

## **UNIDAD 6: ELABORACIÓN DE DIAGRAMAS DE COMPORTAMIENTO**

**Módulo Profesional: Entornos de desarrollo**

## ÍNDICE

RESUMEN INTRODUCTORIO.....	3
INTRODUCCIÓN.....	3
CASO INTRODUCTORIO .....	4
1. DIAGRAMAS DE CLASES: NOTACIÓN .....	5
1.1 Diagrama de casos de uso.....	6
1.2 Diagramas de interacción .....	7
1.2.1 Diagrama de secuencia .....	7
1.2.2 Diagrama de colaboración.....	9
1.3 Diagrama de estados .....	11
1.4 Diagrama de actividad .....	13
RESUMEN FINAL .....	17

## RESUMEN INTRODUCTORIO

En la presente unidad se estudiarán conceptos relacionados con otros tipos de diagramas complementarios a los diagramas de clase de la unidad anterior. En este sentido, se analizarán los diagramas de casos de uso y sus diferentes elementos, los diagramas de interacción, que engloban a su vez a los diagramas de secuencia y de colaboración, los diagramas de estados y los diagramas de actividad. Todos ellos permiten 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

Como se ha estudiado en las unidades anteriores, todo el proceso de desarrollo de software conlleva la ejecución de un conjunto de fases que, de forma ordenada, crean el ciclo de vida de la aplicación. A medida que se van procesando las diferentes etapas, surge la necesidad de diseñar un conjunto de diagramas que denotan los distintos comportamientos del software.

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, Juan ha tomado las riendas del proyecto y ha realizado 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, es necesario retomar el diseño de todos los diagramas realizados anteriormente y volver a estudiar uno por uno el cumplimiento de los requerimientos. Es imprescindible, por tanto, analizar de nuevo los diagramas de casos de uso, secuencia, colaboración, estado, actividad, clases, objetos, componentes e implementación.

Al finalizar la unidad el alumnado:

- Conocerá los distintos diagramas UML para las fases del desarrollo de software (diagramas de casos de uso, secuencia, colaboración, estado, actividad, clases, objetos, componentes e implementación).
- Será capaz de crear cualquier diagrama UML utilizando software libre o propietario, por ejemplo, a través de ArgoUML.

## 1. DIAGRAMAS DE CLASES: NOTACIÓN

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.

## 1.1 Diagrama de casos de uso

El diagrama de casos de uso representa la forma en como 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, caso de uso y relaciones de uso, herencia y comunicación.

**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 dos tipos:

- **Asociación:** 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.

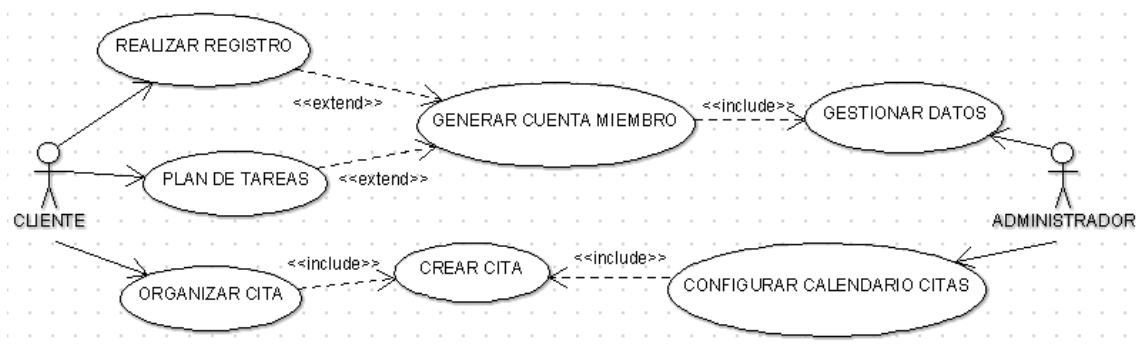
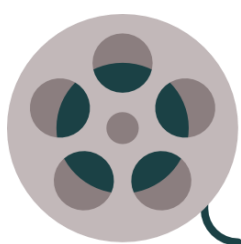


Imagen: Ejemplo de diagrama de casos de uso



### VIDEO DE INTERÉS

En este videotutorial se ve con detalle la creación de un diagrama de casos de uso y las relaciones:

<https://www.youtube.com/watch?v=pOrgwFEK9Ag>

## 1.2 Diagramas de interacción

Los diagramas de interacción se pueden clasificar en dos tipos de diagramas:

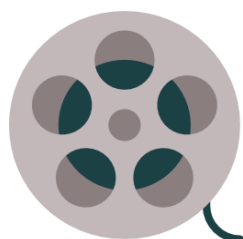
- Diagramas de secuencia
- Diagramas de colaboración

### 1.2.1 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.



#### VIDEO DE INTERÉS

En este videotutorial se aprende a crear un diagrama de secuencia con ArgoUML:

<https://www.youtube.com/watch?v=zaCrRISYMMo>

**Objetos:** 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*.

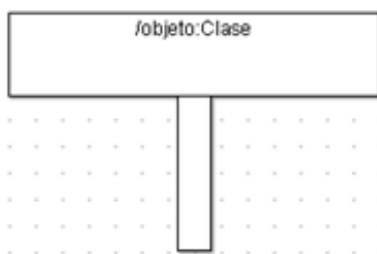


Imagen: Representación de un objeto

**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.



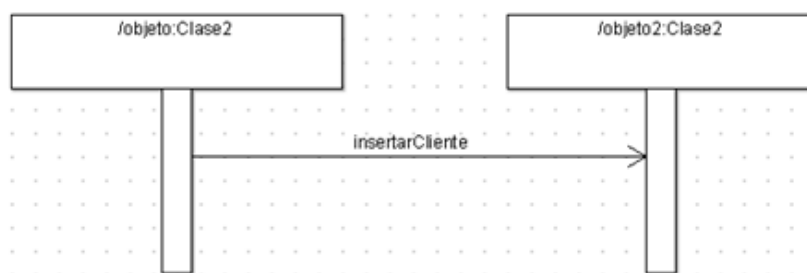


Imagen: Representación de un mensaje



### PARA SABER MÁS

Más sobre los diagramas de secuencia en esta web:

<https://www.youtube.com/watch?v=zgkwAGk0zrs>

### 1.2.2 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

Puede ampliar información sobre los diagramas de colaboración:

<http://ocw.uc3m.es/ingenieria-informatica/metodologia-de-desarrollo-visual/course-files/material-del-tema-7>

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":

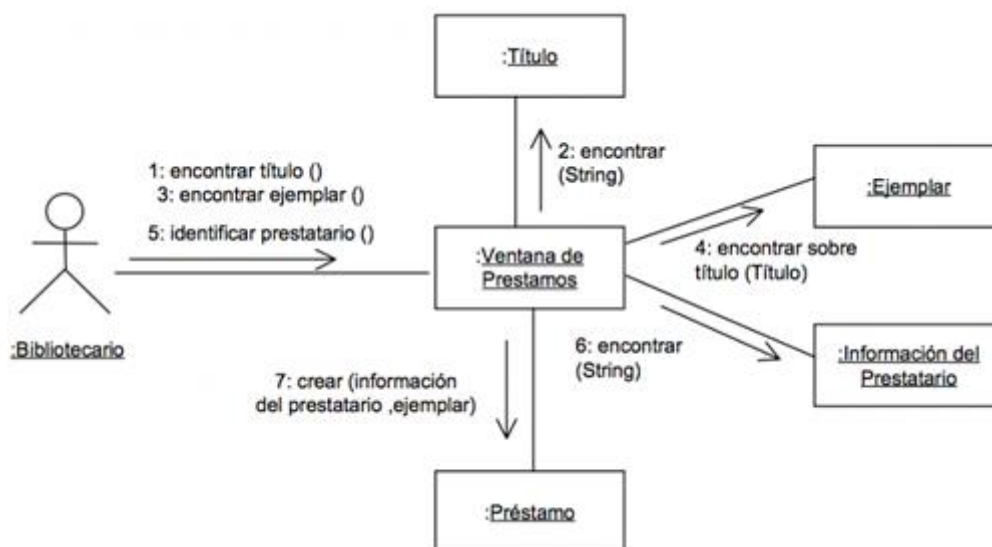


Imagen: Ejemplo de diagrama de colaboración

## 1.3 Diagrama de estados

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:

<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagrama-de-estado-uml/>

**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

Imagen: 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.

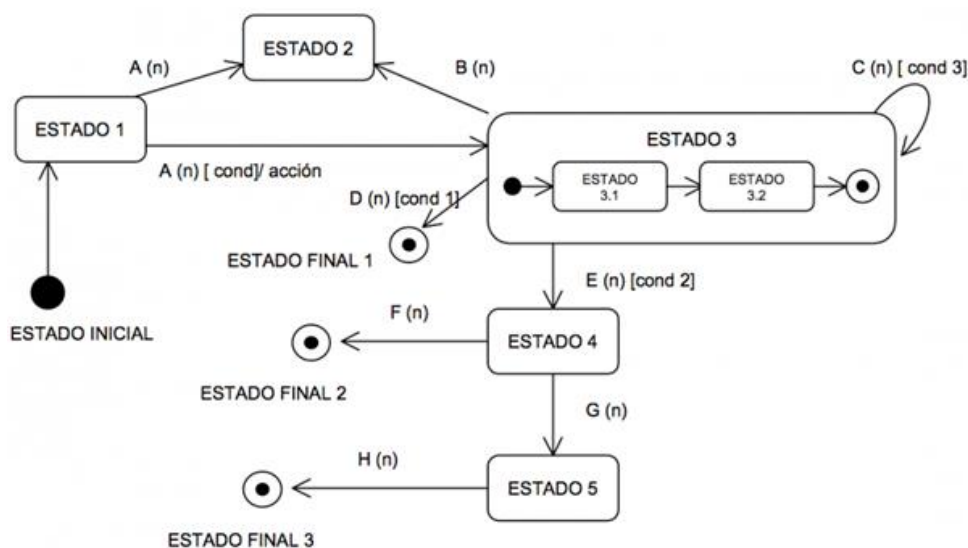
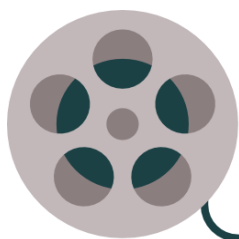


Imagen: Ejemplo de diagrama de estados

## 1.4 Diagrama de actividad

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 y un proceso de negocio (Workflow).



### VIDEO DE INTERÉS

En este videotutorial enseñan a crear un diagrama de actividad utilizando StarUML:

<https://www.youtube.com/watch?v=eG7JuxWJ6XU>

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.

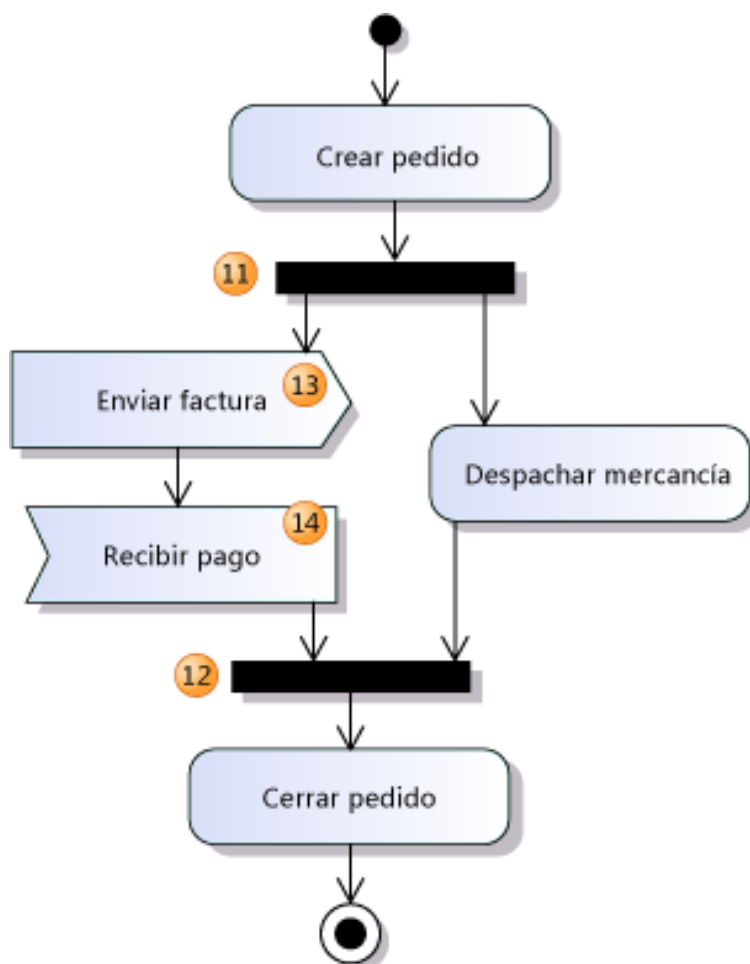


Imagen: Ejemplo de diagrama de actividad



### ARTÍCULO DE INTERÉS

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

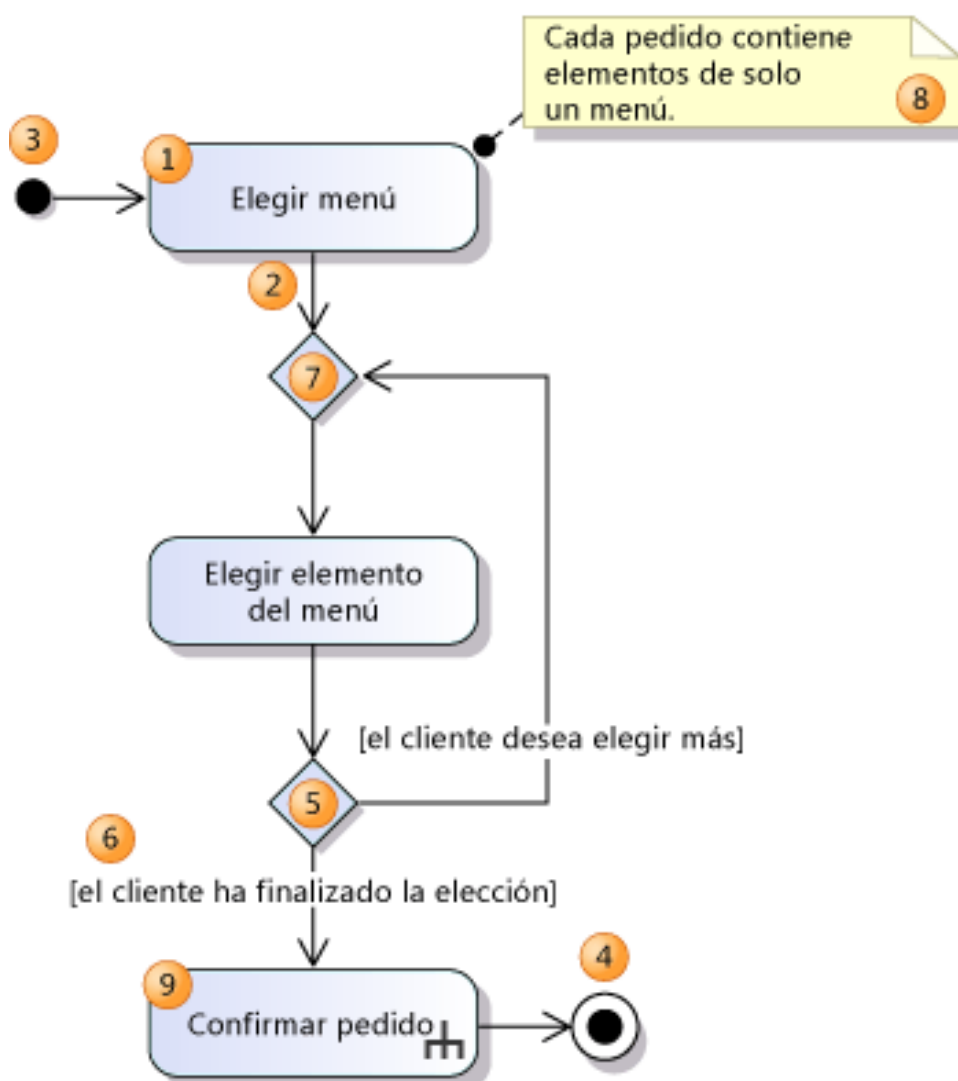
<http://msdn.microsoft.com/es-es/library/dd409360.aspx>



### EJEMPLO PRÁCTICO

Se desea realizar un diagrama de actividad para un caso sencillo. 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:** Diagrama de actividad, en este orden:





## RESUMEN FINAL

El diseño de aplicaciones a través del modelado UML, implica el conocimiento de una serie de diagramas tales como: Diagrama de Casos para Uso, de Secuencia, de Colaboración, de Estado, de Actividad, de Objetos, de Componentes, de Clases y de Implementación. Todos y cada uno de ellos presentan una serie de características y morfologías diferentes, que se han detallado a lo largo de la unidad. No obstante, a la hora de realizar estos diagramas durante las fases del desarrollo de una aplicación, debe existir una conexión directa entre ellos.

En la actualidad existen herramientas que facilitan la funcionalidad necesaria para realizar el diseño de este tipo de diagramas tanto privativas como de código abierto o de software libre (ArgoUML o StarUML). No obstante, existen plugins que pueden ser descargados e instalados a los diferentes IDEs tales como Eclipse o NetBeans que proporcionan funcionalidades para el diseño de diagramas UML. La mayoría de los diseños realizados en esta unidad han sido realizados mediante la herramienta ArgoUML, por su sencillez de uso y su carácter de software libre.