

Guías para comunicar un diseño Casos de uso

**por
*Fernando Dodino***

**Versión 1.3
Agosto 2020**

Índice

[1 Diagrama de Casos de Uso](#)

[Elementos](#)

[1.3 Observaciones sobre las relaciones includes y extends](#)

[2 Ejemplo preliminar](#)

[3 Ejemplos](#)

[3.1 Televisor de Alta Definición de 29"](#)

[3.2 Una posible solución:](#)

[3.3 Salón de Videojuegos](#)

[3.4 Una posible solución:](#)

[4 Resumen](#)

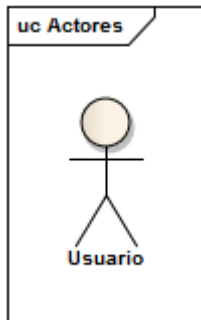
[5 Especificación de un caso de uso](#)

[5.1 Ejemplo Salón de Videojuegos](#)

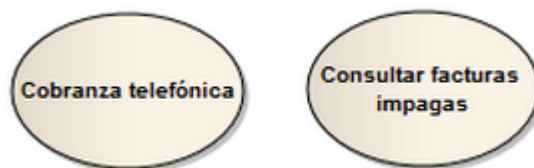
1 Diagrama de Casos de Uso

Este diagrama nos permite definir **qué es lo que el sistema debe hacer**. Surge del relevamiento inicial con el usuario y se va refinando sucesivamente para llegar al desarrollo de cada funcionalidad.

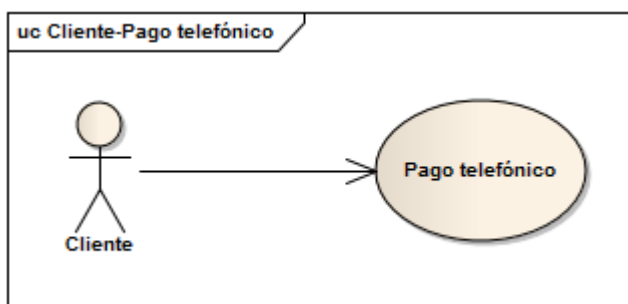
Elementos



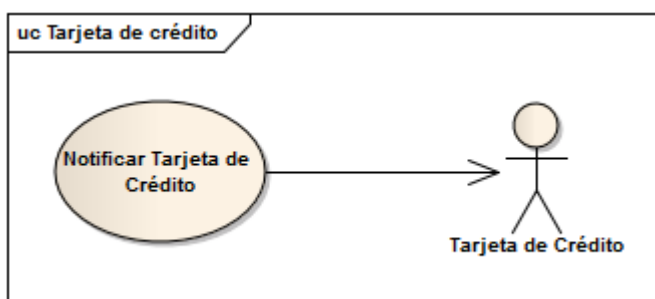
Actores: representan **roles o papeles** que cumplen usuarios o sistemas externos. **Son los que solicitan que el sistema haga determinadas cosas**. Una persona física puede cumplir varios roles a la vez. Por ej.: un usuario puede pedir reportes estadísticos al sistema, ejecutar tareas de administración y de seguridad; cada una de esas tareas responde a perfiles diferentes, **serán entonces 3 actores distintos por más que quien haga los requerimientos sea la misma persona física**.



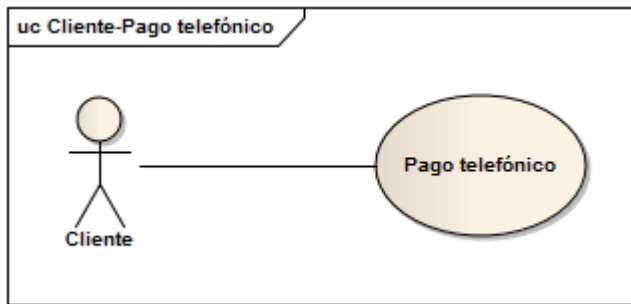
Casos de uso: representan **requerimientos funcionales**, descritos desde el punto de vista del usuario. El nombre puede ser un **verbo o un sustantivo**, no es necesario ponernos estrictos en este aspecto; lo que sí es importante es que con dicho nombre **el usuario final entienda claramente qué es lo que el caso de uso resuelve**.



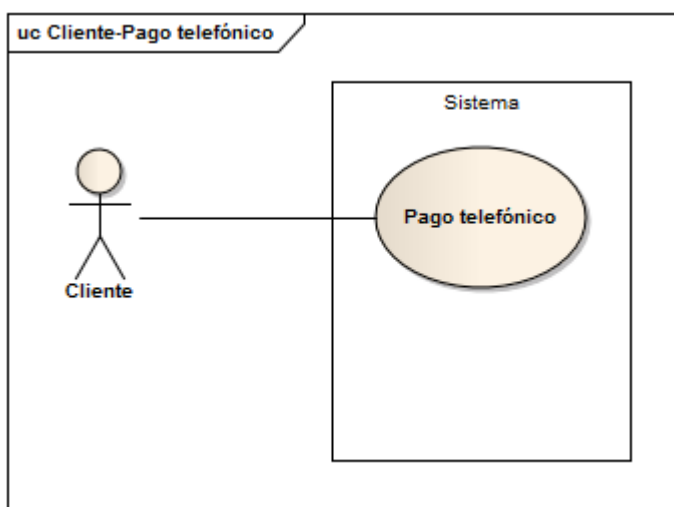
Asociaciones: hay una asociación entre un actor y un caso de uso cuando un actor se comunica con el sistema **participando en ese caso de uso**. Si la flecha de navegación va desde el actor hacia el caso de uso indica que quien **inicia el caso de uso** es dicho actor.



Cuando la flecha tiene la dirección contraria lo que se indica es que el actor toma el resultado del caso de uso ejecutado.

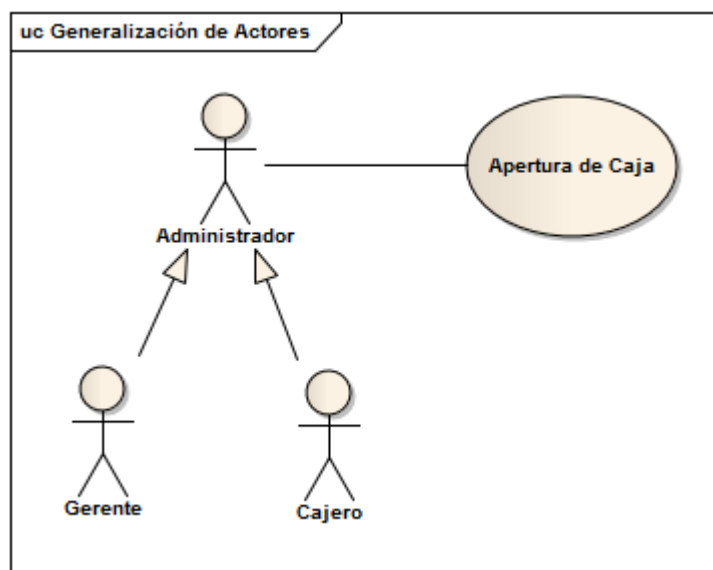


Las flechas son opcionales, podemos dejar la asociación entre actor y caso de uso como se ve en el dibujo de la izquierda. **En ese caso no estamos especificando quién inicia la interacción** (si el sistema a través de otro actor o el cliente). Es tarea del analista funcional decidir el grado de detalle del diagrama ("Sólo documentar lo que queremos comunicar").



Límite del sistema (opcional): los casos de uso se los suele ubicar en un contorno que delimita el conjunto de requerimientos que forman parte del alcance del sistema

Generalización de actores: cuando varios actores tienen funcionalidades en común, es posible abstraer un actor más genérico:



En el ejemplo de arriba, **Gerente y Cajero son actores específicos mientras que Administrador es un actor más general.** Tanto el Gerente como el Cajero podrían representar al administrador de la aplicación de Cobranzas, aunque seguramente el gerente y el cajero se diferenciarán en los casos de uso en los que intervienen.

1.3 Observaciones sobre las relaciones includes y extends

Como advierten Steve Adolph y Paul Bramble en el recomendable libro “Patterns for Effective Use Cases”:

“It’s easy to start a riot at a use case conference. Simply walk up to an open microphone and ask, “What is the difference between includes and extends?” and then run for cover! How to effectively apply the includes and extends relationships in use cases is a question that has perplexed both beginning and experienced use case writers.

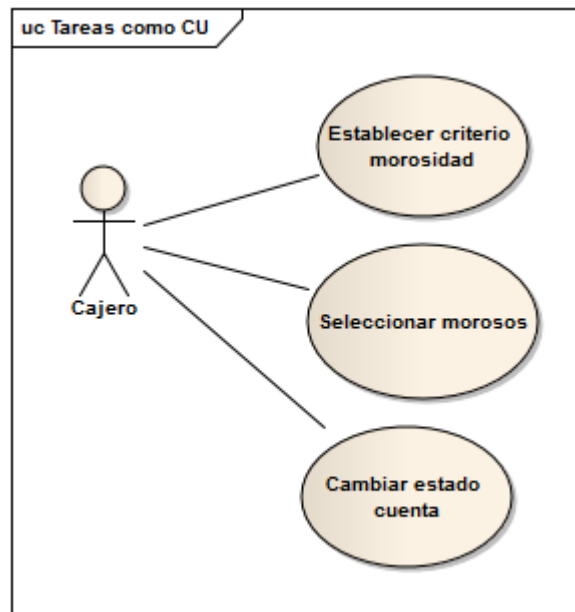
“Includes and extends were originally intended to simplify the use case model by collecting common threads in the use case model. Unfortunately, these simplifying relationships seem to have the opposite effect, making the model more complex and difficult to understand. Many factors contribute to this problem. First, the definitions of includes and extends are ambiguous, leading to a wide variety of interpretations as to what they really mean. Second, software professionals tend to favor formality in the use case model over usability, creating models that follow all the rules, but are hard to read. Finally, instructors tend to over emphasize these techniques when teaching people to write use cases.”

En definitiva, **no recomendamos utilizar las relaciones extends ni includes para “reutilizar” definiciones de análisis, solo complican el entendimiento de este diagrama.**

2 Ejemplo preliminar

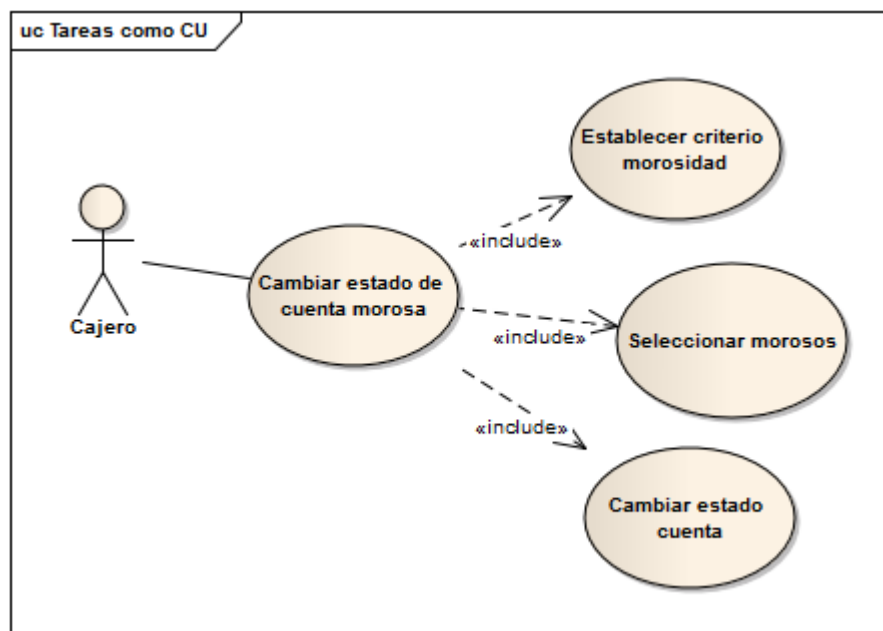
“En el proceso de morosidad interviene el analista de cuentas, quien selecciona los clientes a partir de un criterio (por lo general, el rango mínimo de deuda). El sistema devuelve la lista de clientes morosos encontrados y el analista finalmente puede modificar el estado de cada una de las cuentas”.

¿Cuál es el diagrama de casos de uso que podemos representar? Una tentación al comenzar a armar diagramas de caso de uso es asociar cada tarea como un caso de uso, de la siguiente manera:



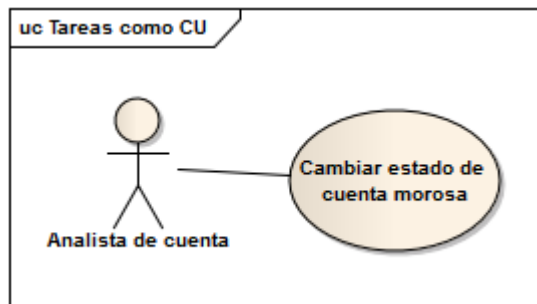
Ejemplo incorrecto 1: Tareas documentadas como casos de uso

o bien



Ejemplo incorrecto 2: Tareas documentadas como casos de uso

No obstante, el diagrama de casos de uso *no es* un diagrama de flujo. Lo que hay que mostrar son las funcionalidades que le dan valor agregado al usuario. El mismo ejemplo de arriba está especificando un solo caso de uso:



De otra manera el diagrama se llenaría de infinitos “casos de uso” que corresponderían en realidad a interacciones entre el usuario y el sistema y que se pueden analizar con el diagrama de Actividades, como veremos más adelante.

Para más información puede verse [Examples of Use Case Antipatterns](#).

3 Ejemplos

3.1 Televisor de Alta Definición de 29"

(idea extractada de www.umliranch.com/lu2-use-cases-exercises/)



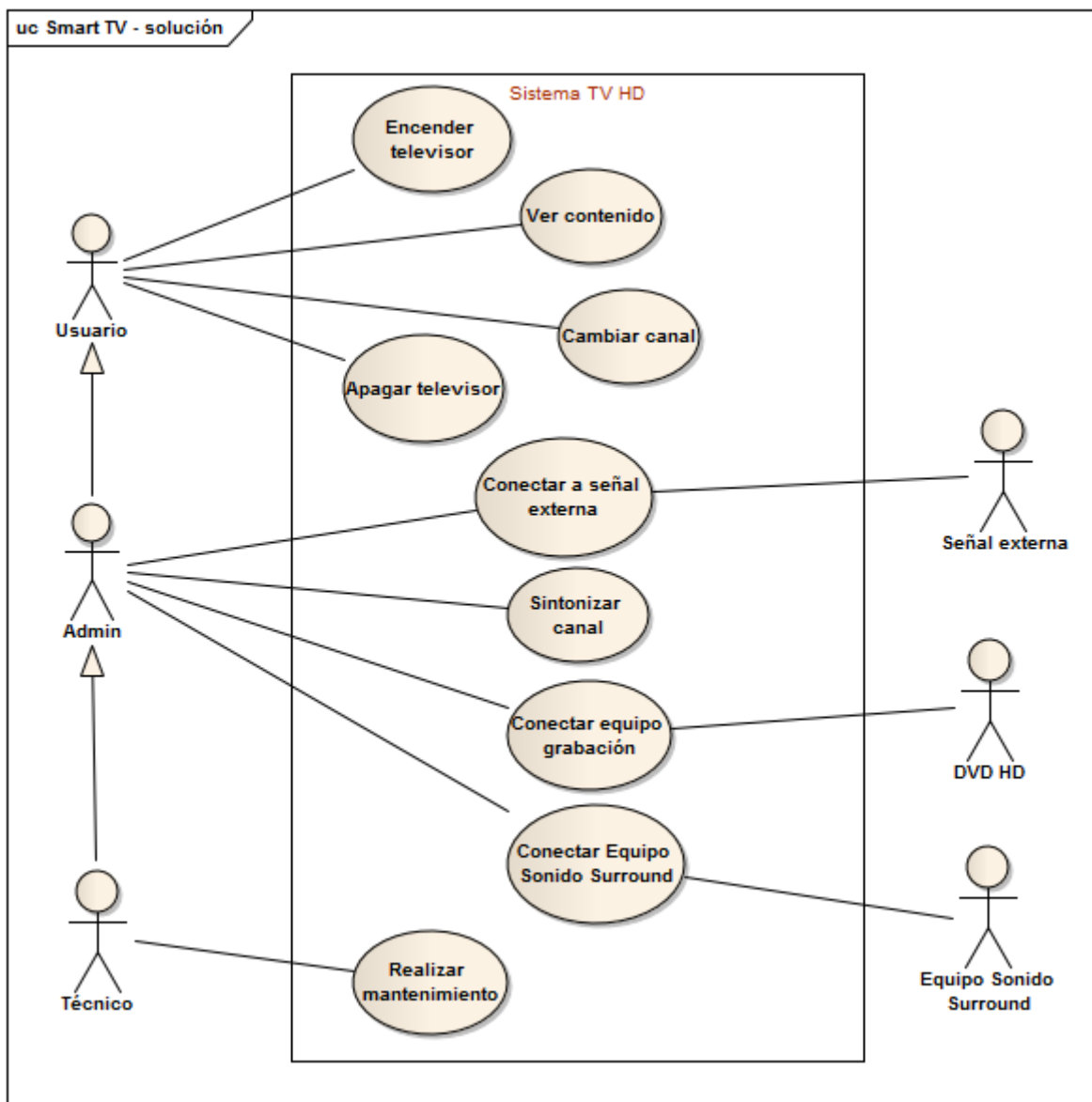
Tomando en cuenta los siguientes requerimientos para un televisor de alta definición de 29", genere el diagrama de casos de uso correspondiente:

1. El usuario podrá encender el sistema presionando el botón del televisor o a través de un control remoto.
2. El usuario podrá cambiar de canal presionando un botón del televisor o a través de un control remoto.
3. El usuario podrá seleccionar cualquier canal disponible presionando un botón o a través de un control remoto.
4. El televisor podrá conectarse tanto a la señal interna que viene de la antena del aparato de TV como a una señal externa.
5. El televisor debe aceptar las siguientes resoluciones de pantalla:
 - 1920 x 1080
 - 1280 x 720
 - 852 x 480
6. El televisor debe poder conectarse a un sistema de sonido Surround.
7. El televisor debe proveer una salida de audio-video para un sistema de grabación, como por ejemplo un DVD de alta definición.
8. El televisor debe superar las 10.000 horas de uso antes de necesitar mantenimiento de un técnico.
9. El televisor debe estar en el mercado para el año 2015 (cuando estén disponibles contenidos en formato "alta definición").
10. El sistema deberá respetar las disposiciones legales que regulan los Derechos de Propiedad Intelectual de los contenidos a visualizar.

Antes de comenzar a bosquejar una solución, tengamos en cuenta que sólo los requerimientos funcionales deben modelarse en un diagrama de casos de uso. Las preguntas que nos pueden ayudar a encontrar una solución son:

- ¿Qué actores intervienen?
- ¿Es posible establecer una generalización para dichos actores?
- ¿Cuáles son las funcionalidades básicas que aparecen?
- ¿Cuáles son las funcionalidades derivadas o especializaciones que surgen de las funcionalidades básicas?
- ¿Qué casos de uso pueden ser incluidos en otros casos de uso?

3.2 Una posible solución:



La aparición de un actor Admin nos permite introducir dos perfiles distintos: por un lado el que accede a las funcionalidades básicas del televisor y por otro el que conecta el televisor al

equipo de audio, al DVD, a la antena del cable, etc. Por último el técnico agrega a las funcionalidades anteriores la posibilidad de realizar el mantenimiento del aparato, según indica el requerimiento n° 8.

Todos los requisitos no funcionales (año de salida al mercado, resoluciones máximas y mínimas, cantidad de horas que deben pasar hasta que se requiera mantenimiento, etc.) no están ni deben estar en el diagrama de casos de uso.

3.3 Salón de Videojuegos



Se trata del típico salón con juegos de video, tejo, pool, bowling, etc. El cliente accede a los juegos mediante una tarjeta que adquiere en las cajas (cuyo valor es de \$ 20) y carga en dicha tarjeta el monto que él desea.

Una vez adquirida, la tarjeta se puede recargar todas las veces que el cliente quiera. Si la tarjeta está defectuosa (siempre que no esté rota), el supervisor del salón puede reemplazarla por una nueva sin costo alguno, transfiriendo el saldo de la tarjeta original.

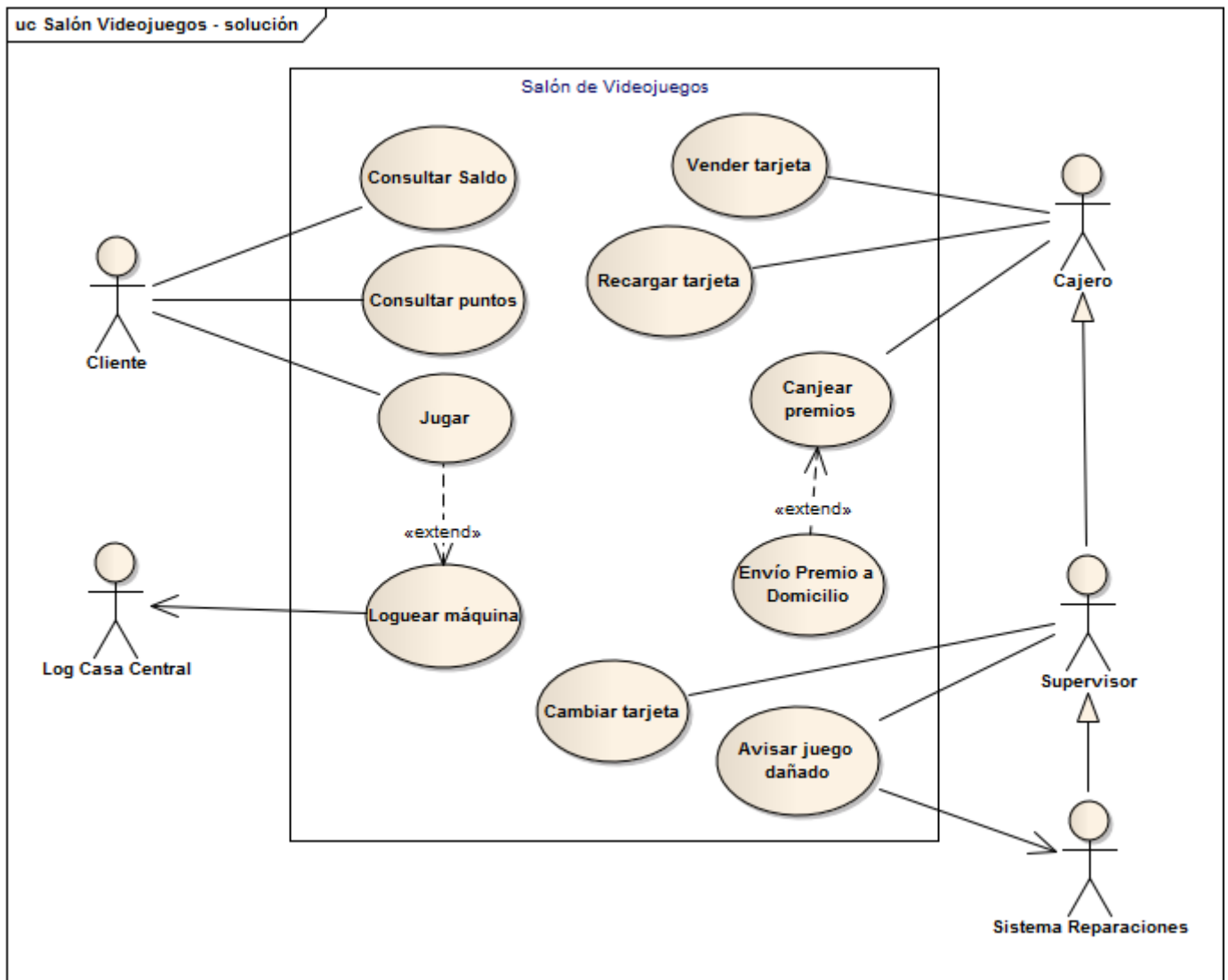
Cada vez que un cliente juega en una máquina, debe pasar la tarjeta. Entonces se le descuenta el monto de cada juego y se notifica a casa central que va llevando las estadísticas de uso de cada juego por sucursal.

Algunas máquinas entregan puntos que se van acumulando en la tarjeta en base al desempeño del cliente en el juego. En la caja se puede canjear los puntos obtenidos por premios reales. En algunos casos, no obstante, el cliente puede solicitar algún premio que está en el catálogo pero no en la sucursal. Entonces se le pide al cliente un domicilio de entrega y en el plazo de las 72 horas se envía el premio a dicho domicilio.

Hay 3 puestos de auto-consulta, donde el cliente puede averiguar los puntos obtenidos o el saldo pendiente de su tarjeta.

Cuando un juego se descompone, los supervisores del salón deben avisar al servicio Técnico de Reparaciones, que está tercerizado.

3.4 Una posible solución:



Como alternativa podríamos incorporar la consulta del Catálogo de Premios, que podría estar en el puesto de Auto-Consulta (en ese caso quien inicia el caso de uso es el Cliente) o bien en la Caja (en ese caso quien inicia el caso de uso es el Cajero). Lo importante para el analista funcional es detectar funcionalidades posibles y consultar con el usuario para llegar al conjunto de requerimientos mínimos que hay que satisfacer para salir a producción.

4 Resumen

¿Quién hace el diagrama de casos de uso? El analista funcional, en base a los relevamientos hechos con el usuario final.

¿Para quién es el diagrama de casos de uso? Claramente, para el usuario final, ya que se capturan todos los requerimientos funcionales que él pide y se dejan registrados como una especie de contrato de lo que el sistema incluye y lo que no. Para el usuario final cada caso de uso es una “caja negra”, sin entrar en detalles internos de la solución.

Para los analistas funcionales y los técnicos, el diagrama de casos de uso permite una visión global del sistema, supone una primera incursión en el dominio del usuario y en muchas metodologías como UP, dirige el proceso de desarrollo en sí. De hecho, parte de la negociación entre el usuario y el analista funcional en una metodología de trabajo iterativa es definir las fechas de entrega para cada uno de los casos de uso.

¿Cuándo queremos hacer un diagrama de casos de uso? Siempre que comenzamos el desarrollo de un software y que debemos acordar con el usuario las funcionalidades de un sistema.

5 Especificación de un caso de uso

Aunque no hay un estándar UML para especificar un caso de uso, la mayoría de los modelos para documentar en detalle cada caso de uso tiene un formato similar al siguiente:

- **Nombre del caso de uso:** la misma descripción que fue dada en el diagrama de casos de uso general. Como comentábamos anteriormente, el caso de uso puede comenzar con verbos o sustantivos, lo que no resulta recomendable es utilizar códigos difíciles de memorizar ("VEN001", "COB-RX-01").
- **Objetivo o Propósito:** representa el resultado observable y valioso para el usuario. Cada **caso de uso debe tener un solo objetivo** (concepto de diseño relacionado: cohesión).
- **Prioridad (opcional):** se puede definir como:
 - Crítica: el caso de uso debe estar para salir a producción (forma parte de la operatoria básica del usuario)
 - Necesaria: la funcionalidad es importante pero puede implementarse en una fase posterior.
 - Deseable: ...
 - etc.
- **Pre-Condiciones:** no son las causas que disparan el caso de uso (que en todo caso pueden definirse en la sección *Trigger*), sino el estado en el que las cosas deben estar para que el caso de uso se ejecute satisfactoriamente. Aquí se pueden anotar una lista de condiciones a cumplir o bien de casos de uso a ejecutar previamente.
- **Escenario:** se define una serie de pasos para completar la ejecución normal de un caso de uso: debe quedar claro qué actor es responsable de resolver cada uno de los pasos.

paso 1: El Actor X realizará la operación Y.

paso 2: El Actor Z realizará la operación T.

... etc ...

Cada paso puede derivar en flujos alternativos o excepciones.

- **Escenarios alternativos:** aquí se define una serie de pasos alternativos al caso de uso original, tanto por extensiones al caso de uso base como en el caso de las excepciones.
- **Post-Condiciones:** es el estado en el que debe quedar el sistema tras la ejecución exitosa del caso de uso. Es algo tan simple como el cumplimiento del objetivo del caso de uso.

Según la bibliografía, podremos encontrar estos puntos opcionales para definir un caso de uso:

- **Versión del documento,** que corresponde al número de iteración por la cual atraviesa.
- **Tiempo** en que tarda en completarse el caso de uso
- **Frecuencia:** cada cuánto se ejecutará el caso de uso
- **Actores:** quienes intervienen en el caso de uso. Se dividen en primarios (aquellos que inician el caso de uso) y secundarios (quienes reciben la respuesta del sistema como resultado de la ejecución del caso de uso). En realidad surge de la documentación del diagrama de casos de uso original, por lo que su especificación sólo es para que el lector no deba revisar dicho diagrama.

- **Trigger:** cuál es la acción que dispara el caso de uso.
- **Issues:** representan los temas que quedaron sin resolver o pendientes de revisión.
- **Notas adicionales:** cualquier otra documentación adicional que no encaje en los puntos anteriores puede utilizarse aquí.

5.1 Ejemplo Salón de Videojuegos

Tomaremos el ejemplo anterior para especificar los siguientes casos de uso:

- Recargar tarjeta
- Canjear premios (como ejemplo para documentar la relación “extends”)
- Jugar (como ejemplo para documentar la relación “includes”)

Revisando los puntos arriba descriptos, podemos detallar los 3 casos de uso como sigue:

Nombre	Recargar tarjeta
Objetivo	Actualizar el saldo de la tarjeta en base al monto pagado por el usuario en caja
Pre-condición	El cliente debe tener una tarjeta en buen estado.
Escenario	El cajero pasa la tarjeta por la lectora. El cliente le entrega al cajero el dinero. El cajero informa al sistema el monto a recargar en la tarjeta.
Post-condición	Se actualizó el saldo de la tarjeta

Nombre	Canjear premios
Objetivo	Canjear los puntos obtenidos en juego por premios
Pre-condición	El cliente debe tener una tarjeta en buen estado y puntos en premios acumulados
Escenario	El cajero pasa la tarjeta por la lectora y le informa al cliente la cantidad de puntos acumulados. El cliente selecciona el premio Flujo alternativo: el cliente selecciona un premio que no está en el catálogo. “Envío Premio a Domicilio”
Post-condición	El cajero entrega el premio al cliente y actualiza la información de la cantidad de puntos que el cliente tiene en su tarjeta.

Nombre	Envío Premio a Domicilio
Objetivo	Enviar a un cliente uno o varios premios que no se encuentren en una sucursal
Pre-condición	El cliente debe tener una tarjeta en buen estado y puntos en premios acumulados. El premio que solicita el cliente no debe estar disponible en la sucursal.
Escenario	El cliente toma los datos del domicilio al cliente. Se despacha la orden de entrega del premio
Post-condición	El cajero actualiza la información de la cantidad de puntos que el cliente tiene en su tarjeta. En el transcurso de las 72 hs. se entrega el premio al cliente.

Nombre	Jugar
Objetivo	Permitir que el cliente juegue en una máquina

Pre-condición	El cliente debe tener una tarjeta en buen estado y saldo en su tarjeta mayor al saldo del juego.
Escenario	El cliente pasa la tarjeta por el dispositivo electrónico del juego. El dispositivo electrónico actualiza el saldo disponible de la tarjeta y activa el juego.
Post-condición	El cliente juega en la máquina.

Nombre	Loguear máquina
Objetivo	Registrar auditoría estadística sobre un juego
Pre-condición	La ejecución del caso de uso "Jugar"
Escenario	La máquina envía a la casa central un registro con información sobre un juego (fecha y hora, juego, cantidad de usuarios simultáneos que jugaron, etc.)
Post-condición	Se registra auditoría en la Casa Central.