



# 6. Diseño orientado a objetos. Elaboración de diagramas de comportamiento.

|       |                        |
|-------|------------------------|
| Autor | ⓧ Xerach Casanova      |
| Clase | Entornos de desarrollo |
| Fecha | @Mar 18, 2021 9:46 PM  |

## 1. Introducción

### 2. Diagramas de casos de uso.

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

##### 2.2.1. Actores

##### 2.2.2. Casos de uso

##### 2.2.3. Relaciones

#### 2.2. Elaboración de casos de uso

#### 2.3. Escenarios

## 3. Diagramas de interacción

### 3.1. Diagramas de secuencia

#### 3.1.1. Representación de objetos, línea de vida y paso de mensajes

### 3.2. Diagramas de colaboración

#### 3.2.1. Representación de objetos

#### 3.2.2. Paso de mensajes

## 4. Diagramas de estados

### 4.1. Estados y eventos

### 4.2. Transiciones

## 5. Diagramas de actividad

### 5.1. Elementos del diagrama de actividad

## Mapa conceptual

## Anexos

Ejercicio resuelto 1 ("ZAPATERÍA TACÓN DE ORO"). Elaboración de un diagrama de casos de uso

Ejercicio resuelto 2 ("QUIJOTE"). Elaboración de un diagrama de casos de uso.

Ejercicio resuelto 3 ("ALQUILER DE PISOS Y LOCALES"). Elaboración de un diagrama de casos de usos.

[Ejercicio resuelto 1 \("Generar pedido"\). Elaboración de un diagrama de secuencias](#)

[Ejercicio resuelto 2 \("Estadio"\). Elaboración de un diagrama de secuencia](#)

[Ejercicio resuelto 3 \("ROPERO"\). Elaboración de un diagrama de secuencia.](#)

[Ejemplo de un diagrama de colaboración](#)

[Ejercicio resuelto 1 \("Generar pedido"\). Elaboración de un diagrama de estados.](#)

[Ejercicio resuelto 2 \("RELOJ"\). Elaboración de un diagrama de estados.](#)

[Ejercicio resuelto 3 \("VIDA LABORAL"\). Elaboración de un diagrama de estados](#)

[Ejemplo de un diagrama de actividad](#)

# 1. Introducción

Un diagrama de clases nos da información estática, pero no nos dice nada acerca del comportamiento dinámico de los objetos que lo forman. Para este tipo de información se utilizan diagramas de comportamiento, los cuales 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.

Cuando se construye un software es necesario conocer los requerimientos del mismo. Se precisa alguna herramienta que ayude a especificarlos de manera clara, sistemática y que el cliente pueda entender.

No basta una lista de requerimientos descritos, ya que se puede inducir a errores de interpretación y se suelen dejar cabos sueltos.

La solución la da los diagramas de casos de uso. Son un elemento fundamental en la etapa de análisis de un sistema y resuelven el problema de la falta de comunicación entre el equipo de desarrollo y el equipo que necesita una solución de software.

Los diagramas de casos de uso nos ayudan a determinar qué puede hacer cada tipo diferente de usuario con el sistema.

Los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista de usuario. Determinan los requisitos funcionales del sistema, es decir, las funciones del sistema que se pueden ejecutar.

Es una visualización gráfica de los requisitos funcionales del sistema que está formado por casos de uso, representados como elipses. Los actores que interactúan con ellos son monigotes. Su principal función es dirigir el proceso de creación del software definiendo lo que se espera de él. Son fáciles de interpretar y útiles para comunicarse con el cliente.

## 2.1. Elementos del diagrama de casos de uso

- Actores.
- Casos de uso.
- Relaciones.

/code

### 2.2.1. Actores

Representan un tipo de usuario del sistema y se entiende por usuario cualquier cosa externa que interactúa con el sistema. Puede ser humano, otro sistema informático o unidades organizativas.



La diferencia entre actores y usuarios es que un usuario puede interpretar diferentes roles según la operación, estos roles representará un actor diferente. Por tanto, cada actor puede ser interpretado por diferentes usuarios.

Tipos de actores:

- **Primarios:** interaccionan con el sistema para explotar su funcionalidad, de forma directa y frecuentemente con el software.
- **Secundarios:** soporte del sistema para que los primarios puedan trabajar. Precisos para alcanzar algún objetivo.
- **Iniciadores:** es posible que haya casos de uso que no sean iniciados por ningún usuario, se considera un actor tiempo o sistema que asume el

arranque del caso.

## 2.1.2. Casos de uso

Se representa mediante un óvalo o elipse y su descripción.

Especifican una secuencia de acciones, incluyendo variantes, que el sistema puede llevar a cabo y que producen un resultado observable de valor para un actor concreto.




El conjunto de casos de uso forma el comportamiento requerido de un sistema. El objetivo principal de elaborar el diagrama de casos de uso no es crear el diagrama en sí, sino la descripción que de cada caso se debe realizar. Esto ayuda al equipo de desarrollo a crear el sistema. Junto al diagrama se crea una tabla con descripción textual, en la que se incluyen al menos los siguientes datos (contrato).

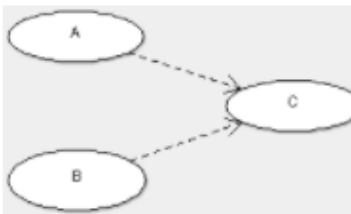

- **Nombre:** nombre del caso de uso.
- **Actores:** aquellos que interactúan con el sistema a través del caso de uso.
- **Propósito:** breve descripción de lo que se espera que haga.
- **Precondiciones:** aquellas que deben cumplirse para que pueda llevarse a cabo el caso de uso.
- **Flujo normal:** flujo normal de eventos que deben cumplirse para ejecutar el caso de uso exitosamente, desde el punto de vista del actor que participa y del sistema.
- **Flujo alternativo:** flujo de eventos que se llevan a cabo cuando se producen casos inesperados o poco frecuentes. No se deben incluir aquí errores como escribir un tipo de dato incorrecto o la omisión de un parámetro necesario.
- **Postcondiciones:** las que se cumplen una vez que se ha realizado el caso de uso.

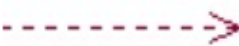
|                          |                      |                        |
|--------------------------|----------------------|------------------------|
| <b>Super Use Case</b>    |                      |                        |
| <b>Author</b>            | usuario              |                        |
| <b>Date</b>              | 26-ago-2011 13:56:56 |                        |
| <b>Brief Description</b> |                      |                        |
| <b>Preconditions</b>     |                      |                        |
| <b>Post-conditions</b>   |                      |                        |
| <b>Flow of Events</b>    |                      |                        |
|                          | <b>Actor Input</b>   | <b>System Response</b> |
| 1                        |                      |                        |

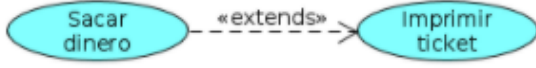

## 2.1.3. Relaciones


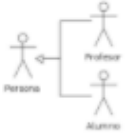
Los diagramas de casos de uso son grafos no conexos. Los nodos son actores y casos de uso y las aristas son las relaciones entre ellos. Estas relaciones representan qué actores realizan las tareas descritas en los casos de uso. Además existen otro tipos de relaciones que se utilizan para especificar relaciones más complejas, como uso o herencia entre casos de uso o actores.

| Relación  | Descripción/ Ejemplo  |
|---|---|
| <b>Asociación</b>   |   |
|  | Representa la relación entre el actor y un caso de uso en el que participa.                               |
|   | <p><b>Ejemplo:</b> Relación entre el caso de uso <i>sacar dinero</i> y el <i>cliente de un banco</i>.</p> |

|   |   |
|---|---|
|  | <p>Esta relación es muy útil cuando se desea especificar algún comportamiento común en dos o más casos de uso, aunque es frecuente cometer el error de utilizar esta técnica para hacer subdivisión de funciones, por lo que se debe tener mucho cuidado cuando se utilice.</p> <p><b>Ejemplo 1:</b></p> <p>Al ejecutar el caso de uso <i>sacar dinero</i>, <b>obligatoriamente</b> se ejecuta el caso de uso <i>validar pin</i> de la tarjeta de crédito.</p>  <p><b>Ejemplo 2:</b></p> <p>Por ejemplo, a la hora de hacer un pedido se debe buscar la información de los artículos para obtener el precio, es un proceso que necesariamente forma parte del caso de uso, sin embargo también forma parte de otros, como son el que visualiza el catálogo de productos y la búsqueda de un artículo concreto, y dado que tiene entidad por sí solo se separa del resto de casos de uso y se incluye en los otros tres.</p> |
|---|---|

| Extensión (extend)  |   |
|---|---|
|  <p>extend</p> | <p>Se trata de una relación entre casos de uso. La ejecución de un caso de uso <b>puede</b> provocar la ejecución del segundo</p>   |
|   | <p>Se utiliza una relación entre dos casos de uso de tipo "extends" cuando se desea especificar que el comportamiento de un caso de uso es diferente dependiendo de ciertas circunstancias.</p> <p>La principal función de esta relación es simplificar el flujo de casos de uso complejos. Se utiliza cuando existe una parte del caso de uso que se ejecuta sólo en determinadas ocasiones, pero no es imprescindible para su completa ejecución. Cuando un caso de uso extendido se ejecuta, se indica en la especificación del caso de uso como un <b>punto de extensión</b>. Los puntos de extensión se pueden mostrar en el diagrama de casos de uso.</p> |

|  |   |
|--|---|
|  | <p><b>Ejemplo 1:</b></p> <p><i>Imprimir ticket</i> es consecuencia del caso de uso <i>sacar dinero</i>, pero su ejecución es opcional a que sea requerida por el cliente.</p>  <pre> graph LR     A([Sacar dinero]) -.-&gt; «extends»  B([Imprimir ticket]) </pre> <p><b>Ejemplo 2:</b></p> <p>Cuando un usuario hace un pedido si no es socio se le ofrece la posibilidad de darse de alta en el sistema en ese momento, pero puede realizar el pedido aunque no lo sea.</p>  <pre> graph TD     Usuario((Usuario)) --&gt; RealizarPedido([Realizar pedido])     RealizarPedido --&gt; RegistrarSocio([Registrar como socio])     RegistrarSocio -.-&gt; «extends»  RealizarPedido </pre> |
|--|---|

| Generalización  |  |
|---|--|
|  | <p>Se utiliza para representar relaciones de herencia entre casos de uso o actores. No se contemplan generalizaciones combinadas entre actores y casos de uso.</p> <p>Se utiliza cuando se tiene uno o más casos de uso que son especificaciones de un caso más general.</p>   |
|   | <p><b>Por ejemplo, entre actores:</b> tanto <i>profesor</i> como <i>alumno</i> son casos particulares del actor <i>persona</i>.</p>  <pre> graph LR     Persona((Persona)) --&gt; Profesor((Profesor))     Persona --&gt; Alumno((Alumno)) </pre> <p><b>Ejemplos, entre casos de uso:</b></p> <p>Un ejemplo de generalización de casos de uso sería la compra de artículos en un comercio, pudiendo considerarse la compra de alimentos o de bebidas. Ambos tipos de compras tendrán las características heredadas del caso de uso compra general, más las particulares definidas para cada caso.</p> |

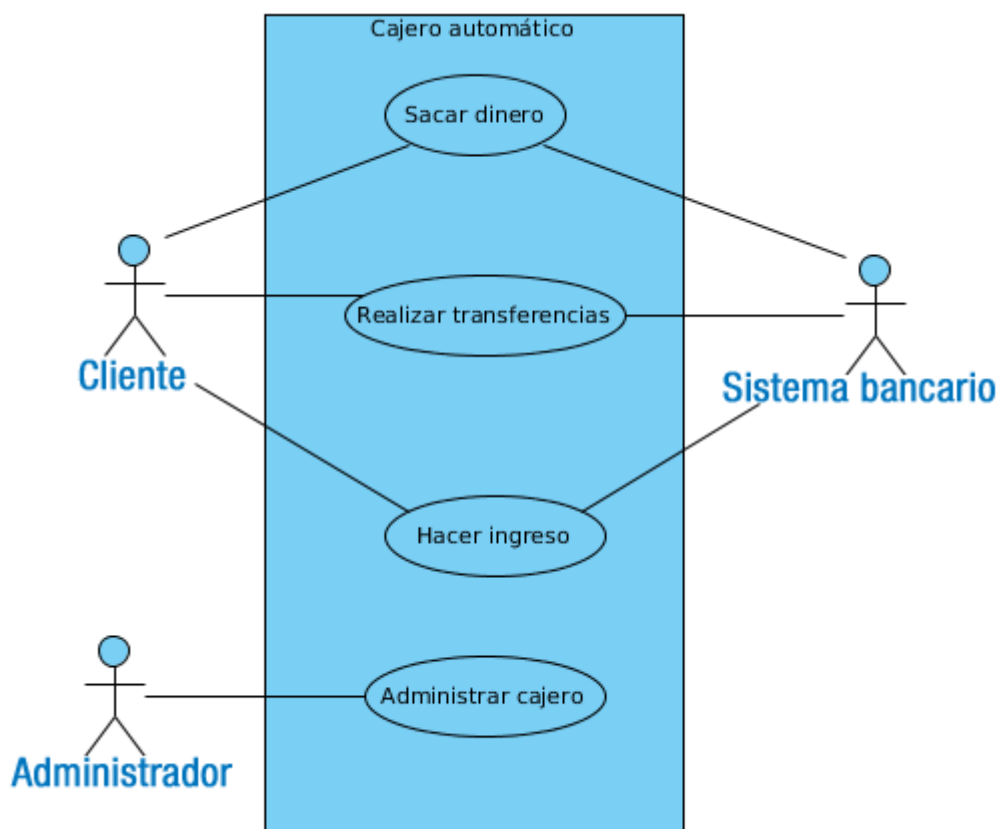
## 2.2. Elaboración de casos de uso

En los diagramas de casos de uso se hace una abstracción de la realidad en la que representamos qué cosas se pueden hacer en un sistema y quién las va a hacer.

Necesitamos implementar diagramas cuya información permita al equipo de desarrollo tomar decisiones adecuadas en la fase de análisis y diseño, y que sean útiles en la fase de implementación.

Partimos de una descripción detallada del posible problema a resolver y tratamos de detectar aspectos como:

- Usuarios que interactúan para obtener a los actores.
- Tareas que realizan estos actores para determinar los casos de uso más genéricos.
- Refinar el diagrama analizando los casos de uso más generales, para detectar casos relacionados por inclusión, extensión y generalización.



## 2.3. Escenarios

Un caso de uso especifica un comportamiento deseado, pero no impone como se llevará a cabo.



Un escenario es una ejecución particular de un caso de uso que se describe como una secuencia de eventos. un caso de uso es una generalización de un escenario.

### **Ejempplo**

Para el caso de uso hacer pedido se establecen diferentes escenarios, uno podría ser:

1. El usuario inicia el pedido.
2. Se crea el pedido en estado "en construcción".
3. Se selecciona un par de zapatos "Lucía" de piel negros, del número 39.
4. Se selecciona la cantidad 1.
5. Se recupera la información de los zapatos y se modifica la cantidad a pagar sumándole 45 .  
€
6. Se selecciona un par de botas "Aymara" de ante marrón del número 40.
7. Se selecciona la cantidad 1.
8. Se recupera la información de las botas y se modifica la cantidad a pagar sumándole 135 €.
9. El usuario acepta el pedido.
10. Se comprueba que el usuario es, efectivamente socio.
11. Se comprueban los datos bancarios, que son correctos.
12. Se calcula el total a pagar añadiendo los gastos de envío.
13. Se realiza el pago a través de una entidad externa.
14. Se genera un pedido para el usuario con los dos zapatos que ha comprado, con el estado "pendiente".

Los escenarios pueden y deben posteriormente documentarse mediante diagramas de secuencia.

## **3. Diagramas de interacción**

También se hace necesario buscar la forma de representar como circula la información, los objetos que participan en los casos de uso, los mensajes que

se envían, el momento en que se producen...

Los diagramas de interacción son vistas del sistema que muestran como grupos de objetos interactúan para un comportamiento. Captan la ejecución de los casos de uso, representando a los actores que participan y los mensajes que se pasan.

Existen dos: diagramas de secuencia y diagramas de colaboración, en el que la anotación es distinta en ambas aunque la información es la misma.

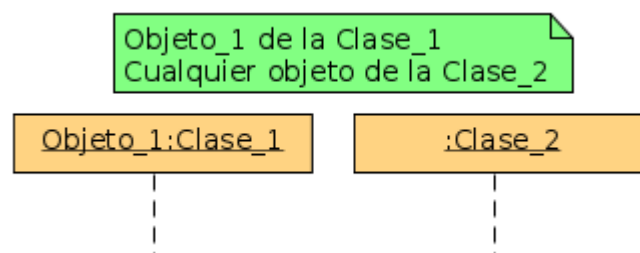
## 3.1. Diagramas de secuencia

En estos diagramas los objetos/actores que forman parte del escenario de un caso de uso, se representan mediante rectángulos distribuidos horizontalmente en la zona superior del diagrama a los que se asocia una línea temporal vertical para cada actor de las que salen los diferentes mensajes que se pasan entre ellos.

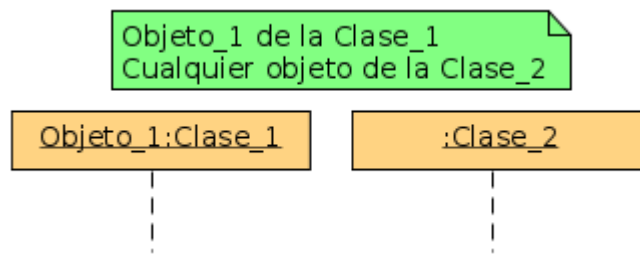
El equipo de desarrollo se puede hacer una idea de las distintas operaciones al ejecutar determinada tarea y su orden.

### 3.1.1. Representación de objetos, línea de vida y paso de mensajes

#### Representación de objetos y línea de vida



Los objetos son rectángulos y se distribuyen horizontalmente en la parte superior, por cada objeto se identifica su nombre seguido del símbolo de dos puntos y después el nombre de su clase. Si no se indica nombre del objeto, se considera que para el propósito del diagrama es válido cualquier objeto de la clase.



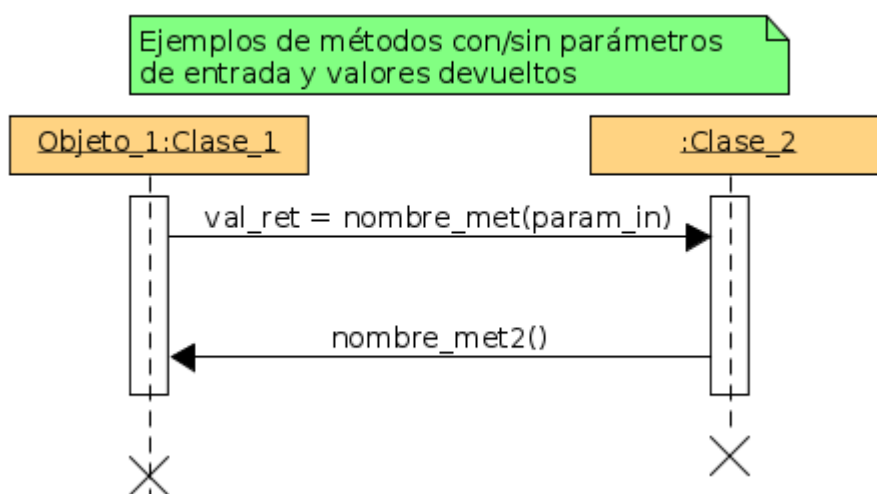
De cada rectángulo sale una línea discontinua que representa el paso del tiempo (línea de vida), esta línea representa la vida del objeto mientras es relevante en el diagrama, cuando deja de serlo se marca una cruz.

Una línea de vida puede ser encabezada por otro tipo de instancias como el sistema o un actor que aparecerán con su propio nombre. Usaremos el sistema para representar solicitudes al mismo, como por ejemplo pulsar una ventana o una llamada a una subrutina.

### Paso de mensajes (invocación de métodos)

Los mensajes significan invocación de métodos y se representan con flechas horizontales que van de una línea de vida a otra. Los mensajes se dibujan desde el objeto que envía el mensaje hasta el que lo recibe, pudiendo ser el mismo objeto emisor y receptor.

El orden en el tiempo va determinado por su posición vertical, un mensaje que se dibuja debajo de otro indica que se envía después. Los mensajes tienen un nombre y pueden incluir argumentos de entrada, valores devueltos e información de control (condición o iteración).

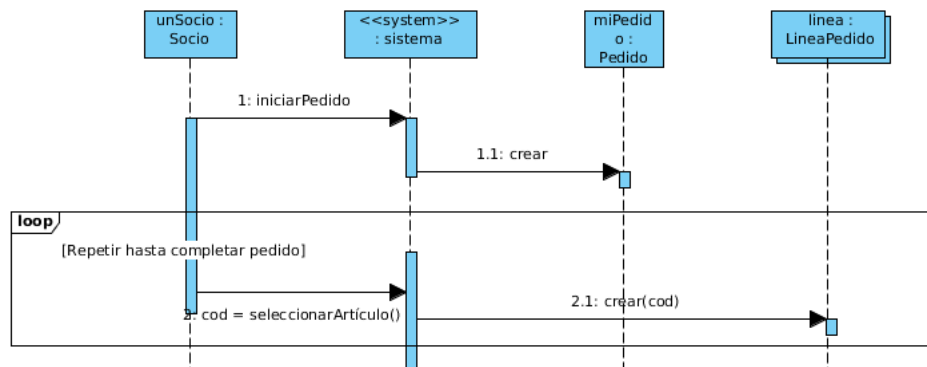


Una notación alternativa para recoger valores devueltos por los métodos es dibujar una línea de puntos finalizada en flecha, desde el objeto destinatario

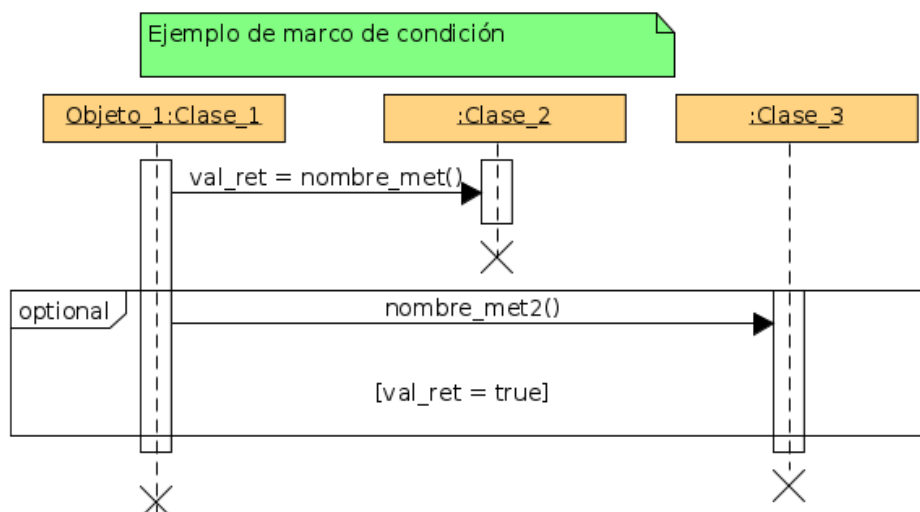
del mensaje al que lo ha generado, acompañado del valor devuelto.

## Condiciones e iteraciones

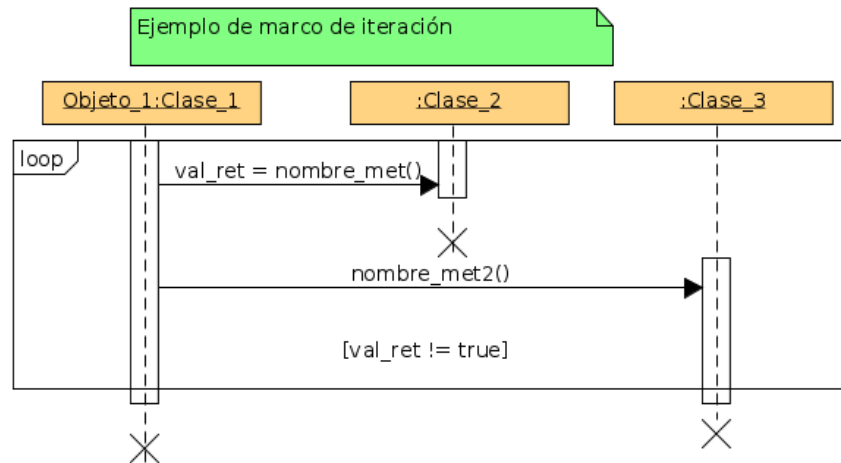
Las secuencias de control se pueden representar usando marcos, normalmente se nombra el marco con el tipo de bucle a ejecutar y la condición de parada. También se pueden representar flujos de mensajes condicionales en función de un valor determinado.



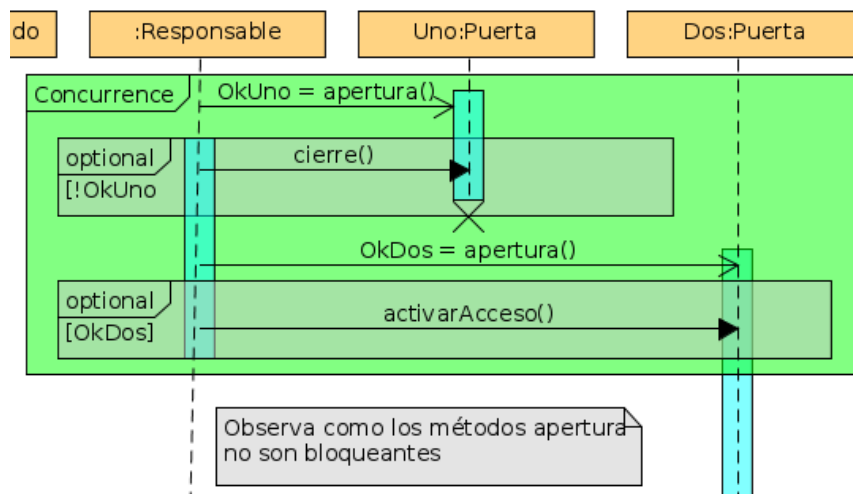
La expresión a evaluar para la condición o iteración se representa entre corchetes:



Combinando varios marcos opcionales es posible representar diferentes alternativas en la ejecución de un diagrama de secuencia. Para el caso de una iteración, tenemos el siguiente ejemplo.



Por defecto los métodos son bloqueantes, se entiende que el proceso del diagrama de secuencia completa cada método antes de continuar con el siguiente, es una secuencia de métodos en el tiempo. Pero en ocasiones se producen situaciones en las que se desea mostrar varios procesos en paralelo (conurrencia), se puede reflejar mediante el uso de marcos con la etiqueta concurrence.

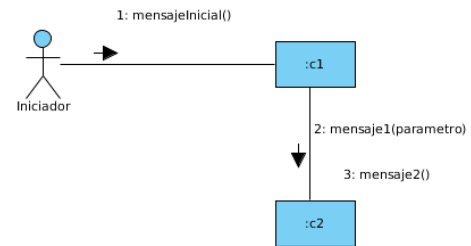


Se puede completar el diagrama añadiendo etiquetas y notas en el margen izquierdo que aclare la operación que se está realizando.

## 3.2. Diagramas de colaboración

Muestran una secuencia de ejecución de uno o varios casos de uso, al igual que los diagramas de secuencia.

La notación es muy similar y la principal diferencia radica en el modo de mostrar el orden de mensajes intercambiados entre objetos.



En el diagrama de colaboración se hace mediante el etiquetado de mensajes. Las interacciones entre objetos se describen en forma de grafo, en el que los nodos son objetos y las aristas son enlaces entre objetos a través de los cuales se envían mensajes entre ellos.

Permiten una mejor organización visual que los diagramas de secuencia, además son más fáciles de seguir.

UMLet no dispone de herramientas para la elaboración de diagramas de colaboración directamente. No obstante, no resulta complicado generarlos a partir de los símbolos disponibles para otros diagramas: representación de objetos mediante cajas, paso de mensajes mediante líneas, información de los métodos mediante descripciones textuales; todos ellos disponibles en los diagramas de secuencia de UMLet.

### 3.2.1. Representación de objetos

Un objeto puede ser cualquier instancia de las clases definidas en el sistema, pero también pueden incluirse objetos como la interfaz del sistema o el propio sistema, ayudándonos a modelar las operaciones que se llevan a cabo.

Los objetos se representan mediante rectángulos con sus nombres en el interior.

- **NombreClase:** se puede utilizar el nombre de la clase al que pertenece el objeto que participa en la interacción.
- **NombreObjeto:** se usa el nombre del objeto que participa en la interacción, normalmente subrayado.
- **:nombreClase:** cuando se colocan dos puntos delante del objeto significa que es un objeto genérico de esa clase.

Clase

:objeto

:Clase

objeto.clase

- **:nombreObjeto:nombreClase:** hace referencia al objeto concreto que se nombre añadiendo la clase a la que pertenece.

### 3.2.2. Paso de mensajes

Para hacer posible el paso de mensajes se necesita una asociación entre los objetos, que se hará mediante una línea que los una y una flecha que indique la dirección.

También se pueden incluir parámetros en mensajes, valores devueltos, condicionales y bucles.

La sintaxis de un mensaje es la siguiente:

- [Secuencia] [\*] [Condición] {valorDevuelto} : mensaje (argumentos de entrada)
- [Secuencia] [\*] [Condición] mensaje (argumentos de entrada) : {valorDevuelto}

**Secuencia:** representa el nivel de anidamiento del envío del mensaje dentro de la iteración. Los mensajes se numeran para indicar el orden en el que se envían y si es necesario se puede indicar anidamiento incluyendo subrangos.

\*: indica que el mensaje es iterativo.

**Condición de guarda:** debe cumplirse para que el mensaje pueda ser enviado.

**Valor devuelto:** lista de valores devueltos por el mensaje. Estos valores se pueden utilizar como parámetros de otros mensajes. los corchetes indican que es opcional.

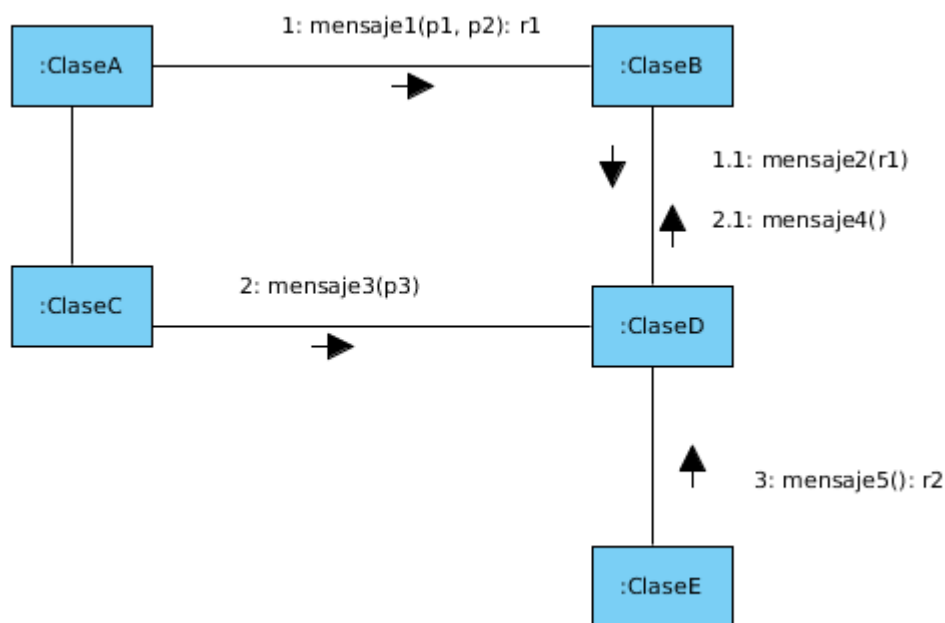
**Mensaje:** nombre del mensaje.

**Argumentos:** parámetros que se pasan al mensaje.

La enumeración de los mensajes se puede hacer de dos maneras:

- **Numeración simple:** Empieza en 1 y se va incrementando en 1 y no hay ningún nivel de anidamiento.
- **Numeración decimal:** se muestran varios niveles de subíndices para indicar anidamiento en operaciones. Por ejemplo, 1 es el primer mensaje, 1.1 es el primer mensaje anidado en el mensaje 1, 1.2 el segundo mensaje...

Como se ve en el ejemplo, se puede usar la misma asociación para enviar varios mensajes. Vemos que hay dos mensajes anidados, el 1.1 y el 2.1, se ha usado el nombre de los mensajes para indicar el orden real en el que se envían.



Los mensajes 1, 1.1 y 2 tienen parámetros y los mensajes 1 y 3 devuelven un resultado.

Se contempla la bifurcación en la secuencia añadiendo una condición en la sintaxis del mensaje:

[Secuencia][\*][CondiciónGuarda]{valorDevuelto} : mensaje (argumentos)

Cuando tenemos una condición se repite el número de secuencia y se añaden las condiciones necesarias, como vemos en la imagen según la condición se enviará el mensaje 1 o el 2, pero no ambos, por lo que coinciden en número de secuencia.

La iteración se representa mediante un \* al lado del número de secuencia, pudiendo indicarse entre corchetes la condición de parada del bucle.

## 4. Diagramas de estados

Los diagramas de estados permiten analizar como evoluciona el estado de un objeto a lo largo del tiempo, es decir, su comportamiento transitado por una serie de estados.

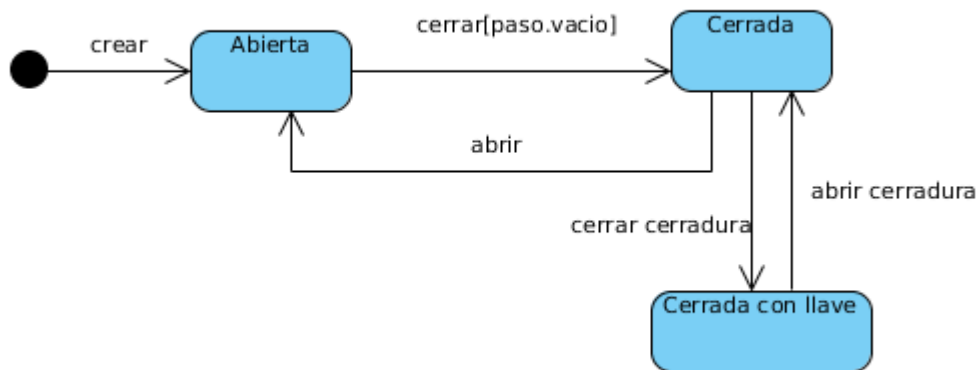
Modelan el comportamiento dinámico de los objetos en respuesta a los eventos.

En relación con el diagrama de estados se cumple que



- Un objeto está en un estado concreto en un cierto momento, que principalmente viene determinado por los valores de sus atributos.
- La transición de un estado a otro es momentánea y se produce cuando ocurre un evento.

Ejemplo de estados de una puerta:



## 4.1. Estados y eventos

Un estado es una situación en la vida del objeto en la que satisface cierta condición, realiza alguna actividad o espera algún evento.

Existen tres tipos de estados:

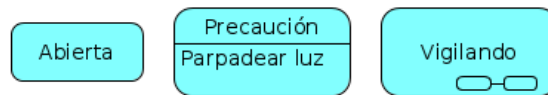
- **Estado inicial.** Punto de partida del diagrama de estados, corresponde a los valores de los atributos de la clase en el momento de instanciar.
- **Estado final.** Estado en el que se encuentra el objeto una vez finalizada la secuencia de eventos.
- **Estado intermedio.** Cualquiera de los estados intermedios entre los dos anteriores.



Los estados se representan mediante una caja y admite variantes, la información mostrada en los estados suele ser:

- Nombre de estado.

- Nombre de estado y acción/actividad asociada al objeto. En un semáforo en estado precaución se produce parpadeo de luz.
- Estado con subestados. En el ejemplo se indica que el estado vigilando tiene asociados, si se trata de un vigilante de seguridad, el estado vigilando poría tener relacionados "de ruta a pie" o "visionado de cámaras".



Un evento es un acontecimiento que dispara una transición entre dos estados del objeto. Existen eventos externos e internos, según el agente que los produzca:

- **Señales (excepciones).** La recepción de una señal producida por una situación excepcional en el sistema.
- **Llamadas:** la recepción de una petición para invocar una operación, normalmente un evento de llamada se maneja por el método de un objeto.
- **Paso de tiempo:** se genera como consecuencia del cumplimiento de un temporizador.
- **Cambio de estado:** se genera por el cambio en el estado o el cumplimiento de una condición.

## 4.2. Transiciones

Una transición de un estado A a un estado B se produce cuando se origina el evento asociado y se satisface cierta condición especificada, lo cual hace que se ejecute la acción de salida de A, la acción de entrada a B y la acción asociada a la transición.

La notación de una transición tiene tres partes, todas optativas:

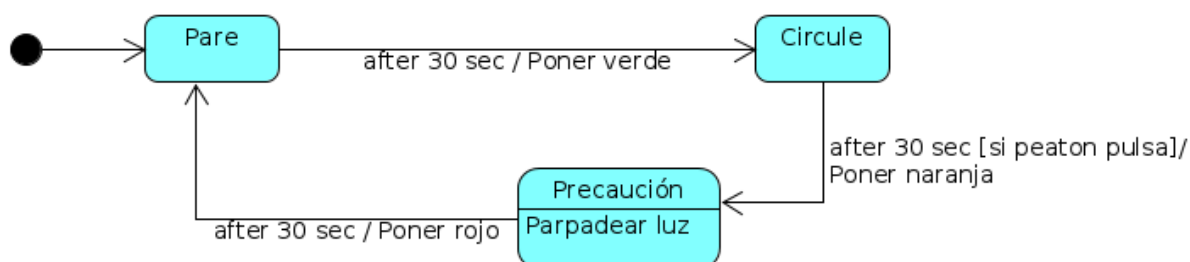
Evento(argumentos)[condición]/acción

Elementos de una transición:

- **Evento:** cuando se produce un evento afecta a todas las transiciones que lo contienen en su etiqueta.
- **Condición:** expresión evaluable como verdadera o falsa, si es falsa no se dispara.

- **Acción.** Conjunto de actuaciones que lleva asociada la transición. Pueden incluir llamadas a operaciones de objetos, creación o destrucción de objetos.

Por ejemplo en el diagrama de estados de un semáforo:



## 5. Diagramas de actividad

Los diagramas de actividad son una especialización del diagrama de estados, organizado en torno a las acciones en lugar de los objetos, se compone de una serie de actividades y representa como se pasa de unas a otras.

Las actividades se enlazan por transiciones automáticas. Es decir, cuando una actividad termina se desencadena el paso a la siguiente.

Resultan útiles cuando se quieren representar solo las acciones que tienen lugar, prescindiendo de quien las genera.

Se usan para modelar el flujo de control entre actividades, en el que se puede distinguir cuales ocurren secuencialmente a lo largo del tiempo y cuales se pueden usar concurrentemente. Permite visualizar la dinámica del sistema desde otro punto de vista complementando al resto de diagramas.

Es un grafo conexo en el que los nodos son estados que pueden ser de actividad o acción y los arcos son transiciones.

### 5.1. Elementos del diagrama de actividad

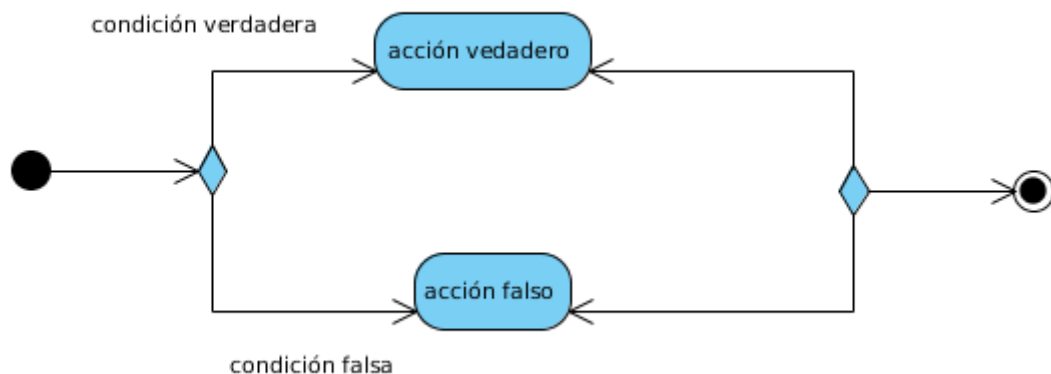
- **Estados:** de actividad y de acción
  - **De actividad:** elemento compuesto cuyo flujo de control se compone de otros estados de actividad y acción.
  - **De acción:** representa la ejecución de una acción atómica, que no se puede descomponer ni interrumpir, normalmente la invocación de una

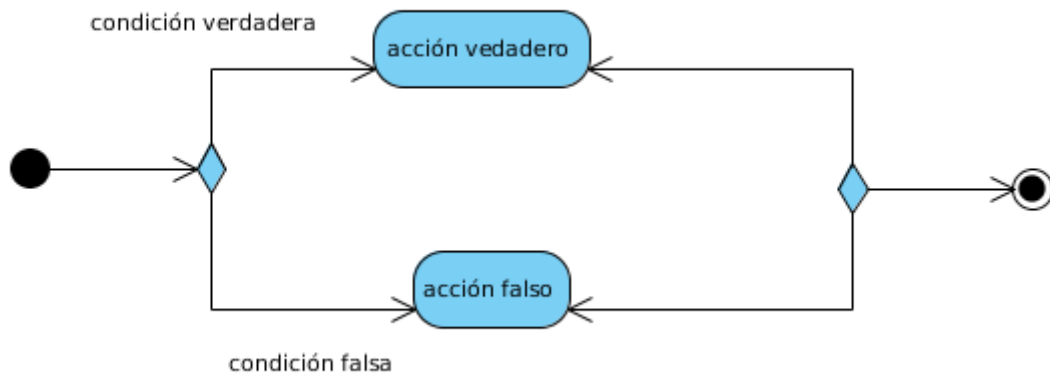
operación. El tiempo de ejecución normalmente es un tiempo insignificante.

- **Otros tipos de estado:** Inicial y Final.

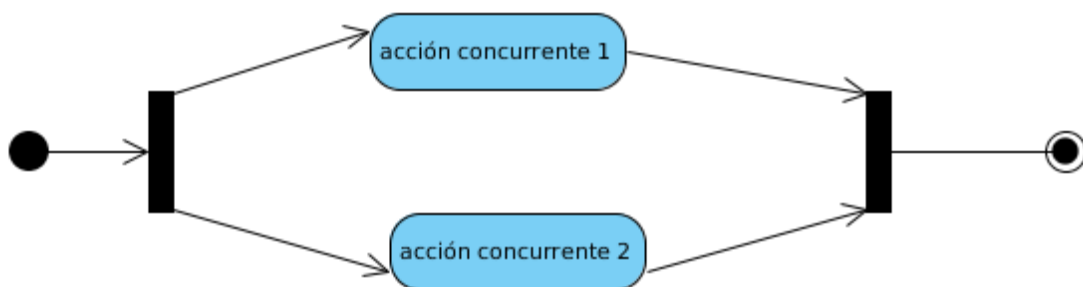


- **Transiciones:** relación entre dos estados que indica que un objeto en el primer estado realiza ciertas acciones y pasará al segundo estado cuando ocurra un evento específico y satisfaga ciertas condiciones. Se representa mediante una línea dirigida del estado inicial al siguiente.
  - **Secuencial o sin disparadores.** Al completar la acción del estado de origen se ejecuta la acción de salida, sin ningún retraso, el control sigue por la transición y pasa al siguiente estado.
  - **Bifurcación (Decision node).** Especifica caminos alternativos, elegidos según el valor de una expresión booleana. Las condiciones deben cubrir todas las posibilidades, puede usarse la palabra else y pueden utilizarse para lograr el efecto de iteraciones.

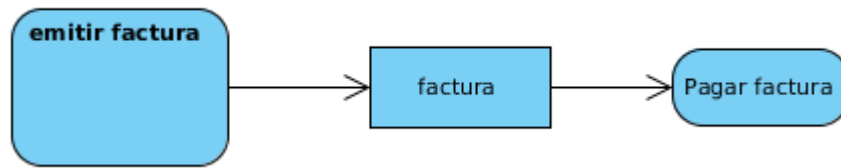




- **Fusion (Merge node):** Redirigen varios flujos de entrada en un único flujo de salida. No requiere tiempo de espera ni sincronización.
- **Division (Fork node).** Expresa la sincronización o ejecución paralela de actividades. Las actividades invocadas después de una división son concurrentes.



- **Union (Join node).** En la unión de flujos entrantes se sincronizan, es decir, cada uno espera hasta que todos los flujos de entrada han alcanzado la unión.
- **Objetos:** Manifestación concreta de una abstracción o instancia de una clase, cuando interviene un objeto no se utilizan flujos de eventos habituales, sino flujos de objetos, que se representan con flecha igualmente y permiten mostrar objetos que participan dentro del flujo de control asociado a un diagrama de actividades. Se puede indicar como cambian los valores de sus atributos, su estado o sus roles.



Utilizamos carriles o calles para ver quienes son los responsables en realizar distintas actividades y especificar que parte de la organización es responsable de una actividad.

- Cada calle tiene un nombre único dentro del diagrama.
- Puede ser implementada por una o varias clases.
- Las actividades de cada calle se consideran independientes y se ejecutan concurrentemente a las otras calles.

## Mapa conceptual

## Anexos

### Ejercicio resuelto 1 ("ZAPATERÍA TACÓN DE ORO"). Elaboración de un diagrama de casos de uso

#### Descripción del problema: "El tacón de oro".

La zapatería Tacón de oro ha decidido crear un espacio web para ampliar su línea de negocio, así sus usuarios podrán adquirir los artículos: zapatos, bolsos y complementos que se venden en la tienda.

Los usuarios del sistema navegarán por la web para ver los artículos, zapatos, bolsos y complementos que se venden en la tienda. De los artículos nos interesa su nombre, descripción, material, color, precio y stock. De los zapatos nos interesa su número y el tipo. De los bolsos nos interesa su tipo (bandolera, mochila, fiesta). De los complementos (cinturones y guantes) su talla.

Los artículos se organizan por campañas para cada temporada (primavera/verano y otoño/invierno) de cada año.

Los artículos son de fabricación propia, pero, opcionalmente, pueden venderse artículos de otras firmas. De las firmas nos interesa saber su nombre, CIF y domicilio fiscal. La venta de artículos de firma se realiza a través de

proveedores, de forma que un proveedor puede llevar varios artículos de diferentes firmas, y una firma puede ser suministrada por más de un proveedor. Los artículos pertenecen a una firma solamente. De los proveedores debemos conocer su nombre, CIF, y domicilio fiscal.

Los usuarios pueden registrarse en el sitio web para hacerse socios. Cuando un usuario se hace socio debe proporcionar los siguiente datos: nombre completo, correo electrónico y dirección.

Los socios pueden hacer pedidos de los artículos. Los usuarios pueden consultar todos los productos que tienen a su disposición, pero para realizar compras han de registrarse como socios.

Para comprar productos, se generan pedidos. Un pedido está formado por un conjunto de detalles de pedido que son parejas formadas por artículo y la cantidad. De los pedidos interesa saber la fecha en la que se realizó y cuanto debe pagar el socio en total. El pago se hace a través tarjeta bancaria, cuando se va a pagar una entidad bancaria comprueba la validez de la tarjeta. De la tarjeta interesa conocer el número.

Las campañas son gestionadas por el administrativo de la tienda que se encargará de dar de baja la campaña anterior y dar de alta la nueva siempre que no haya ningún pedido pendiente de cumplimentar.

Existe un empleado de almacén que revisa los pedidos a diario y los cumplimenta. Esto consiste en recopilar los artículos que aparecen en el pedido y empaquetarlos. Cuando el paquete está listo se pasa al almacén a la espera de ser repartido. Del reparto se encarga una empresa de transportes que tiene varias rutas preestablecidas. Según el destino del paquete (la dirección del socio) se asigna a una u otra ruta. De la empresa de transportes se debe conocer su nombre, CIF y domicilio fiscal. Las rutas tienen un área de influencia que determina los destinos, y unos días de reparto asignados. Se debe conocer la fecha en la que se reparte el pedido. Si se produce alguna incidencia durante el reparto de algún pedido se almacena la fecha en la que se ha producido y una descripción.

Los socios pueden visualizar sus pedidos y una vez comprobados, puede cancelarlos (siempre y cuando no hayan sido cumplimentados por el empleado de almacén) o confirmar la compra. Las compras deberán ser abonadas a través de una entidad bancaria. Así mismo los socios puede modificar sus datos personales.

**Se pide:**

- Diagrama de casos de uso. Identifica los actores y casos de uso, incluye relaciones de asociación, identifica generalizaciones (de actores y de casos de uso). Si consideras alguna relación tipo include o extend, justifica su uso.
- Del diagrama que hayas obtenido en el apartado anterior, agrupa todos los casos de uso que hayas considerado en el proceso de gestión de pedidos en uno sólo. Para este apartado, se entiende que el pedido ya ha sido dado de alta y que el proceso normal de un pedido es que termine siendo comprado. Desarrolla su notación escrita (tabla del caso de uso).

### **Solución:**

Los diagramas de casos de uso quedan encuadrados en la fase de análisis de los proyectos, se entienden como una puesta en común entre el cliente y el analista sobre como entender requisitos funcionales.

Al utilizarse UML, se entiende que ambos interlocutores conocen la notación propia de estos diagramas y las reglas sobre como combinar los diferentes símbolos, de forma que no hay ambigüedades que sí podrían darse mediante una descripción escrita.

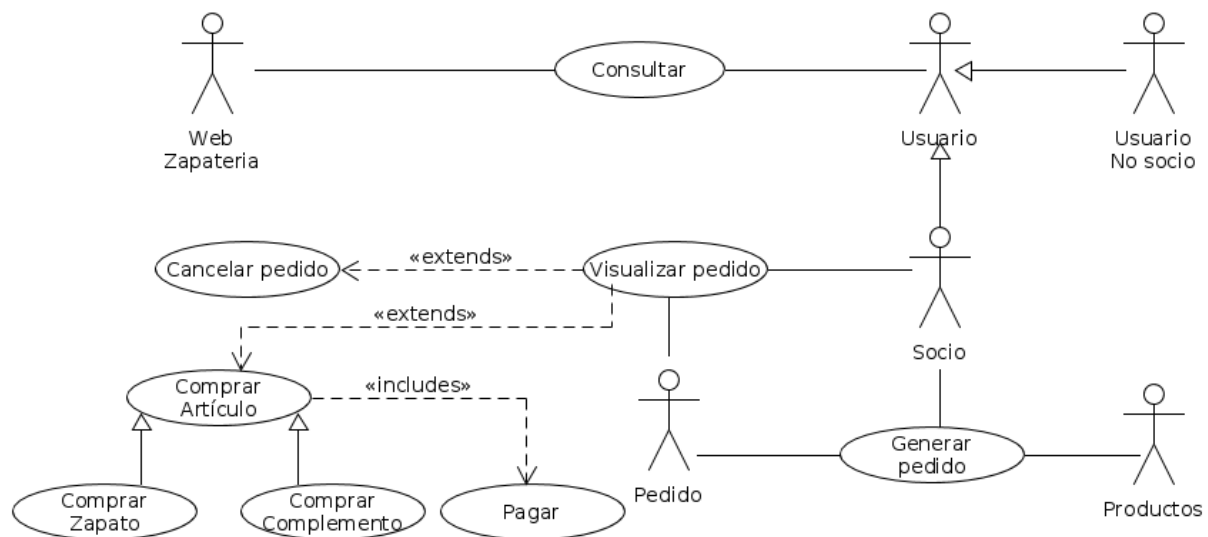
Completar un diagrama de casos de uso supone versionar el diagrama tantas veces como sean necesarias hasta que la vista del proyecto que se presenta al cliente garantiza que el requisito funcional ha sido entendido.

En nuestros ejercicios, sólo hay una descripción que no permite ir refinando el diagrama mediante sucesivas consultas con el cliente, por lo tanto, las soluciones propuestas podrán ser diversas, todas ellas válidas siempre que reflejen razonablemente lo propuesto en el enunciado.

A la hora de valorar el ejercicio, lo que no es interpretable, es que el uso de la notación sea el correcto y que se muestre una variedad de los símbolos que conocemos para estos diagramas.

Con estas consideraciones, se propone la siguiente solución:





### Actores que participan en el problema.

Zapatería (web), usuario (que podrá ser socio o no socio), productos/artículos (que podrán ser zapatos o complementos), y pedidos.

### Generalizaciones.

Se observan dos posibles generalizaciones, que en el desarrollo del diagrama podrán ser expresadas en relación a actores o en relación a los casos de uso en los que participan. En la solución propuesta se ha decidido hacer una generalización de actores -- usuario socio o no socio -- y de casos de uso -- comprar zapatos o complementos (que podría haber sido una generalización de productos) --.

### Relaciones extends.

De la ejecución del caso de uso visualizar pedido se pueden derivar los casos de uso cancelar pedido y comprar artículo. Se trata de casos de uso que no tienen porque siempre llevarse a cabo.

### Relaciones includes/use.

Cuando se realiza la compra de un artículo, siempre habrá que pagarlo.

### Notas:

- Puesto que todos los usuarios pueden consultar la web de la zapatería, la relación consultar web la hace el actor usuario (padre de la relación de generalización), en cambio, generar pedido sólo puede llevarse a cabo por los socios, así que es éste el que participa en este caso de uso.

- También podría haberse considerado el actor entidad bancaria, partícipe del caso de uso pagar.

Como se indicaba en los contenidos del apartado 2.2, lo más importante en la elaboración de un diagrama de casos de uso, no es el diagrama en sí, sino la documentación de los casos de uso que es lo que permitirá desarrollar otros diagramas que ayuden en la codificación del sistema, y la elaboración de los casos de prueba de caja negra.

A modo de ejemplo vamos a desarrollar la documentación del caso de uso **Generar Pedido**, ya que, por su complejidad abarca todos los apartados que hemos visto. El ejemplo se hará con la herramienta Visual Paradigm for UML, aunque tu puedes usar la herramienta que consideres más oportuna

Los datos que debemos incluir para elaborar la documentación del caso de uso eran:

- **Nombre:** nombre del caso de uso.
- **Actores:** aquellos que interactúan con el sistema a través del caso de uso.
- **Propósito:** breve descripción de lo que se espera que haga.
- **Precondiciones:** aquellas que deben cumplirse para que pueda llevarse a cabo el caso de uso.
- **Flujo normal:** flujo normal de eventos que deben cumplirse para ejecutar el caso de uso exitosamente.
- **Flujo alternativo:** flujo de eventos que se llevan a cabo cuando se producen casos inesperados o poco frecuentes. No se deben incluir aquí errores como escribir un tipo de dato incorrecto o la omisión de un parámetro necesario.
- **Postcondiciones:** las que se cumplen una vez que se ha realizado el caso de uso.

Para incluir el nombre, actores, propósito, precondiciones y postcondiciones abrimos la especificación del caso de uso. Esto da lugar a la aparición de una ventana con un conjunto de pestañas que podemos rellenar:

- En la pestaña "**General**" rellenamos el nombre "**Hacer pedido**", y tenemos un espacio para escribir una breve descripción del caso de uso, por ejemplo:

*"El cliente visualiza los productos que están a la venta, que se pueden seleccionar para añadirlos al pedido. Puede añadir tantos artículos como desee, cada artículo añadido modifica el total a pagar según su precio y la cantidad seleccionada.*

*Cuando el cliente ha rellenado todos los productos que quiere comprar debe formalizar el pedido.*

*En caso de que el cliente no sea socio de la empresa antes de formalizar la compra se le indica que puede hacerse socio, si el cliente acepta se abre el formulario de alta, en caso contrario se cancela el pedido.*

*En caso de que se produzca algún problema con los datos bancarios se ofrecerá la posibilidad de volver a introducirlos.*

*Al finalizar un pedido se añade al sistema con el estado pendiente."*

- En la pestaña "**Valores etiquetados**" encontramos un conjunto de campos predefinidos, entre los que se encuentran el autor, precondiciones y postcondiciones, que podemos rellenar de la siguiente manera:

**Autor:** *usuario.*

**Precondiciones:** *Existe una campaña abierta con productos de la temporada actual a la venta.*

**Postcondiciones:** *Se ha añadido un pedido con un conjunto de productos para servir con el estado "pendiente" que deberá ser revisado.*

También se pueden incluir otros datos como la complejidad, el estado de realización del caso de uso la complejidad que no hemos visto en esta unidad.

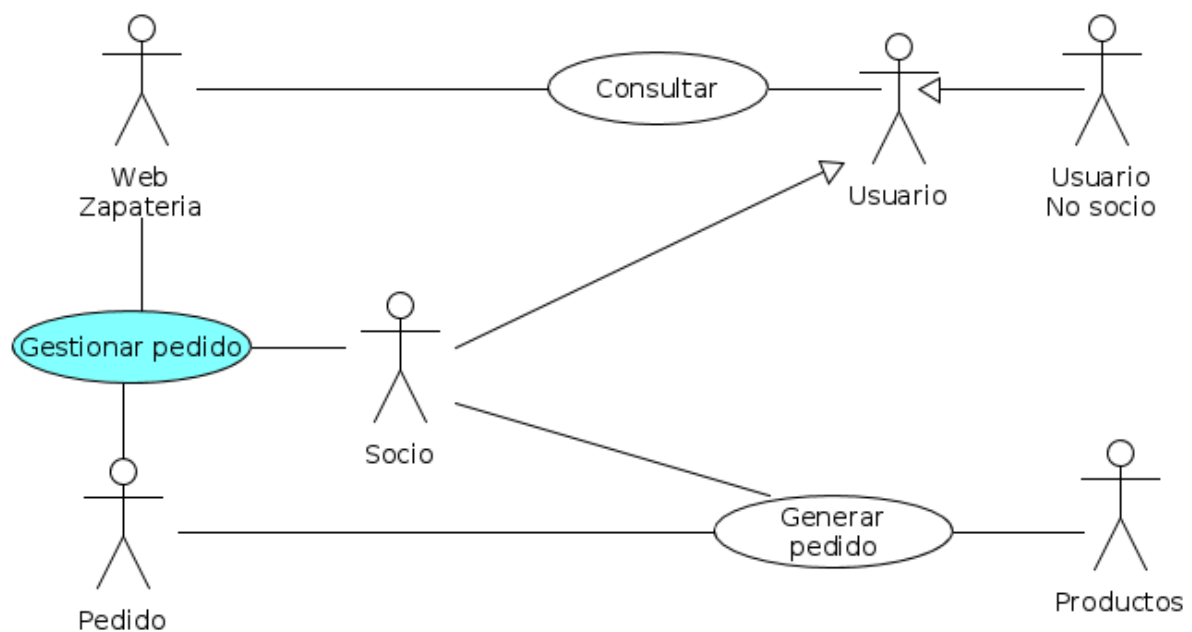
Para incluir el resto de los datos en el caso de uso hacemos clic en la opción "**Open Use Case Details...**" del menú contextual, lo que da lugar a la aparición de una ventana con una serie de pestañas. En ellas se recupera la información de la especificación que hemos rellenado antes, además, podemos rellenar el flujo de eventos del caso de uso, en condiciones normales usaríamos la pestaña "Flow of events", sin embargo como esta opción solo está disponible en la versión profesional, utilizaremos la pestaña "**Descripción**", que está disponible en la versión community. Para activarla pulsamos el botón "**Create/Open Description**".

Podemos añadir varias descripciones de diferentes tipos, pulsando el botón "**Nuevo**". Para añadir filas al flujo de eventos pinchamos en el botón. En

principio añadimos la descripción principal, luego añadiremos otras alternativas.

### **Tabla gestionar pedido.**

Una vez adaptado el diagrama para obtener el caso de uso gestionar pedidos, podría quedar como sigue:



La tabla resultante sería:

|                            |  |  |
|----------------------------|--|--|
| <b>Caso de uso</b>         | Gestionar pedidos.   |  |
| <b>Actores</b>             | Principal: usuario-socio. Secundarios: pedido, zapatería.                    |  |
| <b>Propósito</b>           | Gestión de los pedidos ya dados de alta en el sistema.                       |  |
| <b>Pre-condiciones</b>     | El pedido ha sido generado y está en modo activo. El usuario debe ser socio. |  |
| <b>Flujo normal</b>        | El flujo normal finaliza con la compra del pedido.                           | Visualizado del pedido.  |
|                            |  | Compra del pedido.   |
|                            |  | Pago del pedido.   |
|                            |  | Fin de flujo.  |
| <b>Flujos alternativos</b> | Cancelación del pedido.  | Visualizado del pedido.  |
|                            |  | Cancelación del pedido. El pedido desaparece de la lista de pedidos activos. |
|                            |  | Fin de flujo.  |
|                            | Imposibilidad de efectuar el pago.   | Visualizado del pedido.  |
|                            |  | Compra del pedido.   |
|                            |  | Error en la operación de pago. La compra del pedido no se completa.          |
|                            |  | Fin de flujo.  |

|                                |   |
|--------------------------------|---|
| <b>Post-condiciones</b>        | Las post-condiciones son estados en los que ha de quedar el sistema siempre, sea cual sea el flujo ejecutado (normal o alternativo). Para este enunciado no se identifica ninguna post-condición en particular. |
| <b>Requerimientos trazados</b> | Los que correspondan en el documento de requisitos. Recuerda que el tipo de requisitos que se consideran en los diagramas de casos de uso son siempre de tipo funcional.  |
| <b>Puntos de inclusión</b>     | Ninguno. Este diagrama no dispone de otros casos de uso con los que establecer estas relaciones.  |
| <b>Puntos de extensión</b>     | Ninguno. Este diagrama no dispone de otros casos de uso con los que establecer estas relaciones.  |

Esta es la descripción principal, en ella se describe el flujo normal de eventos que se producen cuando se ejecuta el caso de uso sin ningún problema.

### Flujo de eventos normal para el caso de uso Hacer Pedido.

|                          |  |
|--------------------------|--|
| <b>Use Case</b>          | Hacer pedido   |
| <b>Author</b>            | usuario  |
| <b>Date</b>              | 26-ago-2011 13:56:56   |
| <b>Brief Description</b> | EL usuario selecciona un conjunto de artículos, junto con la cantidad de los mismos, para crear el pedido. Cuando se formaliza se comprueba que el usuario sea socio. A continuación se comprueban los datos bancarios, se realiza el cobro y se crea el pedido. |
| <b>Preconditions</b>     | Existe un catálogo de productos disponibles para pedir.<br>El usuario está registrado.<br>Los datos bancarios son correctos.   |
| <b>Post-conditions</b>   | Se crea un pedido con los datos del usuario que lo realiza y los artículos solicitados.  |

| Flow of Events |    | Actor Input   | System Response   |
|----------------|----|---|---|
|                | 1  | Inicia el pedido.   |   |
|                | 2  |   | Se crea un pedido en estado "en construcción".  |
|                | 3  | Selecciona un artículo.                                     |   |
|                | 4  | Selecciona una cantidad.                                    |   |
|                | 5  |   | Recupera la información del artículo para obtener el precio y modifica el precio total del pedido.        |
|                | 6  | El proceso se repite hasta completar la lista de artículos. |   |
|                | 7  | Se acepta el pedido.  |   |
|                | 8  |   | Se comprueba si el usuario es socio, si no lo es se le muestra un aviso para que se registre en el sitio. |
|                | 9  |   | Se comprueban los datos bancarios con una entidad externa.  |
|                | 10 |   | Se genera: calcula el total, sumando los gastos de envío.   |
|                | 11 |   | Se realiza el pago a través de la entidad externa.  |
|                | 12 |   | Se almacena la información del pedido con el estado "pendiente".  |

Añadimos un par de descripciones alternativas para indicar que hacer cuando el usuario no es socio y cuando los datos bancarios no son correctos:

**Flujo alternativo para el caso de uso Hacer Pedido cuando el usuario no está registrado.**

|                          |  |
|--------------------------|--|
| <b>Author</b>            | usuario  |
| <b>Date</b>              | 26-ago-2011 18:14:35   |
| <b>Brief Description</b> | Una vez que se han seleccionado los artículos y se han introducido los datos del socio, al hacer la comprobación de los datos bancarios con la entidad externa se produce algún error, se da la posibilidad al usuario de modificar los datos o de cancelar el pedido. |
| <b>Preconditions</b>     | Existe un catálogo de productos disponibles para pedir.<br>El usuario está registrado.<br>Los datos bancarios no son correctos.  |
| <b>Post-conditions</b>   | Se crea un pedido con los datos del usuario que lo realiza y los artículos solicitados.  |

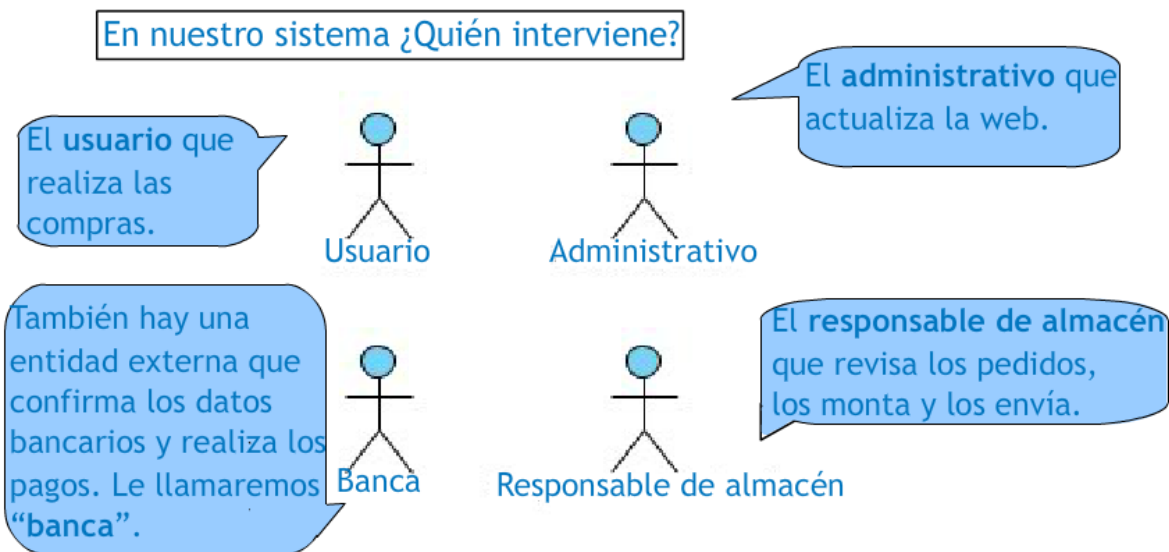
| Flow of Events |    | Actor Input   | System Response  |
|----------------|----|---|--|
|                | 1  | Inicia el pedido.   |  |
|                | 2  |   | Se crea un pedido en estado "en construcción".   |
|                | 3  | Selecciona un artículo.                                     |  |
|                | 4  | Selecciona una cantidad.                                    |  |
|                | 5  |   | Recupera la información del artículo para obtener el precio y modifica el precio total del pedido. |
|                | 6  | El proceso se repite hasta completar la lista de artículos. |  |
|                | 7  | Se acepta el pedido.  |  |
|                | 8  | Acepta el pedido.   |  |
|                | 9  |   | Se comprueban los datos bancarios con una entidad externa, fallando la comprobación.               |
|                | 10 |   | Se solicitan los datos de nuevo.   |
|                | 11 | Introduce los datos de nuevo.                               |  |
|                | 12 |   | Se repite el proceso hasta que se acepten los datos bancarios o se cancele la operación.           |
|                | 13 |   | Se genera: calcula el total, sumando los gastos de envío.  |
|                | 14 |   | Se realiza el pago a través de la entidad externa.   |
|                | 15 |   | Se almacena la información del pedido con el estado "pendiente".                                   |

**Debes conocer:**



## Primeros pasos

- Antes de elaborar el diagrama tienes que leer con detenimiento el documento con la especificación del problema a resolver y asegurarte de que entiendes la idea central del problema, crear una tienda virtual en la que se puedan realizar pedidos de los productos a la venta (zapatos). El proceso se centra en el pedido, desde poner a disposición del cliente los artículos en venta, pasando por la selección de artículos a pedir, la cumplimentación de toda la información necesaria para el pedido, pago, confección del pedido, envío y reajuste del stock en almacén, todo ello, a través de la web.



## Identificar funcionalidades

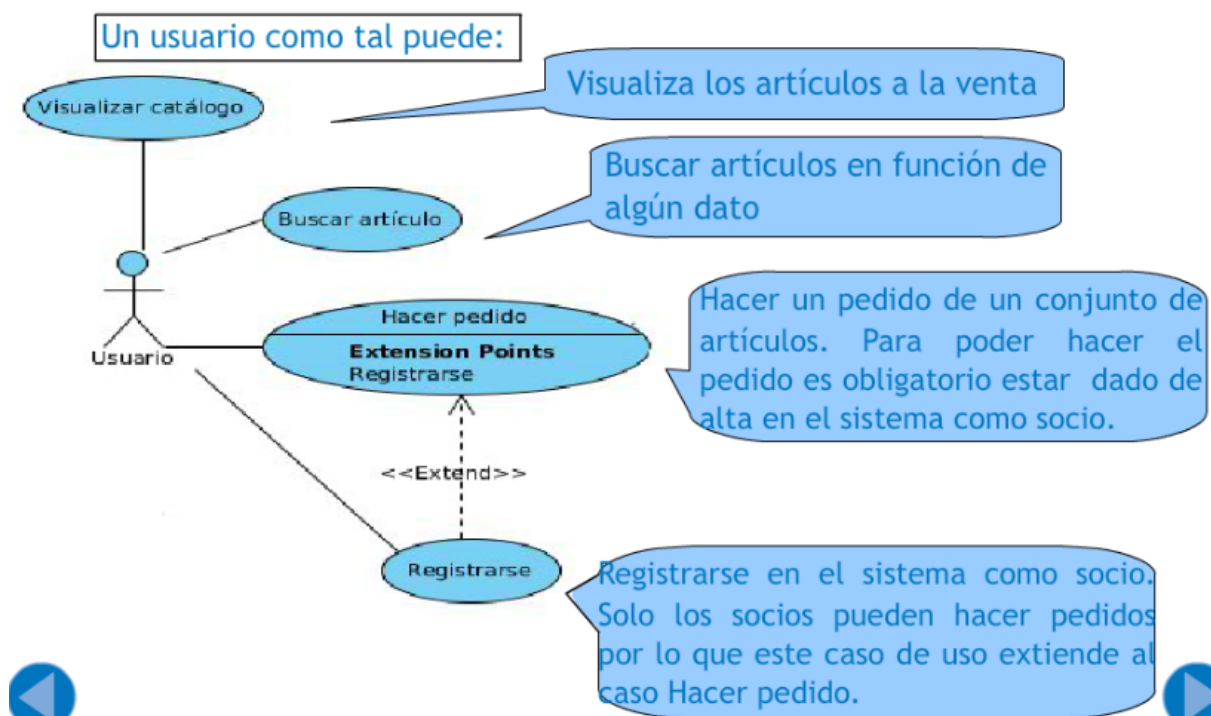
- Para facilitar la creación del diagrama vamos a ir sacando funcionalidades para casa usuario.

- Debemos recordar que una caso de uso representa una interacción de un actor con el sistema, que está relacionado con los requisitos funcionales de la aplicación final y que, en definitiva representa tareas que llevará a cabo el sistema.

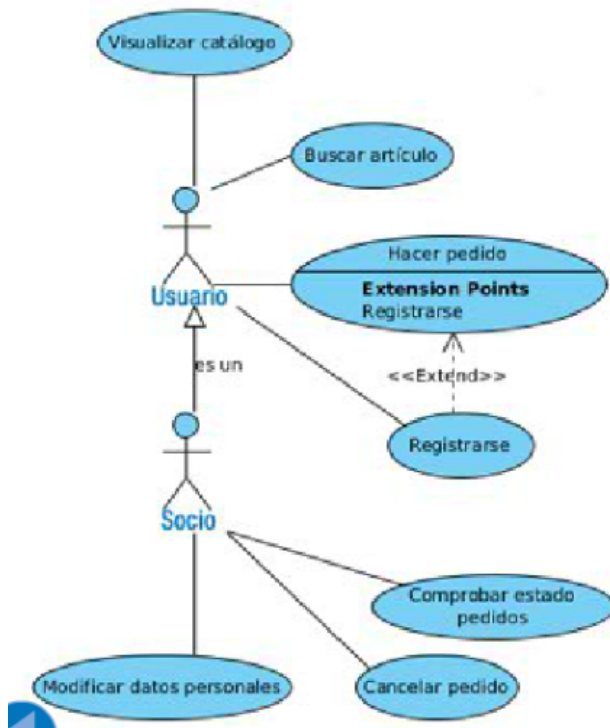
## Funcionalidades del usuario

- Cuando una persona se conecta al sistema lo primero que podrá hacer será **visualizar** el catálogo de la temporada.
- También puede hacer un **pedido** con uno o varios artículos del catálogo, para ello visualizará los artículos de forma que pueda seleccionar algunos de ellos e indicar la cantidad que quiere comprar.
- También puede hacer **búsquedas** por datos concretos de artículos.
- Cualquier persona que acceda al sistema puede **darse de alta** para ser socio.
- Así mismo, si es socio, podrá **comprobar el estado** de sus pedidos y **cancelarlos**.

## Funcionalidades del usuario



## Funcionalidades del usuario



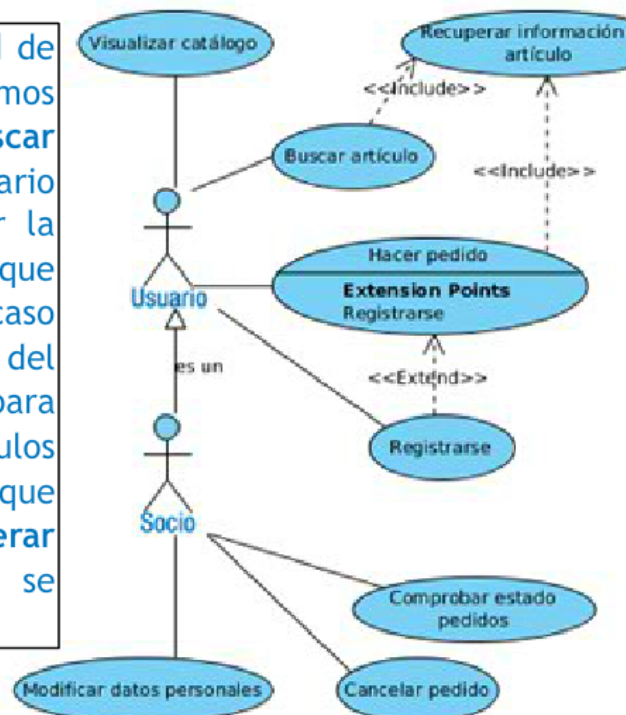
Al hacerse socio la identidad del usuario con respecto del sistema cambia, por lo que surge un nuevo tipo de actor que hereda de usuario:

Un socio es un usuario que se ha registrado en el sistema. Los socios almacena cierta información, como sus datos personales y bancarios, email, etc.

El socio, además, reúne una serie de funcionalidades propias (el usuario no puede hacerlas), como comprobar el estado de sus pedidos, o cancelarlos, así como modificar sus datos en el sistema

## Funcionalidades del usuario

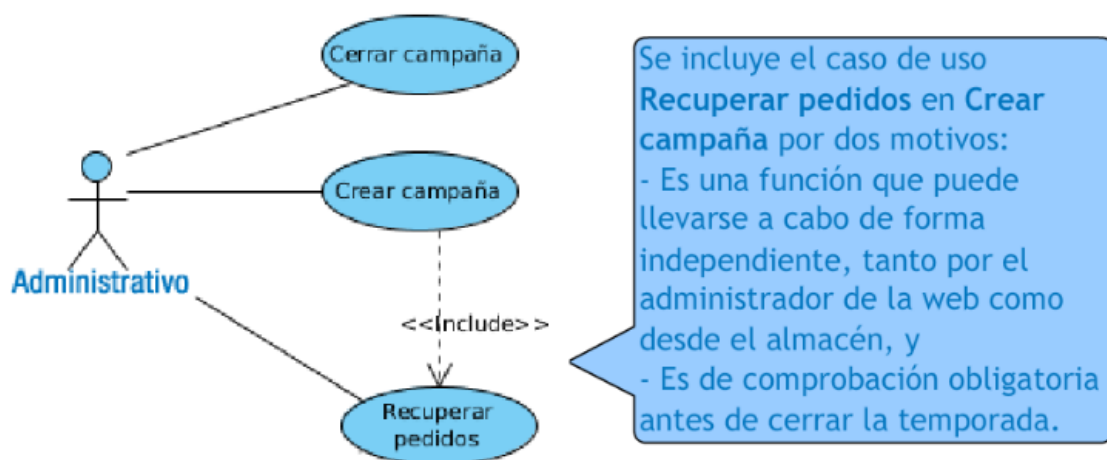
Al revisar un poco la funcionalidad de los casos de uso del usuario podemos comprobar que en los casos **Buscar artículo** y **Hacer pedido** es necesario buscar en el sistema y recuperar la información de un artículo del que tenemos algún dato, en el primer caso para obtener todos los datos del artículo buscado y en el segundo para recuperar el precio de los artículos que se añaden al pedido, por lo que extraemos el caso uso "Recuperar información de artículo" que se incluye en los otros dos.



## Funcionalidades del administrador

- El objetivo del administrador web es gestionar los contenidos de la web, en concreto de las diferentes campañas, ya que cada temporada se debe cerrar la campaña antigua, retirando los artículos de la temporada anterior y abrir la temporada nuevo, añadiendo sus artículos. Para que se pueda cerrar una temporada es necesario que en el almacén se hayan gestionado todos sus pedidos, por lo que es obligatorio comprobarlo, antes de cerrar.

## Funcionalidades del administrador



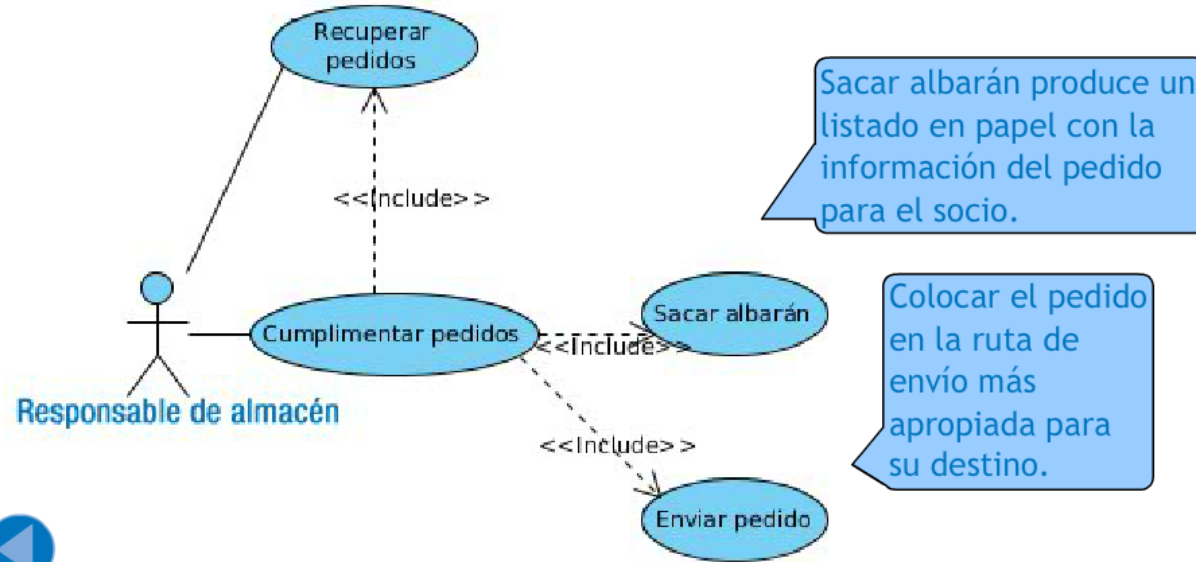
## Funcionalidades del responsable de almacén

- Es el encargado de leer los pedidos de los usuarios y cumplimentarlos, está será su única función, si bien, es una función complicada, ya que implica realizar una serie de tareas:

- Seleccionar el pedido más antiguo.
- Buscar los artículos a servir.
- Empaquetarlos junto con un albarán para el socio.
- Colocarlos en su ruta de envío.

## Funcionalidades del responsable de almacén

El primer paso es recuperar la lista de pedidos sin procesar. Esta tarea recupera el pedido más antiguo para ser procesado.



## Ejercicio resuelto 2 ("QUIJOTE"). Elaboración de un diagrama de casos de uso.

La empresa Quijote se dedica a la venta de material informático puerta a puerta, ofrece sus productos a los clientes en sus propios domicilios. Sus empleados se organizan en dos grandes grupos: vendedores y publicitarios. Los publicitarios tratan de facilitar el acceso de los vendedores a los clientes para que éstos les hagan llegar los catálogos de productos y realicen las operaciones de ventas.

Los publicitarios anualmente encargan a la consultora Sancho un estudio de sus resultados, y en función de los datos que desprenda pueden realizar un análisis de mercado. Gracias a la información obtenida en el análisis se hacen campañas publicitarias en radio y televisión.

La política de la empresa Quijote se basa en tener grandes profesionales en sus filas, por lo que todos sus empleados reciben formación periódicamente.

### Se pide:

- Diagrama de casos de uso. Identifica los actores y casos de uso, incluye relaciones de asociación, identifica generalizaciones. Si consideras alguna relación tipo include o extend, justifica su uso.

- Si agrupamos todos los casos de uso que hayas considerado en el proceso relacionado con los empleados de marketing/publicitarios en uno sólo. Desarrolla la notación escrita del caso de uso.

### **Resolución:**

Los diagramas de casos quedan encuadrados en la fase de análisis de los proyectos, se entienden como una puesta en común entre el cliente y el analista sobre como entender requisitos funcionales.

Al utilizarse UML, se entiende que ambos interlocutores conocen la notación propia de estos diagramas y las reglas sobre como combinar los diferentes símbolos, de forma que no hay ambigüedades que sí podrían darse mediante una descripción escrita.

Completar un diagrama de casos de uso supone versionar el diagrama tantas veces como sean necesarias hasta que la vista del proyecto que se presenta al cliente garantiza que el requisito funcional ha sido entendido.

En nuestros ejercicios, sólo hay una descripción que no permite ir refinando el diagrama mediante sucesivas consultas con el cliente, por lo tanto, las soluciones propuestas podrán ser diversas, todas ellas válidas siempre que reflejen razonablemente lo propuesto en el enunciado.

A la hora de valorar el ejercicio, lo que no es interpretable, es que el uso de la notación sea el correcto y que se muestre una variedad de los símbolos que conocemos para estos diagramas.

Con estas consideraciones, se propone la siguiente solución:

### **Actores que participan en el problema.**

Empleado (que podrá ser ventas o publicitario), cliente y consultora.

### **Generalizaciones.**

Se observan dos posibles generalizaciones, que en el desarrollo del diagrama podrán ser expresadas en relación a actores o en relación a los casos de uso en los que participan. En la solución propuesta se ha decidido hacer una generalización de actores -- empleado de ventas y empleado publicitario -- y de casos de uso -- lanzar campaña por radio o por televisión --.

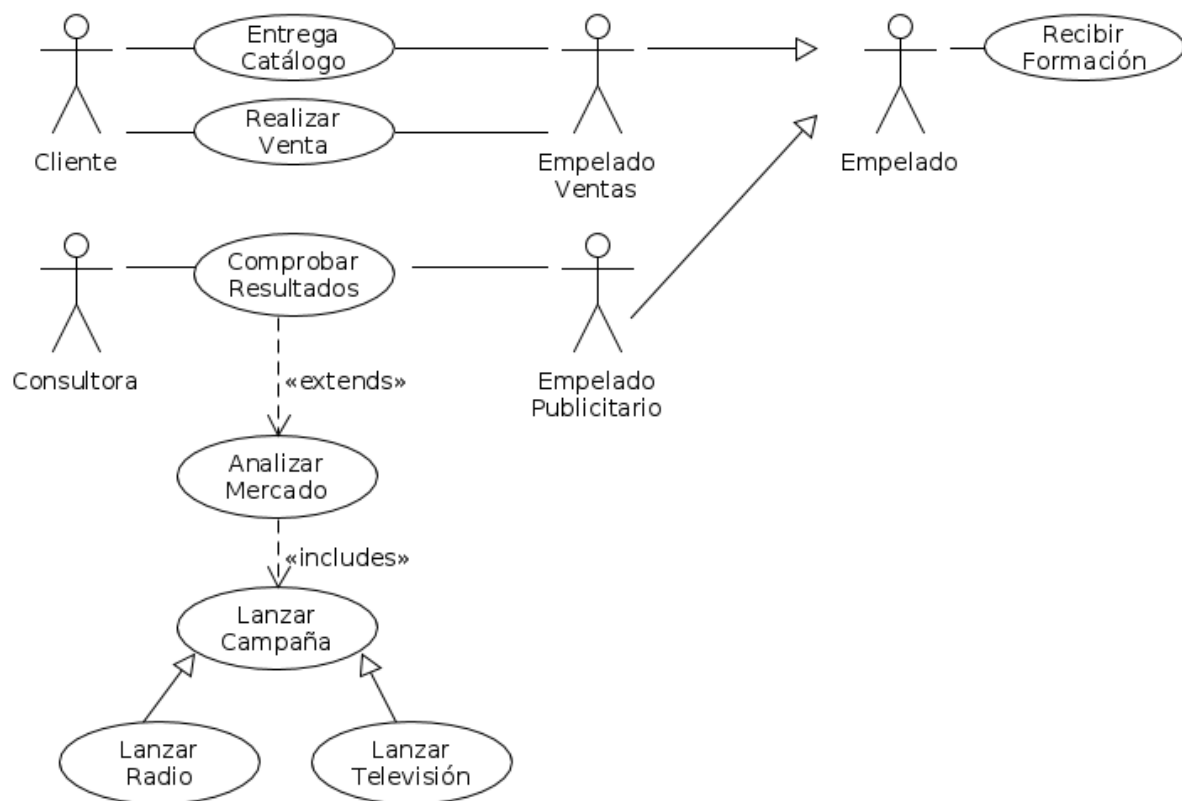
### **Relaciones extends.**

De la ejecución del caso de uso comprobar resultados se puede derivar el caso de uso analizar mercado.

### **Relaciones includes/use.**

Cuando se realiza un análisis de mercado, siempre se lanza una campaña publicitaria.

Quedando el siguiente diagrama.



### **Tabla promocionar Quijote.**

Una vez adaptado el diagrama para obtener el caso de uso promocionar Quijote, podría quedar como sigue:



La tabla resultante sería:



|                        |  |   |
|------------------------|--|---|
| <b>Caso de uso</b>     | Promocionar Quijote.   |   |
| <b>Actores</b>         | Empleado marketing (principal), consultor (secundario).  |   |
| <b>Propósito</b>       | Promocionar la empresa para ganar nuevos clientes.   |   |
| <b>Pre-condiciones</b> | Seleccionar consultora con experiencia en el sector.   |   |
| <b>Flujo normal</b>    | Como flujo normal vamos a suponer que del estudio se genera un análisis de mercado, sería igualmente válida la decisión contraria. | El empleado solicita comprobar los resultados del año en curso a la consultora.                         |
|                        |  | La consultora concluye que se precisa un análisis de mercado de acuerdo al estado actual de la empresa. |
|                        |  | Se lanza una campaña publicitaria en radio.   |
|                        |  | Se lanza una campaña publicitaria en televisión.  |
|                        |  | Fin de flujo.   |



|                            |  |   |
|----------------------------|--|---|
| <b>Flujos alternativos</b> | El análisis de mercado no muestra potenciales clientes.  | El empleado solicita comprobar los resultados del año en curso a la consultora.   |
|                            |  | La consultora concluye que se precisa un análisis de mercado de acuerdo al estado actual de la empresa, pero no se detectan clientes futuros. |
|                            |  | Fin de flujo.   |
|                            | El presupuesto para las campañas publicitarias disponible no es suficiente   | El empleado solicita comprobar los resultados del año en curso a la consultora.   |
|                            |  | La consultora concluye que se precisa un análisis de mercado de acuerdo al estado actual de la empresa.                                       |
|                            |  | Se lanza una campaña publicitaria en radio.   |
|                            |  | Se suspende la campaña publicitaria en televisión por falta de fondos.  |
|                            |  | Fin de flujo.   |
| <b>Post-condiciones</b>    | No se identifican en este ejercicio. Serían aquellas condiciones que se deben cumplir <u>siempre</u> al finalizar el caso de uso sea cual sea el flujo ejecutado.        |   |
| <b>Requisitos trazados</b> | Los que correspondan en el documento de requisitos. Recuerda que el tipo de requisitos que se consideran en los diagramas de casos de uso son siempre de tipo funcional. |   |
| <b>Puntos de inclusión</b> | Ninguno. Este diagrama no dispone de otros casos de uso con los que establecer estas relaciones.   |   |
| <b>Puntos de extensión</b> | Ninguno. Este diagrama no dispone de otros casos de uso con los que establecer estas relaciones  |   |

## Ejercicio resuelto 3 ("ALQUILER DE PISOS Y LOCALES"). Elaboración de un diagrama de casos de usos.

Una empresa de alquiler de pisos y locales desea diseñar un sistema que cumpla los siguientes requisitos:

- Los propietarios previa identificación en el sistema, podrán dar de alta o baja un piso o un local. También podrán modificar los datos de ese piso o local.
- Los futuros inquilinos también deben identificarse antes de poder usar el sistema. Al acceder se les presenta un menú donde pueden elegir la acción a realizar:

1. listar pisos y locales disponibles.
2. solicitar el alquiler de un piso o local.

Para alquilar un local se le pedirá su email y para alquilar un piso su número de teléfono

### **Diagrama de casos de uso**

1. Identificar los actores.
2. Identificar los casos de uso.
3. Implementar con UMLet el diagrama de casos de uso.

### **Resolución:**

#### **1.-Identificar los actores.**

Los actores identificados son : Propietario e Inquilino.

**Propietario**El propietario será el responsable de la gestión de los pisos o locales mediante el alta, la baja y las modificaciones de los mismos. Una vez identificado en el sistema podrá gestionar pisos o locales.

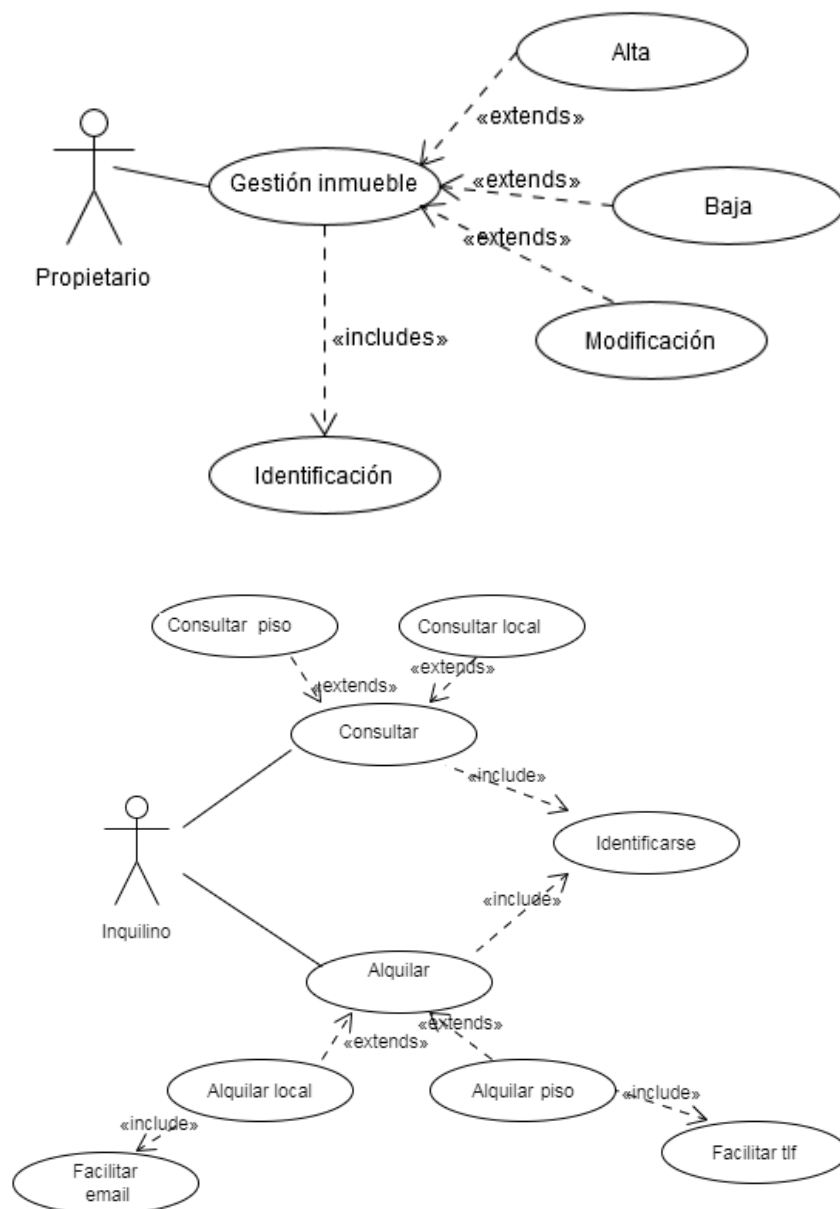
**Inquilino**El inquilino consultará los pisos o locales y también podrá solicitar el alquiler. Debe identificarse en el sistema.

#### **2.-Identificar los casos de uso.**

El propietario podrá dar de alta, baja dar de baja o modificar un inmueble. Si se ofrece un menú podemos usar extends con estos casos de uso y uno llamado gestión inmueble. El inquilino siempre que quiera gestionar un inmueble se identifica.

El inquilino podrá consultar o alquilar. En base a lo que elija podrá ser local o piso. Cuando alquile deberá facilitar teléfono o email según proceda. También debe identificarse para realizar las acciones.

#### **3.-Implementar con UMLet el diagrama de casos de uso.**

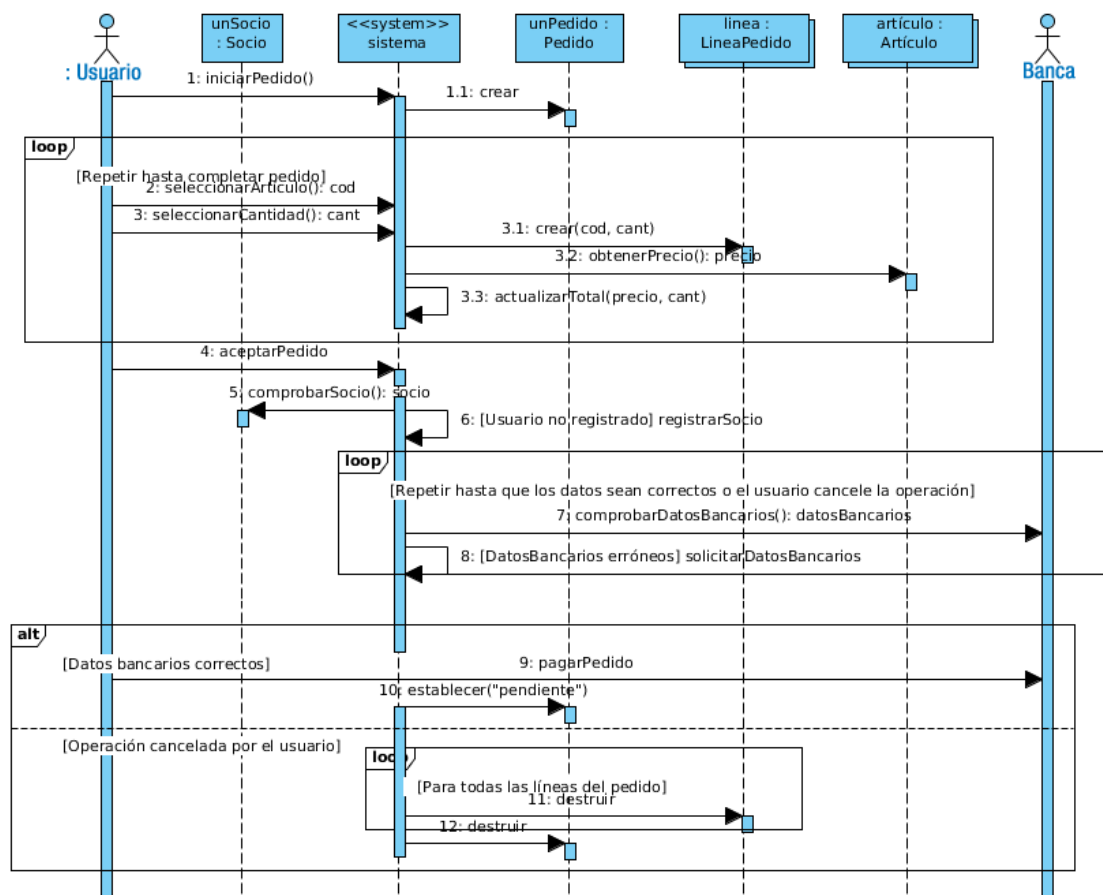


## Ejercicio resuelto 1 ("Generar pedido"). Elaboración de un diagrama de secuencias

Vamos a generar el diagrama de secuencia que representa el caso de uso **"Generar pedido"** del diagrama de casos de uso del ejercicio resuelto 1 "ZAPATERÍA TACÓN DE ORO" (en el punto 2 de "Los diagramas de casos de uso"). En dicho diagrama se establece la secuencia de operaciones que se llevarán a cabo entre los diferentes objetos que intervienen en el caso de uso.

Este es el diagrama ya terminado, en el se han incluido todas las entidades (actores, objetos y sistema) que participan en el diagrama, y se han descrito todas las operaciones, incluidos los casos especiales, como es el registro de usuarios o la gestión de los datos bancarios. También incluye el modelado de

acciones en bucle, como es la selección de artículos y de acciones regidas por condición, como es la posibilidad de cancelar el pedido si hay problemas con la tarjeta de crédito.

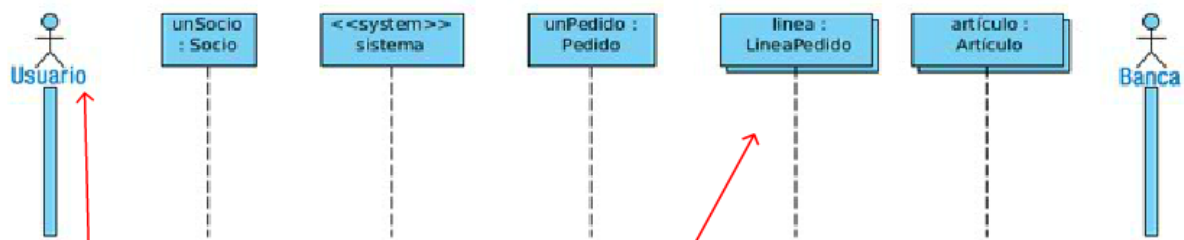


**Debes conocer:**

## Crear el diagrama de secuencia

- El diagrama de secuencia que vamos a tratar es el del caso de uso Hacer pedido, que hemos descrito en el apartado anterior. Si no recuerdas bien cual era la descripción del caso te invito a que la repases en el punto 2.4 de los contenidos de la unidad.
- Este es un diagrama muy completo que incluye bastantes elementos de este diagrama, como bucles o condicionantes, veamos como incluirlos con la herramienta Visual Paradigm, como siempre, debes saber que puedes investigar y utilizar otras herramientas que sirvan para este mismo propósito.

## Incluir entidades



### Actores.

El primer paso es incluir las entidades participantes. Tenemos dos actores, el usuario que inicia el caso de uso y la banca que es un actor secundario que realiza una operación por nosotros como es comprobar que los datos de la tarjeta sean correctos.

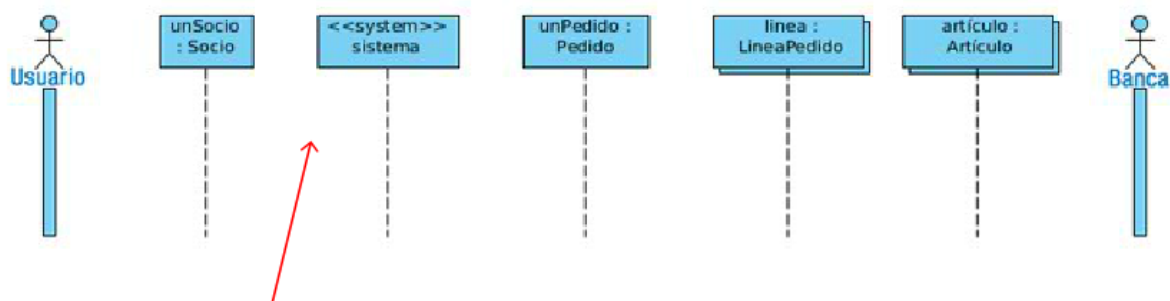
### Objetos.

También necesitamos que intervengan cuatro objetos de nuestro sistema: un socio, un pedido con sus líneas de pedido, y los artículos a pedir.

Para **nombrar** estas entidades en la especificación indicamos su nombre y seleccionamos en el clasificador de base, que muestra la lista de clases del sistema la clase a la que pertenece, así aparece automáticamente su nombre en el formato adecuado.

Indicaremos que las líneas de pedido y los artículos son un conjunto de objetos marcando en la especificación la casilla “**Multi-object**”, en la zona inferior del cuadro de diálogo.

## Incluir entidades



### Sistema.

Por último también interviene el sistema en si mismo, que utilizaremos para las operaciones relacionadas con la interfaz gráfica y las que se lanzan directamente para crear objetos, recuperar información del sistema como los datos de un artículo, o comunicarnos con entidades externas como la banca.

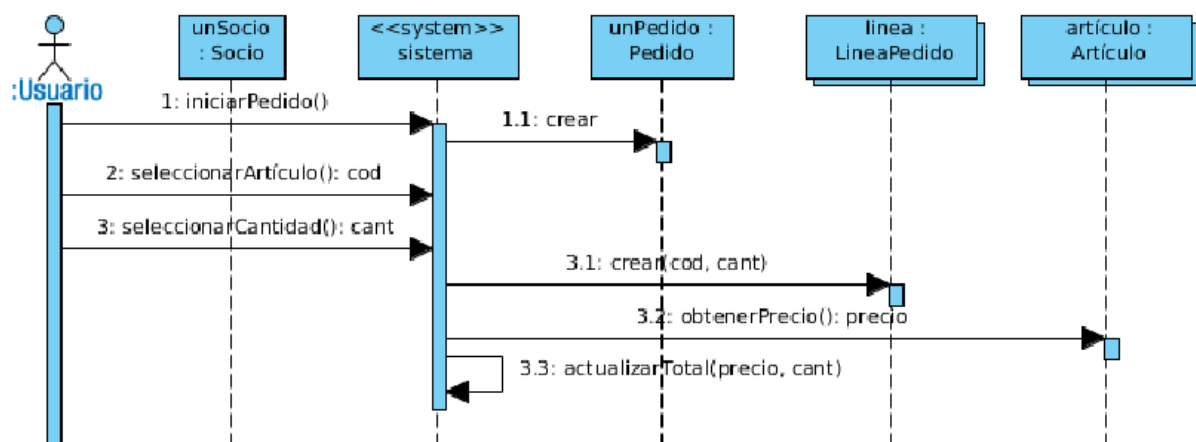
Se añade como una línea de vida más, a la que en su especificación indicamos que su nombre es sistema, y en la pestaña **Estereotipos** seleccionamos **System**.

## Mensajes

Los mensajes pueden devolver un valor, que podemos escribir en la especificación del mismo en la opción return value.

## Mensajes

En esta imagen vemos los mensajes correspondientes al apartado de añadir los datos del pedido. Se selecciona el artículo y la cantidad, se crea una nueva línea de pedido y se recuperan los datos del artículo para obtener su precio y actualizar el total. Puesto que este proceso lo podemos repetir en más de una ocasión lo meteremos en un bucle.

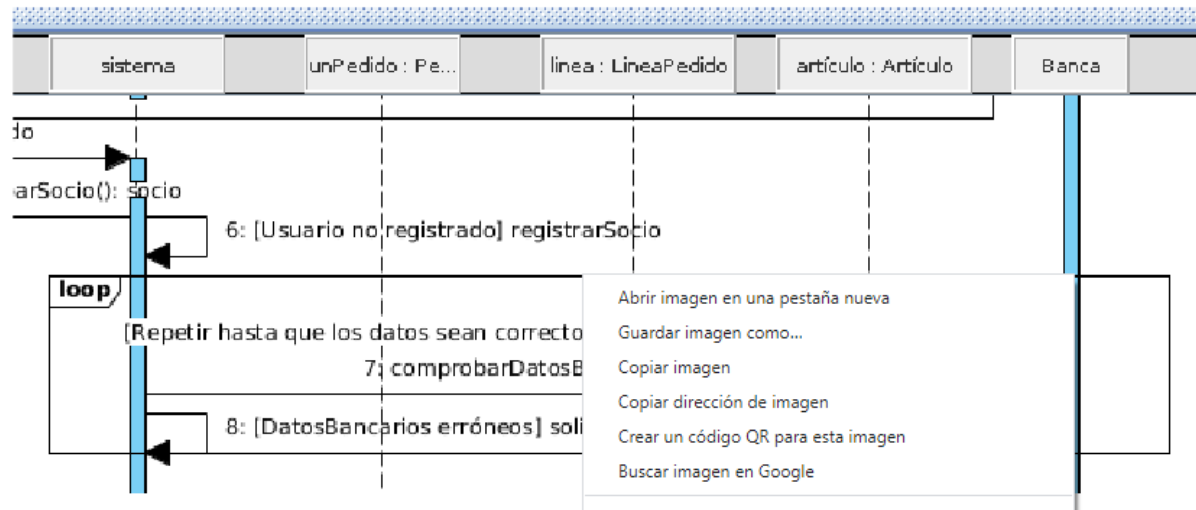


## Mensajes

Para añadir un bucle seleccionamos la opción Loop Combined Fragment.

## Mensajes

Añadiremos condiciones de guarda cuando queramos indicar que un mensaje se enviará sólo si se cumple cierta condición, por ejemplo, sólo registraremos a un usuario si no es socio ya.



## Mensajes

También se pueden incluir condiciones más elaboradas que implique una bifurcación en el flujo de eventos, como es el caso de la comprobación de la tarjeta, si todo marcha bien se finalizará la creación del pedido, si no, el usuario puede cancelar la operación y terminar sin guardar nada.

## Ejercicio resuelto 2 ("Estadio"). Elaboración de un diagrama de secuencia

Se pretende desarrollar un programa que dé respuesta a las necesidades de un estadio de fútbol en uno de sus partidos. Tras varias entrevistas con el responsable, se ha llegado al acuerdo de que los requisitos funcionales se pueden recoger en el caso de uso Gestión del estadio donde hay que considerar actividades como:

- **Control de puntos de acceso.** Se dispone de 2 puertas, a las que el responsable del estadio dará orden de apertura. Cada puerta hace un test interno y devuelve al responsable Ok o Ko. Si una puerta no abre, queda inactiva para el resto del partido. El proceso de apertura se realiza de forma simultánea en ambas puertas. Al finalizar el partido, el responsable del estadio dará orden de cierre de las puertas que se han usado durante el partido.

- **Control de acceso de aficionados.** Si la puerta está disponible, el responsable da orden de iniciar el acceso al estadio. Mientras haya aficionados, el operario de la puerta valida la entrada, si es correcta le da acceso al campo, en caso contrario avisa al responsable del estadio que ha habido un intento de acceso con entrada falsa.

**Nota:**

- Considera en este diagrama que la puerta uno está averiada y la puerta dos operativa.

**Se pide:**

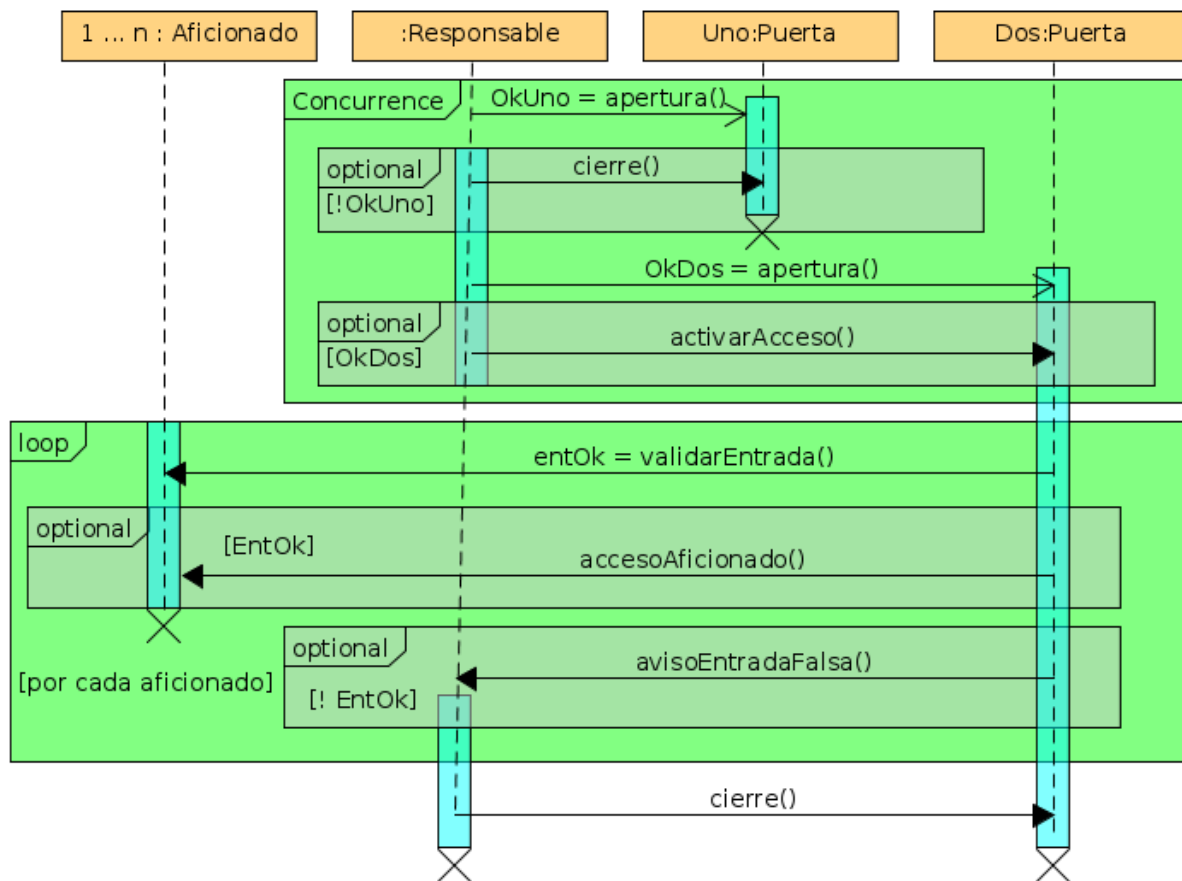
Desarrollar el diagrama de secuencia resultante del caso de uso planteado.

**Atiende a aspectos como:**

- Identificación de los objetos involucrados en el diagrama de secuencia.
- Que el diagrama recoja la secuencia de mensajes intercambiados, tomando en consideración las funcionalidades descritas en el enunciado.
- Uso correcto de la notación vista para este tipo de diagramas.
- Que los diagramas generados sean visualmente útiles. Un diagrama de secuencia debe dar una idea clara/rápida de la secuencia de acciones que se derivan de la ejecución del caso de uso que representa.

Resolución:





## Ejercicio resuelto 3 ("ROPERO"). Elaboración de un diagrama de secuencia.

Se pretende crear un programa para la gestión de ciertas funciones de una discoteca. En particular, el depósito de los abrigos en el ropero.

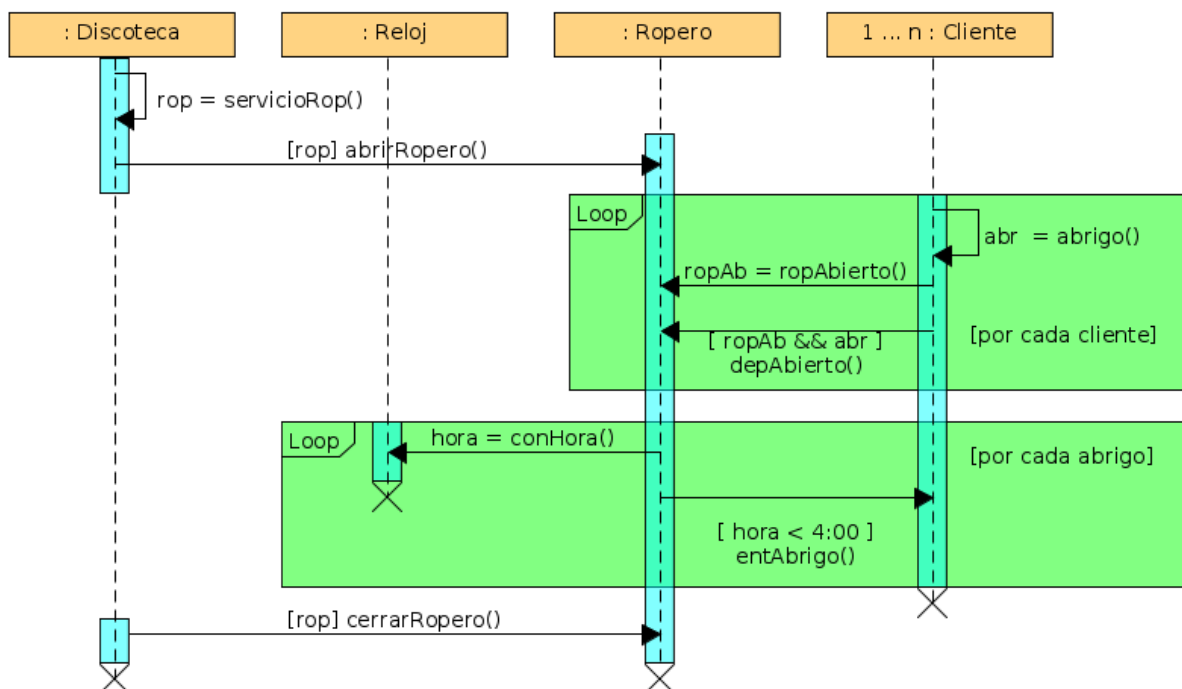
Algunas consideraciones:

- El responsable de la discoteca cada día decide si abre el vestíbulo, y es el encargado de su apertura y cierre.
- El propio ropero informa a los clientes si proporciona o no servicio de guardarropa.
- Si está abierto, a primera hora recoge los abrigos de los clientes que tengan abrigo.
- Al finalizar la jornada, mientras haya abrigos en el ropero se devuelven siempre que sea antes de las 4 de la mañana.

- Se pide desarrollar el diagrama de secuencia resultante del caso de uso planteado.

Considera aspectos como:

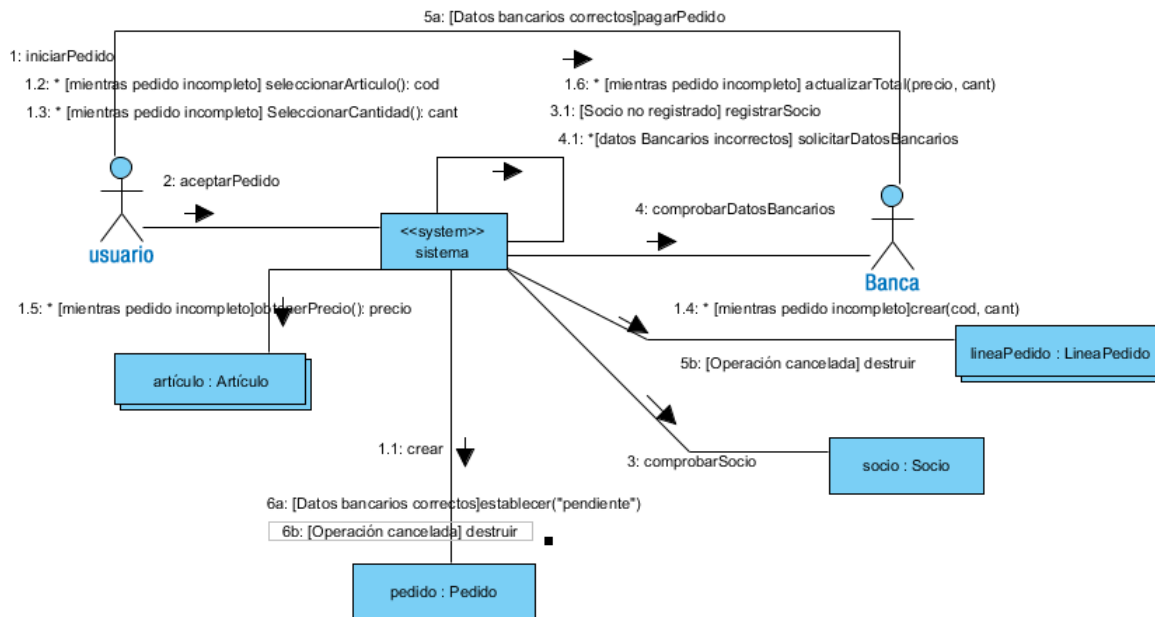
- Identificación de los objetos involucrados.
- Que el diagrama recoja la secuencia de mensajes intercambiados, tomando en consideración las funcionalidades descritas en el enunciado.
- Uso correcto de la notación vista para este tipo de diagramas.



## Ejemplo de un diagrama de colaboración

A continuación se muestra un diagrama de colaboración de ejemplo.

Este es el diagrama de colaboración que representa el caso de uso "**Generar pedido**" del diagrama de casos de uso del ejercicio resuelto 1 "ZAPATERÍA TACÓN DE ORO" (en el punto 2 de "Los diagramas de casos de uso"). Se ha creado siguiendo el diagrama de secuencia, por lo que no te debe ser muy difícil seguirlo, de hecho algunas aplicaciones para la creación de estos diagramas permiten la obtención de uno a partir de otro. Debes tener en cuenta que la aplicación modifica un poco la signatura de los mensajes, el valor devuelto se representa al final precedido de dos puntos.



Los aspectos más destacados son los siguientes:

- Las actividades que se repiten o pueden repetirse se marcan con un asterisco y su condición.
- Las condiciones de guarda se escriben en el mismo nombre del mensaje.
- El flujo alternativo de eventos según si el usuario cancela el pedido o no, obliga a modificar los números de secuencia de los mensajes 5 y 6, pasando a tener los mensajes 5a y 6a y 5b y 6b, según la condición. Puedes modificar el número de secuencia de los mensajes abriendo la especificación del diagrama, y seleccionando la pestaña Mensajes, donde puedes editar los números de secuencia haciendo doble clic sobre ellos.
- Al objeto "sistema" se le ha asignado el estereotipo system.

## Ejercicio resuelto 1 ("Generar pedido"). Elaboración de un diagrama de estados.

Para ejemplificar la creación de un diagrama de estados vamos a ver el que representa el caso de uso "**Generar pedido**" del diagrama de casos de uso del ejercicio resuelto 1 "ZAPATERÍA TACÓN DE ORO" (en el punto 2 de "Los diagramas de casos de uso") que cumple con las condiciones que hemos visto al principio, tiene un comportamiento significativo en tiempo real, ya que su

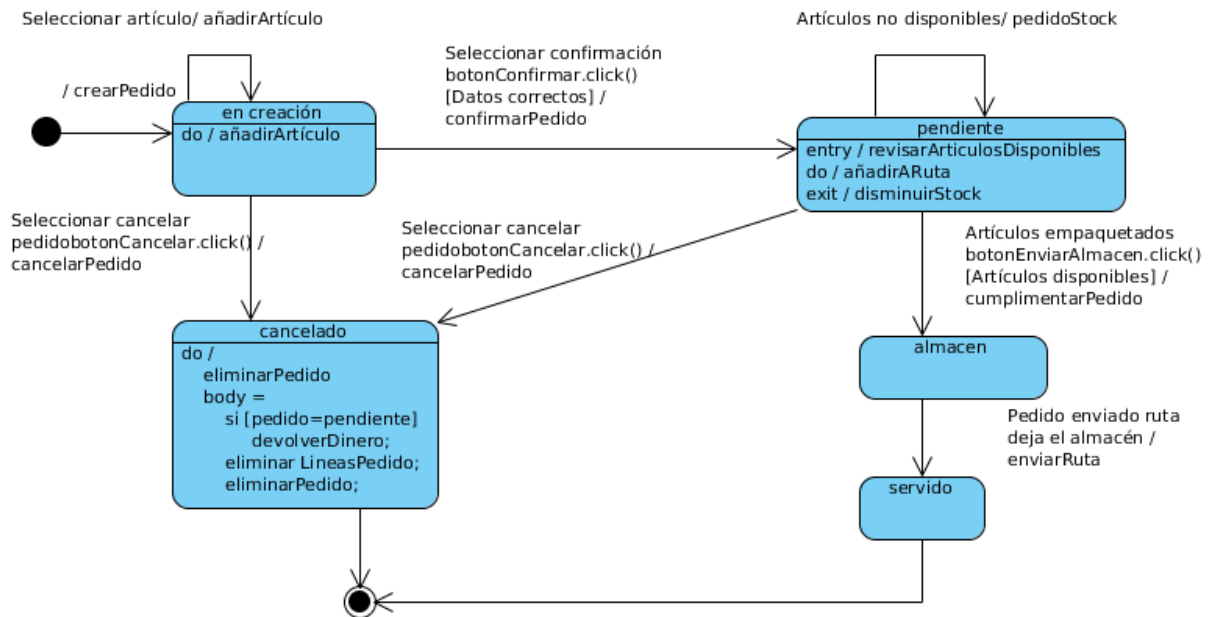
situación tanto física, como el sistema, va evolucionando conforme pasa el tiempo, y participa en varios casos de uso (como Hacer pedido y Cumplimentar pedido).

Los diferentes estados en los que puede estar un pedido son:

- **En creación:** es cuando se están seleccionando los productos que formará el pedido.
- **Pendiente:** está en este estado desde que se confirma el pedido hasta que se selecciona para preparar su envío.
- **En almacén:** está en este estado cuando es elaborado el paquete y se ha asignado a una ruta, hasta que se envía a través de la ruta que le corresponde.
- **Servido:** Cuando el pedido es enviado. En este caso se envía una señal física desde el almacén cuando el transporte abandona el almacén.
- **Cancelado:** puede llegarse a esta situación por dos motivos, o bien se cancela mientras se está haciendo por problemas con la tarjeta de crédito, o bien porque, una vez pendiente de su gestión el usuario decide cancelarlo, la diferencia fundamental entre ambos es que en el segundo caso hay que devolver el importe pagado por el pedido al socio que lo ha comprado.

Las transiciones entre estados se producen por llamadas a procedimientos en todos los casos, no intervienen cambios de estado o el tiempo, ni señales.

El diagrama quedaría de la siguiente manera:



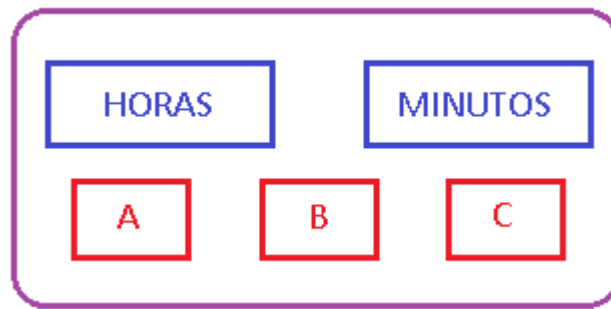
En las **transiciones** se ha incluido el nombre de la transición, el evento que la dispara (normalmente hacer clic en algún botón de la interfaz), si existe condición de guarda se pone entre corchetes y la acción a realizar para llegar al siguiente estado junto al símbolo /. En todos los casos el evento de disparo es de tipo llamada (incluye la llamada a una función o pulsar un botón de la interfaz), salvo en el caso de pedido enviado que se controla por una señal que se envía cuando el transporte abandona el almacén.

A los **estados** se les ha añadido la acción a realizar, apartado do/ y en algunos casos la acción de entrada, por ejemplo en el caso del estado pendiente, se debe revisar que los artículos a enviar tengan disponibilidad y la de salida, en el ejemplo disminuir el stock.

**Nota:** para incluir las condiciones de guarda en el diagrama debes rellenar el apartado "Guard" de la especificación, si necesitas añadir alguna acción puedes hacerlo rellenando el apartado "Effect". Los eventos de disparo.

## Ejercicio resuelto 2 ("RELOJ"). Elaboración de un diagrama de estados.

La siguiente figura muestra un **reloj digital** cuyo comportamiento se describe a continuación:



El reloj se enciende y está visualizando las horas y minutos.

Funciones de reloj:

Pulsado de A durante tres segundos: parpadea la hora. Para evitar cambios de hora involuntarios, si el tiempo de pulsado es inferior a tres segundos no se activa la función.

El botón B no funciona, si no se ha pulsado antes el botón A durante 3 segundos.

De tal forma que si el reloj está en el estado en el que la hora está parpadeando:

1. Si se pulsa el botón B incrementa la hora en una unidad.
2. Si se pulsa el botón A, pasará al estado de poder cambiar los minutos. Los minutos parpadearán. No se precisa mantener pulsado el botón porque se entiende que se está modificando la hora de forma voluntaria.

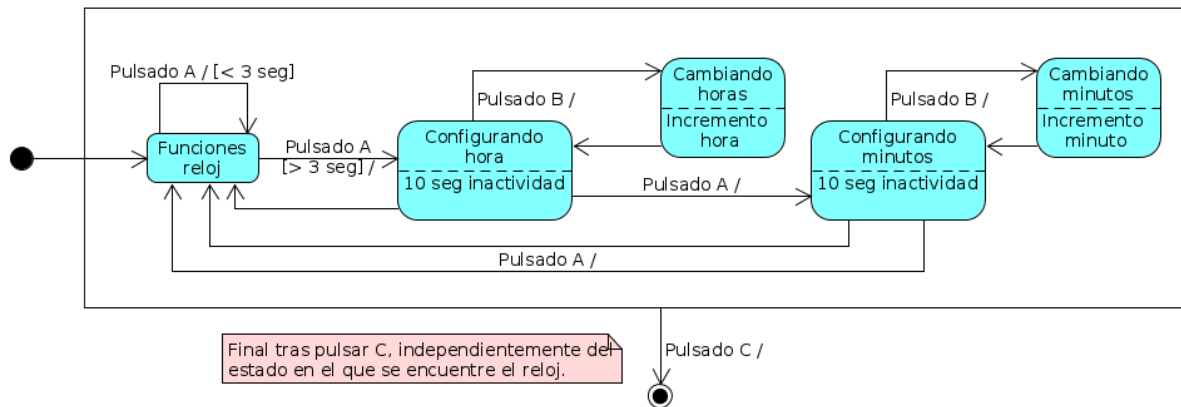
Si está el reloj en el estado de poder cambiar los minutos:

1. Pulsado del botón B: cada pulsación del botón B incrementa los minutos en una unidad
2. Si pulsamos el botón A, finaliza el modo configuración y vuelve a mostrar la hora.

Pulsado de C: apaga del reloj sin tener en consideración el estado en el que se encuentre.

Cuando el reloj está en modo configuración de horas o minutos, tras 10 segundos de inactividad abandona la configuración y pasa a modo funciones de reloj.

**Resolución:**



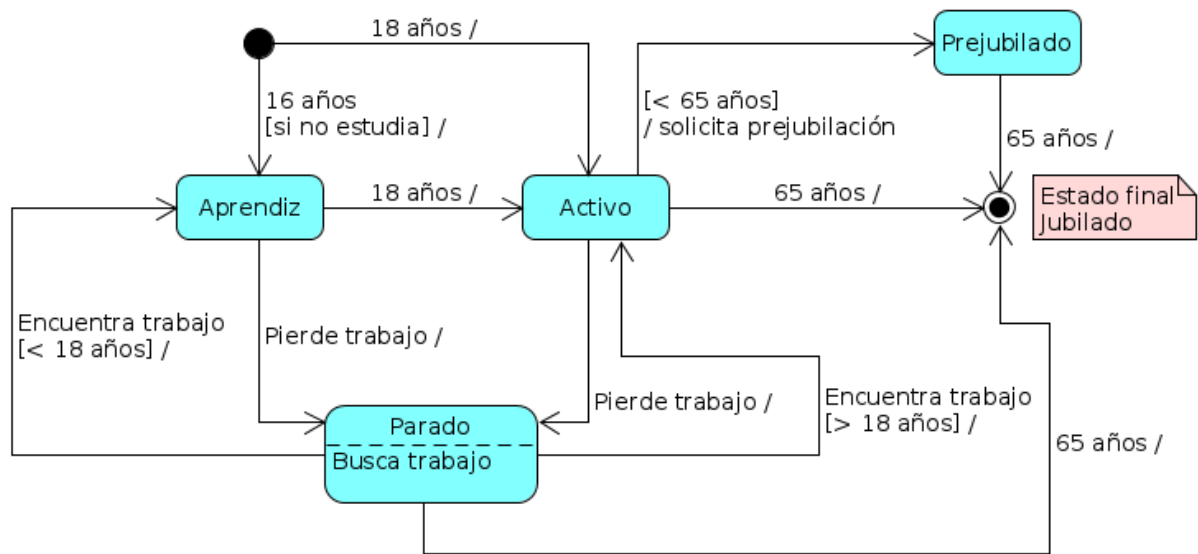
- Las transiciones sin eventos, aunque pueden tener condiciones, se producen al finalizar la actividad del estado.
- Es posible salir de un estado tanto por un evento externo, como por el fin de la actividad del estado.
- El evento pulsado C genera el apagado del reloj con independencia del estado en el que se encuentre.

## Ejercicio resuelto 3 ("VIDA LABORAL"). Elaboración de un diagrama de estados

Crea un diagrama de estados que muestre la **evolución de un empleado** a lo largo de su vida laboral.

Se considera que el proceso transcurre entre los 16 y 65 años de edad y se plantean los siguientes **estados**:

- **Preempleado.** Anterior a los 16 años. Se considera el estado inicial.
- **Aprendiz.** Es el periodo comprendido entre los 16 y 18 años para aquellas personas que han decidido no continuar sus estudios.
- **Activo.** El trabajador se encuentra en activo y con contrato en vigor.
- **Parado.** El trabajador ha perdido el empleo, su tarea principal es la búsqueda de un nuevo trabajo.
- **Prejubilado.** El trabajador solicita dejar de estar activo, pero no ha alcanzado la edad de 65 años. Desde el estado de parado no se considera la opción de solicitar la prejubilación.
- **Jubilado.** El trabajador ha cumplido los 65 años y pasa a disfrutar de un merecido descanso. Se considera el estafo final.



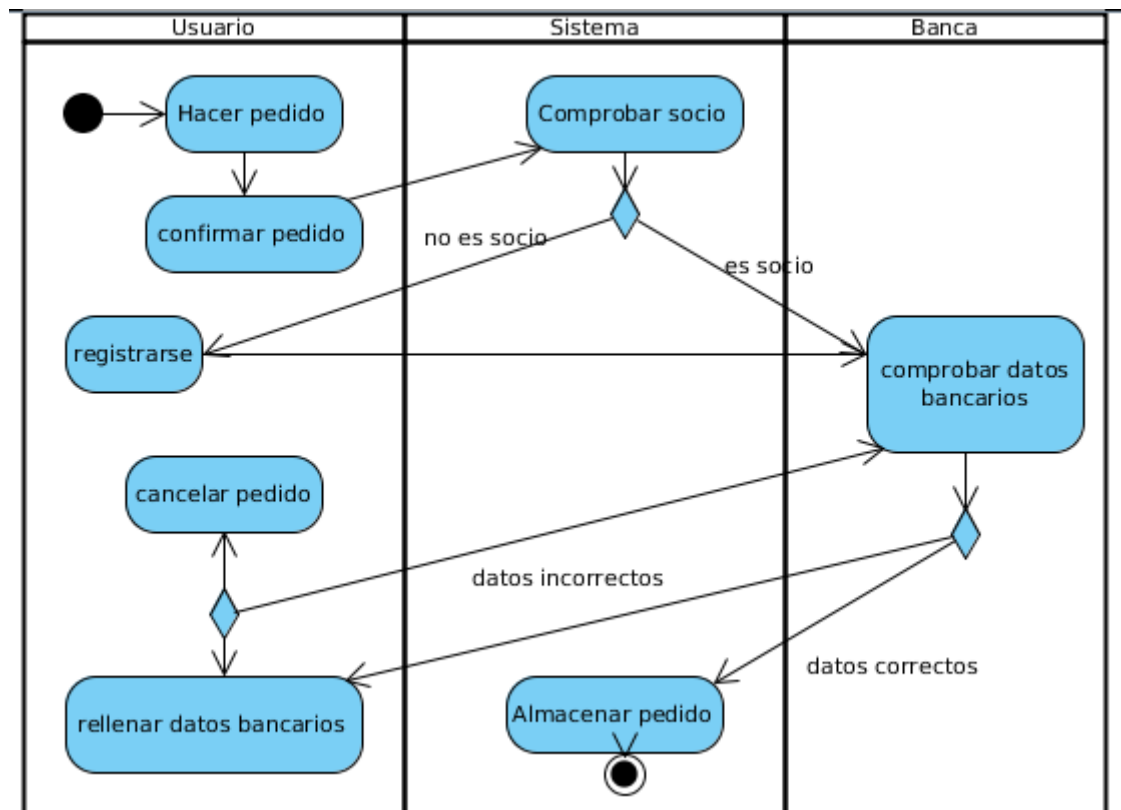
## Ejemplo de un diagrama de actividad

El siguiente diagrama de actividad representa el caso de uso "Generar pedido" del diagrama de casos de uso del ejercicio resuelto 1 "ZAPATERÍA TACÓN DE ORO" (en el punto 2 de "Los diagramas de casos de uso"), en el aparecen los elementos que hemos visto en las secciones anteriores.

- En las bifurcaciones se ha añadido la condición que indica si se pasa a una acción o a otra.
- Las acciones Seleccionar artículo y Seleccionar cantidad se han considerado concurrentes.







ota: Para añadir las calles en Visual Paradigm se utiliza la herramienta del panel "Vertical Swimlane".