

# CONTORNOS DE DESENVOLVEMENTO

## TEMA 6: ELABORACIÓN DE DIAGRAMAS DE COMPORTAMIENTO



## Índice

---

<b>1.</b>	<b>Diagramas de casos de uso.....</b>	<b>3</b>
1.1	Os diagramas de comportamento de UML .....	3
1.2	Os diagramas de casos de uso .....	7
<b>2.</b>	<b>Diagramas de interacción .....</b>	<b>16</b>
2.1	Os diagramas de interacción de UML.....	16
2.2	Diagramas de secuencia .....	16
2.3	Diagramas de comunicación.....	21
<b>3.</b>	<b>Diagramas de estados.....</b>	<b>28</b>
3.1	Propósito e función dos diagramas de estados .....	28
3.2	Elementos .....	28
<b>4.</b>	<b>Diagramas de actividades.....</b>	<b>33</b>
4.1	Propósito e función dos diagramas de actividades. ....	33
4.2	Elementos .....	33
4.3	Recomendacións e exemplos.....	36

### Material complementario e vídeos:

<https://wirtzIDE.blogspot.com/>



Fernando Rodríguez Diéguez  
rdf@fernandowirtz.com  
Versión 2021-06-08

Traballo derivado por: Fernando Rodríguez Diéguez. rdf@fernandowirtz.com

Con Licenza Creative Commons BY-NC-SA (recoñecemento - non comercial - compartir igual) a partir dos documentos orixinais:

© Xunta de Galicia. Consellería de Cultura, Educación e Ordenación Universitaria.

Autores: María del Carmen Fernández Lameiro, Máximo Fernández López Andrés del Río Rodríguez

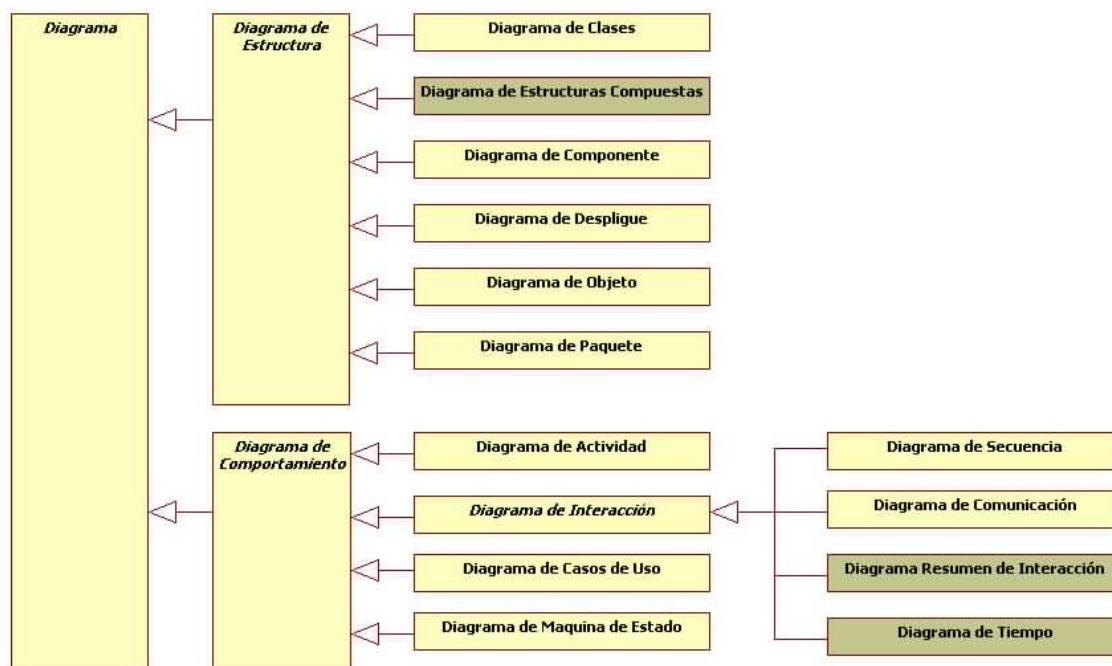
Licenza Creative Commons BY-NC-SA (recoñecemento - non comercial - compartir igual).

Para ver unha copia desta licenza, visitar a ligazón <http://creativecommons.org/licenses/by-nc-sa/3.0/es/>.

# 1. Diagramas de casos de uso.

## 1.1 Os diagramas de comportamento de UML

Tal e como dixemos na unidade anterior, o conxunto de diagramas de UML subdivídese habitualmente -figura<sup>1</sup>- en dous grandes subconxuntos: *diagramas estruturais (ou de estrutura)* e



*diagramas de comportamento.*

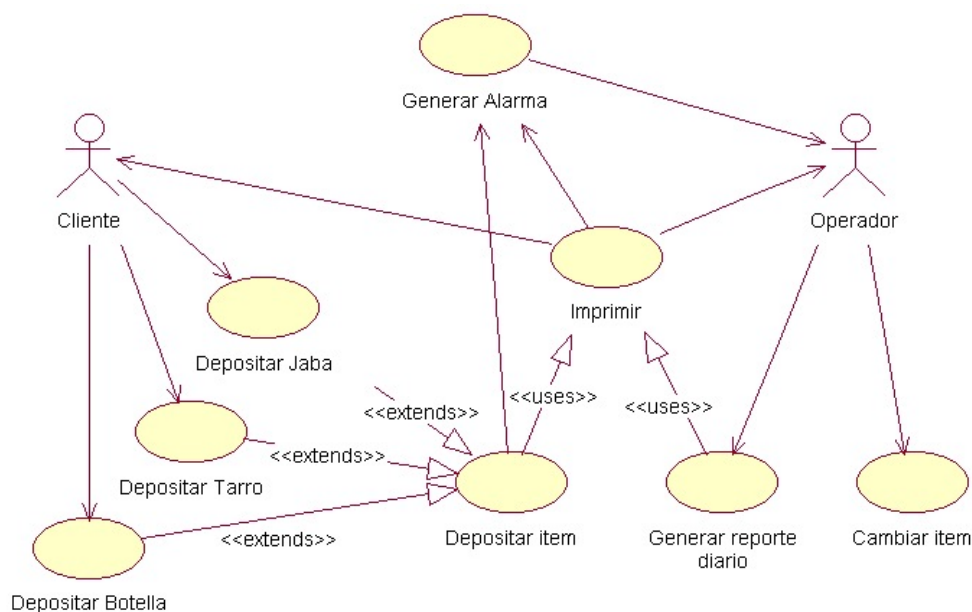
(Os diagramas sinalados nunha cor distinta foron incorporados na versión 2 de UML polo que resultan ser dos menos coñecidos e empregados.)

Xa comentamos que os diagramas estruturais *presentan elementos estáticos do modelo*, tales como clases, paquetes ou compoñentes; en tanto que os diagramas de comportamento *mostran a conduta en tempo de execución do sistema*, tanto visto como un todo como das instancias ou obxectos que o integran.

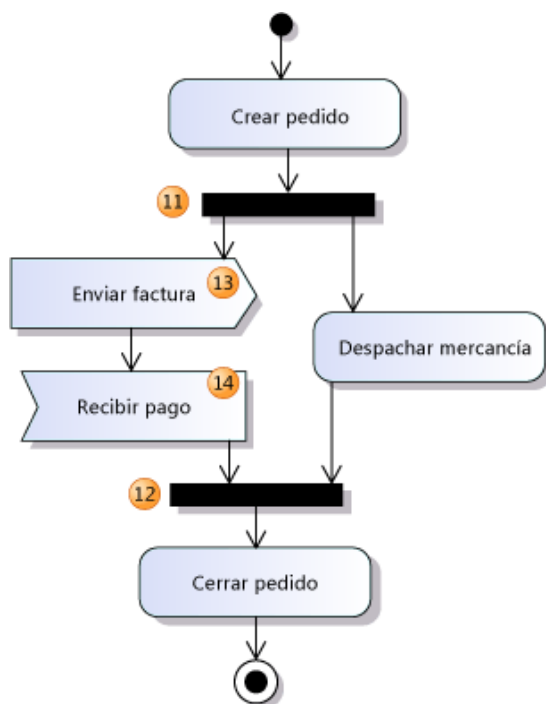
Deste xeito podemos dicir que os diagramas de comportamento permítennos visualizar, especificar, construír e documentar os aspectos dinámicos dun sistema tales como o fluxo de mensaxes ó longo do tempo, os estados polos que pode pasar un obxecto ó longo do seu ciclo de vida ou a interacción dun usuario cun sistema. En maior ou menor medida e dependendo do tipo de sistema que esteamos modelizando, todos estes aspectos deben de quedar reflectidos e definidos na fase de deseño do noso proxecto. Para elo poderemos usar os seguintes diagramas UML:

<sup>1</sup> <http://synergix.wordpress.com/2008/07/20/tipos-de-diagramas-en-uml/>

**1. Diagramas de casos de uso:** mostra as relacións entre os actores e o suxeito (sistema), e os casos de uso:<sup>2</sup>



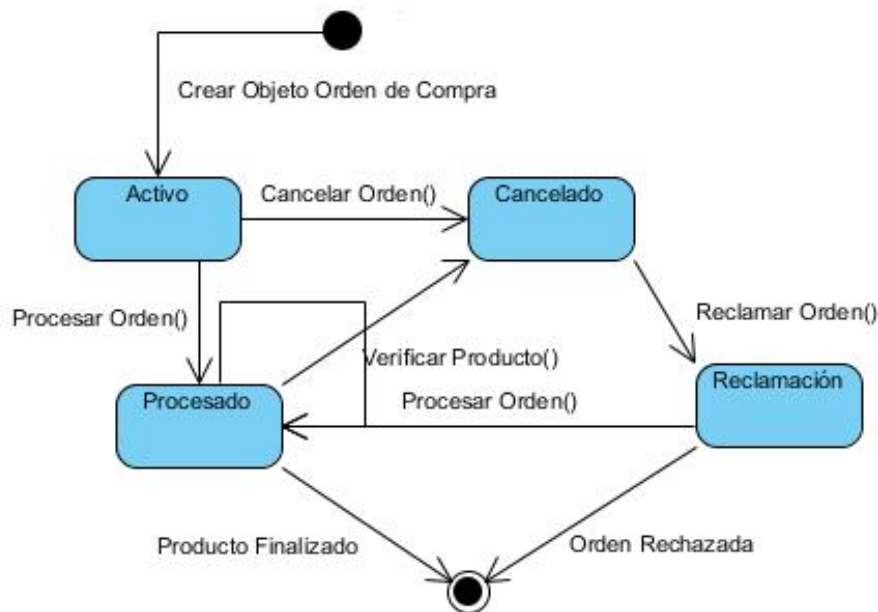
**2. Diagramas de actividade:** representa os procesos de negocios de alto nivel, incluídos o fluxo de datos. Tamén pode utilizarse para modelar lóxica complexa e/ou paralela dentro dun sistema:<sup>3</sup>



<sup>2</sup> <http://users.dcc.uchile.cl/~psalinas/uml/casosuso.html>

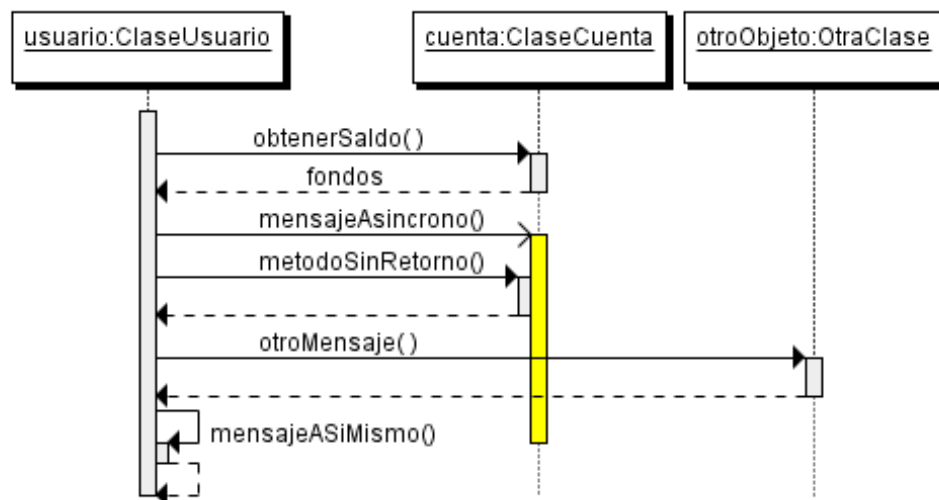
<sup>3</sup> <http://msdn.microsoft.com/es-es/library/dd409360.aspx>

**3. Diagramas de máquinas de estado:** ilustra como un elemento, moitas veces unha clase, pódese mover entre estados que clasifican o seu comportamento, de acordo con disparadores de transicións, gardas de restricións e outros aspectos:<sup>4</sup>



**4. Diagramas de interacción,** que se subdividen en:

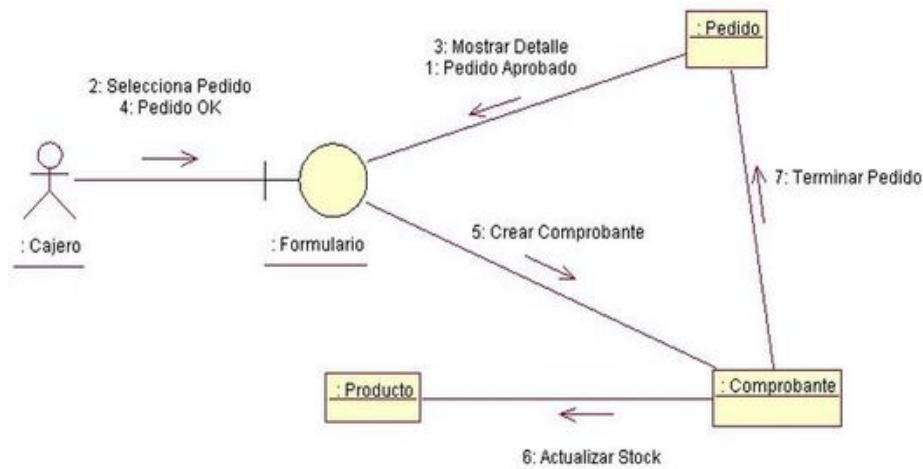
**4.1. Diagramas de secuencia:** representa unha interacción, poñendo o foco na secuencia das mensaxes que se intercambian, xunto coas súas correspondentes ocorrencias de eventos nas Liñas de Vida:<sup>5</sup>



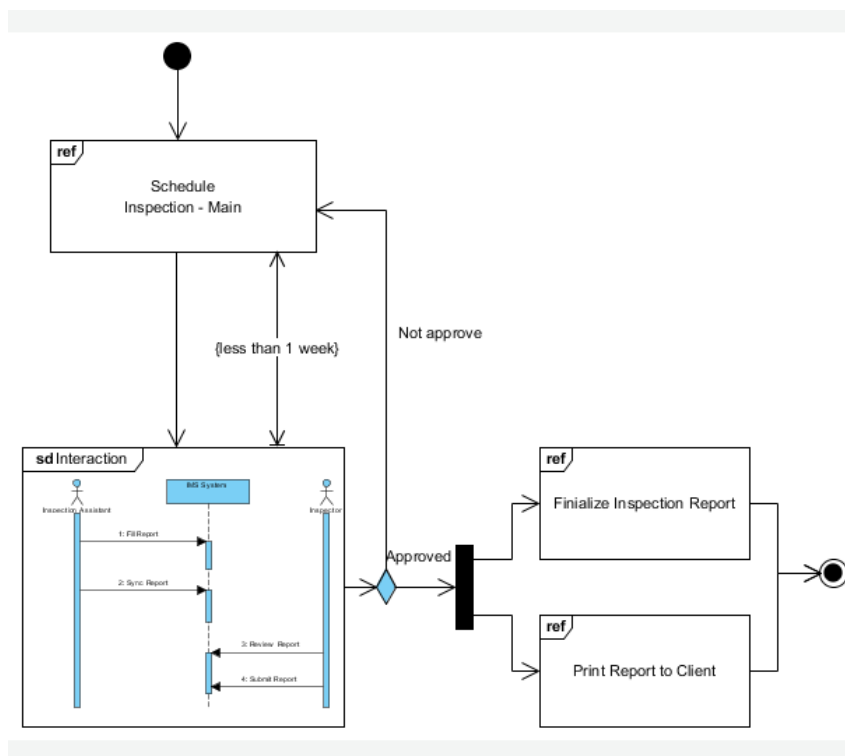
<sup>4</sup> <http://www.elclubdelprogramador.com/2012/04/03/uml-diagrama-de-estados/>

<sup>5</sup> <https://ingenieriasoftware2011.wordpress.com/2011/07/15/diagramas-de-secuencia/>

**4.2 Diagramas de comunicación:** enfoca a interacción entre liñas de vida, onde é central a arquitectura da estrutura interna e como ela se corresponde coa pasaxe de mensaxes. A secuencia das mensaxes dáse a través dun esquema numerado:<sup>6</sup>



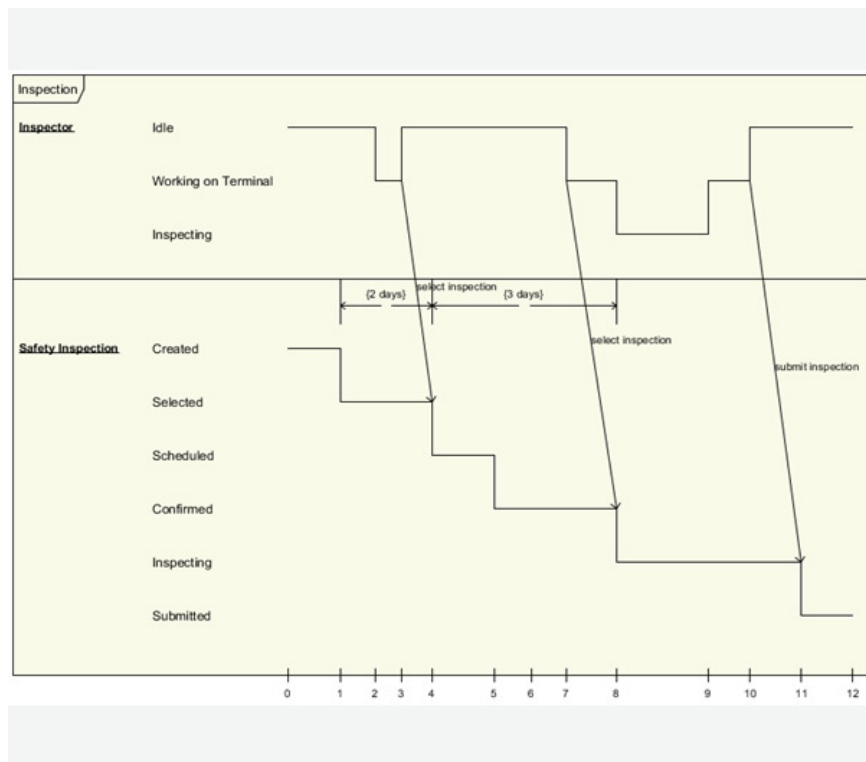
**4.3. Diagramas de resumen de interacción:** enfocanse á revisión do fluxo de control, onde os nodos son interaccións ou ocorrencias de interaccións:<sup>7</sup>



**4.4. Diagramas de tempo:** O propósito primario do diagrama de tempos é mostrar os cambios no estado ou a condición dunha liña de vida ao longo do tempo lineal.:<sup>8</sup>

<sup>6</sup> <http://diagramasumlrickolmososati102.weebly.com/diagramas-de-colaboracioacuten.html>

<sup>7</sup> <http://www.visual-paradigm.com/VPGallery/diagrams/InteractionOverviewDiagram.html>



Nas seguintes actividades iremos coñecendo algúns dos principais diagramas desta lista comezando primeiramente cos *diagramas de casos de uso*.

## 1.2 Os diagramas de casos de uso

### Introdución

Os diagramas de casos de uso permítennos modelar e documentar os requisitos funcionais do sistema dende o punto de vista do usuario. Comprender tal punto de vista non sempre é doado e non obstante é clave para xerar sistemas que sexan tanto útiles como funcionais; isto é, que cumpran cos requisitos e que sexa doado traballar con eles.

A principal vantaxe dos diagramas de casos de uso é a facilidade para elaboralos e interpretalos; isto fai que sexan especialmente útiles na comunicación entre o analista e o cliente tanto para establecer os requisitos iniciais do sistema como para comprobar que unha vez acabado o sistema cumpre cos requisitos iniciais.

Un diagrama de casos de uso debe dicir QUE comportamento se espera do sistema de maneira entendible polo cliente e sen utilizar tecnicismos pero non COMO se levará a cabo ese comportamento. Para a descrición do COMO empregaremos outros diagramas como os de secuencia.

### Elementos

Os diagramas de casos de uso están compostos por actores (participantes no sistema), casos de

<sup>8</sup> <http://www.visual-paradigm.com/VPGallery/diagrams/TimingDiagram.html>

uso (diferentes usos do sistema) e relacións entre eles.

- **Actor.** Un actor representa un rol ou papel que xoga un usuario ou outro sistema (un actor non ten que ser necesariamente unha persoa) no sistema a modelar. É importante destacar o uso da palabra rol, pois con isto especificase que un Actor non necesariamente representa a unha persoa en particular, senón máis ben o labor que realiza fronte ao sistema.

O nome dos actores son substantivos relacionados coas regras de negocio e non co cargo ou posición no sistema; represéntanse cun “monicreque” co nome debaixo.



Algunhas características:

- Un actor está fóra dos límites do sistema.
  - Un mesmo actor pode intervenir en varios casos de uso.
  - Nun caso de uso poden intervenir varios actores.
  - Normalmente os actores inician un caso de uso.
  - Soen colocarse os actores principais na parte superior do diagrama.
- **Caso de uso.** Un caso de uso describe unha tarefa que se vai realizar no sistema despois dunha orde dalgún axente externo como un actor ou outro caso de uso.  
Debe ter un nome que empezará por un verbo e represéntase mediante unha elipse co nome dentro.



Algunhas características:

- Os casos de uso soen representarse de forma descendente situando os máis importantes na parte superior.
  - O límite do sistema define o ámbito no que se producen os casos de uso.
  - A acción correspondente ao caso de uso non pode ser nin excesivamente xenérica nin demasiado específica.
- **Relacións.** Unha relación indica a actividade ou fluxo de información. Pode ser de varios tipos:
    - *Asociación.* Indica a invocación dende un actor ou un caso de uso a outro caso de uso.

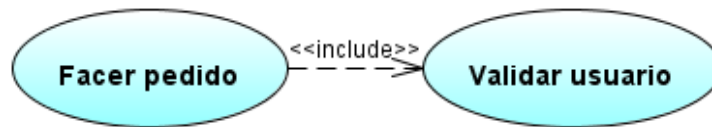


Represéntase mediante unha liña que os une.

- *Inclusión.* Dáse entre dous casos de uso e indica que un ten a funcionalidade doutro como parte integrante súa e sen el non podería funcionar ben. Represéntase cunha

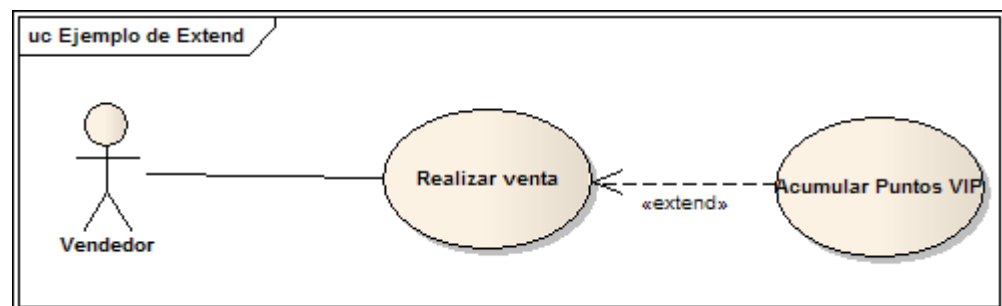


frecha descontinua coa punta da frecha no caso de uso que se inclúe e co estereotipo <<include>>.



O caso de uso incluído poderá incluírse noutros casos de uso e así evítase duplicidade de interaccións en distintos casos de uso.

- *Extensión*. Dáse entre dous casos de uso e indica que un caso de uso amplía a funcionalidade doutro. Representase cunha frecha descontinua coa punta da frecha no caso de uso que estende e co estereotipo <<extend>>. Sirve para modelar partes opcionais do sistema, variacións do comportamento normal dun caso de uso, subfluxos que só se executan en determinadas circunstancias ou varios fluxos que se poden inserir nun punto.



Unha das diferenzas básicas é que no caso do <<extend>> hai situacións en que o caso de uso de extensión non é indispensable que ocorra, e cando o fai ofrece un valor extra (estende) ao obxectivo orixinal do caso de uso base. En cambio no <<include>> é necesario que ocorra o caso incluído, tan só para satisfacer o obxectivo do caso de uso base. Exemplo: Podes "Realizar Venda" sen "Acumular Puntos de Cliente VIP", cando non es un cliente VIP. Pero, se es un cliente VIP si acumularás puntos. Polo tanto, "Acumular Puntos" é unha extensión de "Realizar Venda" e só se executa para certo tipo de vendas, non para todas.

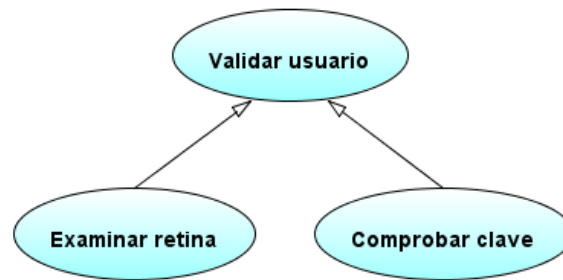
Nalgúns casos pode ser útil especificar os chamados *puntos de extensión* que aclaran en que circunstancias ou puntos un caso será estendido por outro.

A aplicación de cada extensión decídese durante o desenvolvemento dun escenario.

- *Xeneralización*. Dáse entre casos de uso e incluso entre actores e indica que un caso de uso fillo herda a especificación do caso de uso pai e especialízase. Non hai regras específicas en canto ás modificacións ou ampliacións posibles nas especificacións herdadas pero é intuitivo pensar que poden ser modificadas e/ou ampliadas. Representase cunha liña continua coa punta da frecha no caso de uso pai.

Pode pensarse que a xeneralización é unha relación parecida á extensión pero na

xeneralización poderíase substituír ao pai polo fillo e o sistema non se vería afectado en canto ás regras de negocio.



- *Escenario.* Un **escenario** é unha instancia concreta dun caso de uso. Un caso de uso pode ter varios escenarios.

### Recomendacións para a elaboración dos diagramas

Os pasos a seguir poden ser:

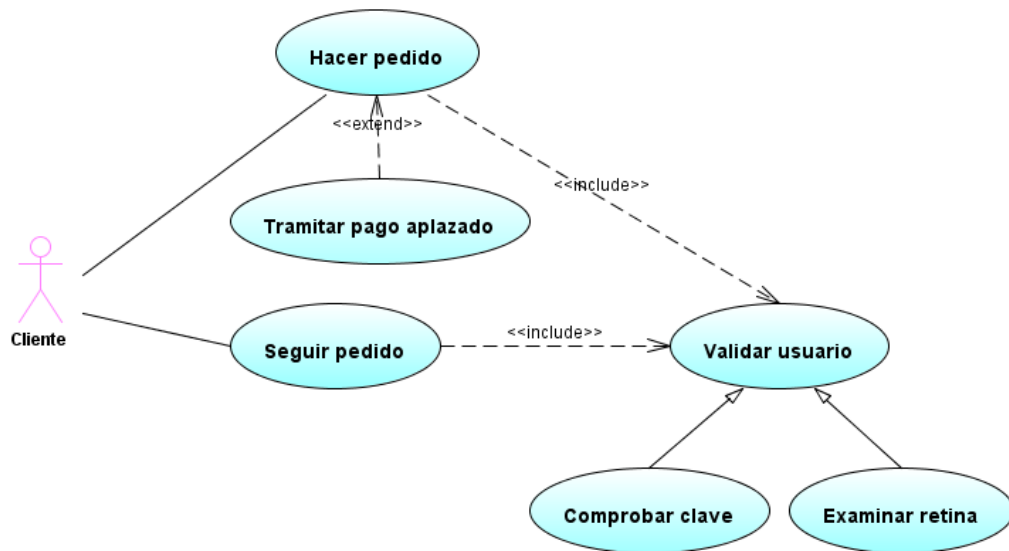
- a) *Identificar os actores.* Os actores son os roles que un usuario ou usuarios do sistema levan a cabo nalgún momento do tempo. Tamén poden ser outros sistemas cos que o 'sistema' en proceso de modelado ten interacción. Exemplo: Para un sistema de vendas (directas e por catálogo), os nosos actores poden ser: Vendedor, Cliente, Supervisor de Vendas.
- b) *Identificar os obxectivos xerais ou as responsabilidades* e a partir deles os casos de uso do sistema. Todos os actores na contorna a modelar teñen metas ou obxectivos, responsabilidades ou accións que desexan realizar ou obter do sistema. A partir deles definiremos os casos de uso do sistema. Por exemplo: para o sistema de Vendas, o Vendedor ten como meta, obxectivo ou responsabilidade: ofrecer produtos, pechar venda, gañar moito diñeiro vía cobrar comisións.
- c) Se é necesario, *establecer un proceso iterativo* no que os casos de usos se amplían afondando na súa descrición, buscando etapas comúns e alternativas que poderemos representar usando outros caso de uso relacionados polas relacións inclúe, xeneraliza e estende.
- d) *Describir cada caso de uso* (véxase apartado seguinte).

Algunhas recomendacións para deseñar este tipo de diagramas poden ser:

- Deben describirse a nivel esencial e non dependen da interface nin dos detalles de implementación.
- Debe evitarse representar máis de dous niveis de relacións de casos de uso no mesmo diagrama. Se aparece esta situación pode ser que esteamos facendo unha descomposición funcional dos casos de uso e esa é unha operación que se fará na fase de deseño.
- Soen colocarse os casos incluídos (<<include>>) á dereita dos caso que os inclúen e as extensións e xeneralizacións arriba.
- Centrarse na escritura en vez de no debuxo, é dicir, trátase de identificar aos actores e os casos de uso e ter unha axuda visual dos requisitos máis que de facer un debuxo.

- Non se deben de incluír as operacións CRUD (Create, Read, Update, Delete) excepto se son relevantes.
- Non se recomenda utilizar a inclusión para facer descomposición funcional.

Por exemplo:



### Descrición de cada caso de uso

Pódese describir un caso de uso utilizando un texto estruturado informal ou utilizando algún modelo xa deseñado e debería contemplar os seguintes aspectos:

- Identificación do caso de uso mediante nome e/ou clave e/ou número de control.
- Resúmen do que resolve este caso de uso.
- Actores que interveñen: nome e breve descrición se é necesario.
- Condicións previas que teñen que cumprirse antes de que se inicie o caso de uso.
- Condicións que se deben cumprir despois de terminado o caso de uso.
- Fluxo principal e alternativos que poden representarse de forma textual numerando cada paso, cun diagrama de fluxo, cun diagrama de secuencia ou cun diagrama de estados.
- Requirimentos iniciais que non teñen impacto na funcionalidade.
- Diagramas de navegación que ilustran o fluxo entre as pantallas para ese caso de uso.

Por exemplo a descrición do caso de uso “Realizar venta nun TPV” podería ser:

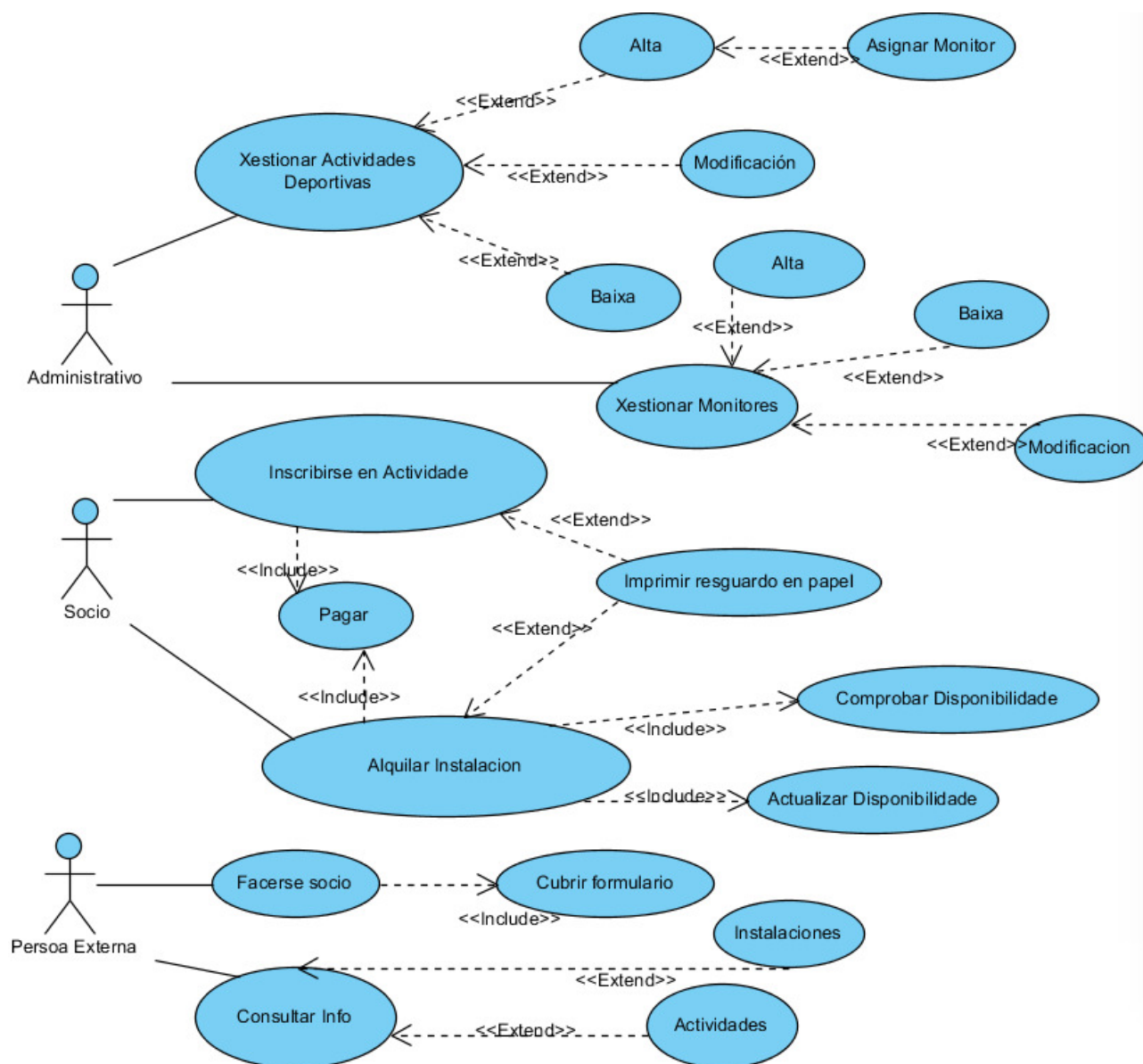
<b>Caso de uso: Realizar venta nun TPV</b>	
<b>Resumen</b>	Permite que un caixeiro poida rexistrar nun TPV os artigos que compra un cliente, xerar o ticket e rexistrar o cobro do importe da compra.
<b>Actor principal</b>	Caixeiro
<b>Actores secundarios</b>	<b>Administración:</b> quere ter rexistradas as transaccións e satisfacer os clientes. <b>Cliente:</b> quere artigos e o ticket de compra.
<b>Condicións previas</b>	O caixeiro ten que identificarse no sistema.
<b>Condicións posteriores</b>	Rexístrase a venta. Calcúlase o imposto. Actualízase contabilidade e inventario.
<b>Fluxo principal</b>	<ul style="list-style-type: none"> <li>▪ Cliente chega ao TPV cos artigos que desexa.</li> <li>▪ Caixeiro inicia nova venta.</li> <li>▪ Caixeiro introduce identificador de cada artigo.</li> <li>▪ Sistema rexistra liña de venta e o sistema presenta ao cliente: descrición de artigo, prezo e suma parcial.</li> <li>▪ Caixeiro repite pasos 3 e 4 ata que finalice entrada de artigos.</li> <li>▪ Sistema presenta total e caixeiro informa ao cliente de importe total a pagar.</li> <li>▪ Cliente paga e sistema xestiona o pago.</li> <li>▪ Sistema presenta ticket.</li> </ul>
<b>Fluxos alternativos</b>	<ul style="list-style-type: none"> <li>▪ O identificador do artigo non é válido. <ul style="list-style-type: none"> <li>– O sistema sinala o erro e rexeita a entrada.</li> </ul> </li> <li>▪ Cliente pide eliminar un artigo da compra. <ul style="list-style-type: none"> <li>– Caixeiro introduce identificador a eliminar.</li> <li>– Sistema actualiza a suma.</li> </ul> </li> <li>▪ Cliente non pode pagar compra. <ul style="list-style-type: none"> <li>– Caixeiro elimina toda a compra.</li> <li>– Sistema actualiza a suma poñéndoa a 0.</li> </ul> </li> </ul>
<b>Requisitos especiais</b>	<ul style="list-style-type: none"> <li>▪ Interface de usuario con pantalla táctil en monitor de pantalla plana. Texto visible a un metro de distancia.</li> <li>▪ Tempo de resposta para autorización de pago con tarxeta de crédito de 30 segundos no 90% dos casos.</li> </ul>
<b>Comentarios</b>	<ul style="list-style-type: none"> <li>▪ O identificador do produto pode estar en calquera formato.</li> <li>▪ Existe un lector de tarxetas de crédito para o pago con tarxeta.</li> <li>▪ Existe un escáner para lectura de código de barras dos produtos.</li> </ul>





## Tarefa 6.1. Interpretación dun diagrama de casos de uso

Dado o seguinte diagrama de casos de uso, redacta un posible escenario que se adapte á información recollida no diagrama:





### Tarefa 6.2. Elaboración dun diagrama de casos de uso

Representa a través dun diagrama de Casos de Uso o seguinte escenario:

- Deséxase desenvolver unha pequena aplicación para levar o control dos socios dun vídeo club. Esta aplicación poderá ser manexada tantos polo clientes como polos empregados do vídeo club.
- Os clientes poderán, logo de proceso de autenticación, darse de baixa e modificar os seus datos persoais. Ademais, se aínda non son socios, poderán solicitar darse de alta como socios.
- Os empregados do vídeo club poderán, logo de proceso de autenticación, dar de alta, de baixa e modificar datos de socios; consultar os datos dos socios e solicitar un listado de socios con pagamentos pendentes.
- Representa o modelo con Visual Paradigm. Non hai que facer a táboa de descrición.



### Tarefa 6.3. Elaboración dun diagrama de casos de uso

- Describir os casos de uso para realizar unha chamada telefónica dende un teléfono fixo ou dende un móbil. No caso do móbil deberíase de contemplar que é obrigatorio o acendido do móbil e o desbloqueo do teclado.
- Representa o diagrama con Visual Paradigm. Non hai que facer a táboa de descrición.



### Tarefa 6.4. Elaboración dun diagrama de casos de uso

- Describir os casos de uso dun dispensador automático de produtos alimenticios como bocadillos, latas, e bebidas ou outros produtos como xabón e deterxente nos que se pode:
  - Elixir o produto a comprar. Se o produto é alimenticio comprobarase a data de caducidade.
  - Pagar o produto en efectivo ou con tarxeta de crédito. Podes considerar
- Representa o diagrama con Visual Paradigm. Non hai que facer a táboa de descrición.



### Tarefa 6.5. Elaboración dun diagrama de casos de uso

- Describir os casos de uso dun sistema de consulta de horarios de tren nun sitio web detallándoos gradualmente:
  1. Describir os casos de uso para consultar horarios sen detalle. Terase en conta que hai que teclear datos e que despois verase o horario de trens que coincida cos datos tecleados.
  2. Detallar o diagrama anterior para contemplar que se teclearán os datos: estación de saída, de chegada, data e hora de saída, posibilidade de teclear unha estación intermedia e posibilidade de indicar o tempo de espera nesa estación intermedia.
  3. Detallar máis o diagrama anterior para que despois de ver os horarios correspondentes aos datos tecleados, se poida seleccionar un deles e opcionalmente se poida mercar o billete. Contemplaranse pero non se detallarán a existencia de diferentes tarifas, tipos de pago e envío do billete.
- Representa os diagramas con Visual Paradigm. Non hai que facer a táboa de descrición.



### Tarefa 6.6. Elaboración dun diagrama de casos de uso

a) Realiza un diagrama de casos de uso para o seguinte escenario:

O dono dun hotel pídenos desenvolver un programa para consultar os cuartos dispoñibles e poder reservar cuartos no seu hotel.

- O hotel posúe tres tipos de cuartos: simple, dobre e de matrimonio, e dous tipos de clientes: habituais (dispoñerán dun desconto) e esporádicos.
- Unha reserva almacena datos do cliente, do cuarto reservado, a data de comezo e o número de días que será ocupado o cuarto.
- O recepcionista do hotel debe poder facer as seguintes operacións:
  - Obter un listado dos cuartos dispoñible de acordo ao seu tipo.
  - Preguntar polo prezo dun cuarto de acordo ao seu tipo.
  - Preguntar polo desconto ofrecido aos clientes habituais.
  - Preguntar polo prezo total para un cliente dado, especificando o seu número de reserva, tipo de cuarto e número de noites.
  - Debuxar en pantalla a foto dun cuarto de acordo ao seu tipo.
  - Reservar un cuarto especificando o seu número e o nome do cliente.
  - Eliminar unha reserva especificando o número do cuarto.
- O administrador pode usar o programa para:
  - Cambiar o prezo dun cuarto de acordo ao seu tipo.
  - Cambiar o valor do desconto ofrecido aos clientes habituais.
  - Obter información de todas as reservas realizadas.

*Pista: Os tipos de cuarto poden xerar herdanza nos casos de uso, pero para simplificar o diagrama podemos representalos como unha inclusión do caso de uso "Elixir tipo de cuarto"*

Representa o modelo con Visual Paradigm. Non hai que facer a táboa de descrición.

b) Realiza o diagrama de clases do escenario anterior con Visual Paradigm. Inclúe unicamente os parámetros e métodos principais.

## 2. Diagramas de interacción

---

### 2.1 Os diagramas de interacción de UML

Os diagramas de interacción inclúense dentro dos chamados diagramas de comportamento de UML e permítenos describir a maneira en que colaboran grupos de obxectos para certo comportamento. Habitualmente, os diagramas de interacción captan o comportamento dun só caso de uso, representando os obxectos e as mensaxes que se pasan entre eles dentro de dito caso de uso.

Existen cinco tipos de diagramas de interacción (a partir de UML 2.X soamente catro):

- *Diagrama de secuencia*: Mostra as mensaxes que son pasadas entre obxectos nun escenario.
- *Diagrama de comunicación*: Mostra as interaccións entre os participantes facendo énfase na secuencia de mensaxes.
- *Diagrama de colaboración*: (Soamente en UML 1.X) Mostra as interaccións organizadas arredor dos roles.
- *Diagrama de (visión de conxunto ou resumo de) interacción*: Trátase de mostrar de forma conxunta diagramas de actividade e diagramas de secuencia.
- *Diagrama de tempo*: Pon o foco nas restricións temporais dun obxecto ou un conxunto de obxectos.

De todos eles os mais empregados, e que son os que veremos na presente actividade, son os *diagramas de secuencia* e os *diagramas de comunicación*. O emprego destes dous diagramas vains servir para:

- Ilustrar e comprobar o comportamento dun conxunto de obxectos (sistema ou subsistema)
- Axudar a descubrir os obxectos do sistema.
- Axudar a descubrir os métodos de ditos obxectos.

Os diagramas de secuencia poñen en primeiro plano os aspectos temporais, mentres que os diagramas de comunicación amosan os vínculos entre clases.

### 2.2 Diagramas de secuencia

#### Introdución

Mostran, para un escenario dado, a interacción dun grupo de obxectos a través do tempo mediante o envío de mensaxes entre eles. Son os diagramas máis utilizados para representar as interaccións entre obxectos.

Normalmente utilízanse para modelar casos de uso do sistema e neste caso a secuencia pode expresar tanto un caso de uso completo como unha alternativa dun caso concreto. Polo tanto, a menos que se modele un sistema moi pequeno, a dinámica completa dun sistema representarase mediante un conxunto de diagramas de secuencia. Tamén son unha boa ferramenta para explorar a lóxica dunha operación complexa, ou os elementos implicados na prestación dun servizo, ou axudar a detectar cales van ser as clases máis complexas de implementar e decidir cales de elas van a necesitar diagramas de estado.



## Elementos

Os compoñentes deste diagrama son:

- Participantes que se colocan na parte superior do diagrama ao longo do eixo X. Poden ser actores ou obxectos:
  - **Actor.** Normalmente inicia a secuencia e colócase á esquerda. O símbolo é o mesmo có dos diagramas de casos de uso.
  - **Obxecto.** É a instancia dunha clase participante na secuencia. O símbolo é un rectángulo coa notación “función:Clase” que representa a función dun obxecto seguida do nome da clase. A función é opcional se só participa unha instancia dunha clase. A *liña de vida do obxecto* é unha liña descontinua vertical dende o símbolo do obxecto ata o final da vida que se representa cunha aspa.

Os participantes teñen un *período de actividade* dentro da secuencia que indicará que o obxecto está executando unha acción, e isto represéntase mediante un rectángulo sobre a liña de vida que empeza cando o obxecto recibe unha mensaxe e termina cando devolve a última resposta. A parte superior deste rectángulo alíñase co comezo da acción. A parte inferior alíñase coa terminación e pode marcarse cunha mensaxe de retorno.

- **Accións** entre actores e obxectos ou **mensaxes** entre obxectos. Colócanse ao longo do eixo ‘Y’ en orde de sucesión no tempo dende arriba cara abaixo. As mensaxes poden ir numeradas en secuencia para facilitar a comprensión do diagrama. Se unha mensaxe se envía antes de que remate o tratamento da mensaxe precedente é posible utilizar a numeración composta. Un obxecto pode enviar mensaxes a si mesmo. As mensaxes poden corresponder a :

- **Mensaxes síncronas** que son as utilizadas con maior frecuencia e correspóndense a mensaxes que deben de finalizar para que o emisor poida continuar a actividade. Represéntanse cunha liña de ida continua con punta de frecha ata o receptor e unha liña de volta descontinua con punta de frecha ata o emisor. As liñas poden levar o nome do método invocado e os valores dos parámetros.
- **Mensaxes asíncronas** nas que o emisor non espera a que o destinatario termine a activación para seguir coa súa actividade. Isto utilízase por exemplo con obxectos que poden funcionar en paralelo (sistemas multi-thread).

Un tipo especial de mensaxes son as de creación e destrución de obxectos que se estereotipan con <<create>> e <<destroy>> respectivamente.

## Recomendacións

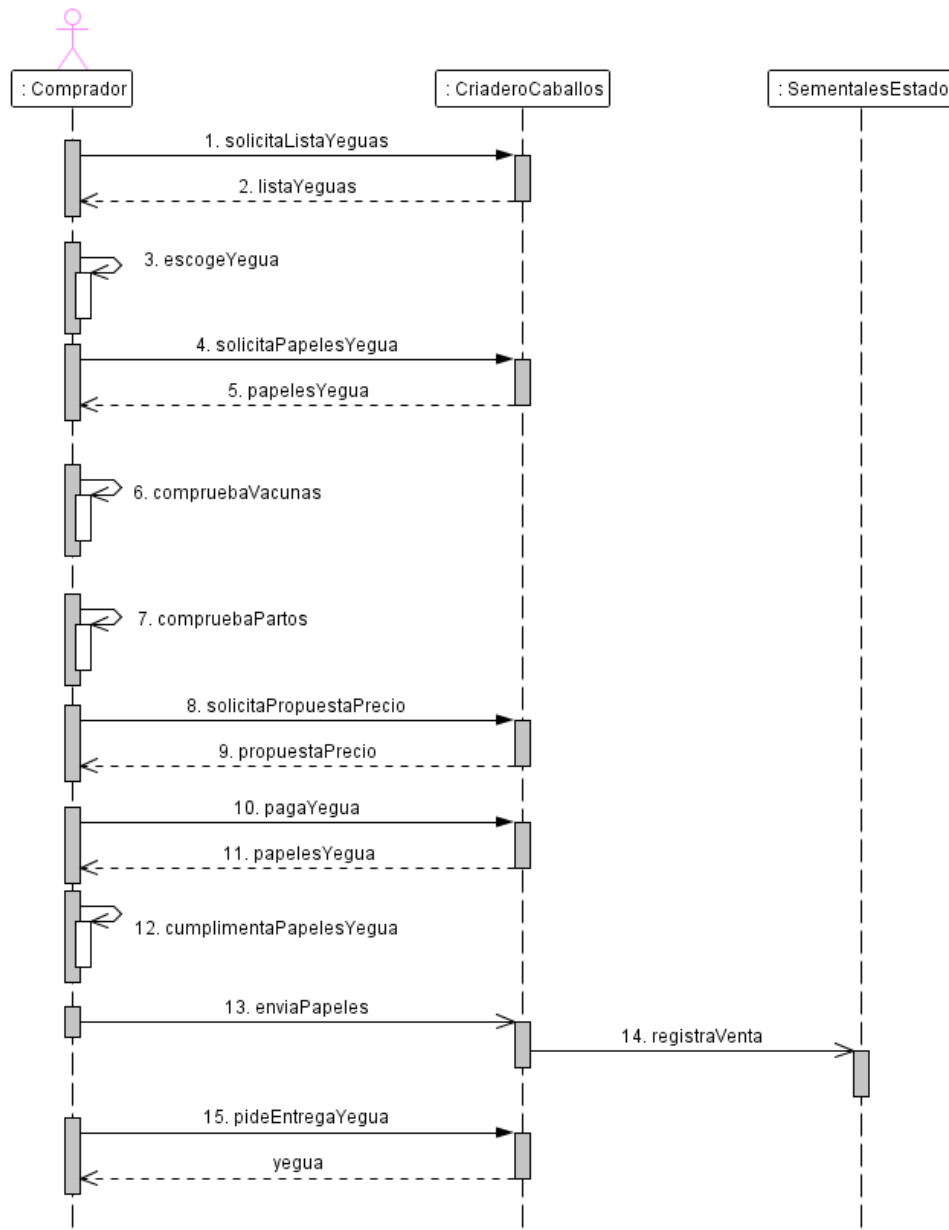
A continuación enumeramos unha serie de recomendacións a ter en conta na elaboración de diagramas de secuencia:

- A orde entre as mensaxes e os participantes debe ser sempre de esquerda a dereita e de arriba a abaixo.
- O nome dos actores debe ser consistente cos diagramas de casos de uso.
- O nome das clases debe ser consistente cos diagramas de clases.
- Incluír notas nas secuencias.
- Só incluír a aspa de destrución do obxecto nos casos en que proporcione información sobre cando debe destruírse.

- Nos parámetros das mensaxes é máis conveniente usar nomes claros que os tipos dos mesmos.

## Exemplo

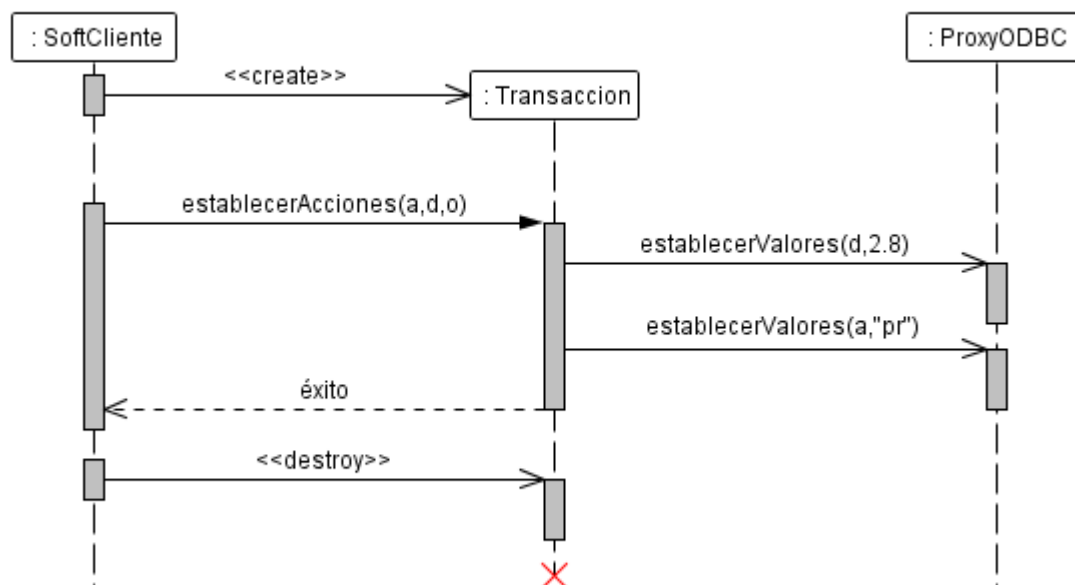
Exemplo de diagrama de secuencia do escenario da compra dunha egua<sup>9</sup>.



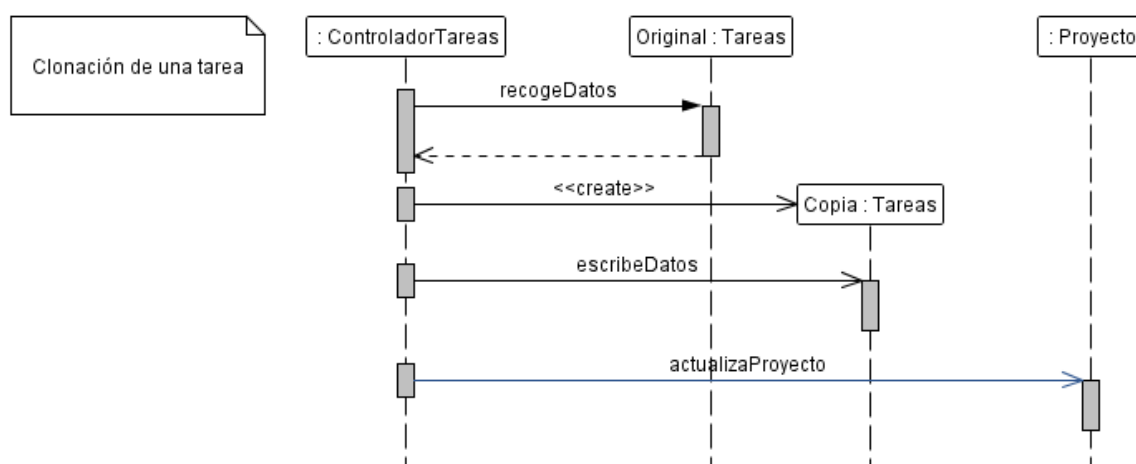
Se o diagrama de secuencia se fai na fase de deseño pode poñerse máis información sobre as mensaxes como por exemplo a información (parámetros) que vai necesitar o obxecto receptor para levar a cabo a actividade. Por exemplo<sup>10</sup> no caso dunha aplicación que fai unha transacción cunha base de datos podemos especificar os parámetros necesarios nun escenario concreto:

<sup>9</sup> DEBRAUWER, Laurent, VAN DER HEYDE, Fien. *UML 2: Iniciación, exemplos y ejercicios corregidos [3ª edición]*. Ediciones ENI

<sup>10</sup> GARCIA MOLINA, Jesús. *El lenguaje unificado de modelado*. Dpto. Informática y sistemas. Universidad de Murcia. <http://dis.um.es/~jnicolas/CSW/transparencias/Capitulo01p04.pdf>

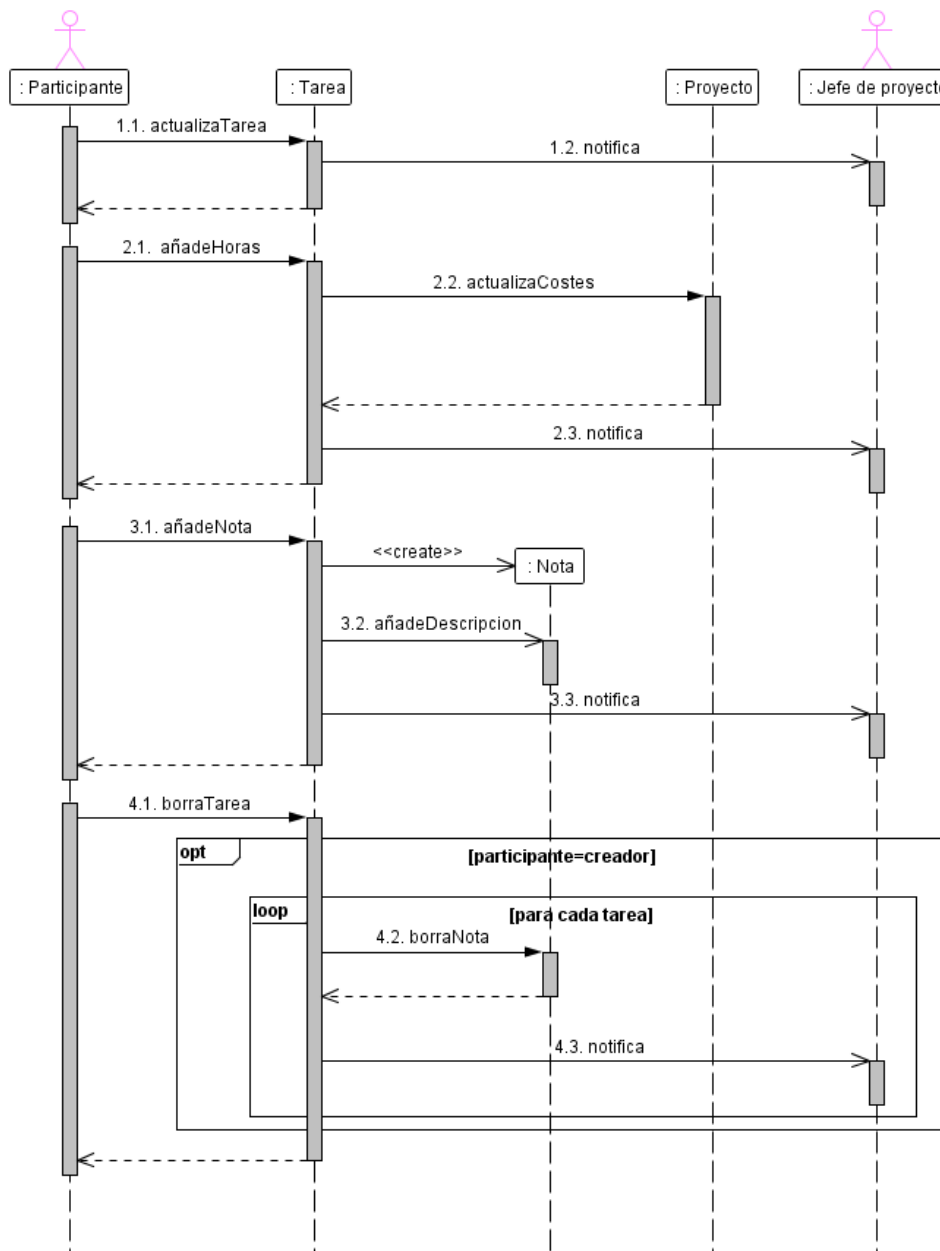


Se a secuencia traballa con obxectos da mesma clase pero con roles diferentes pódense etiquetar. Por exemplo<sup>11</sup> no caso dun obxecto encargado de facer unha copia dun obxecto tarefa vemos que temos dúas instancias do obxecto Tarefa, un para a orixinal e outro para a copia:



<sup>11</sup> AYCART PÉREZ, David. GIBERT GINESTA, Marc HERNÁNDEZ MATÍAS, Martín, MAS HERNÁNDEZ, Jordi. *Ingeniería de software en entornos de SL*. Universitat Oberta de Catalunya.

Segundo a notación de UML é posible expresar condicionais e bucles nos diagramas de secuencia. Moitos expertos non o aconsellan xa que implica incluír lóxica nuns diagramas que só deberían de representar mensaxes entre participantes na secuencia, pero a veces é imprescindible para reflectir correctamente a secuencia de mensaxes. O marco de iteración é unha caixa transparente que envolve a condición ou bucle e que ten na esquina superior esquerda o tipo de iteración (bucle=loop ou condición=opt) e ao seu carón a condición entre corchetes. Vexamos un exemplo<sup>12</sup>:



<sup>12</sup> AYCART PÉREZ, David. GIBERT GINESTA, Marc HERNÁNDEZ MATÍAS, Martín, MAS HERNÁNDEZ, Jordi. *Ingeniería de software en entornos de SL*. Universitat Oberta de Catalunya.



## 2.3 Diagramas de comunicación

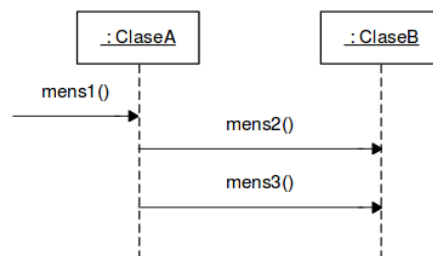
### Introdución

O diagrama de comunicación aparece en UML 2.0 e é a versión abreviada do diagrama de colaboración de UML 1.x.

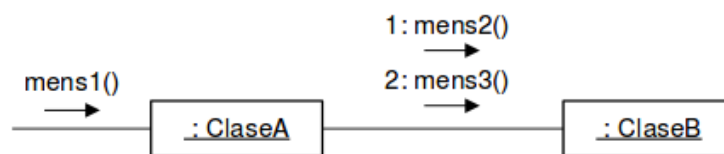
Serve para destacar a organización dos obxectos que participan na interacción. Algúns autores o consideran como unha variación do diagrama de secuencia na que os obxectos non están en filas e columnas senón que distribuídos libremente e coas mensaxes numeradas para seguir a secuencia de mensaxes. A diferenza do diagrama de secuencia non contempla retornos; só contempla as mensaxes que se intercambian os obxectos.

Os diagramas de comunicación constrúense normalmente a partir dos diagramas de secuencia colocando en primeiro lugar os obxectos, despois os enlaces e finalmente as mensaxes.

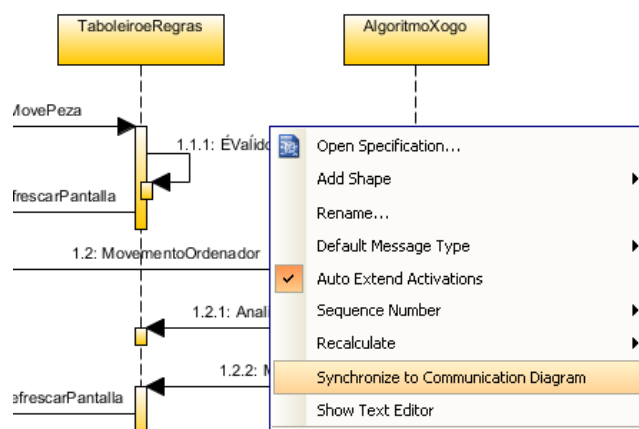
Diagrama de secuencia:



O seu diagrama de comunicación equivalente:



A equivalencia entre os dous tipos de diagramas permite que algunhas aplicacións como *StarUML* e *Visual Paradigm* convertan diagramas de secuencia en diagramas de comunicación e ao revés.



Outras como *NetBeans* non contemplan este tipo de diagramas.

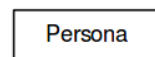
## Elementos

Os principais elementos que poden aparecer nun diagrama de comunicación son os seguintes<sup>13</sup>:

- **Instancias de clases.** As instancias represéntanse igual que nos diagramas de secuencia:



- **Clases.** As clases represéntanse co nome da clase dentro dun rectángulo. Corresponden a unha clase non a unha instancia.

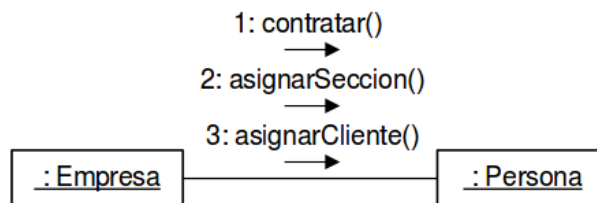


Clase Persona

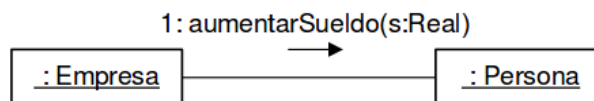
- **Enlaces.** Representa unha conexión entre instancias que indica navegabilidade e visibilidade entre elas.



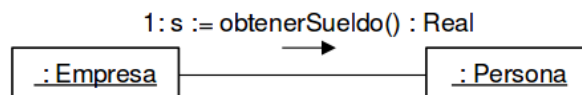
**Mensaxes.** As mensaxes son representadas mediante unha frecha etiquetada. Unha mensaxe está asociada a un enlace e ten asignado un número de secuencia que determina a orde de ocorrencia.



- **Parámetros.** Os parámetros móstranse entre paréntese á dereita do nome da mensaxe. Pódese mostrar ademais o seu tipo.



- **Tipo de retorno.** O valor de retorno pode ser mostrado á esquerda da mensaxe, cun := no medio. Pódese mostrar ademais o tipo do valor de retorno.



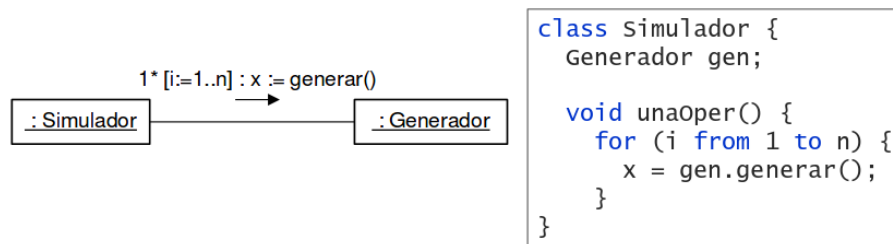
A sintaxe das mensaxes é a seguinte:

[ret :=] mensaxe([param [: TipoParam]]) [: TipoRet]

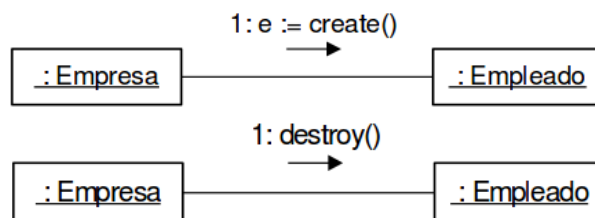
<sup>13</sup> [https://eva.fing.edu.uy/pluginfile.php/60058/mod\\_resource/content/1/pavan-teorico11-diseno-diagramas\\_comunicacion.pdf](https://eva.fing.edu.uy/pluginfile.php/60058/mod_resource/content/1/pavan-teorico11-diseno-diagramas_comunicacion.pdf)

Onde:

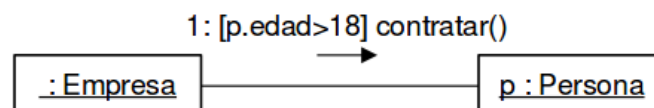
- *ret* almacena o resultado da operación (opcional)
  - *mensaxe* é o nome da mensaxe enviada (e da operación invocada)
  - *param* son argumentos usados no envío
  - *TipoParam* é o tipo de cada parámetro (opcional)
  - *TipoRet* é o tipo do valor de retorno da operación (opcional)
- **Iteracións.** As iteracións indícanse mediante un asterisco (\*) a continuación do numero de secuencia da mensaxe. Isto expresa que a mensaxe é enviada en forma repetida (nun bucle) ao receptor



- **Creación e destrución de instancias.** Correspondese co envío das mensaxes *create* e *destroy*:



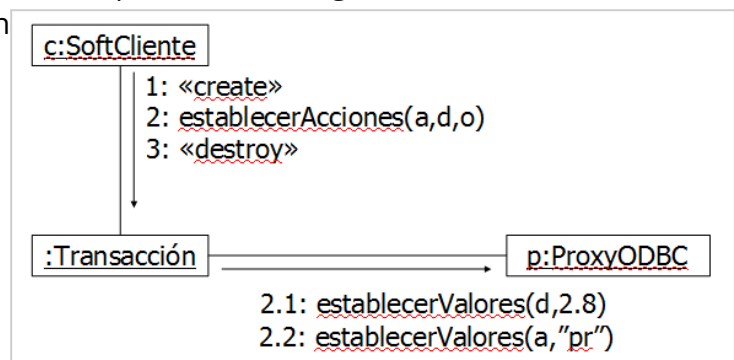
**Mensaxes condicionais.** Unha mensaxe condicional é enviada unicamente se a súa condición é satisfeita. A condición móstrase entre paréntese rectos ([ ]) á esquerda da mensaxe.



Hai que mencionar tamén que aínda que estas son as recomendacións que fai o estándar respecto a notación a empregar, podémonos atopar, igual que pasa con outros diagramas, con variacións a dita nomenclatura ben sexa porque os diagramas empregan nomenclatura de versións anteriores de UML ou porque o software empregado para a representación non segue fielmente as recomendacións oficiais.

## Exemplos

- Exemplo do diagrama de comunicación correspóndete ó diagrama de secuencia dunha transacción cunha base de datos visto en apartados anteriores:



Exemplo<sup>14</sup> de caso de uso sinxelo para un módulo de planificación de rutas.

Diagrama de secuencia:

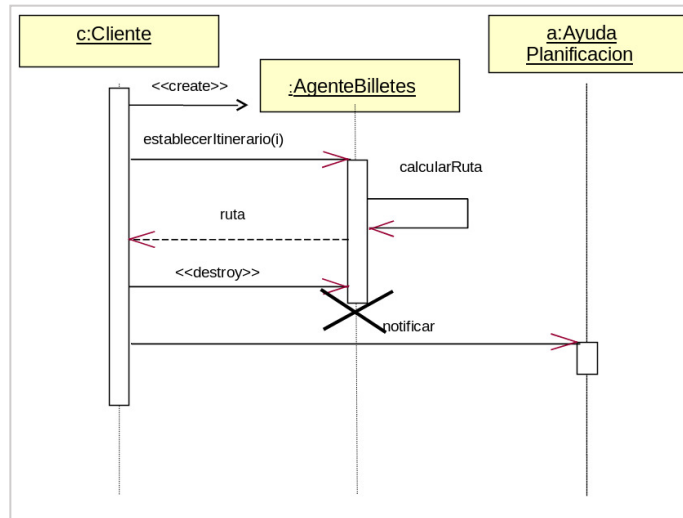
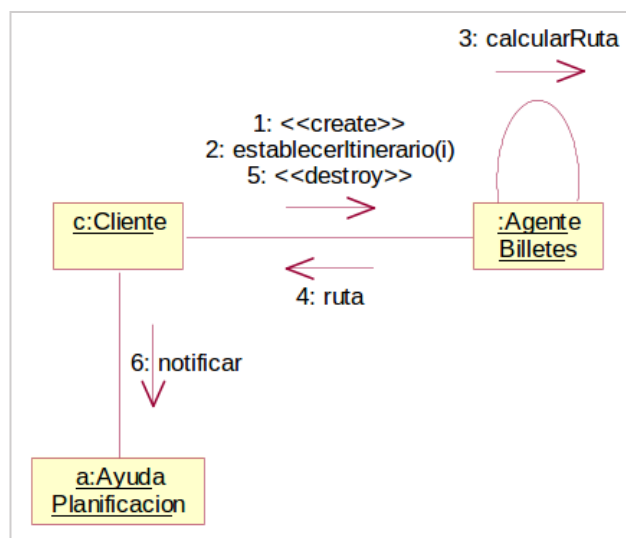


Diagrama de comunicación:



Exemplo dun caso de uso para a preparación e entrega dos distintos produtos (items) dun pedido:

Diagrama de secuencia:

<sup>14</sup> GARCIA MOLINA, Jesús. *El lenguaje unificado de modelado*. Dpto. Informática y sistemas. Universidad de Murcia. <http://dis.um.es/~jnicolas/CSW/transparencias/Capitulo01p04.pdf>



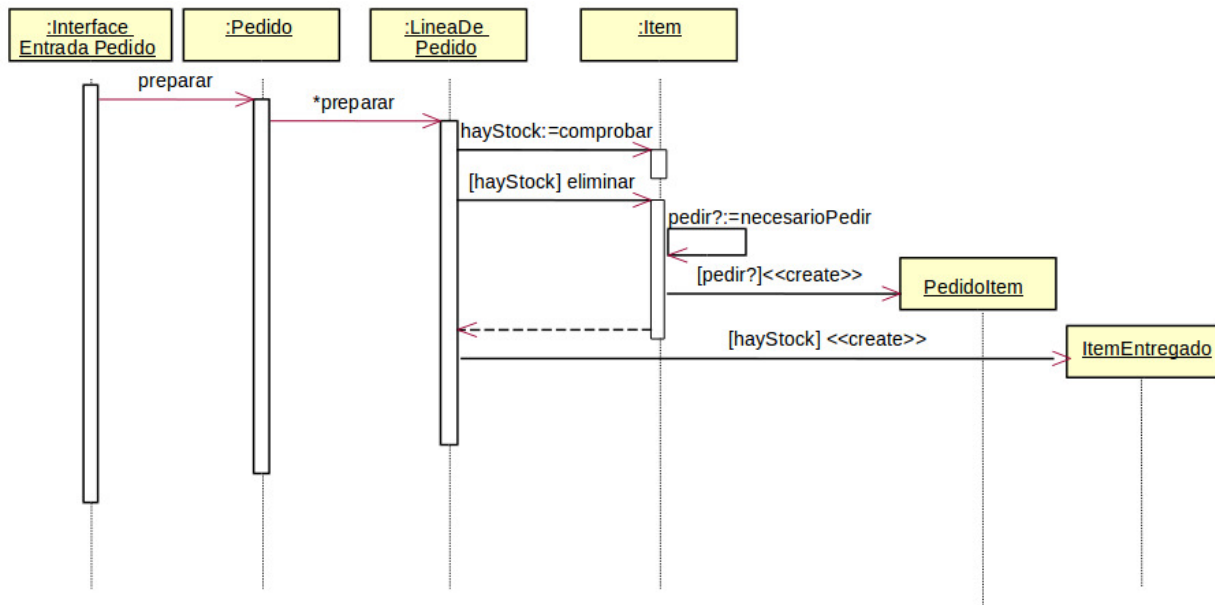
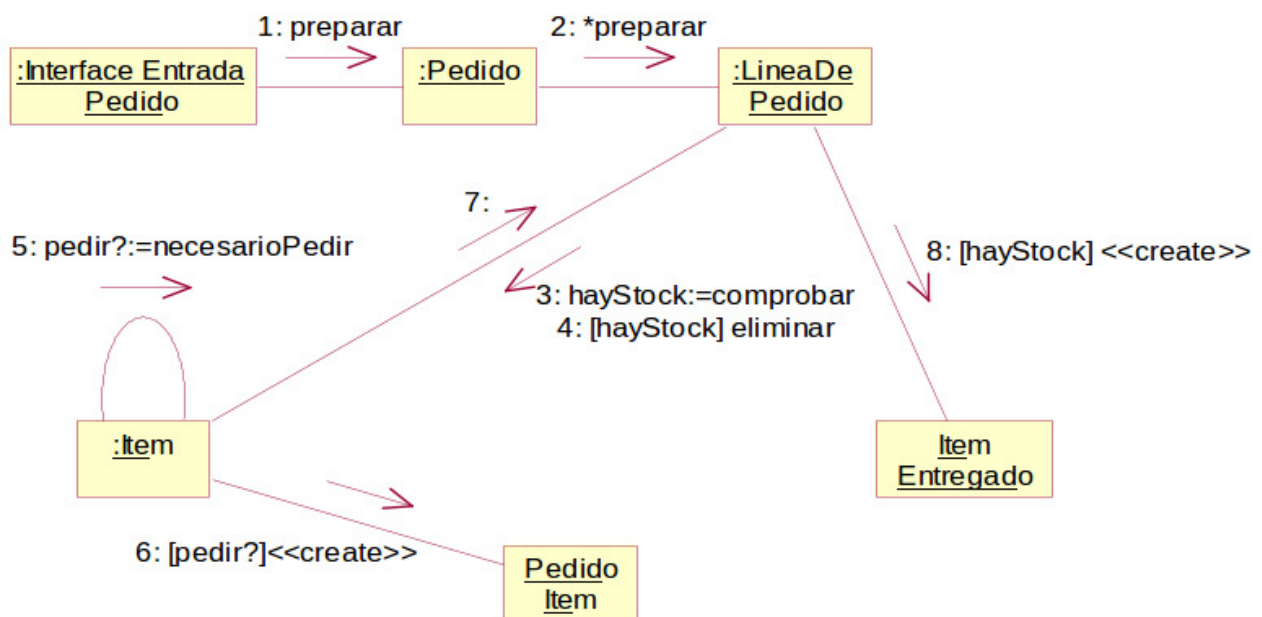


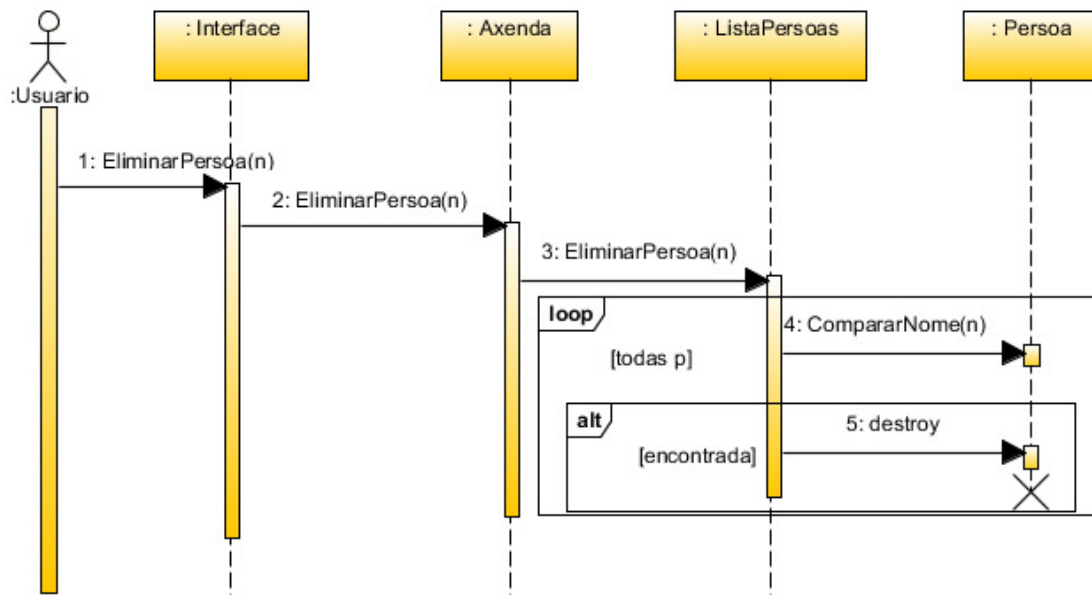
Diagrama de comunicación:





## Tarefa 6.7. Interpretación dun diagrama de secuencia

Dado o seguinte diagrama de secuencia<sup>15</sup>, describe brevemente o escenario que representa:



## Tarefa 6.8. Elaboración dun diagrama de secuencia

Utilizando Visual Paradigm representa o diagrama da tarefa anterior e engádelle o seguinte suposto:

- Antes de poder eliminar unha persoa debe de comprobarse se existen citas con dita persoa e de ser así hai que eliminalas antes de poder eliminar a persoa.

Os novos obxectos que se necesitan son *:ListaCitas* e *:Cita*

<sup>15</sup> Exemplo tomado de apuntamentos do Dpto. de Lenguajes y Sistemas Informáticos, Escuela Universitaria de Ingeniería de Vitoria-Gasteiz.



### Tarefa 6.9. Elaboración dun diagrama de secuencia

Mediante algunha ferramenta software de manexo de diagramas UML debuxa un diagrama de secuencia que represente un caso de uso dun programa de Xadrez<sup>16</sup>. O caso de uso en cuestión será o movemento dunha peza por parte do usuario e incluírase tamén o movemento seguinte realizado por parte do programa.

Como estamos nunha fase previa de deseño soamente será necesario incluír no diagrama, aparte do propio usuario, tres módulos: un para a interface de usuario, outro para conter o taboleiro e regras do xadrez (movementos válidos e demais) e outro para o algoritmo de xogo do ordenador.

Non é necesario incluír situacións erróneas, movementos inválidos, xaques, etc posto que poñer todos os detalles pode dar lugar a un diagrama que non se entenda ou difícil de ler. (Nunha situación de deseño real o diagrama pode acompañarse cun texto no que se detallen todas estas situacións erróneas e particularidades).



### Tarefa 6.10. Elaboración dun diagrama de comunicación

Elabora o diagrama de comunicación para o diagrama de secuencia da tarefa 3 (Xadrez). Faino primeiro sen a axuda automática do entorno software e despois fai que VisualParadigm xere un automaticamente a partir do diagrama de secuencia. Compara os dous diagramas centrándote nas diferenzas atopadas.



### Tarefa 6.11. Elaboración dun diagrama de comunicación

Elabora o diagrama de comunicación para o diagrama de secuencia da primeira tarefa (Axenda). Faino primeiro sen a axuda automática do entorno software e despois fai que VisualParadigm xere un automaticamente a partir do diagrama de secuencia.

Repite o exercicio agora co diagrama ampliado da segunda tarefa.

---

<sup>16</sup> <http://www2.uah.es/jcaceres/capsulas/DiagramaSecuencia.pdf>

## 3. Diagramas de estados

### 3.1 Propósito e función dos diagramas de estados

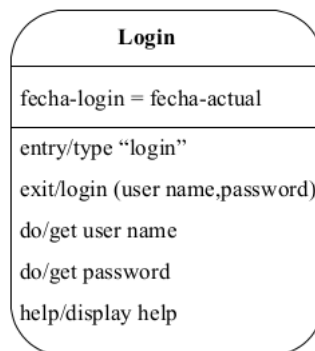
Os diagramas de estados, tamén chamados diagramas de máquina de estados, vannos servir para mostrar os diferentes estados polos cales pasa un obxecto durante o seu ciclo de vida e a maneira en que o obxecto cambia dun estado a outro en resposta a os eventos que chegan a el.

Dentro dun proxecto software, os diagramas de estado normalmente empréganse sempre para mostrar o comportamento dun único obxecto ó longo do seu ciclo de vida pero tamén poden ser útiles para representar a secuencia de eventos que poden afectar a un sistema nun determinado caso de uso. Neste caso falamos de *diagramas de estado para casos de uso* e unicamente soen empregarse en casos de uso complexos con moitos eventos que deben de sucederse seguindo un orden establecido (por exemplo cando estamos utilizando un procesador de texto).

### 3.2 Elementos

Nun diagrama de estados basicamente aparecen dous tipos de elementos:

- **Estados.** Un estado é unha situación na vida dun obxecto. Identifica un período de tempo do obxecto no cal o obxecto está esperando algunha operación, ten certo estado característico ou pode recibir certo tipo de estímulos. Representábase mediante un rectángulo cos bordos redondeados, que pode ter tres compartimentos: un para o nome, outro para o valor característico dos atributos do obxecto nese estado e outro para as accións que se realizan ao entrar, saír ou estar nun estado (*entry*, *exit* ou *do*, respectivamente).



En moitos diagramas de estado omítense os dous compartimentos inferiores.

Márcanse tamén os estados iniciais e finais mediante os símbolos ● e ●◉ , respectivamente.

- **Transicións.** Unha transición é unha relación entre dous estados que indica que un obxecto que estea no primeiro estado realizará certas accións e entrará no segundo estado cando ocorra un evento determinado e se satisfagan unhas condicións específicas. Graficamente represéntase cunha liña continua con punta de frecha no estado destino e normalmente acompáñase cun texto indicando o nome do evento que desencadea a transición.

Pode existir unha transición que teña o mesmo estado orixe e destino.

O nome do evento pode vir acompañado cunha serie de parámetros có seguinte formato:

*nome-evento* '['*lista-de-argumentos*']' '['*condicion-de-garda*']' '/' *acción*  
'^' *cláusula-envío*

Onde:

- *nome-evento* e *lista-de-argumentos* describen o evento que dá lugar á transición e forman o que se denomina *event-signature*.

- *condicion-de-garda* é unha condición (expresión booleana) adicional ao evento e necesaria para que a transición aconteza.

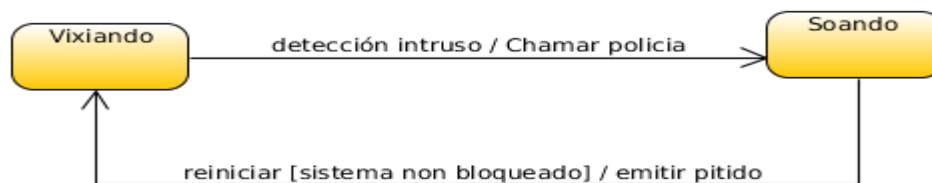
Se a *condicion-de-garda* se combina cunha *event-signature*, entón para que a transición se dispare teñen que suceder dúas cousas: debe acontecer o evento e a condición booleana debe ser verdadeira.

- *acción* é unha acción que se executa cando se dispara a transición e que, por exemplo, pode ser unha chamada a unha operación.

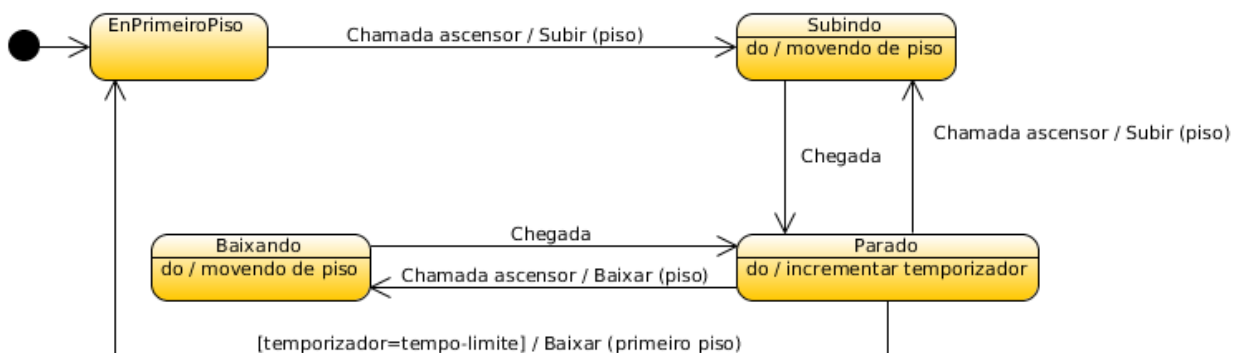
É posible ter unha ou varias *acción* nunha transición de estado, as cales delimitáanse co carácter "/".

- *cláusula-envío* é unha acción adicional que se executa co cambio de estado, por exemplo, o envío de eventos a outros paquetes ou clases.

Exemplo:



Imos ver un exemplo<sup>17</sup> dun diagrama de estado para un ascensor:



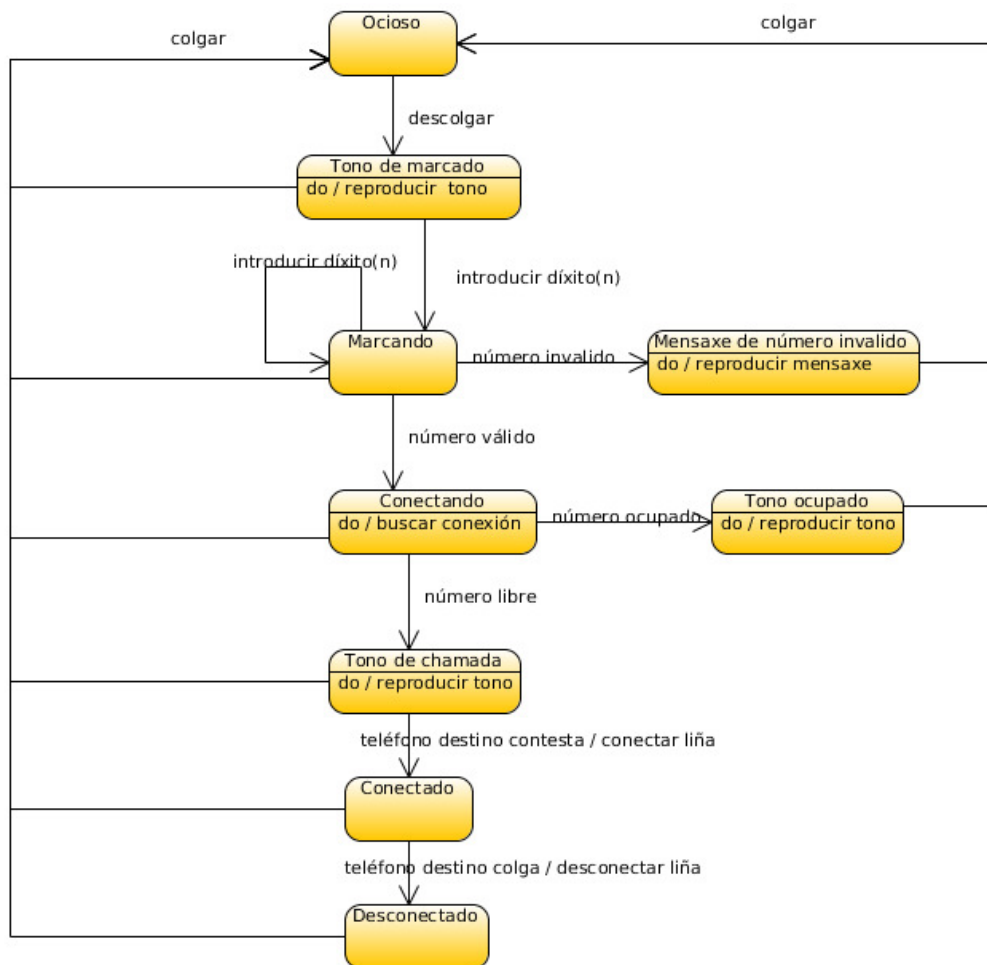
<sup>17</sup> OTERO VIDAL, M<sup>a</sup> Carmen. *Diagramas de estado*. Dpto. de Lenguajes y Sistemas Informáticos. Universidad del País Vasco.



O ascensor empeza estando no primeiro piso. Pode subir ou baixar. Se o ascensor está parado nun piso, acontece un evento de tempo superado despois dun período de tempo e o ascensor baixa ao primeiro piso. Este diagrama de estado non ten un punto de finalización (estado final).

O evento da transición entre os estados *EnPrimeiroPiso* e *Subindo* ten un argumento, *piso*. O mesmo sucede cos eventos das transicións entre *Parado* e *Subindo* e entre *Parado* e *Baixando*. O estado *Parado* asigna o valor cero ao atributo temporizador, logo o incrementa continuamente ata que aconteza o evento *Chamada ascensor* ou ata que a condición *[temporizador = tempo-límite]* se converta en verdadeira.

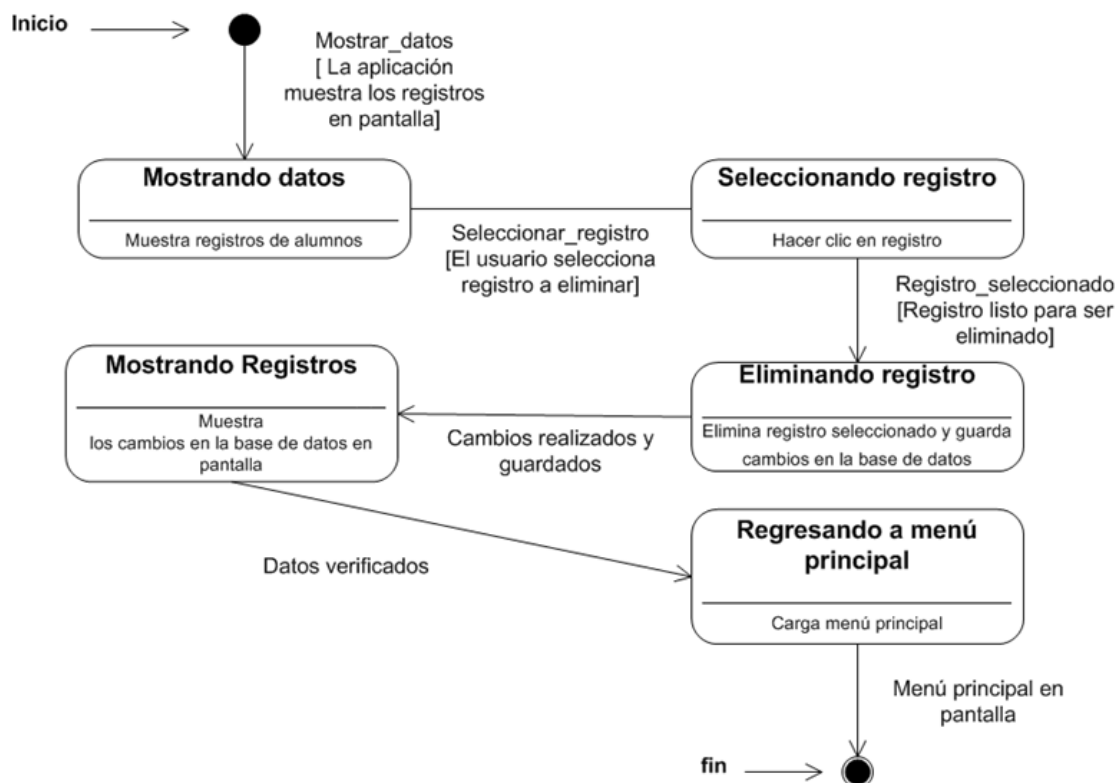
Outro exemplo, neste caso dos estados polos que pasa un teléfono nunha chamada telefónica:





### Tarefa 6.12. Interpretación dun diagrama de estados

Dado o seguinte diagrama de estados, redacta un posible escenario que se adapte á información recollida no diagrama<sup>18</sup>:



### Tarefa 6.13. Elaboración dun diagrama de estados

Representa a través dun diagrama de estados o funcionamento dun semáforo con luces para vehículos e para peóns segundo as seguintes indicacións:

- Os posibles estados do semáforo serán:
  - Roxo-verde*. Roxo para os vehículos, verde para os peóns.
  - Roxo-verde!*. Roxo para os vehículos, verde intermitente para os peóns.
  - Amarelo-roxo*. Amarelo para os vehículos, roxo para os peóns.
  - Verde-roxo*. Verde para os vehículos, roxo para os peóns.
- O semáforo estará en verde para os vehículos ata que alguén pulse o botón para os peóns. Cando isto ocorre o semáforo porase en amarelo durante 4 segundos e logo porase en roxo para os vehículos e verde para os peóns. Os peóns terán o semáforo en verde durante 20 sg. Pasado este tempo porase intermitente para os peóns durante 5 sg e logo porase de novo a roxo para os peóns e verde para os vehículos.

<sup>18</sup> <http://gzloluna8sm.blogspot.com.es/2010/06/diagramas-de-estados.html>



#### Tarefa 6.14. Elaboración dun diagrama de estados

Dado o seguinte escenario:

- Unha biblioteca ten copias de libros. Estes últimos caracterízanse polo seu nome, tipo (novela, teatro, poesía, ensaio), editorial, ano e autor.
- Os autores caracterízanse polo seu nome, nacionalidade e data de nacemento.
- Cada copia ten un identificador, e pode estar na biblioteca, prestada, reservada (un máximo de dous días), con atraso ou en reparación.
- Os lectores poden ter un máximo de 3 libros en préstamo.
- Cada libro prestase un máximo de 30 días. Por cada día de atraso, imponse unha "multa" de dous días sen posibilidade de coller un libro.

Realiza un posible diagrama de estados para a clase "copia".



#### Tarefa 6.15. Elaboración dun diagrama de estados

Realiza un diagrama de estados que represente o estado dunha multa de tráfico (imposta, comunicada, recorrida, anulada, confirmada, cobrada, en embargo), tendo en conta o seguinte:

- Cando se crea unha multa, automaticamente envíase unha notificación ao condutor.
- Se correos devolve asinada a notificación de recibo ou certifica que o condutor rexeitou a notificación, a multa pasa a notificada.
- Aos 30 días sen recorrer, pasa automaticamente a confirmada. Se chega antes un recurso, pasa a recorrida ata que a autoridade competente decide confirmar ou anular a sanción, salvo que transcorran 6 meses sen decisión e nese caso pasa tamén a anulada.
- Se a multa está confirmada e transcorren 6 meses, iníciase automaticamente o procedemento de embargo con comunicación simultánea ao interesado.
- En calquera momento apropiado, a multa se pode pagar e pasa ao estado cobrada.

## 4. Diagramas de actividades

---

### 4.1 Propósito e función dos diagramas de actividades.

Os diagramas de actividades son unha variante dos diagramas de estados nos que os estados representan a execución de actividades ou accións e as transaccións só se disparan cando se finaliza a execución das actividades ou accións.

Cando se empeza a traballar con este tipo de diagramas non sempre resulta doado diferenciar un estado dunha actividade. Os estados son xeralmente atómicos (non se poden dividir) e pásase dun estado a outro cando se produce algún evento. As actividades, por contra, non son atómicas (poden dividirse noutras actividades ou accións) e xeralmente pásase dunha a outra cando a actividade termina.

Facendo un símil coas persoas, un estado podería ser *triste*, e non o podemos dividir en estados mais sinxelos, e normalmente solo poderemos cambiar de estado - a *contento* por exemplo - cando suceda algún evento, por exemplo recibir unha boa noticia. Pola contra, un exemplo de actividade podería ser *facen a colada*, que podemos dividir en actividades ou accións mais sinxelas (seleccionar roupa, metela na lavadora, poñer a lavadora, etc) e, normalmente cambiaremos a outra actividade, *facen a comida* por exemplo, cando terminemos a primeira.

A diferenza doutros diagramas que soen empregarse para funcións moi concretas, os diagramas de actividades son mais versátiles e poden empregarse para distintas tarefas dentro do proceso de análise e deseño dun proxecto e a distinto nivel de detalle. Por exemplo:

- Para describir un proceso de negocio ou un fluxo de traballo entre os usuarios e o sistema.
- Para describir os pasos que se levan a cabo nun caso de uso.
- Para describir un método, unha función ou unha operación de software.

En todos estes casos, debuxar un diagrama de actividades pode axudar a mellorar o proceso que se está representado. Se o diagrama de actividade do proceso resulta ser moi complexo, pódense buscar posibilidades para simplificalo.

### 4.2 Elementos

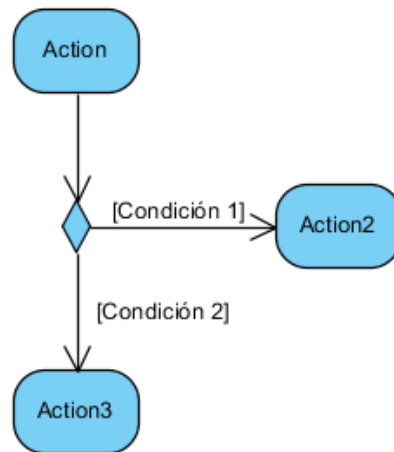
Os diagramas de actividade poden incluír os seguintes elementos:

- **Actividade/Acción.** Identifícase cun paso ou conxunto de pasos na execución do proceso que se estea representando. A diferenza entre actividades e accións é que as primeiras poden ser subdivididas en outras actividades ou accións e as segundas non (considéranse atómicas, e dicir, indivisibles).

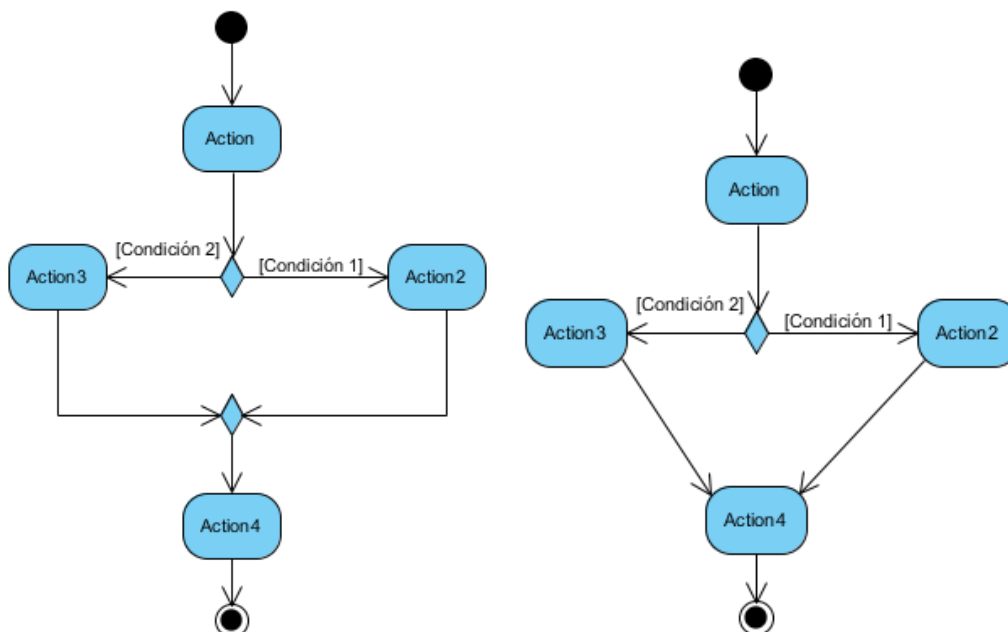
Representáanse mediante un rectángulo cos bordos redondeados. Dentro poñerase o nome da actividade ou acción aínda que tamén pode aparecer unha expresión que a identifique.

Algunhas veces as accións representáanse cunha figura en forma de píldora ou cápsula para diferenciarlas das actividades.

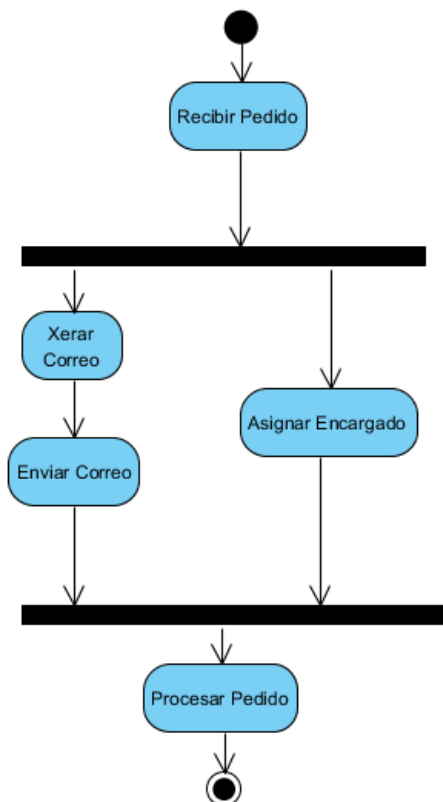
- **Transición ou fluxo.** Determina a actividade que vai a continuación de outra. Representase cunha frecha entre unha actividade e outra.
- **Bifurcación.** Marca a existencia de fluxos alternativos. En cada fluxo de saída indícase unha condición de garda. Representase cun rombo que ten unha transición de entrada e 2 ou máis de saída.



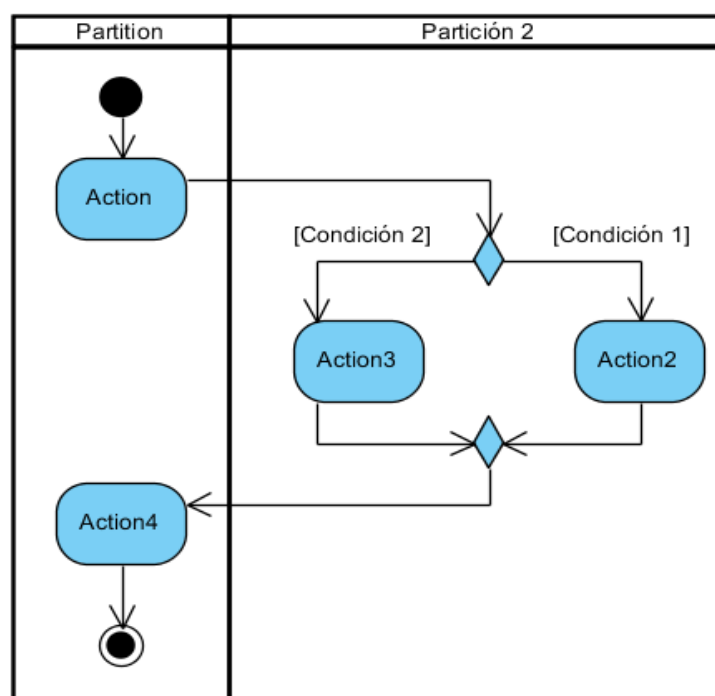
As bifurcacións poden fusionarse despois nun único fluxo. Isto pode vir representado de varias formas:



- **Divisións e unións.** Utilízanse para representar estados que se poden realizar en paralelo. Gráficamente represéntanse cunha barra de sincronización que é unha liña ancha horizontal ou vertical.



- **Particións.** Pódense facer particións nun diagrama de actividades para identificar as accións que teñen algunha característica en común. Por exemplo que se levan a cabo por un mesmo actor:



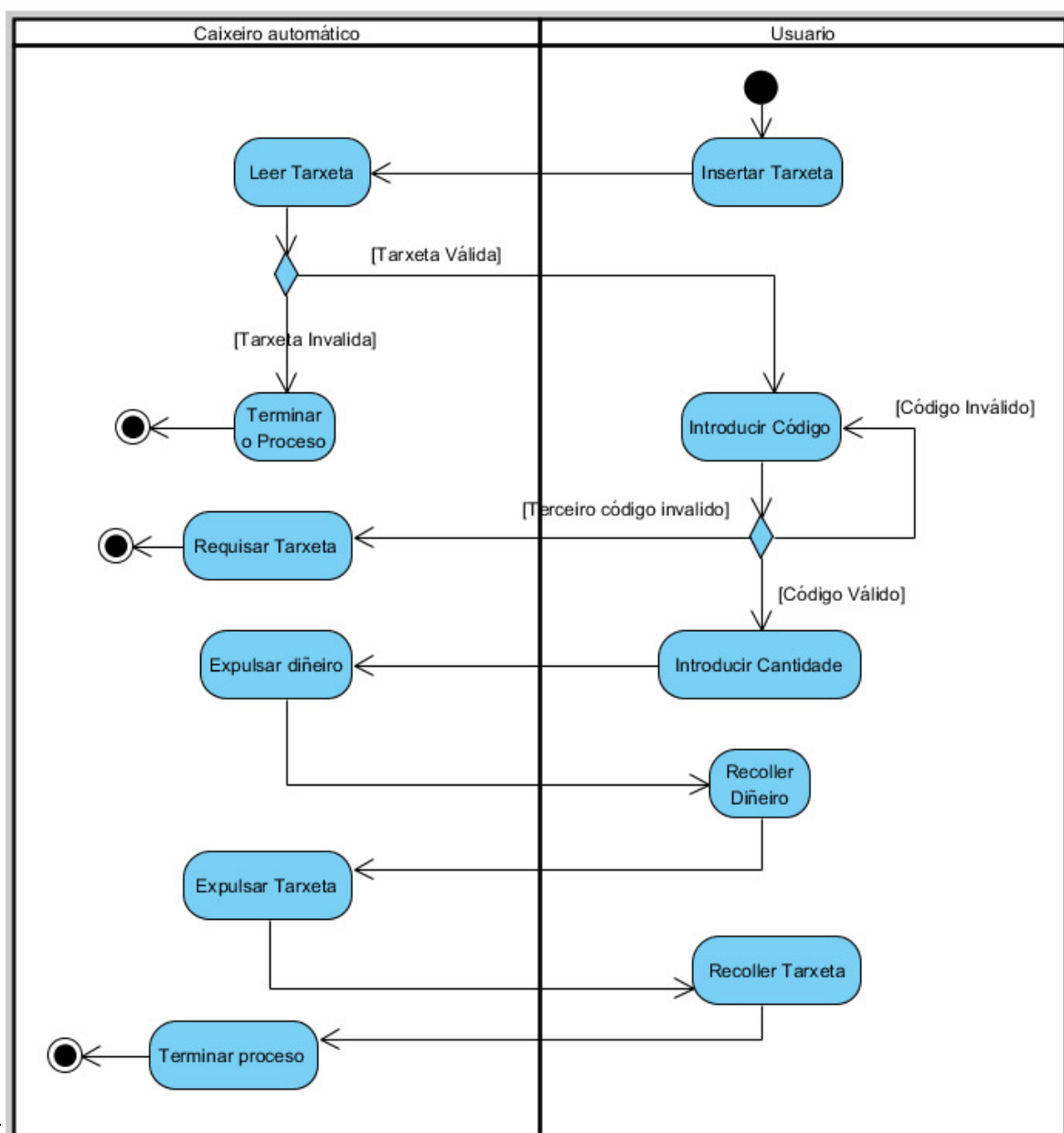


## 4.3 Recomendacións e exemplos

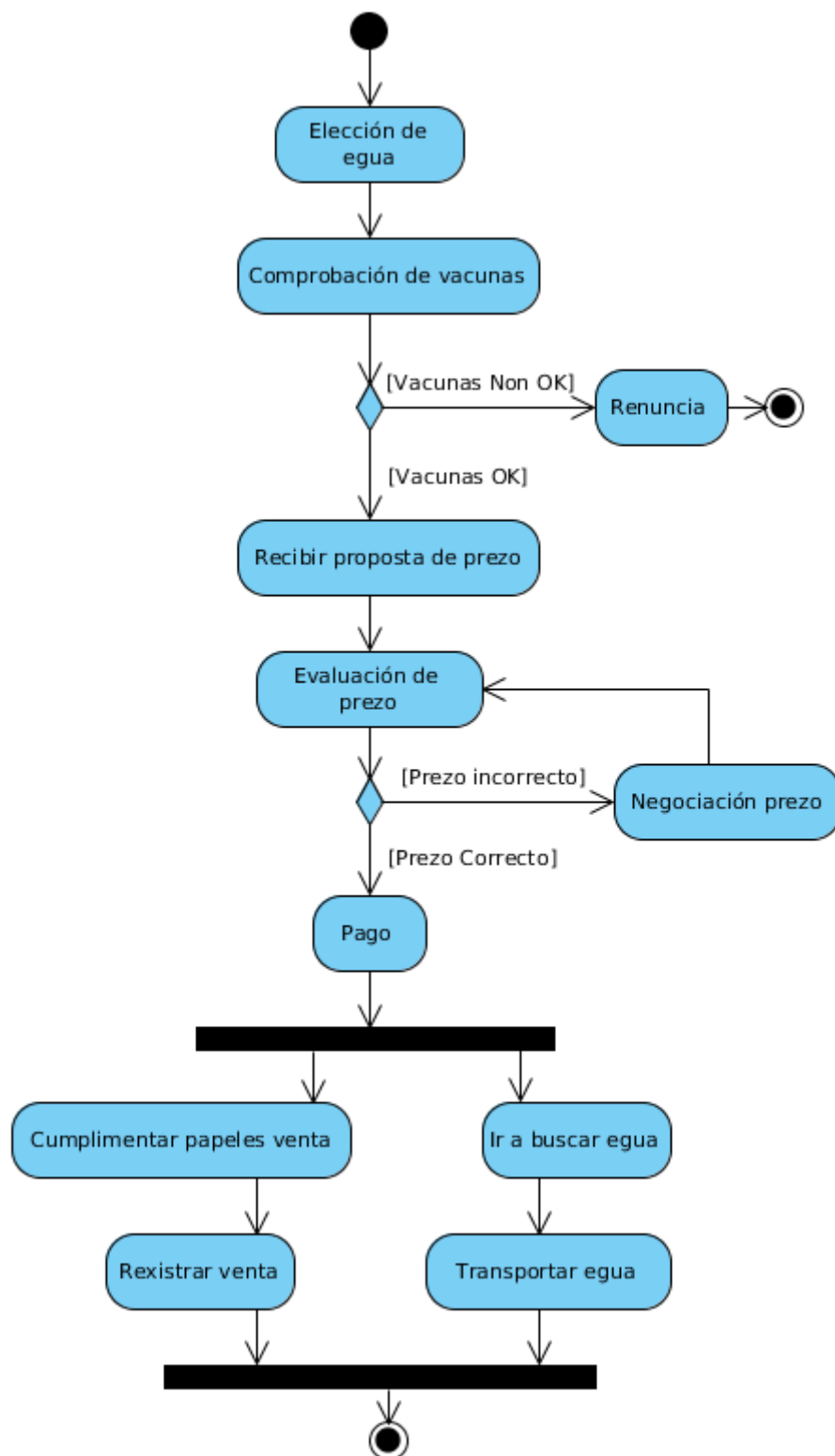
Imos ver agora unha serie de recomendacións que compre ter en conta a hora de elaborar un diagrama de actividades.

- Primeiro, tendo en conta que os diagramas de actividade soen estar asociados a unha clase, a unha operación ou a un caso de uso, débese seleccionar correctamente a operación que será representada; recoméndase seleccionar operacións importantes que poidan ser custosas repetitivas e que causen dificultades no proceso.
- En segundo lugar, determinar onde empeza e onde remata o ciclo que se quere representar.
- En terceira, observar varias veces a operación, para dividila nos seus elementos e identificalos claramente.
- Finalmente, cos datos anteriores e seguindo a secuencia de elementos, constrúese o diagrama.

Exemplo 1:



Exemplo 2<sup>19</sup>:

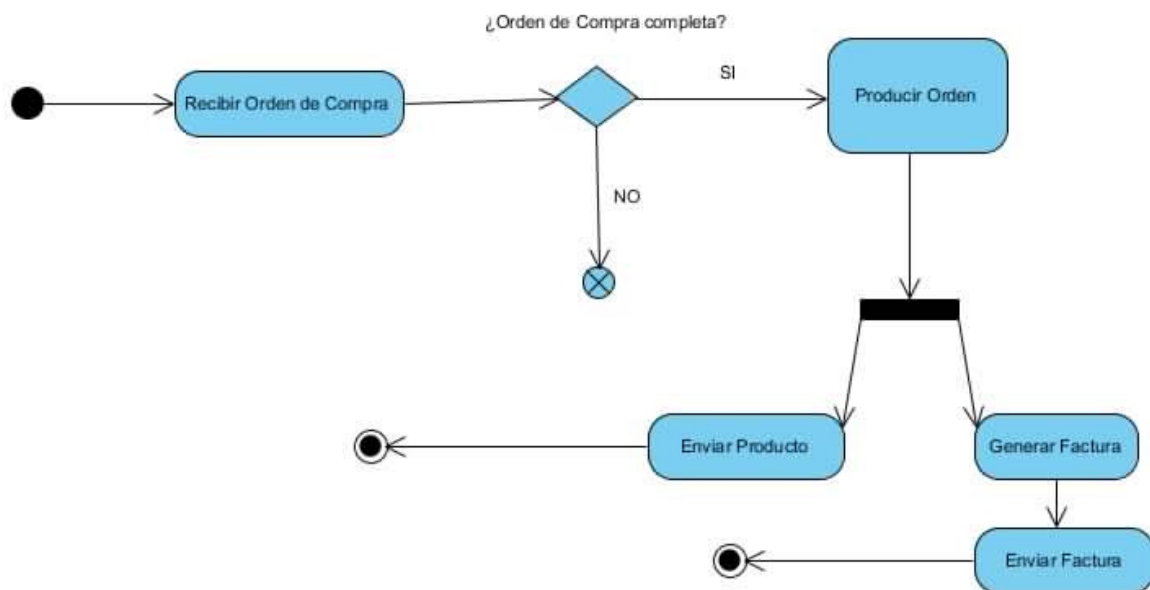


<sup>19</sup> DEBRAUWER, Laurent, VAN DER HEYDE, Fien. *UML 2: Iniciación, ejemplos y ejercicios corregidos [3ª edición]*. Ediciones ENI



### Tarefa 6.16. Interpretación dun diagrama de actividades

Dado o seguinte diagrama de actividades, redacta un posible escenario que se adapte á información recollida no diagrama:



### Tarefa 6.17. Elaboración dun diagrama de actividades

Representa a través dun diagrama de actividades o seguinte escenario<sup>20</sup>:

- Para construír un edificio, un investidor deberá contratar primeiro un arquitecto para armar os planos e logo contratar aos obreiros que traballarán na obra. Logo disto é posible comezar coa construción do edificio.
- Ao iniciar a construción todos os obreiros xuntos constrúen os alicerces. Para construír unha planta todos os obreiros xuntos colocan o piso. Logo, algúns colocan as paredes, outros as canalizacións, e outros a electricidade. Unha vez finalizadas todas estas tarefas pódese construír outra planta arriba da última planta construída (repetindo os pasos anteriores), ou simplemente poñer o teito e concluír a construción (dependendo do especificado nos planos).

Representa o modelo con Visual Paradigm.



### Tarefa 6.18. Elaboración dun diagrama de actividades

Representa a través dun diagrama de actividades o proceso que se segue nunha empresa de deseño gráfico para imprimir un folleto solicitado por un cliente:

- O cliente redacta as especificacións e as características desexadas para o folleto e as envía por correo electrónico ó deseñador gráfico da empresa.
- O deseñador realiza un bosquexo do folleto e llo envía o cliente.
- O cliente revísao e propón correccións. Este proceso repítese ata que o cliente finalmente acepta ó deseño realizado.
- O deseñador realiza entón a maqueta final para impresión.
- O responsable de impresión recibe a maqueta final e imprime os folletos.