



# Elaboración de diagramas de comportamiento

## Entornos de desarrollo



# Índice



## 7.1. Notación de los diagramas de clases

7.1.1. Asociaciones

7.1.2. Relaciones

## 7.2. Diagramas de secuencia

7.2.1. Elementos de un diagrama de secuencia

7.2.2. Línea de vida de un objeto

7.2.3. Envío de mensajes

## 7.3. Diagramas de colaboración

7.3.1. Objetos

7.3.2. Envío de mensajes

## 7.4. Diagramas de estados

7.4.1. Sucesos y acciones del sistema

7.4.2. Estado en reposo, standby o modo seguro

7.4.3. Subestados

## 7.5. Diagramas de actividades

7.5.1. Decisiones

7.5.2. Concurrencia



## Introducción

Los diagramas de clases ya vistos anteriormente permiten una visión general de los elementos y procesos que los POO llevarán a cabo, por lo que son de gran utilidad para conocer el funcionamiento básico del programa, pero esta visión no siempre es suficiente.

Con el fin de solucionar las faltas de información que los diagramas de clases que se producen surgen otros tipos de diagramas más especializados, con los que poder obtener el resto de información en función de nuestras necesidades.

Este conjunto de diagramas, de los cuales estudiaremos solo algunos, permite mostrar diferente información en función de su enfoque, ya sea la duración de los procesos, la secuenciación, las relaciones que ocurren entre los diferentes elementos, etc.

Jugando con la inclusión de diferentes diagramas podremos tener la visión de los POO que requerimos.

## Al finalizar esta unidad

- + Conoceremos la función y utilidad de los diferentes diagramas, como los de uso, estados, etc.
- + Estudiaremos los diferentes elementos que conforman cada uso de los diagramas.
- + Describiremos las relaciones entre elementos que pueden darse en función del tipo de diagrama.
- + Definiremos el uso que se puede hacer de cada uno de los diagramas.

# 7.1.

## Notación de los diagramas de clases

Este tipo de diagrama se realiza en los programas orientados a objetos realizados con UML. El empleo de estos diagramas, uno de los diversos diagramas de comportamiento existente que UML emplea, permite a los analistas una visión clara de la interacción ocurrida entre los usuarios de un sistema y el propio sistema en función del caso de uso que se hace de este.

Estos tres elementos forman los diagramas de uso, y cada uno de ellos hace referencia a:



Diagrama 1. Elementos del diagrama de uso

- > **El sistema:** Son los elementos que van a codificarse en un diagrama de uso completo siempre aparecerán dentro del rectángulo formado por la figura "Sistema". Se suele incluir el nombre del sistema realizado en la parte superior izquierda o derecha del rectángulo. Es posible en grandes proyectos que existan más de un sistema o subsistemas.
- > **Caso de uso:** Son las funcionalidades del sistema que los actores realizan, generalmente aparecen con su nombre en su interior, haciendo referencia a su función.
- > **El actor:** Es la persona, normalmente identificado por su nombre o rol, que hace uso del sistema mediante los casos de uso. Sus nombres o roles se ponen debajo, puede poseer más de uno, y se une con líneas a los casos de uso que desarrolla.

Generalmente las formas empleadas se mantienen con el fin de que el diagrama posea un fuerte impacto de visualización y sea lo más intuitivamente sencillo para los usuarios.

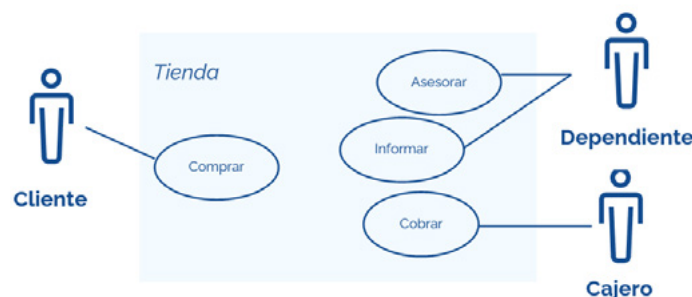


Diagrama 2. Ejemplo de diagrama de caso de uso de una tienda



Un diagrama de uso deberá contar como mínimo con uno de cada uno de los tres elementos mostrados, aunque es posible y probable que se complique mucho más con la inclusión de una gran cantidad de elementos en los sistemas más complejos.

Generalmente se busca la mayor sencillez en estos diagramas, llegando a descartar cualquier tipo de nota adjunta, ya que restaría claridad y sencillez al diagrama.

Este tipo de documento con diagramas se suele emplear en la etapa de análisis de un programa, antes de su realización o para su revisión, con la intención de mostrar una imagen clara de las funciones de sistemas a los desarrolladores y posibles clientes, con el fin de asegurar la idoneidad de la idea y encontrar las mejores maneras de elaborar ese programa con el fin de llevar a cabo la acción de la manera más eficiente.

### 7.1.1. Asociaciones

La relación existente entre los diferentes actores y los casos de uso realizados es denominada asociación.

Cada una de estas asociaciones implican puntos de contacto entre el mundo real, representado por el actor pertinente, y el sistema, por lo que son muestras de interactividad con el sistema.

### 7.1.2. Relaciones

Existen tres tipos de posibles relaciones dentro de este tipo de diagramas:

#### Include o inclusión

Es la empleada en caso de que un caso de uso requiera de otro caso de uso para su realización, es decir, la realización de un caso de uso que englobe la funcionalidad de otros casos de uso más simples.

Se marca con una línea intermitente y la etiqueta {inc}.

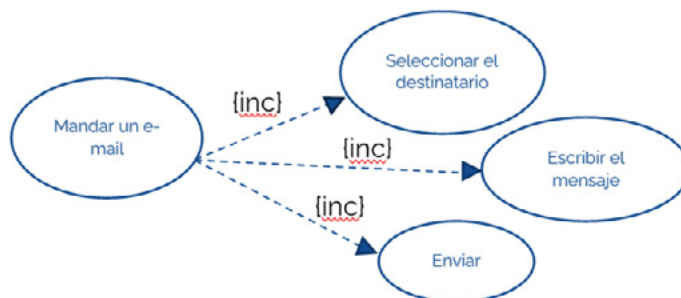


Diagrama 3. Ejemplo de relación include



### Extends o extensión

Una extensión es una situación donde dos casos de uso comparten funcionalidad, pero con uno de ellos poseyendo un matiz extra. Podemos explicar esto de la siguiente forma: "B" cumple la función de "A", pero, además, lo hace de cierta forma específica o incluye un elemento extra.

Este tipo de relación se marca con el empleo de una línea discontinua y la etiqueta [extends].



Diagrama 4. Relación de extensión

### Generalización

En estas relaciones se buscan elementos en común, tanto entre actores como entre casos de uso. Este tipo de relación nos permite crear una jerarquía englobando los diferentes elementos con un elemento general a la cabeza y los elementos especializados subordinados a él. Se marca con una flecha continua con un triángulo en blanco.



Diagrama 5. Ejemplos de relaciones de generalización



# 7.2.

## Diagramas de secuencia

Dentro de los distintos diagramas los de secuencia se especializan en mostrar la temporalización de las acciones del sistema. Con ellos podremos conocer no solo la duración de los diferentes elementos del diagrama, sino también la secuencia en la que dichas acciones se desarrollan, esto permite una mejor visión general al analista o desarrollador.

### 7.2.1. Elementos de un diagrama de secuencia

Algunos de los elementos, como los actores coinciden con los diagramas vistos con anterioridad, pero otros se modifican, o se incluyen elementos nuevos para mostrar la temporalidad. Los elementos destacados son:

#### Objetos y actores

Son los elementos principales, ya que son los que realizan la acción propiamente dicha. El icono de actor sigue siendo el mismo, y el de objeto es un rectángulo vertical, pudiendo cambiar su tamaño para expresar su duración.

#### Mensaje a otro objeto

Marca mediante una flecha las interacciones que ocurren entre diferentes objetos. Al mensaje se le añade una etiqueta con el nombre del método que se empleará por el siguiente objeto. El periodo de ejecución se muestra sobre una línea discontinua con el tamaño del rectángulo.

#### Mensaje del mismo objeto

Es posible que el mensaje sea enviado y recibido por el mismo objeto, en cuyo caso se sigue marcando con una flecha.

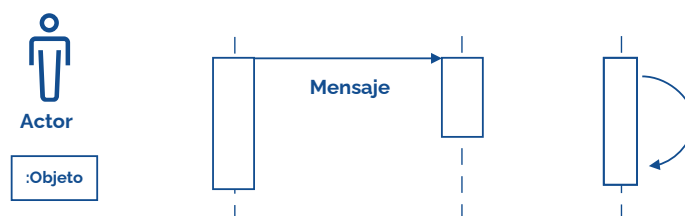


Diagrama 6. Ejemplos de los elementos de los diagramas de secuencia

Por formato el nombre de los objetos debe subrayarse y comenzar con dos puntos ":".





### 7.2.2. Línea de vida de un objeto

La línea de vida se marca con la línea discontinua vertical, la cual representa el tiempo, siempre en dirección descendente, por lo que lo que aparezca en la parte superior ocurrirá antes y las acciones se irán realizando siguiendo la línea discontinua, siempre de forma descendente.

El tamaño ocupado por el rectángulo será proporcional a la duración de la acción, de modo que, usando diversas líneas de forma paralela somos capaces de ver la duración y secuenciación de diversas acciones que suceden simultáneamente.

### 7.2.3. Envío de mensajes

Podemos encontrar tres tipos de mensajes en este tipo de diagramas:

- > **Simple:** Son los que pasan la acción a otro objeto, es decir envían el mensaje y no vuelven a ejecutar la acción, permanecerán así hasta la recepción de otro mensaje. Podemos comparar esta secuencia con una carrera de relevos.
- > **Sincrónico:** El elemento que envía el mensaje debe esperar la respuesta para poder seguir realizando su función.
- > **Asincrónico:** El remitente no requiere ni espera respuesta para poder seguir realizando su función. El receptor ejecutará su función con la finalización de la comunicación.

### Representación recursiva

La recursividad es el proceso producido cuando un proceso se envía un mensaje a sí mismo. Existen diversas formas de evitar este suceso, pero suelen ser más lentas en su procesamiento.

Este tipo de suceso se marcan con una flecha curva que señala un proceso superpuesto al actual.

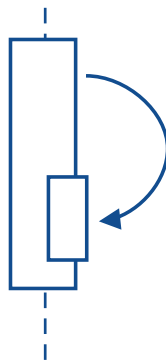


Diagrama 7. Ejemplo de un mensaje recursivo.





# 7.3.

## Diagramas de colaboración

Permite una clara visualización de como interactúan los diferentes elementos de un diagrama dentro de un sistema, posee una apariencia similar al diagrama de secuencia, marcando la secuenciación, pero no la temporalidad, por lo que muchos lo consideran el mismo diagrama, pero en otro formato y con menos información.

### 7.3.1. Objetos

Los objetos siguen siendo iguales, salvo que ya no proporcionan su tamaño a su duración. Se siguen escribiendo con subrayado y dos puntos, aunque es posible encontrarlos sin estos últimos.

### 7.3.2. Envío de mensajes

La secuenciación será marcada por las líneas marcadas por los mensajes. Los mensajes propiamente dichos seguirán el mismo esquema que el diagrama anterior, empleando simplemente flechas.



Diagrama 8. Ejemplo de diagrama de colaboración



# 7.4.

## Diagramas de estados

En ocasiones no es relevante conocer la duración de un proceso, sino el estado en el que queda un objeto tras una acción.

Existen numerosos procesos que cambian el estado de los objetos, unos pocos entre ellos son:

- > Fin de un proceso que deja un objeto inactivo.
- > Acción que activa o desactiva un objeto, sin que se soluciones con el tiempo.
- > Sucesos específicos que desencadenan otros que es necesario apagar o encender.

Los estados con un rectángulo redondeado, mientras que el inicio y final de esos estados se marcan con los círculos uno relleno que marca el inicio de esos estados y otro relleno rodeado por un segundo círculo que marca el final de los estados. El proceso se sigue marcando con flechas.

El estado se marca mediante un rectángulo redondeado, el cual contendrá diversa información:

- > **Nombre de estado:** Nombre de identificación del estado.
- > **Variables de estado:** Almacenan información sobre las posibilidades de estado, su función de almacenamiento de información es similar a los atributos de las clases.
- > **Sucesos y acciones de las actividades:** Generalmente se usan estos tres:
  - » Entrada.
  - » Salida.
  - » Hacer.

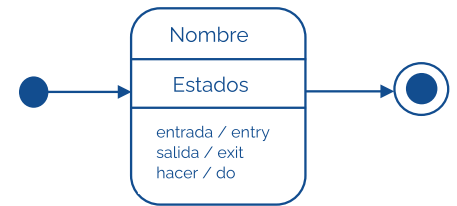


Diagrama 9. Ejemplo de los iconos de estado y su información

### 7.4.1. Sucesos y acciones del sistema

Los cambios de estado pueden ocurrir por reacción a acciones externas, pero son comunes los sucesos donde son los propios sistemas los que provocan sus cambios de estado por procesos que ellos mismos ejecutan. Podemos poner por ejemplo la conexión a la red de nuestro móvil, el cual, tras encontrar una red WIFI a la que tiene acceso, desconecta el consumo de datos propio para comenzar a trabajar con los del WIFI.

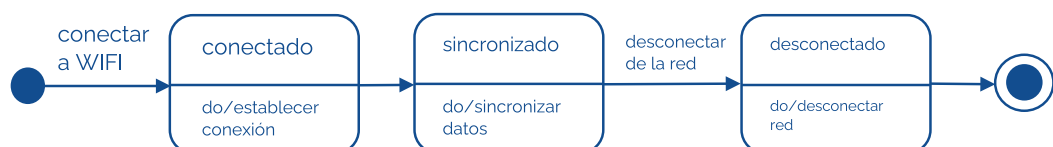


Diagrama 10. Ejemplo de diagrama de estado y sus sucesos, Desconexión de la red



### 7.4.2. Estado en reposo, standby o modo seguro

Algunos de los estados más conocidos son el modo reposo o standby, ya que, para disminuir el consumo de energía muchos dispositivos, como los móviles, tras un periodo de inactividad pasan por un proceso de underclocking. Este tipo de proceso, por ejemplo, en móviles, generalmente continua hasta la intervención de un usuario, generalmente presionando un botón o la pantalla táctil.

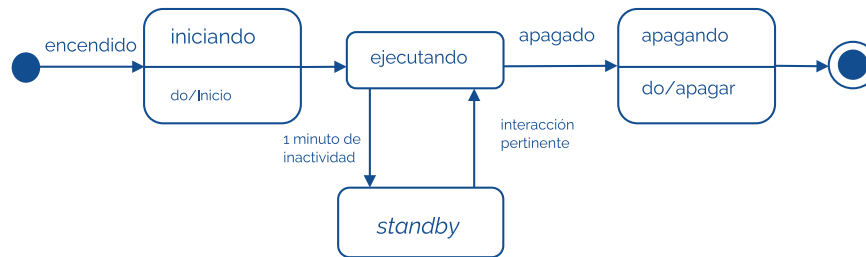
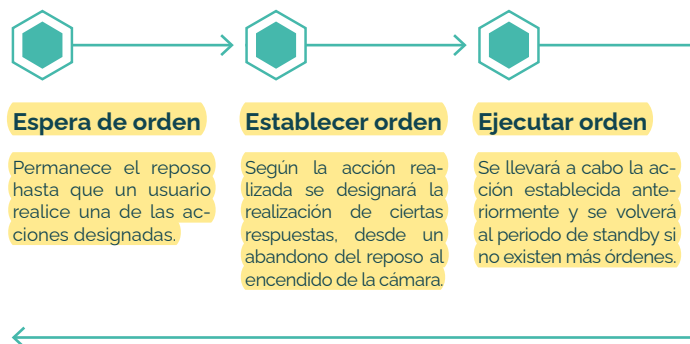


Diagrama 11. Ejemplo de diagrama de estados en un suceso de standby

### 7.4.3. Subestados

Es posible que un estado sea en realidad un conjunto de subestados como ocurre en el proceso de estado de standby, donde podemos encontrar subestados a la espera de ser realizados, generalmente estos tres:





# 7.5.

## Diagramas de actividades

Los diagramas de actividades funcionan de una manera semejante a los de flujo, permitiendo que sea visible a simple vista el funcionamiento de un proceso a través de su esquema. Se elimina toda la información superflua como la duración de los procesos y sus características, centrándose solo en la secuenciación de los distintos procesos, así como los procesos de causa y efecto de cada acción.

Al igual que en los diagramas de estado se comienza con un círculo negro y se acaba con un doble círculo, manteniendo los elementos unidos mediante flechas, aunque posee elementos de unión específicos además de las flechas.

### 7.5.1. Decisiones

Las decisiones son un tipo de unión que deriva en dos elementos diferentes en función de las circunstancias que rodean al proceso, es decir, es posible que una misma acción de lugar a dos resultados totalmente diferentes, esto ocurrirá dependiendo de las circunstancias del proceso inicial. Siempre se escogerá una sola de las posibles soluciones.

### 7.5.2. Concurrencia

Otro tipo de unión son las concurrencias, las cuales ocurren cuando un proceso desencadena dos o más procesos simultáneamente. Este tipo de suceso se marca con una línea gruesa, perpendicular a las flechas, con estas apuntándola o surgiendo de ella en el caso de que se trate de una concurrencia a la inversa, donde dos procesos se convierten en uno.

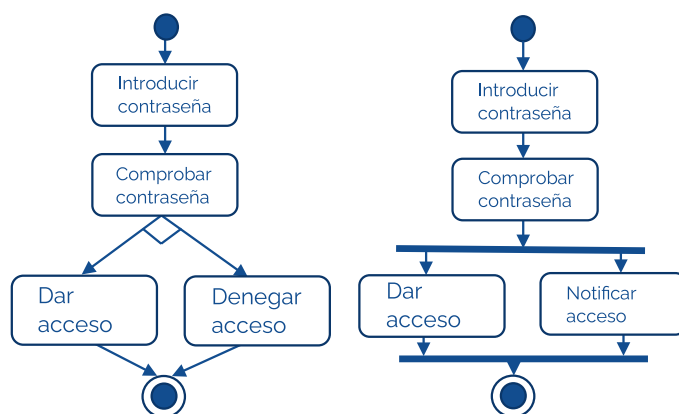


Diagrama 12. Ejemplos de diagramas de actividad con los procesos de decisiones y concurrencia



 [www.universae.com](http://www.universae.com)

