EEDD06

Diseño orientado a objetos.

Elaboración de diagramas de comportamiento.

# Introducción.

Los diagramas de clases nos dan infomración de la estructura del programa, pero no de cómo funciona. PAra eso están los diagramas de comportamiento. Los diagramas de comportamiento incluyen:

* Diagramas de casos de uso.
* Diagramas de actividad.
* Diagramas de estados.
* Diagramas de interacción.
  + Diagramas de secuencia.
  + Diagramas de comunicación/colaboración.
  + Diagramas de interacción.
  + Diagramas de tiempo.

# 2. Diagramas de casos de uso.

NBos ayudan a determinar qué puede hacer cada tipo distinto de usuario en el sistema. Los casos de uso determinan los requisitos funcionales del sistema. En el diagrama los casos se representan con elipses y los usuarios con monigotes.

Definen lo que se espera que pueda hacer el software.

## 2.1. Elementos del diagrama de casos de uso.

Los elementos del diagrama son los actores (monigotes), los casos de uso (elipses) y relaciones (líneas).

### 2.1.1. Actores

Tipos de actores:

* Primarios: Trabajan directamente con el software.
* Secundarios: soportan el sistema para que los primeros puedan trabajar.
* Iniciadores: hay casos de uso que no son iniciados por ningún usuario, como un actor “Tiempo” o “sistema”.

### 2.1.2. Casos de uso.

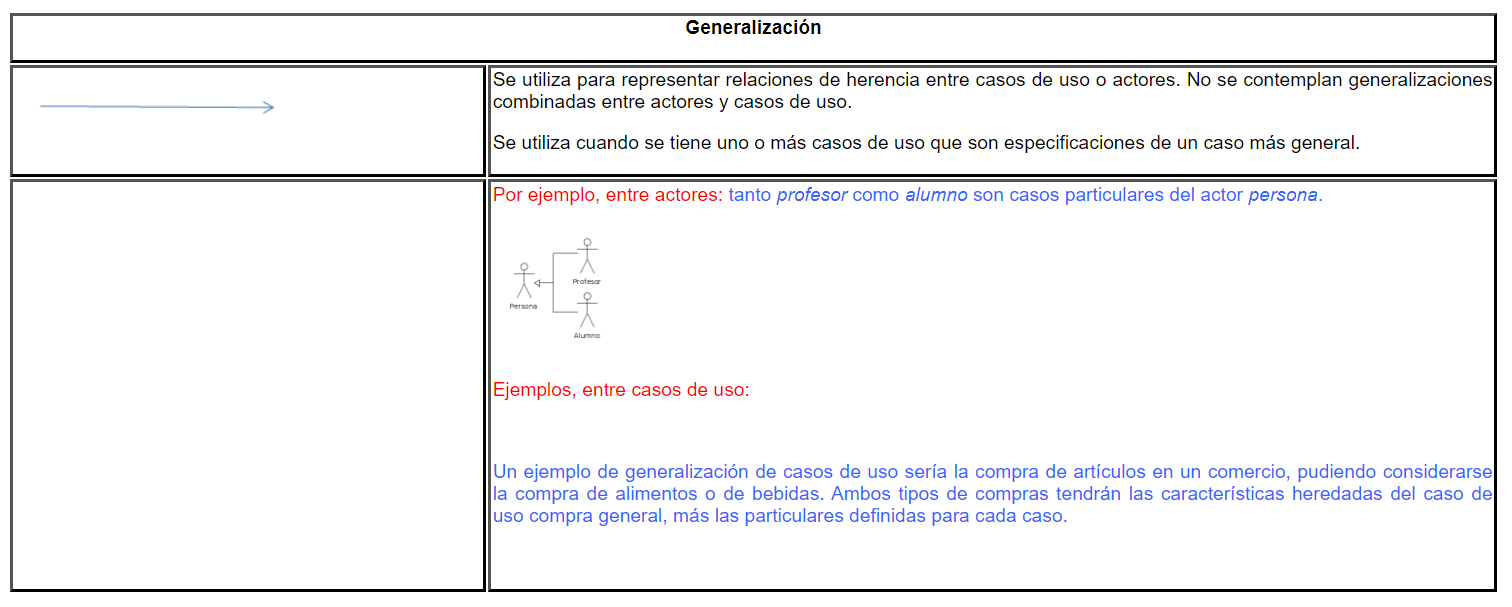
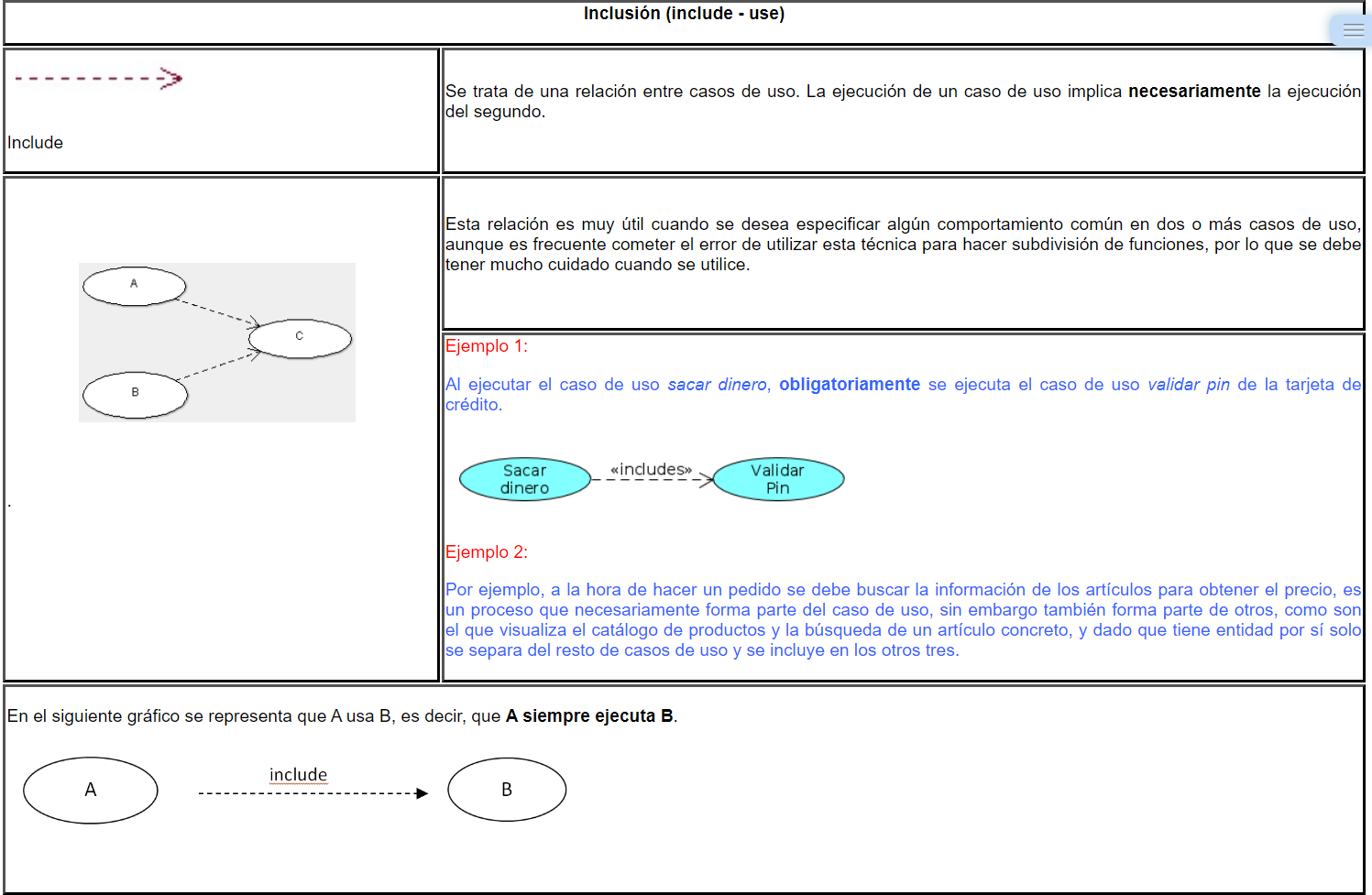
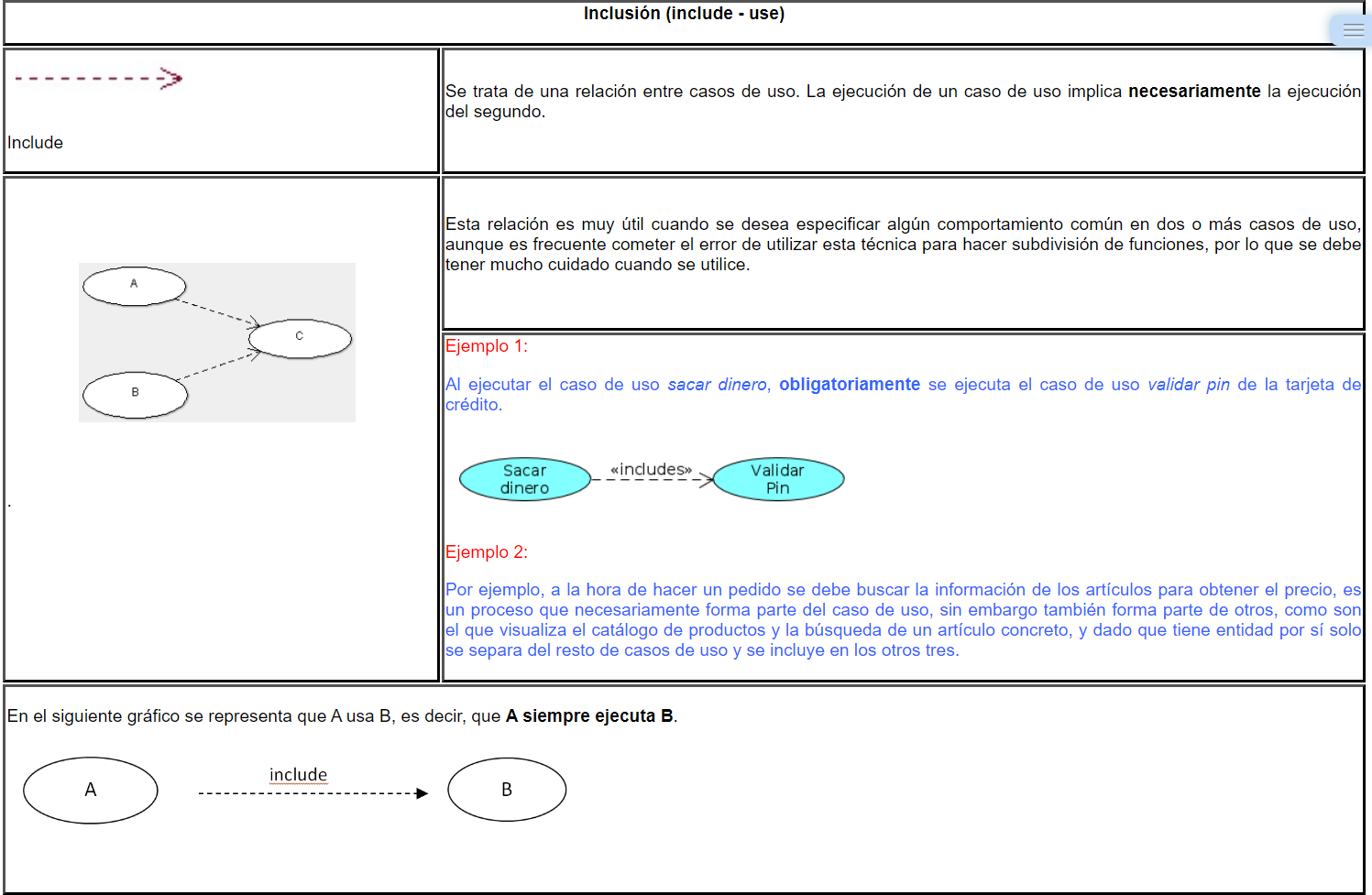
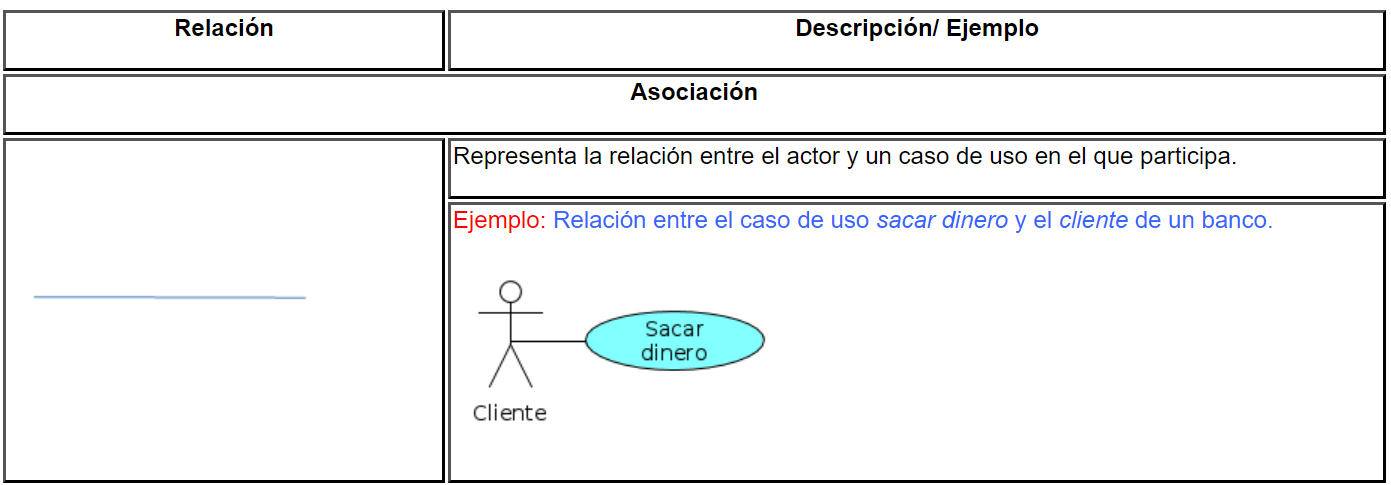
Especifica una secuencia de acciones, incluyendo variantes que el sistema puede llevar a cabo y producen un resutado observable.

El conjunto de casos de uso es el “comportamiento requerido”. Junto al diagrama se crea una tabla para cada usecase con loa siguiente información:

* Nombre dle usecase.
* Actores.
* Propósito: lo que se espera que haga.
* Precondiciones.
* Flujo normal: pasos que realiza el programa bajo condiciones normales.
* Flujo alternativo: pasos a realizar cuando se producen casos inesperados. Cosas como un tipo de dato incorrecto no se incluyen.
* Postcondiciones: condiciones que se deben cumplir al terminar.

### 2.1.3. Relaciones.

Los nodos son los actores y casos de uso y las aristas son las relaciones. Existen distintos tipos de elementos.



## 2.2. Elaboración de casos de uso.

En los diagramas de casos se hace una abstracción de la realidad en la que representamos qué se puede hacer en nuestor sistema y quién lo va a hacer.

Partimos de una descripción detallada y tratamos de detectar aspectos como:

* Usuarios que van a interactuar, para obtener los actores.
* Tareas que relaizan los actores para obtener usecases generales.
* Refinar el diagrama para obtener casos por inclusión, extensión y generalización.

## 2.3. Escenarios.

En la ejecución particular de un caso de uso descrito como una secuencia de eventos.

Por ejemplo, para realizar un pedido de zapatos y botas:

1. El usuario inicia el pedido.
2. Se crea un pedido en estado “En construcción”.
3. Se seleccionan la zapatillas “Lucía” en talla 39.
4. Se selecciona la cantidad “1”.
5. Se obtiene la información delos zapatos y se añaden 45€ a la cuenta.
6. Se seleccionan las botas “Pepe” tamaño 46.
7. Se selecciona la cantidad “1”.
8. Se obtiene la informaicón y se ñaaden 80€ a la cuenta.
9. El usuario acepta el pedido.
10. Se comprueba que el usuario es un socio
11. Se comprueba que los datos bancarios son correctos.
12. Se calcula el total a pagar: cuenta+gastos de envío.
13. Se realiza el pago mediante una entidad externa.
14. Se genera un pedido con el estado “pendiente”.

# 3. Diagramas de interacción.

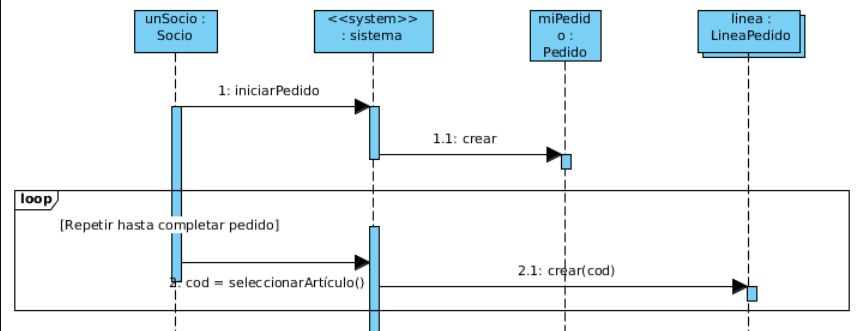
Son diagramas que muestran cómo un grupo de objetos interactúan para una tarea concreta.

Hay dos tipos: diagramas de **secuencia** y **colaboración**.

## 3.1. Diagramas de secuencia.

Los actores se representan mediante rectángulos distribuidos horizontalmente en la parte superior del diagrama. De cada actor sale una línea temporal donde se van indicando las diferentes acciones que realizan entre ellos.

Ejemplo:



En este caso el usuario inicia el pedido y envía la información al sistema. El sistema procesa la información y crea el (objeto?) pedido. Después se ejecuta un bucle de: seleccionar artículo, añadir a pedido. No viene en el diagrama, pero una vez se sale del bucle el usuario empezaría el proceso de pagar, el sistema procesaría los datos de pago (o contactaría con otro sistema para que lo procese) y si todo va bien aceptaría el pedido.

## 3.2. Diagramas de colaboración

Muestran la secuencia de colaboración de uno o varios casos de uso. La principal diferencia con los diagramas de secuencia es que mientras los de secuencia siguen un orden de arriba a abajo, los de colaboración lo hacen mediante etiquetas numéricas.

Los diagramas de colaboración permiten una mejor organización visual de los objetos. Tienen forma de grafo en el que los nodos son objetos y las aristas son mensajes.

### 3.2.1. Representación de objetos.

Los objetos son instancias de las clases, aunque también lo son elementos de la interfaz del sistema o el propio sistema.

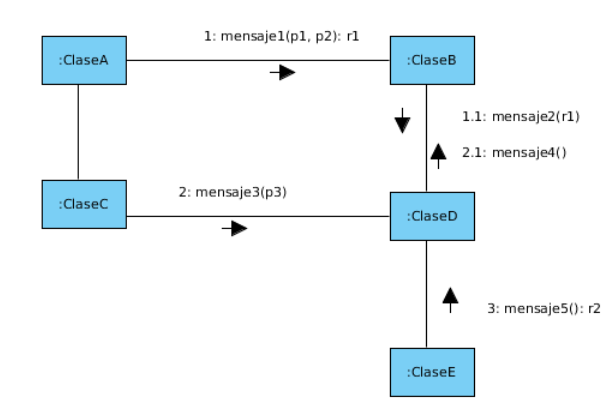
Se representan mediante rectángulos con alguno de estos nombres:

* NombreClase: para hacer referencia a la clase.
* NombreObjeto: nombre concreto del objeto que interactúa.
* :nombreClase : hace referencia a un objeto genérico de la clase.
* NombreClase:nombreObjeto: se especifica el objeto y la clase a la que pertenece.

### 3.2.2. Paso de mensajes.

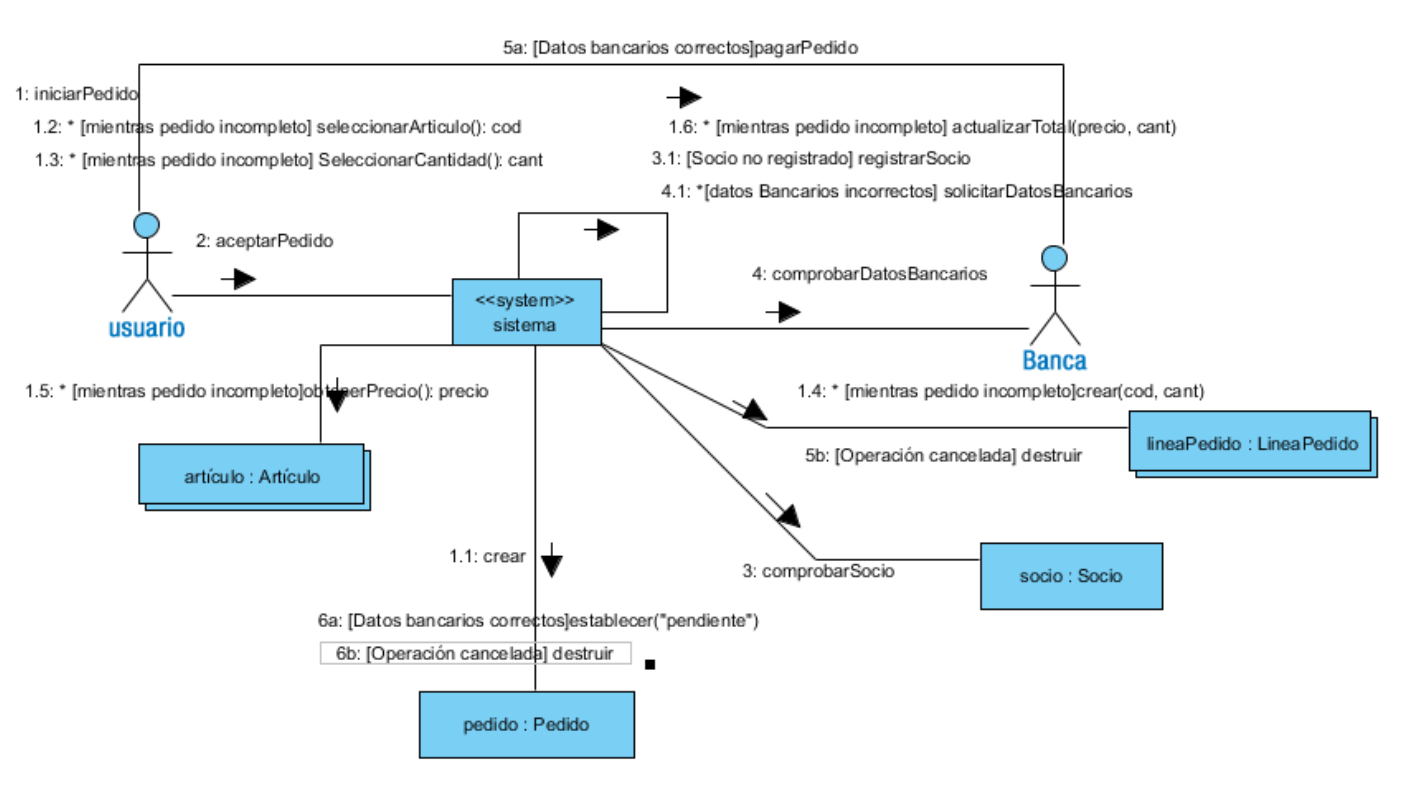
La sintaxis es la siguiente:

* [Secuencia] [\*] [Condición] {ValorDevuelto} : mensaje(args)
* [Secuencia] [\*] [Condición] mensaje(args) : {ValorDevuelto}
  + **Secuencia:** nivel de anidamiento dentro de la interacción. Los mensajes se numeran para indicar el orden en el que se envían.
  + **\***: Indica que el mensaje es iterativo.
  + **Condición:** debe cumplirse para que se envíe el mensaje.
  + **ValorDevuelto:** opcional. El valor que se recibe de vuelta.
  + **Mensaje:** Nombre del mensaje.
  + **Argumentos.**



### 3.2.3. Ejemplo de un diagrama de colaboración.

* Las actividades que se repiten o pueden repetirse se marcan con un asterisco y su condición.
* La condición se escribe en el mismo nombre del mensaje.
* Ya que se puede aceptar o cancelar el pedido se forman dos flujos, 5 y 6.
* Al objeto “sistema” se le ha asignado el nombre **<<system>>**.



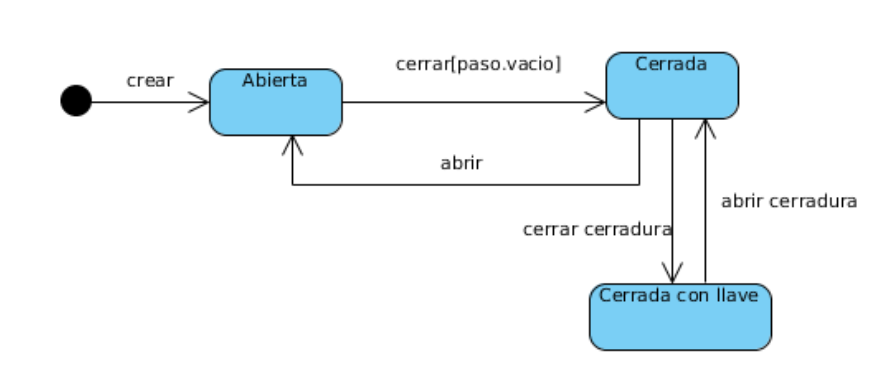
# 4. Diagramas de estados.

PErmiten analizar la evolución del estado de un objeto a lo largo del tiempo.

Modela el comportamiento de los objetos en respuesta a ventos. Se cumple que:

* Un objeto está en un estado concreto que viene determinado por el valor de sus atributos.
* La transición de un estado a otro es momentánea y se produce cuando ocurre un determinado evento.

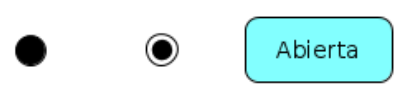
Diagrama de estados para una puerta:



## 4.1. Estados y eventos.

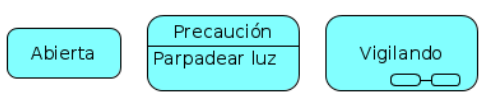
Un estado es una situación en un objeto que satisface cierta condición, realiza alguna actividad o espera al´gun evento. Hay 3 tipos de estado:

1. Inicial: objeto recién creado.
2. Final: cuando finaliza la secuencia de eventos que pueden proporcionar transacciones entre estados.
3. Intermedio: cualquier estado intermedio.



Los estados se representan con cajas, y suelen mostrar:

* Nombre del estado.
* Nombre del estado y acción/actividad asociada al estado.
* Estado con subestados.



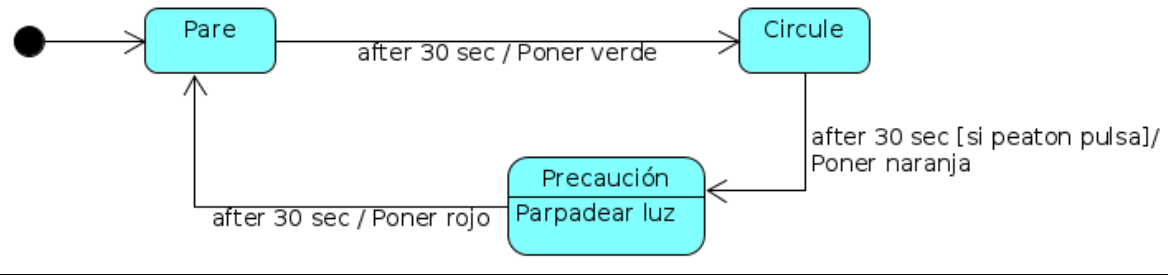
Tipos de eventos:

* Señales (excepciones): recepción de una señal producida por una situación extraordinaria en el sistema.
* Llamadas: recepción de una petición para realizar una operación. Normalmente es un evento de llamada manenajdo por un objeto.
* Paso de tiempo.
* Cambio de estado.

## 4.2. Transacciones.

Es el paso de un estado A a B cuando ocurre un evento y se satisface cierta condición. La notación es:

Evento(argumentos) [condición] / Acción:



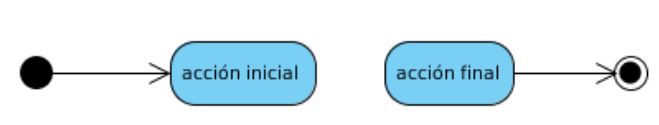
# 5. Diagramas de actividad

Es una especialidad del diagrama de estados, que organiza las acciones y representa cómo se pasa de unas a otra.s Distingue entre las acciones qeu ocurren secuencialmente y concurrentemente. Los nodos son los estados, que pueden ser de actividad o de acción, y los arcos son transacciones entre estados.

## 5.1. Elementos del diagrama de actviidad.

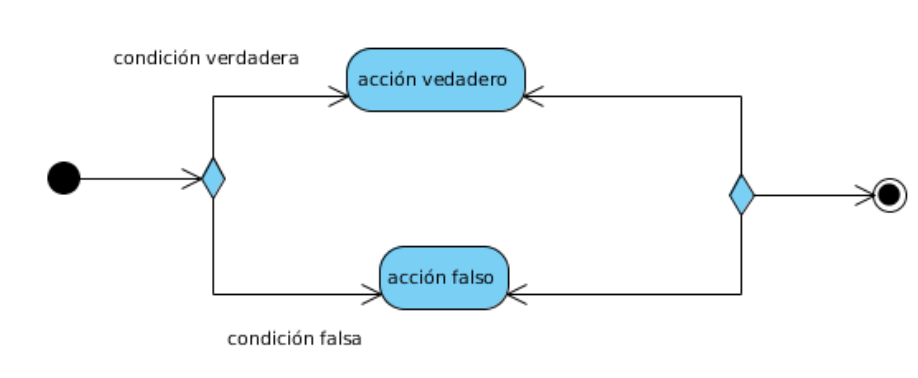
Estados:

* De actividad: elemento compuesto de otros estados de actividad y de acción.
* De acción: representan la ejecución de una acción simple.
* Inicial y final.



Transacciones: relación entre dos estados y cómo se llega a ellos.

* Secuencial: cuando termina la acción de origen, se ejecuta la otra.
* Bifurcación: caminos alternativos dependiendo de una condición.



* Fusión: vario flujos de entrada a uno de saida.
* División: una entrada genera varias saidas.
* Unión: dos flujos entrantes esperan a que todos hayan alcanzar una unión.

Objetos: representan a isntancias de una clase. Se usan flujos de objetos para indicar cómo cambia el valor de sus atributos.

