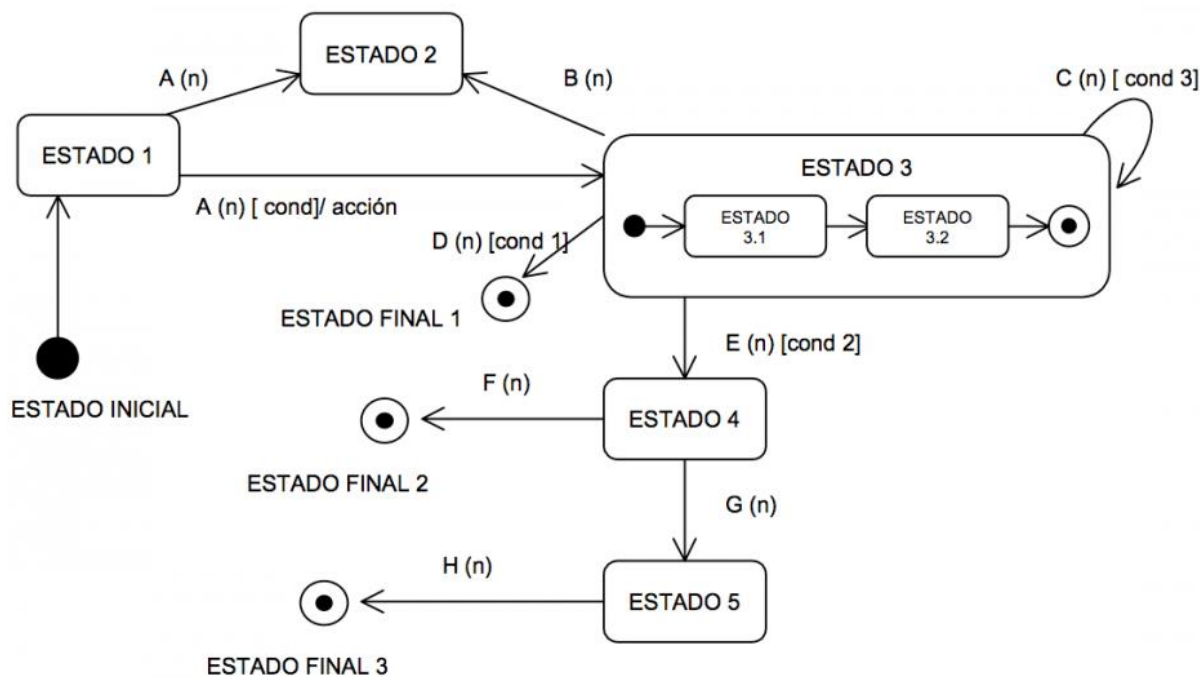
estado inicial



estado final

Estado inicial y final

**Transición.** Una transición se representa con una flecha continua que une dos estados y que se dirige al estado al que cambia el componente. Junto a ella se coloca una etiqueta que debe contener al menos el nombre del evento que provoca la transición. Según el nivel de detalle, puede presentar otros elementos con el formato siguiente: nombre-evento (parámetros) [condición] /acción.



Ejemplo de diagrama de estados

En programación orientada a objetos, dependiendo de los sucesos que ocurran, los objetos o el sistema quedarán en un estado concreto.

Hasta ahora se han podido observar los estados de forma simplificada, tal y como se observa en la figura.



Ejemplo de Simplificación de estado

Fuente: Moreno, J.C. (2018). *Entornos de desarrollo*. Madrid: Síntesis

En la figura se puede observar que el rectángulo representa el estado, el punto sólido sería el punto inicial de la serie de estados, y el punto tipo diana sería la finalización de la serie de estados, las flechas son las transiciones.

Por otro lado, se pueden encontrar con los estados tal y como se ven en la imagen, en él se pueden observar:

- Nombre del estado: identifica el estado.
- Las variables del estado, son similares a los atributos de una clase, ya que almacenan información.

- Actividades (conocidas como sucesos o acciones): son 3, entrada, salida y hacer.



Ejemplo de información de un icono de estado

Fuente: Moreno, J.C. (2018). *Entornos de desarrollo*. Madrid: Síntesis

A continuación, se mostrará un ejemplo de diagrama de estado en el que un wearable se conecta a un móvil por bluetooth y realiza una sincronización de datos. Al finalizar la sincronización de información, se libera la conexión para evitar el consumo de batería y recursos.



Ejemplo de diagrama de estado

Fuente: Moreno, J.C. (2018). *Entornos de desarrollo*. Madrid: Síntesis

Los eventos en los diagramas de estado representan estímulos externos o internos que provocan cambios en el estado de un objeto. Pueden ser desencadenados por acciones del usuario, cambios en el entorno, o por el propio sistema. Los eventos pueden ser simples, como presionar un botón, o complejos, como recibir un mensaje de otro sistema.

Las señales son una forma especial de evento que se utiliza para representar la comunicación entre objetos en un sistema, pueden ser enviadas por un objeto y recibidas por otro, lo que desencadena un cambio en el estado o el comportamiento del receptor. A menudo se utilizan para modelar sistemas distribuidos o eventos asíncronos.

## 5. DIAGRAMA DE ACTIVIDAD

*En la aplicación para la tienda de ropa, tienes que actualizar y crear nuevos diagramas de actividades que muestren las operaciones que se pasan entre los objetos y su interacción, mostrando la secuencia de actividades y el flujo de trabajo desde el punto de inicio hasta el final detallando las decisiones que llevan por el flujo.*

Es un diagrama UML que muestra el flujo de control entre actividades, es decir, es aquel que muestra las operaciones que se pasan entre los objetos en su interacción. Este diagrama es usado para mostrar la secuencia de actividades, ya que, como se comentó, muestran el flujo de trabajo desde el punto de inicio hasta el punto final detallando muchas de las rutas de decisiones que existen en el progreso de eventos contenidos en la actividad.

La versatilidad de estos diagramas permite detallar situaciones donde el proceso paralelo puede ocurrir en la ejecución de algunas las actividades. En base a lo anterior el diagrama de actividad es considerado una especialización del diagrama de estados, organizado respecto de las acciones y usado para especificar:

- Un método.
- Un caso de uso.
- Un proceso de negocio (Workflow).



### VÍDEO DE INTERÉS

Visualiza cómo crear un diagrama de actividad:

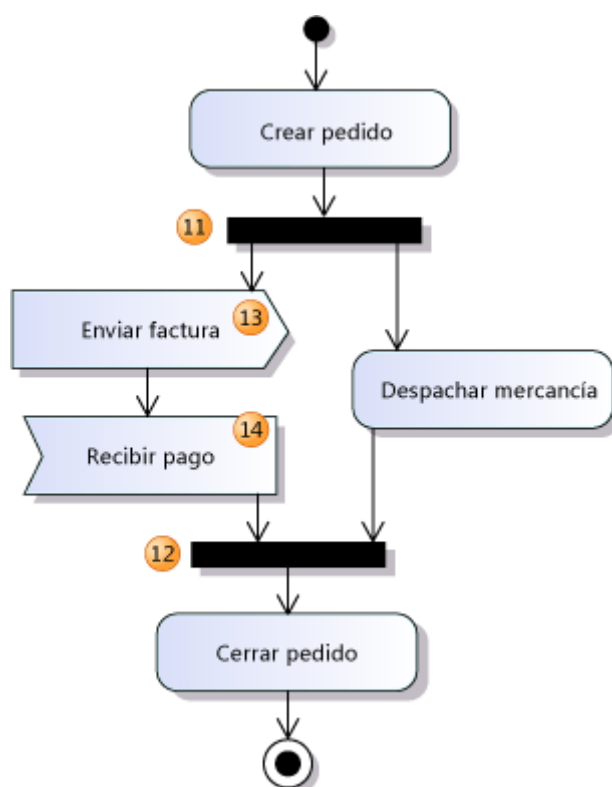


Estos diagramas están compuestos por:

- **Actividades:** Las actividades son las especificaciones de una secuencia parametrizada de comportamiento, en si la actividad es un estado con una acción interna y uno o más transiciones de salida que automáticamente

preceden a la terminación de la acción interna. Las actividades en si misma producen una acción que pueden desencadenar cambios en el estado del sistema, se puede decir entonces que es un elemento compuesto cuyo flujo de control se compone de otros estados de actividad y de acción. Gráficamente es representado con un rectángulo con las puntas redondeadas adjuntando todas las acciones, flujos de control y otros elementos que constituyen la actividad.

- **Acción:** Representa un solo paso dentro de una actividad, la ejecución de una acción atómica, normalmente la invocación de una operación. Gráficamente está representado por rectángulos con las puntas redondeadas.
- **Transiciones o flujo de control:** Es la relación entre dos estados que indica que un objeto en el primer estado realizará ciertas acciones y pasará al segundo estado cuando ocurra un evento específico y satisfaga así ciertas condiciones. Gráficamente se representa con una línea con una punta de flecha.
- **Nodo Inicial:** Es el nodo que indica el comienzo o inicio de una actividad. Gráficamente se representa como un círculo negro.
- **Nodo Final:** Es el nodo que indica el fin ya sea de actividades o de acciones. El nodo final de actividad se representa gráficamente como un círculo con un punto dentro del mismo. El nodo final de flujo se representa gráficamente como un círculo con una cruz dentro del mismo.
- **Nodos de Decisión y Combinación:** Son nodos que representan decisiones o combinaciones en el flujo de control, ejecutando condiciones que cambiarán o no el flujo de la actividad. La representación gráfica de estos nodos de decisión y combinación es la de un rombo.
- **Nodos de Bifurcación y Unión:** Indican el comienzo y final de hilos actuales de control, siendo la bifurcación la especificación de caminos alternativos, elegidos según el valor de alguna expresión booleana y la unión la sincronización de todos los flujos de entrada hasta que todos hayan alcanzado la unión. La representación de bifurcación y unión son una barra horizontal o vertical.



Ejemplo de diagrama de actividad



### ARTÍCULO DE INTERÉS

En la web de Microsoft se muestra más información sobre el diagrama de actividad:



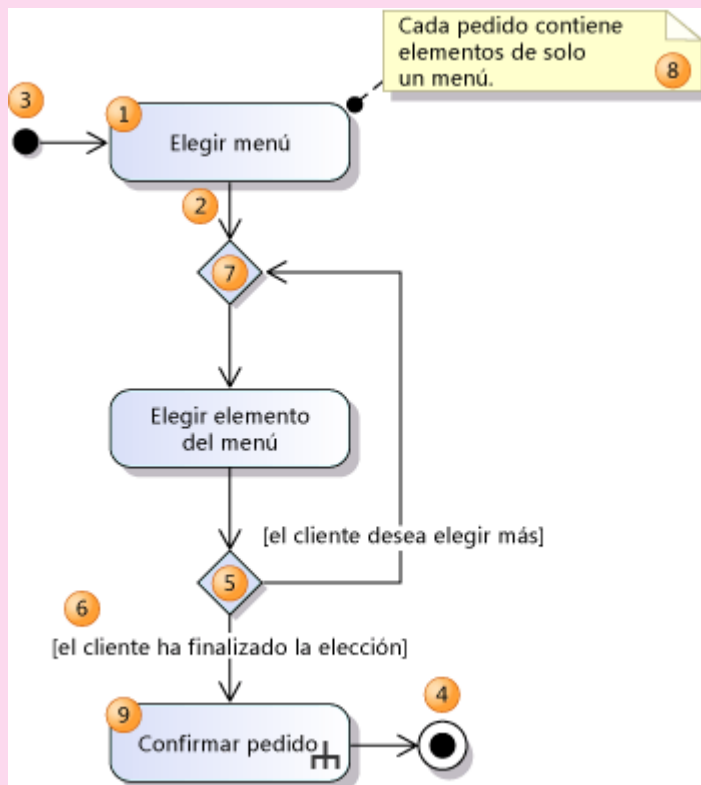


### EJEMPLO PRÁCTICO

Un cliente elige un pedido, el cual solo contiene elementos de un menú. Se pueden elegir varios elementos y, finalmente, se confirma el pedido.

#### Solución:

El diagrama de actividad, sería este:



Cómo puede observarse en el diagrama, el cliente elige el menú, pudiendo elegir un elemento del menú, a partir de ahí, puede volver a elegir otro elemento del menú o confirmar pedido, cuando se termina de elegir los distintos elementos del menú, se confirma el pedido.

## RESUMEN FINAL

Como se ha podido comprobar durante la unidad, los diagramas UML (Unified Modeling Language) son herramientas visuales utilizadas en el desarrollo de software para representar diferentes aspectos de un sistema software. En esta unidad se han podido desarrollar algunos de los diagramas más usados en el desarrollo de software como los diagramas de casos de uso, diagramas de secuencia, diagramas de colaboración, diagramas de estado y diagramas de actividad.

Los diagramas de casos de uso representan las interacciones entre actores externos y el sistema, los actores son roles que interactúan con el sistema. Estos casos de uso son acciones que el sistema realiza en respuesta a la interacción del usuario. Además, ayudan a identificar requisitos del sistema y definir su funcionalidad.

Los diagramas de secuencia muestran cómo las diferentes partes de un sistema interactúan entre sí y en qué orden. Se centra en la secuencia temporal de mensajes intercambiados entre objetos.

Los diagramas de colaboración son similares al diagrama de secuencia, pero se centra en las relaciones entre objetos y la estructura estática. Muestran cómo los objetos colaboran para lograr un objetivo común.

Los diagramas de estado describen el comportamiento de un objeto a lo largo del tiempo en respuesta a los estímulos recibidos. Representa los diferentes estados que puede tener un objeto y las transiciones entre ellos, es útil para modelar sistemas que tienen comportamientos basados en estados, como sistemas de control o de eventos. Los diagramas de actividad, muestran el flujo de control de actividades dentro de un proceso o función. Se utiliza para modelar el comportamiento dinámico de un sistema, centrándose en las acciones que se realizan, es útil para visualizar procesos empresariales, flujos de trabajo y algoritmos.

En resumen, los diagramas UML son herramientas poderosas para visualizar, especificar, construir y documentar sistemas de software. Cada tipo de diagrama tiene su propio propósito y se utiliza en diferentes etapas del proceso de desarrollo para comprender y comunicar aspectos específicos del sistema.