

Universidad de Buenos Aires  
Facultad de  
Ciencias Exactas y Naturales  
Departamento de Computación

Ingeniería de Software I  
Primer Cuatrimestre de 2012

## Trabajo práctico 2

Modelos de comportamiento del sistema de software  
para CentralMarket

### Grupo 1

| Integrante        | LU     | Correo electrónico          |
|-------------------|--------|-----------------------------|
| Abregú, Angel     | 082/09 | angelj_a@hotmail.com        |
| Cammi, Martín     | 676/02 | martincammi@gmail.com       |
| De Sousa, Mariano | 389/08 | marian_sabianaa@hotmail.com |
| Méndez, Gonzálo   | 843/04 | gemm83@hotmail.com          |
| Raffo, Diego      | 423/08 | enanodr@hotmail.com         |

# Índice

# 1. Propuesta de servicios

## 1.1. Introducción

Tras haber realizado una serie de iteraciones con los directivos de CentralMarket sobre la última presentación del sistema y en base a las correcciones y ajustes que nos han comentado quieren, hemos armado una nueva serie de documentos para detallar mejor el comportamiento del sistema. A tal efecto hemos revisado una serie de puntos con los directivos y se han modificado algunos:

- Los agentes *Gerente de proyecto* y *Desarrollador* que figuraban en el diagrama de objetivos de la documentación revisada con el cliente no serán tenidos en cuenta en esta etapa ya que forman parte intrínseca de la construcción del mismo y no de su comportamiento.
- El agente *Cliente* será modelado en esta etapa a través del usuario quien es el que interactúa directamente con el sistema)
- De las diferentes opciones propuestas para el tipo de descarga (no streaming) los directivos han optado por la descarga directa.
- Del diagrama de objetivos, la rama 3.1 (Soportar TV, PC, tablet y mobile) no será modelada en esta etapa ya que un equipo especial de CentralMarket se ocupará de definir esas necesidades.
- Del mismo modo la Situación 6 mencionada en la documentación anterior sobre la *Actualización de la interfaz de usuario* no será incluida en esta nueva presentación ya que CentralMarket se ocupará también en definir
- En el modelado a continuación asumiremos que todos los actores que interactuen con el sistema ya tienen creada una cuenta previa para poder loguearse y realizar sus acciones
- Consultamos a la analista de CentralMarket al respecto de los tipos de publicidad y tipos de posicionamiento y nos comentó que dejáramos la definición de estos tipos a ellos con lo cual asumiremos que ya están definidos en el sistema
- La analista también hizo énfasis en que es importante que el Administrador de contenidos tenga una forma de "poner online" los contenidos que ya han sido verificados, es por eso que le hemos otorgado la opción de "Habilitar" los contenidos cuando quiere ponerlos online
- Los directivos nos mencionaron que quieren poder ver claramente cual es el flujo de validación de los contenidos, para asegurarse de brindar la mejor calidad aunque para los contenidos en si quieren poder asegurar no llegar a generar inconsistencias en el sistema. Es por eso que decidimos presentarles todo lo referente a revisión de contenidos en un *Casos de Uso* y un *Diagramas de actividad* y con relación a los contenidos en si agregar también un *Modelo Conceptual*

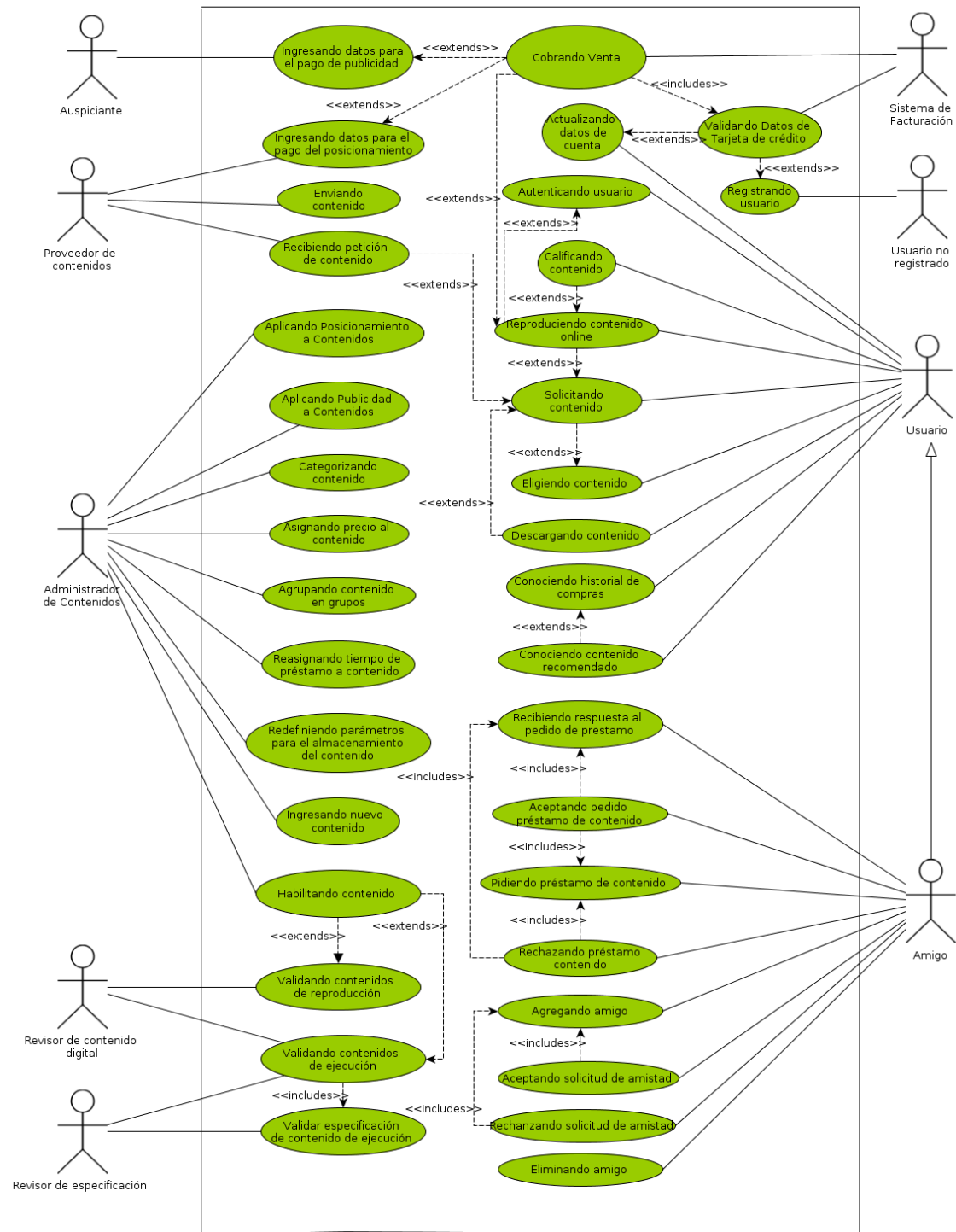
- Al consultar sobre los paquetes los directivos nos comentaron que en una primera instancia no implementarían paquetes, quizás más adelante, así que no quieren demasiada información sobre ellos, decidimos no agregarlos en el *Modelo conceptual* para simplificar la descripción de los contenidos ya que en un futuro puede extenderse.

## 2. Diagramas

A continuación presentamos una serie de modelos que representan el comportamiento acordado con los directivos de CentralMarket. Presentaremos las funcionalidades con tres tipos de diagramas: de *Casos de Uso*, de *Actividad* y *Modelo Conceptual*.

Un diagrama de *Casos de Uso* permite identificar claramente la interacción del sistema con diferentes actores. A continuación se muestran todos los actores relevantes al sistema y las acciones que pueden realizar con el sistema.

## 2.1. Diagrama de Casos De Uso



## 2.2. Descripción de los casos de uso.

---

**Caso de uso: Eligiendo contenido.****Actores:** Usuario.**Pre:** El usuario está  $\frac{1}{2}$  registrado y autenticado.**Post:** El usuario eligió  $\frac{1}{2}$  un contenido.**Caso normal**

1. El usuario entra a la búsqueda de contenido en el sistema.
2. El usuario ingresa un término en la búsqueda de acuerdo al contenido que quiera buscar y/o marca los filtros que necesite.
3. El sistema ofrece una lista básica de contenidos acorde al usuario y los filtros completados.
4. El usuario ve que el contenido a buscar forma parte de lista seleccionada y accede a al detalle.
5. El usuario decide si va a consumir el contenido, de ser así, extiende caso de uso "Solicitando contenido"
6. **Fin Caso de uso**

**Caso alternativo**

4.1 - El usuario no encuentra el contenido que está buscando y decide hacer una búsqueda diferente, va al paso 2.

---

**Caso de uso: Solicitando contenido.****Actores:** Usuario.**Pre:** El usuario está  $\frac{1}{2}$  autenticado y ha elegido un contenido.**Post:** El usuario solicitó  $\frac{1}{2}$  un contenido.**Caso normal**

1. El sistema muestra al usuario, dado el tipo de contenido, cuáles son las modalidades para consumirlo (descarga y/o streaming)
2. El usuario selecciona la modalidad que va utilizar.
3. El sistema valida que el dispositivo está soportado para iniciar el proceso de descarga o streaming.
4. Si la cuenta no tiene asociada una tarjeta, el sistema pide una, el usuario ingresa los datos.
5. Extiende Caso de Uso "Cobrando venta"
6. Si es descarga, extiende Caso de Uso "Descargando contenido"; si el contenido es soportado y es streaming extiende Caso de Uso "Reproduciendo contenido online"
7. **Fin Caso de uso**

**Caso alternativo**

- 3.1 - El sistema muestra mensaje de error indicando que el dispositivo no está soportado. Ir a 7.
- 4.1 - Si el contenido es gratuito, ir a 6.
- 6.1 - De no ser válida la tarjeta, el sistema muestra mensaje de error y vuelve a 4.
- 6.2 - De no ser exitoso el cobro, el sistema muestra mensaje de error.



---

**Caso de uso: Descargando contenido**

**Actores:** Usuario.

**Pre:** dispositivo soportado, usuario autenticado, usuario eligió modalidad de descarga.

**Post:** El contenido fue descargado.

**Caso normal**

1. El sistema actualiza el estado de la cuenta, marca a la cuenta como descargando el contenido
2. El sistema envía un paquete de datos (del archivo) al usuario, por el canal de comunicación.  
3 - Continúa con el siguiente paquete cuando este se envió exitosamente. Repite el paso 2, hasta enviar todos los paquetes.
3. El sistema avisa al usuario que el contenido se descargó exitosamente.
4. El sistema actualiza en la cuenta que se descargó el contenido (ya no está marcada como *descargando el contenido*)
5. **Fin Caso de Uso.**

**Caso alternativo**

1.1 En caso de que en el estado de la cuenta haya algún contenido descargándose ó un contenido distinto en reproducción ó pausado, el sistema muestra un mensaje de error, y vuelve a la pantalla de selección de contenido.

3.1 En caso de que un paquete no se haya logrado enviar por algún error, el sistema reintenta el envío del mismo paquete antes de continuar con el siguiente (i.e. vuelve al paso 2)

---

**Caso de uso: Reproduciendo contenido online****Actores:** Usuario.**Pre:** Dispositivo soportado, usuario autenticado, usuario eligi $\frac{1}{2}$  modalidad streaming .**Post:** Se reprodujo el contenido online (finaliz $\frac{1}{2}$  con un stop) .**Caso normal**

1. El sistema carga el estado de la cuenta
2. Si no habi $\frac{1}{2}$ a contenido en reproduc $\frac{1}{2}$ n, dej $\frac{1}{2}$  asentado en el estado de la cuenta que el i $\frac{1}{2}$ ltimo estado es el inicio, y contini $\frac{1}{2}$ a con el paso 4. Si el contenido estaba pausado, el sistema muestra un mensaje al usuario, preguntando si desea continuar la reproduc $\frac{1}{2}$ n desde donde se lo dej $\frac{1}{2}$  o no.
3. El usuario acepta reproducir desde el estado en que se dej $\frac{1}{2}$ .
4. El sistema marca el estado de la cuenta como *reproduciendo contenido* y empieza a reproducir desde el i $\frac{1}{2}$ ltimo estado.
5. El usuario termina de ver el contenido (o detiene la reproduc $\frac{1}{2}$ n)
6. El sistema marca en el estado de la cuenta *no hay contenido reproduc $\frac{1}{2}$ ndose/descarg $\frac{1}{2}$ ndose*
7. En caso de que el usuario decida calificar el contenido, extiende Caso de Uso *Calificando contenido*
8. **Fin Caso de Uso.**

**Caso alternativo**

2.1 En caso de que en el estado de la cuenta haya alg $\frac{1}{2}$ n contenido descarg $\frac{1}{2}$ ndose i $\frac{1}{2}$  un contenido distinto en reproduc $\frac{1}{2}$ n o pausado, el sistema muestra un mensaje de error, y vuelve a la pantalla de selecc $\frac{1}{2}$ n de contenido 3.1 El usuario cancela, y vuelve a la pantalla de selecc $\frac{1}{2}$ n de contenido 6.1 El usuario pausa la reproduc $\frac{1}{2}$ n del contenido y el sistema marca en el estado de la cuenta como *contenido en pausa*, y sigue en paso 8.

---

**Caso de uso: Calificando contenido****Actores:** Usuario.**Pre:** Usuario autenticado, usuario reprodujo el contenido, usuario no califico  $\frac{1}{2}$  el contenido anteriormente.**Post:** El contenido recibio  $\frac{1}{2}$  calificacio  $\frac{1}{2}$ n del usuario .**Caso normal**

1. El usuario entra al sistema de calificacio  $\frac{1}{2}$ n
  2. El sistema muestra el detalle del contenido anteriormente reproducido y permite que el usuario ingrese un valor del 1 al 10 (10 equivale a excelente, 1 a pñ  $\frac{1}{2}$ simo)
  3. El usuario ingresa la calificacio  $\frac{1}{2}$ n y acepta.
  4. El sistema guarda la calificacio  $\frac{1}{2}$ n dada para el contenido.
  5. El sistema actualiza la lista de recomendados acorde al nuevo registro de calificaciones (Explicado en el diagrama de Clases Conceptuales)
  6. **Fin Caso de Uso.**
-

---

**Caso de uso: Conociendo historial de compras**

**Actores:** Usuario.

**Pre:** Usuario autenticado.

**Post:** Usuario conoce el historial de compras realizadas.

**Caso normal**

1. El usuario entra al historial de compras
2. El sistema muestra el detalle del historial de compras y una solapa con contenido recomendado
3. El usuario navega por el historial, pudiendo ver el detalle de cada una de ellas. En cualquier momento puede decidir ver el contenido recomendado cliqueando en la solapa (extiende caso de uso *Conociendo contenido recomendado*)
4. El usuario cierra el historial
5. El sistema cierra la pantalla del historial
6. **Fin Caso de Uso.**

---

**Caso de uso: Conociendo contenido recomendado**

**Actores:** Usuario .

**Pre:** usuario autenticado.

**Post:** usuario conoce el contenido recomendado por el sistema.

**Caso normal**

Trivial

---

---

**Caso de uso: Registrando Usuario****Actores:** Usuario no registrado.**Pre:** -**Post:** El Usuario se registra.**Caso normal**

1. El sistema pide datos para la registraci3n de la cuenta (email y contrase1a, informaci3n de personal (opcional), datos de tarjeta de cr3dito (opcional)
2. El usuario ingresa los datos pedidos
3. El sistema valida si el email ya existe
4. El sistema verifica que el email no existe, se graba el email y la contrase1a
5. Si el usuario puso datos de la tarjeta de cr3dito, extiende CU Validando datos de tarjeta de cr3dito
6. Se crea una cuenta a nombre del nuevo usuario con sus datos.
7. **Fin Caso de Uso.**

**Caso alternativo**

3.1 - El sistema verifica que el email ya existe, vuelve al punto 1

6.1 - El sistema no puede validar la tarjeta de cr3dito, vuelve al punto 1

---

---

**Caso de uso: Validando datos de tarjeta de crédito****Actores:** Sistema de Facturación  $\frac{1}{2}$ n.**Pre:** El sistema de facturación  $\frac{1}{2}$ n recibió  $\frac{1}{2}$  un pedido de validación  $\frac{1}{2}$ n.**Post:** El sistema de facturación  $\frac{1}{2}$ n realizó  $\frac{1}{2}$  la validación  $\frac{1}{2}$ n.**Caso normal**

1. El sistema le envía  $\frac{1}{2}$  al sistema de facturación  $\frac{1}{2}$ n los datos de la tarjeta, para validar si son correctos
  2. El sistema de facturación  $\frac{1}{2}$ n los valida y emite respuesta acorde al resultado de la validación  $\frac{1}{2}$ n.
  3. **Fin Caso de Uso.**
-

---

**Caso de uso: Autenticando Usuario**

**Actores:** Usuario.

**Pre:** El usuario se haya registrado previamente.

**Post:** El usuario logró  $\frac{1}{2}$  autenticarse exitosamente.

**Caso normal**

1. El sistema pide email y contraseña  $\frac{1}{2}$ a
2. El usuario ingresa email y contraseña  $\frac{1}{2}$ a
3. El sistema valida si el email ya existe en su base de datos
4. El sistema verifica si la contraseña  $\frac{1}{2}$ a ingresada es igual a la anteriormente grabada
5. El sistema autentica al usuario
6. El sistema carga la pantalla principal y libera las funcionalidades del usuario.
7. **Fin Caso de Uso.**

**Caso alternativo**

---

---

**Caso de uso: Autenticando Usuario**

**Actores:** Usuario.

**Pre:** El usuario se haya registrado previamente.

**Post:** El usuario logró  $i_{\frac{1}{2}}$  autenticarse exitosamente.

**Caso normal**

1. El sistema pide email y contraseña  $i_{\frac{1}{2}}$ a
2. El usuario ingresa email y contraseña  $i_{\frac{1}{2}}$ a
3. El sistema valida si el email ya existe en su base de datos
4. El sistema verifica si la contraseña  $i_{\frac{1}{2}}$ a ingresada es igual a la anteriormente grabada
5. El sistema autentica al usuario
6. El sistema carga la pantalla principal y libera las funcionalidades del usuario.
7. **Fin Caso de Uso.**

**Caso alternativo**

- 3.1 - El sistema verifica que el email no existe, vuelve al punto 1
  - 4.1 - El sistema verifica que son distintas las contraseñas  $i_{\frac{1}{2}}$ as, vuelve al punto 1.
  - 6.1 - Si el usuario posee contenido pausado previamente, extiende caso de uso *Reproduciendo contenido online*.
-



---

**Caso de uso: Actualizando datos de cuenta**

**Actores:** Usuario.

**Pre:** El usuario se autenticó exitosamente.

**Post:** Los datos de su cuenta fueron actualizados.

**Caso normal**

1. El usuario abre el sistema de administración de la cuenta.
2. El usuario altera los parámetros deseados
3. El usuario pide guardar los cambios
4. Si el usuario cambia el mail, el sistema verifica que no pertenezca a otro usuario
5. El sistema graba los datos de la cuenta editados
6. **Fin Caso de Uso.**

**Caso alternativo**

- 3.1 En caso de que algún parámetro sea nulo (de los no opcionales), el sistema muestra un mensaje de error y vuelve al paso 2.
  - 4.1 - El mail ya existe y no le pertenece al usuario editando, envía un mensaje de error y vuelve a punto 2
-

---

**Caso de uso: Actualizando datos de tarjeta de crédito**

**Actores:** Usuario.

**Pre:** El usuario se autenticó exitosamente.

**Post:** Los datos de tarjeta de crédito de su cuenta fueron actualizados.

**Caso normal**

1. El usuario abre el sistema de administración de datos de tarjeta de crédito.
2. El usuario cambia (o ingresa si es que no tenía) los datos de la tarjeta de crédito
3. Usa caso de uso: *validando datos de la tarjeta de crédito*
4. Si se pudo validar, se emite mensaje de *cambio exitoso* y se guardan los cambios
5. **Fin Caso de Uso.**

**Caso alternativo**

- 4.1 - Si no se pudo validar, el sistema muestra un mensaje de error. Volver al paso 2.
-

---

**Caso de uso: Pidiendo préstamo de contenido**

**Actores:** Amigo.

**Pre:** El Amigo está autenticado.

**Post:** El sistema graba el pedido de préstamo.

**Caso normal**

1. El usuario entra en la lista de sus amigos
  2. El usuario elige a alguno de sus amigos
  3. El sistema busca el contenido que el amigo puede prestar
  4. El sistema muestra el contenido posible para préstamo de su amigo
  5. El usuario selecciona un contenido del disponible
  6. El sistema envía el pedido de préstamo al amigo elegido.
  7. **Fin Caso de Uso.**
-

---

**Caso de uso: Rechazando pedido préstamo de contenido****Actores:** Amigo.**Pre:** El amigo está  $\frac{1}{2}$  autenticado.**Post:** Se efectúa el rechazo del pedido de préstamo.**Caso normal**

1. El sistema le envía al Amigo una solicitud de pedido de préstamo (incluye caso de uso Pidiendo préstamo de contenido).
  2. El Amigo selecciona que NO quiere prestarlo
  3. El sistema no efectúa el préstamo y guarda en la cuenta del Amigo que realizó el pedido un mensaje de que este fue rechazado. Usa el Caso de Uso: Recibiendo respuesta al pedido de préstamo
  4. **Fin Caso de Uso.**
-

---

**Caso de uso: Recibiendo respuesta al pedido de prestamo**

**Actores:** Usuario.

**Pre:** El usuario pidió  $\frac{1}{2}$  prestamos.

**Post:** El usuario se entera el resultado de su pedido.

**Caso normal**

Trivial

---

---

**Caso de uso: Aceptando pedido préstamo de contenido****Actores:** Amigo.**Pre:** El amigo está  $\frac{1}{2}$  autenticado.**Post:** Se efectúa aceptación del pedido de préstamo y este pasa a estar disponible para el amigo solicitante y deja de estarlo para el amigo prestatante.**Caso normal**

1. El sistema le envía al Amigo una solicitud de pedido de préstamo (incluye caso de uso Pidiendo préstamo de contenido).
  2. El Amigo acepta el préstamo de dicho contenido
  3. El sistema efectúa el préstamo, eliminándolo del contenido disponible para el amigo que aceptó y poniéndolo como disponible en la cuenta del usuario solicitante.
  4. El sistema guarda en la cuenta del Amigo que realizó el pedido que este fue aceptado. Usa el Caso de Uso: Recibiendo respuesta al pedido de préstamo
  5. El sistema guarda la información necesaria para poder saber el tiempo de préstamos y poder hacer la devolución automática.
  6. **Fin Caso de Uso.**
-

---

**Caso de uso: Agregando amigo**

**Actores:** Amigo.

**Pre:** El Amigo se encuentra autenticado.

**Post:** Se crea la solicitud de amistad si el amigo existe..

**Caso normal**

1. El amigo abre el sistema de manejo de amistades.
2. El Amigo busca a la persona (rol de amigo) con la que se quiere contactar y la selecciona
3. El amigo encuentra a la persona y la selecciona.
4. El sistema envía  $\frac{1}{2}$  a solicitud de amistad al usuario elegido como amigo a agregar
5. **Fin Caso de Uso.**

**Caso alternativo**

- 3.1 - El amigo no encuentra a la persona, ir a paso 5.
-

---

–trabajando fin–

**Caso de uso:** Eliminando amigo

**Actores:** Amigo.

**Pre:** El Amigo se encuentra autenticado, tiene al menos un amigo

**Post:** Su amigo es eliminado

**Caso normal**

1. El amigo abre el sistema de manejo de amistades.
2. El amigo busca el amigo a eliminar, y lo elimina
3. El sistema lo borra de la lista de amigos. De esta forma, no podrán ver los contenidos comprados por el otro ni podrán prestarse más contenido entre sí.
4. **Fin Caso de Uso.**

**Caso alternativo**

---

**Caso de uso:** Aceptando solicitud de amistad.

**Actores:** Amigo.

**Pre:** El amigo se encuentra autenticado

**Post:** Se acepta la solicitud de amistad

**Caso normal**

1. El sistema le envía al Amigo una solicitud de amistad (incluye caso de uso Agregando amigo)
2. El amigo lo acepta.
3. El sistema registra la aceptacion, ahora los amigos pueden ver los contenidos comprados de cada uno.
4. avisa a quien emitio la solicitud
5. **Fin Caso de Uso.**

**Caso alternativo**

---



**Caso de uso:** Rechazando solicitud de amistad.

**Actores:** Amigo

**Pre:** El amigo se encuentra autenticado

**Post:** Se rechaza la solicitud de amistad.

**Caso normal**

1. El sistema le envía al Amigo una solicitud de amistad (incluye caso de uso Agregando amigo)
2. El amigo lo rechaza.
3. El sistema registra el rechazo
4. El sistema recuerda el rechazo para que al amigo rechazante no le lleguen más peticiones de amistad, por mas que estas puedan ser realizadas por el amigo que emitio el pedido.
5. **Fin Caso de Uso.**

**Caso alternativo**

---

**Caso de uso:** Aplicando publicidad a contenidos.

**Actores:** Administrador de contenido.

**Pre:** Existen contenidos agregados al sistema.

**Post:** Existen contenidos agregados al sistema.

**Caso normal**

1. El administrador de contenidos ingresa al sistema de administracion de contenido.
2. El administrador de contenidos verifica que se haya efectuado el pago por la publicidad.
3. El administrador de contenidos ingresa el contenido a publicitar en el sistema.
4. El administrador de contenidos aplica la publicidad a los contenidos acordados previamente con el auspiciante.

**Fin Caso de Uso.**

**Caso alternativo**

2.1 - De no estar paga la publicidad, repite el paso 1) al día siguiente.

---

**Caso de uso:** Aplicando Posicionamiento a contenidos

**Actores:** Administrador de contenido.

**Pre:** Existen contenidos agregados al sistema.

**Post:** Se aplico el posicionamiento a los contenidos acordados con el proveedor de contenidos.

**Caso normal**

1. El administrador de contenidos ingresa al sistema de administracion de contenido.
2. El administrador de contenidos verifica que se haya hecho el pago por el posicionamiento.
3. El administrador de contenidos aplica la publicidad a los contenidos acordados previamente con el proveedor de contenidos.
4. **Fin Caso de Uso.**

**Caso alternativo**

2.1 - De no estar pago el posicionamiento, repite el paso 1) al día siguiente.

---

**Caso de uso:** Categorizando Contenido .

**Actores:** Usuario.

**Pre:** Existen contenidos agregados al sistema..

**Post:** El contenido ahora se encuentra categorizado (ej: etiquetas) que el administrador crea pertinentes..

**Caso normal**

1. El administrador de contenidos ingresa al sistema de administracion de contenido.
2. El administrador de contenidos selecciona un contenido y le asigna las categorías a las que pertenecerá.
3. **Fin Caso de Uso.**

**Caso alternativo**

---

**Caso de uso:** Aplicando Precio al contenido.

**Actores:** Administrador de contenidos.

**Pre:** Obtenier nuevo contenido.

**Post:** El contenido ahora tiene un valor monetario (puede ser 0).

**Caso normal**

1. El administrador de contenidos ingresa al sistema de administracion de contenido.
2. El administrador de contenido selecciona un contenido.

3. Selecciona un contenido y le asigna un valor monetario, que puede ser 0 o un valor mayor que cero.

4. **Fin Caso de Uso.**

#### Caso alternativo

---

**Caso de uso:** Agrupando contenido en grupos.

**Actores:** Administrador de contenido.

**Pre:** contenido habilitado (de los contenidos a agrupar)

**Post:** Los contenidos agrupados pueden ser adquiridos en conjunto por los clientes

#### Caso normal

1. El administrador de contenidos ingresa al sistema de administracion de contenido.
2. El administrador de contenido selecciona los contenidos que quiere que formen parte de un grupo o paquete
3. El administrador de contenido le asigna un nombre al grupo (i.e: Grandes éxitos de Stallone)
4. El administrador de contenido le asigna un valor monetario no negativo (puede ser 0) al paquete o grupo de contenidos
5. El administrador de contenido finaliza el armado del grupo
6. El sistema publica el grupo, que ahora puede ser adquirido por los clientes
7. **Fin Caso de Uso.**

#### Caso alternativo

3.1 El nombre de grupo ya existe, el sistema limpia el campo de nombre de grupo a asignar y vuelve a paso 3.

---

**Caso de uso:** Asignando tiempo de préstamo a contenido.

**Actores:** Administrador de contenido.

**Pre:** contenido está en el sistema (puede no estar habilitado)

**Post:**el administrador de contenido asigno un valor tiempo de préstamo (puede ser nulo)

#### Caso normal

1. El administrador de contenidos ingresa al sistema de administracion de contenido.

2. Selecccion de contenido
3. .1 Selecciona el contenido individual al que se quiere cambiar el valor del tiempo de préstamo
4. Le asigna asigna un valor temporal (puede ser 0)
5. **Fin Caso de Uso.**

#### **Caso alternativo**

1.2 - Selecciona la categoría entera de contenido a los que se quiere asignar un mismo valor de tiempo de préstamos

---

**Caso de uso:** Definiendo parámetros de almacenamiento.

**Actores:** Administrador de contenido

**Pre:** contenido/s está/n en el sistema (puede/n no estar habilitado/s)

**Post:** el administrador de contenido asigno los parámetros de almacenamiento a uno o un grupo de contenidos

#### **Caso normal**

1. El administrador de contenidos ingresa al sistema de administracion de contenido.
2. seleccion de contenido
3. .1 El administrador de contenido selecciona el contenido individual al que se quiere cambiar el valor de los parámetros de almacenamiento
4. El administador de contenido le asigna asigna un valor al parámetro de cantidad de descargas necesarias para que se almacene el contenido
5. El administrador de contenido asigna un valor al parámetro del período de tiempo en el que se contabiliza la cantidad de descargas.
6. **Fin Caso de Uso.**

#### **Caso alternativo**

1.2 - El administador de contenido selecciona la categoría entera de contenido a los que se quiere cambiar el valor de los parámetros

---

**Caso de uso:** Habilitando contenido

**Actores:** Administrador de Contenido.

**Pre:** El contenido fue revisado

**Post:** El contenido es marcado como habilitado y los usuarios pueden adquirirlo.

**Caso normal**

1. El administrador de contenidos ingresa al sistema de administracion de contenido.
2. El Administrador de contenido observa la cola de contenido revisado.
3. El Administrador de contenido habilita los contenidos que no hayan tenido error en la revision.
4. Para los contenidos con error, si el contenido es de ejecucion el Administrador de contenido avisa a el/los fabricante/s (ver DA Lograr calidad en contenidos de ejecucion), y si el contenido es de reproduccion el Administrador de Contenido se comunica con el proveedor de contenido (ver DA Lograr calidad en contenidos de reproduccion).
5. El sistema saca los contenidos de la cola
6. El sistema publica los contenidos habilitados con los parámetros y categorías previamente asignados
7. **Fin Caso de Uso.**

**Caso alternativo**

---

**Caso de uso:** Ingresando nuevo contenido.

**Actores:** Administrador de contenido.

**Pre:**

**Post:** El contenido se encuentra en el sistema, pero solo tienen acceso al mismo los revisores de contenido y el administrador de contenido

**Caso normal**

1. El administrador de contenidos ingresa al sistema de administracion de contenido.
2. Ingresar el nuevo contenido en la cola de revision correspondiente al tipo de contenido.
3. El administrador de contenidos le asigna el Genero al contenido.
4. Si el administrador de contenido decide modificar el tiempo de préstamos, extiende CU:  
Asignando tiempo de préstamo
5. Si el administrador de contenido decide modificar las categorías a las que pertenecerá:  
extiende CU: Categorizando contenido

6. Si el administrador de contenido decide asignar el precio al contenido: extiende CU: Asignando precio al contenido
7. Si el Adminsitrador de contenido decide modificar los parámetros para el almacenamiento: extiende CU: Definiendo parámetros de descarga.
8. Todos los parámetros no ingresados, el sistema los asigna por default. Arbitrariamente se decido:
  - Tiempo de préstamo: 3 semanas.
  - Parámetros de almacenamiento: 100 descargas en una semana.
  - Valor monetario: 0 (gratis)

#### 9. Fin Caso de Uso.

### Caso alternativo

---

#### Caso de uso: Ingresando datos para el pago de publicidad

**Actores:** Auspiciante .

**Pre:** –.

**Post:** El auspiciante logra pagar por la publicidad.

#### Caso normal

#### Caso normal

1. El auspiciante se autentica en el sistema
  2. El auspiciante ingresa sus datos. Si no tiene datos personales previos ingresados, los ingresa al sistema el sistema. Si tiene datos personales previos ingresados, no hace nada.
  3. El auspiciante selecciona un tipo de publicidad.
  4. El auspiciante selecciona la categoría  $i \in \frac{1}{2}$  a donde se aplicará  $i \in \frac{1}{2}$  la publicidad.
  5. El auspiciante ingresa su tarjeta de crédito para el pago.
  6. El auspiciante efectúa el pago (Extiende caso de uso *Cobrando Venta*)
  7. El sistema devuelve mensaje de *transacción exitosa*
  8. **Fin Caso de Uso.**
-

**Caso de uso: Ingresando datos para el pago del posicionamiento****Actores:** Proveedor de contenidos.**Pre:** -**Post:** El Proveedor de contenidos logra pagar por el posicionamiento.**Caso normal**

1. El proveedor de contenidos se autentica en el sistema.
2. El proveedor de contenidos sus ingresa datos. Si no tiene datos personales previos ingresados, los ingresa al sistema. Si tiene datos personales previos ingresados, no hace nada.
3. El proveedor de contenidos selecciona un tipo de posicionamiento.
4. El proveedor de contenidos selecciona el posicionamiento.
5. El proveedor de contenidos ingresa su tarjeta de crédito para el pago.
6. El Proveedor de contenidos selecciona si desea que su tarjeta se guarde o no en el sistema.
7. El proveedor de contenidos efectua el pago (Extiende caso de uso *Cobrando Venta*)
8. El Sistema devuelve mensaje de *transacción exitosa*
9. **Fin Caso de Uso.**

**Caso alternativo**

8.1 - El pago no pudo realizarse, El sistema devuelve mensaje de *error en la transacción* vuelve al paso 5

**Caso de uso: Cobrando Venta****Actores:** Sistema de Facturación.**Pre:** El sistema de facturación recibe un pedido de cobro con los datos de la tarjeta de crédito.**Post:** El sistema devuelve el resultado de la operación. Si el resultado es positivo, el sistema de facturación realiza el cobro.**Caso normal**

1. Usa caso de uso: *Validando datos de tarjeta de crédito*.
2. El sistema de facturación verifica si se puede hacer el cobro.
3. Si se puede hacer el cobro lo efectiviza y le envía un mensaje al sistema informando OK.
4. El sistema registra el pago.
5. El sistema muestra un mensaje con el resultado de la transacción.

## 6. Fin Caso de Uso.

### Caso alternativo

1.1 - Si no se puede validar, ir a 5. 3.1 - Si no se puede, no realiza el cobro, Ir a 5

---

### **Caso de uso: Recibiendo petición de contenido**

**Actores:** Proveedor de contenidos.

**Pre:** El usuario solicita un contenido

**Post:** El proveedor de contenido encuentra el contenido en su sistema.

### Caso normal

1. El proveedor de contenidos recibe una petición de contenido.
2. El proveedor de contenido lo busca en su sistema y lo encuentra.

## 3. Fin Caso de Uso.

### Caso alternativo

2.1 - El proveedor de contenido no encuentra el contenido en su sistema. 2.2 - El proveedor de contenido envía a una notificación de fallo al sistema.

---

### **Caso de uso: Enviando contenido**

**Actores:** Proveedor de contenidos

**Pre:** El proveedor de contenido recibe una petición de contenido y lo encuentra en su sistema

**Post:** El proveedor de contenido envía el contenido al sistema.

### Caso normal

1. El proveedor de contenido envía el contenido al sistema.
2. Fin Caso de Uso.

### Caso alternativo

1.1 - Ocurre un fallo en el envío o vuelve al paso 1)

---

### **Caso de uso: Validando contenido de reproducción**

**Actores:** Revisor de contenido digital

**Pre:** El contenido a validar debe ser de reproducción

**Post:** El estado del contenido es validado o con errores

### Caso normal

1. El revisor de contenido digital se loguea al sistema de revisión de contenido digital.



2. El revisor verifica en la cola asignada a  $i_{\frac{1}{2}}l$  si posee algún  $i_{\frac{1}{2}}n$  contenido a validar
3. El revisor elige el contenido y valida que se encuentre en la  $m_{i_{\frac{1}{2}}}x$ ima definici $i_{\frac{1}{2}}n$  posible
4. El contenido cumple la  $m_{i_{\frac{1}{2}}}x$ ima definici $i_{\frac{1}{2}}n$  posible. El revisor pasar a verificar que el contenido no posea cortes.
5. Si el contenido es un video, el revisor valida adem $i_{\frac{1}{2}}s$  que los subt $i_{\frac{1}{2}}t$ ulos sean correctos.
6. Si el contenido forma parte de una temporada o  $i_{\frac{1}{2}}l$ bum, el revisor repite desde el paso 1, hasta haber revisado toda la temporada o  $i_{\frac{1}{2}}l$ bum
7. Se agrega el contenido a la cola de contenido para habilitar (extiende CU Habilitando contenido).

#### 8. Fin Caso de Uso.

#### Caso alternativo

- 2.1 - El revisor no posee ning $i_{\frac{1}{2}}n$  contenido a validar, ir a paso 8.
- 3.1 - El contenido no se encuentra en la  $m_{i_{\frac{1}{2}}}x$ ima definici $i_{\frac{1}{2}}n$  posible. El revisor loguea en el sistema que el contenido posee un error (ver DA Lograr calidad en contenidos de reproduc $i_{\frac{1}{2}}n$ ) (ir a paso 7)
- 4.1 - El contenido posee cortes. El revisor loguea en el sistema que el contenido posee un error (ver DA Lograr calidad en contenidos de reproduc $i_{\frac{1}{2}}n$ ) (ir a paso 7)
- 5.1 - El video tiene subt $i_{\frac{1}{2}}t$ ulos incorrectos, el revisor El revisor loguea en el sistema que el contenido posee un error (ver DA Lograr calidad en contenidos de reproduc $i_{\frac{1}{2}}n$ ) (ir a paso 7)

#### **Caso de uso: Validando especificaci $i_{\frac{1}{2}}n$ de contenido de ejecuci $i_{\frac{1}{2}}n$**

**Actores:** Revisor de especificaci $i_{\frac{1}{2}}n$

**Pre:** El contenido a validar debe ser de ejecuci $i_{\frac{1}{2}}n$ , el contenido se encuentra en la cola de revisi $i_{\frac{1}{2}}n$ , revisor autenticado en el sistema

**Post:** El contenido de ejecuci $i_{\frac{1}{2}}n$  pasa el test de especificaci $i_{\frac{1}{2}}n$  o no.

#### Caso normal

1. El revisor de contenido especificaci $i_{\frac{1}{2}}n$  se loguea al sistema de revisi $i_{\frac{1}{2}}n$  de contenido especificaci $i_{\frac{1}{2}}n$ .
2. El revisor verifica en la cola asignada a  $i_{\frac{1}{2}}l$  si posee algún  $i_{\frac{1}{2}}n$  contenido a validar
3. Como existe al menos uno, el revisor de especificaci $i_{\frac{1}{2}}n$  solicita la especificaci $i_{\frac{1}{2}}n$  de la aplicaci $i_{\frac{1}{2}}n$  al fabricante
4. El revisor confecciona un conjunto de casos de prueba para testear la aplicaci $i_{\frac{1}{2}}n$

5. El revisor verifica que el comportamiento del programa sea el esperado por la especificación  $\frac{1}{2}n$ .
6. El revisor guarda en el sistema si pasó  $\frac{1}{2}$  el test o no.
7. **Fin Caso de Uso.**

#### Caso alternativo

3.1 Como no existe ningún  $\frac{1}{2}n$  contenido, ir a paso 7.

#### **Caso de uso: Validando contenido de ejecución $\frac{1}{2}n$**

**Actores:** Revisor de contenido digital, Revisor de especificación  $\frac{1}{2}n$

**Pre:** El contenido a validar debe ser de ejecución  $\frac{1}{2}n$ , revisor de contenido digital autenticado en el sistema

**Post:** El estado del contenido es validado o con errores

#### Caso normal

1. El revisor de especificación  $\frac{1}{2}n$  valida el cumplimiento de la especificación  $\frac{1}{2}n$ . Incluye CU Validando especificación  $\frac{1}{2}n$  de contenido de ejecución  $\frac{1}{2}n$ .
2. El revisor de contenido digital somete al contenido a un conjunto de eventos de acuerdo al tipo de aplicación  $\frac{1}{2}n$  (arranque, tamaño  $\frac{1}{2}n$  de los archivos abiertos, cantidad de thread en ejecución  $\frac{1}{2}n$ , etc), durante un tiempo establecido con anterioridad.
3. Luego del test, si siempre logró  $\frac{1}{2}$  iniciarse en menos de 30 segundos y nunca estuvo colgado por más  $\frac{1}{2}s$  de 5 segundos, se aprueba el contenido
4. Se agrega el contenido a la cola de contenido para habilitar (extiende CU Habilitando contenido).
5. **Fin Caso de Uso.**

#### Caso alternativo

2.1 - Si el contenido no cumple con la especificación  $\frac{1}{2}n$ , el revisor de contenido digital loguea en el sistema este suceso (ir a paso 4).

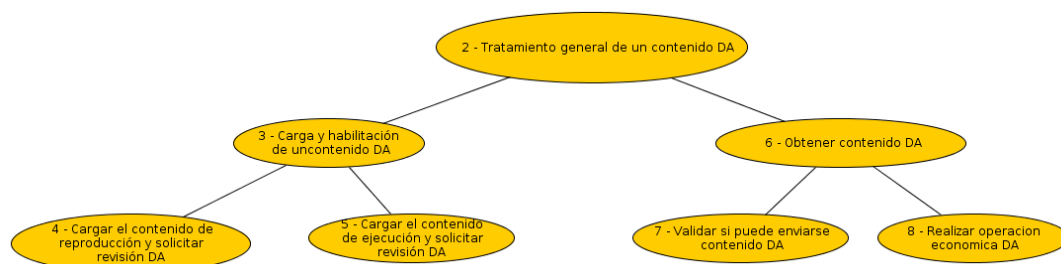
3.1 - Si ocurrió  $\frac{1}{2}$  un cuelgue de duración  $\frac{1}{2}n$  mayor a 5 segundos, o no logró  $\frac{1}{2}$  iniciarse en menos de 30 segundos, se considera rechazada la aplicación  $\frac{1}{2}n$ , y se loguea en sistema el suceso. (ir a paso 4).

## 2.3. Diagrama de Actividad

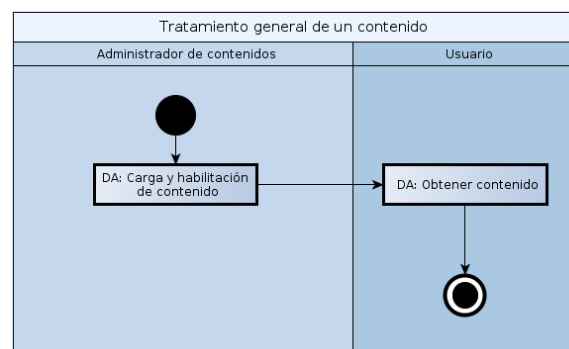
Un diagrama de actividad permite modelar una secuencia de acciones entre los componentes de un sistema. En nuestro caso lo hemos utilizado para describir los flujos principales del sistema y la interacción a través del mismo de los diferentes agentes que intervienen.

Los diagramas a continuación guardan una relación siendo que los diagramas principales utilizan diagramas secundarios para describir el comportamiento. Así, en sí un diagrama un estado (que representa una acción) se encuentra recuadrado con una línea gruesa y precedido por las siglas DA (Diagrama de Actividad) indica que ese estado o acción se describe con mayor detalle en otro Diagrama de Actividad.

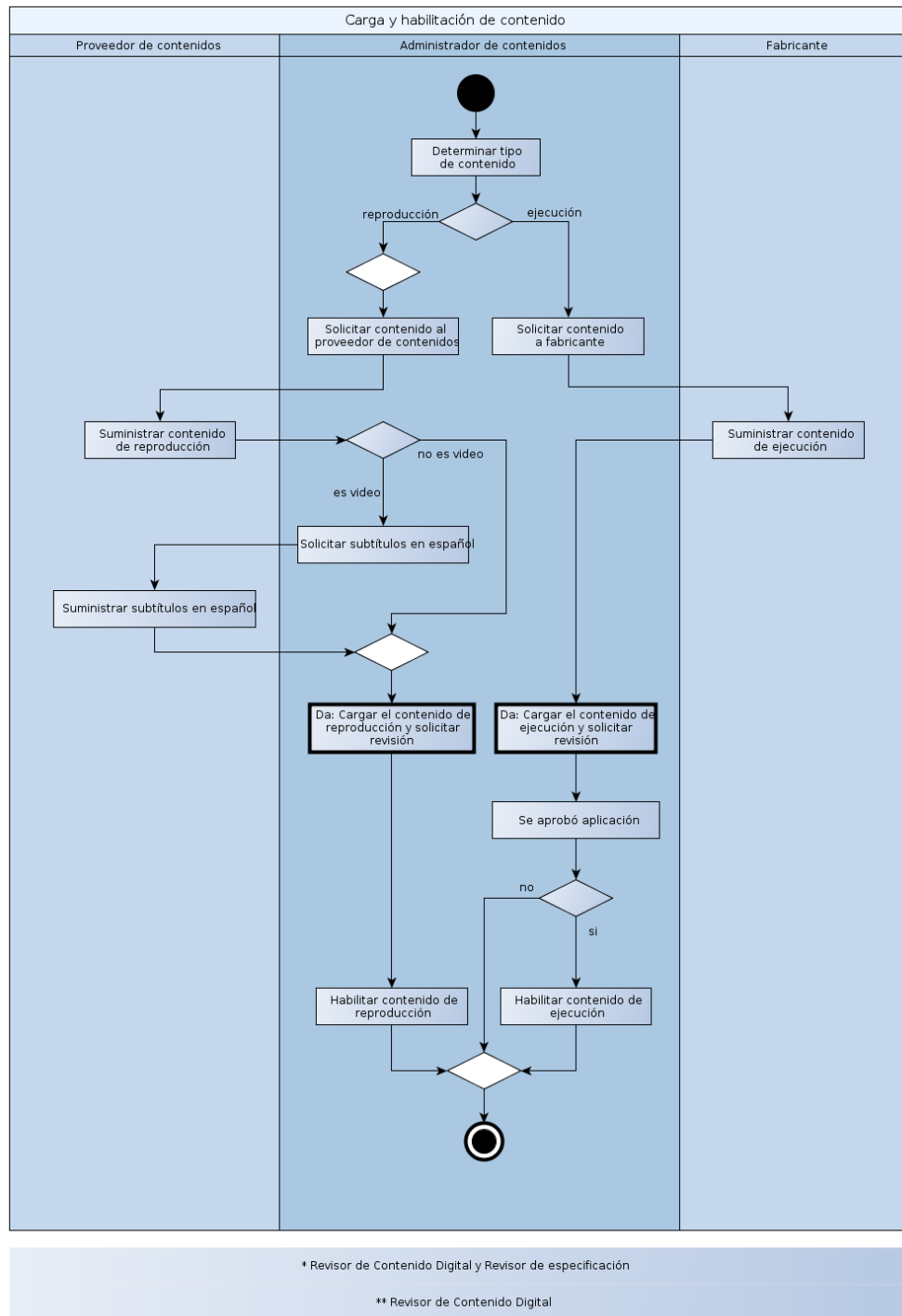
El siguiente diagrama muestra la relación que existe entre los *Diagramas de actividad* para mayor legibilidad de los mismos. Si un *Diagrama de actividad A* tiene una flecha con otro *Diagrama de actividad B* ejemplo:  $(A \rightarrow B)$  significa que diagrama A aparece mencionado en el diagrama B. De esta forma un árbol de la forma en que se utilizan los diagramas es el siguiente:



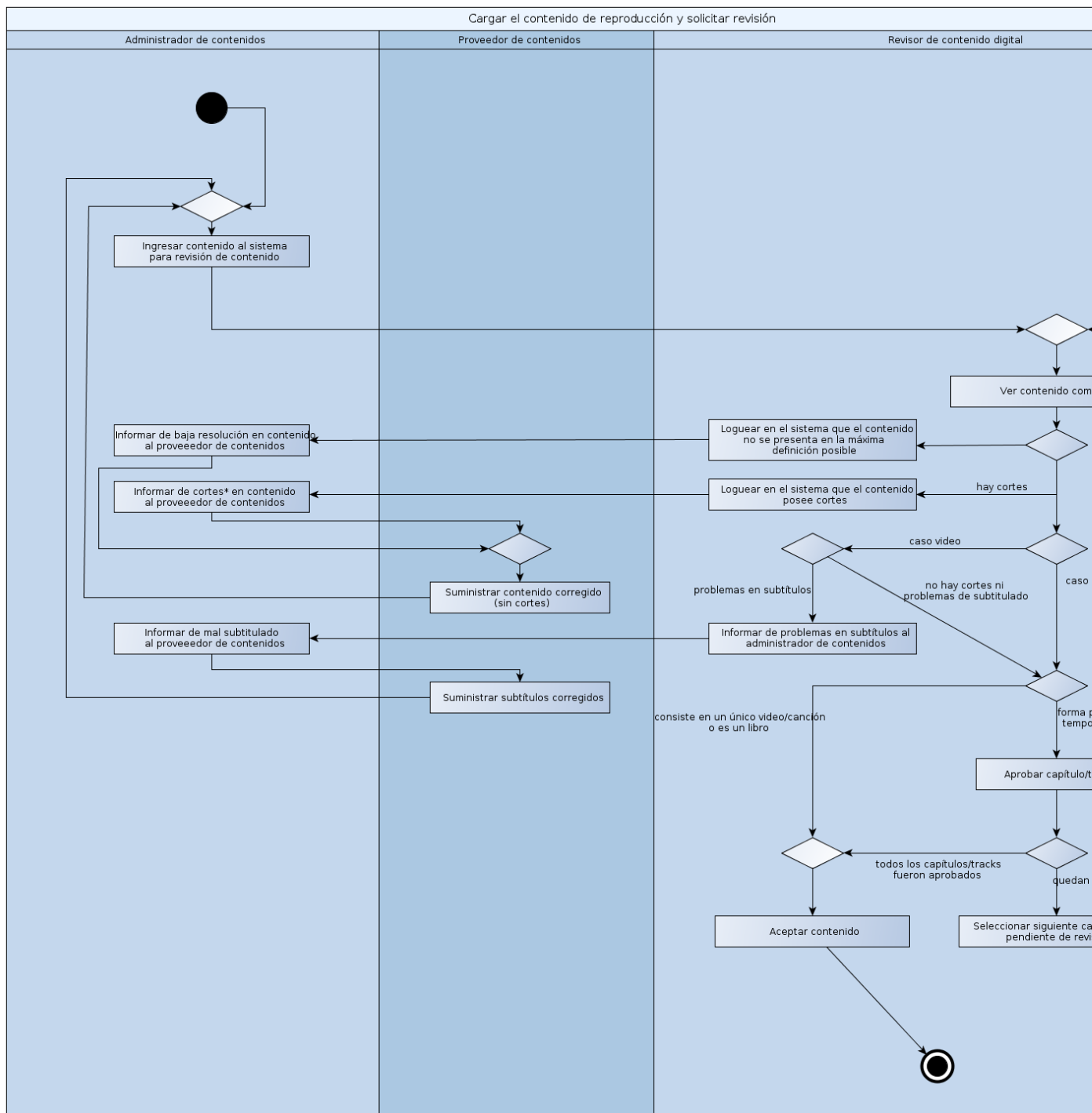
### 2.3.1. Tratamiento general de un contenido



### 2.3.2. Carga y habilitación de contenido



### 2.3.3. Cargar el contenido de reproducción y solicitar revisión

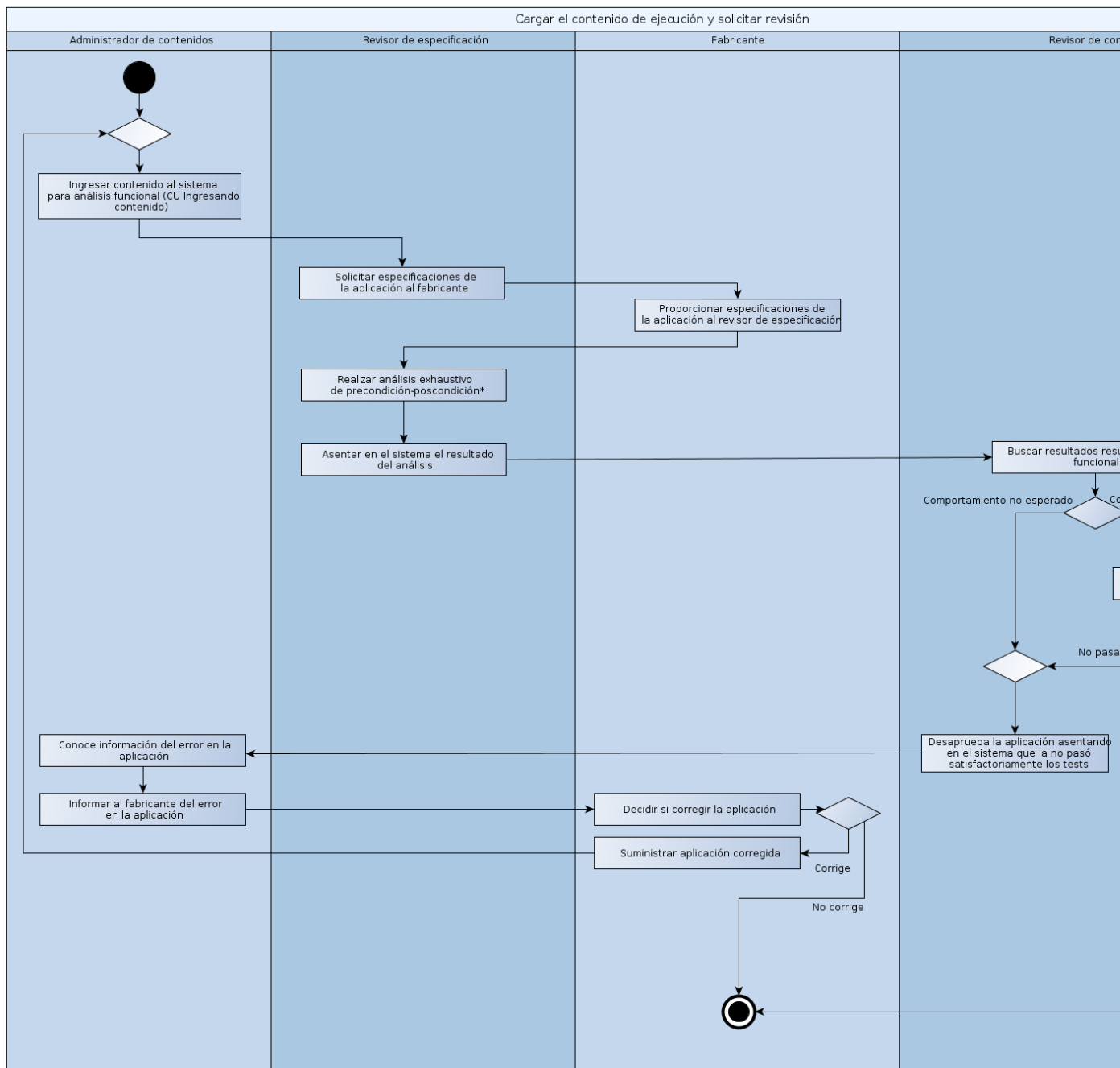


Categorizar se refiere a hacer que el contenido forme parte de un grupo o no, y si será pago ó gratuito.

Hemos pintado algunos rombos de blanco para indicar que serán usados para unificar distintos flujos

\* Decimos que hay cortes en un libro (caso no trivial), cuando falta alguna página o hay errores en la decodificación del formato en que se encuentra (ej: PDF, DJVU,...)

### 2.3.4. Cargar el contenido de ejecución y solicitar revisión

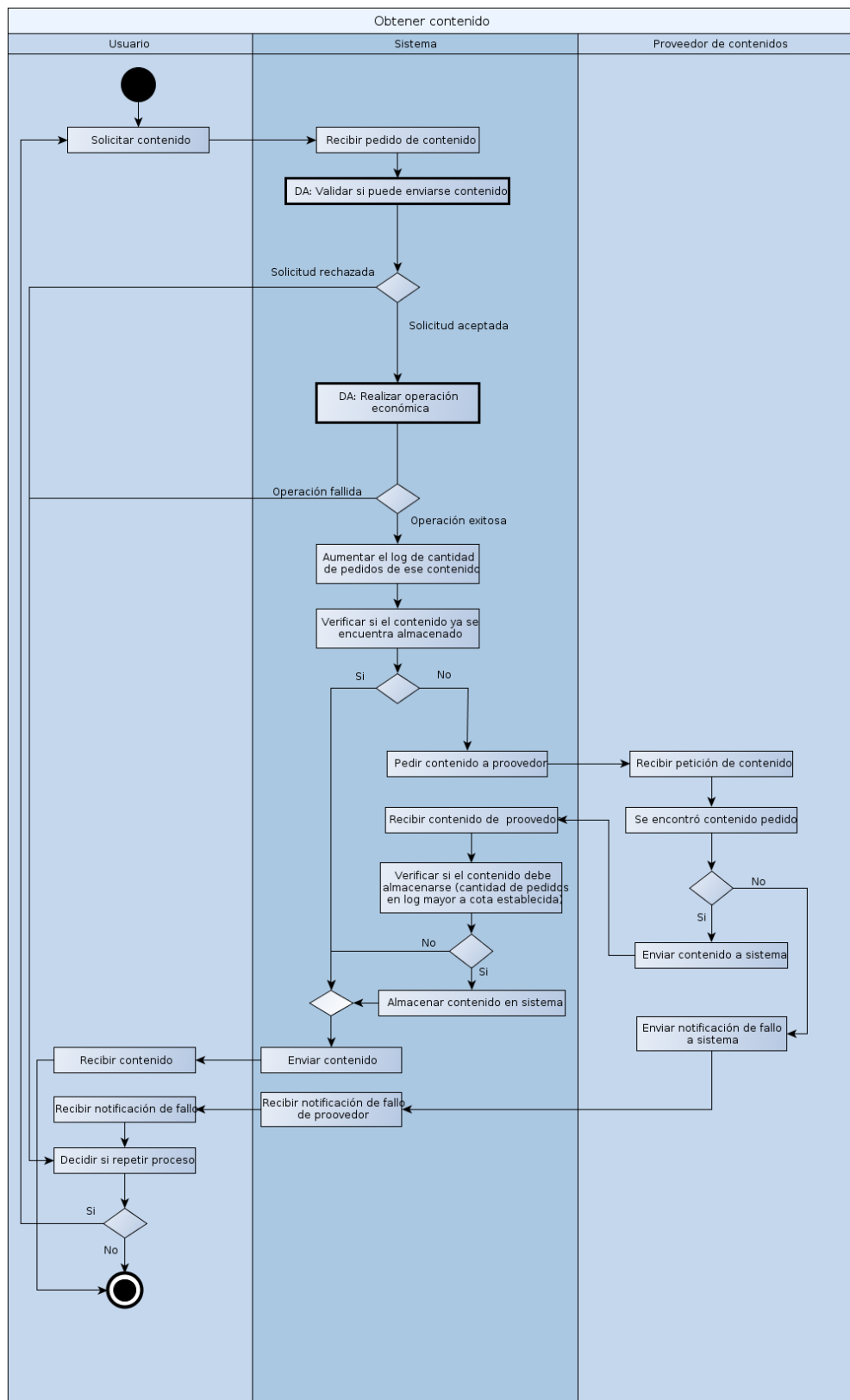


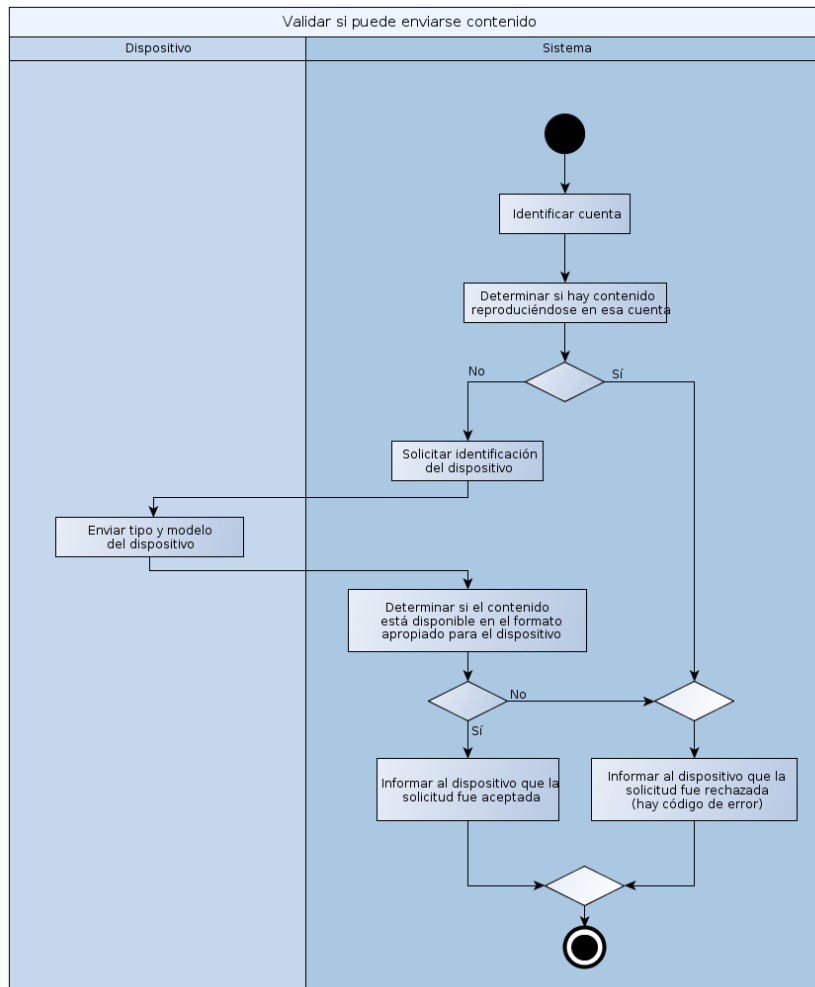
\* El análisis de precondición-poscondición consiste en verificar que la aplicación cumpla con su especificación (la cual se ha obtenido previamente)

\*\* Se considera que una aplicación pasa el test de estrés, si se inicia en menos de 30 segundos, y nunca se cuelga por más de 5 segundos

Llamamos "fabricante" a la empresa o particular que ha desarrollado la aplicación

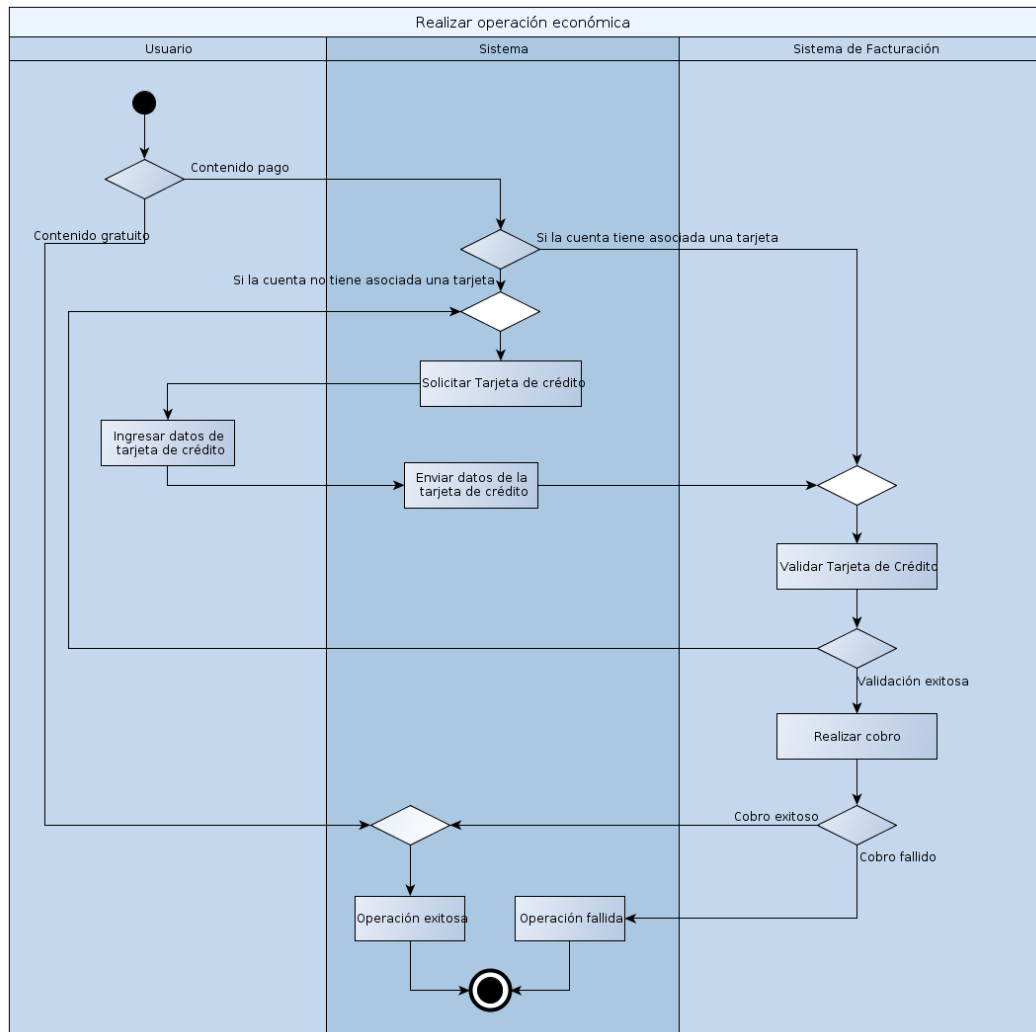
## 2.3.5. Obtener contenido



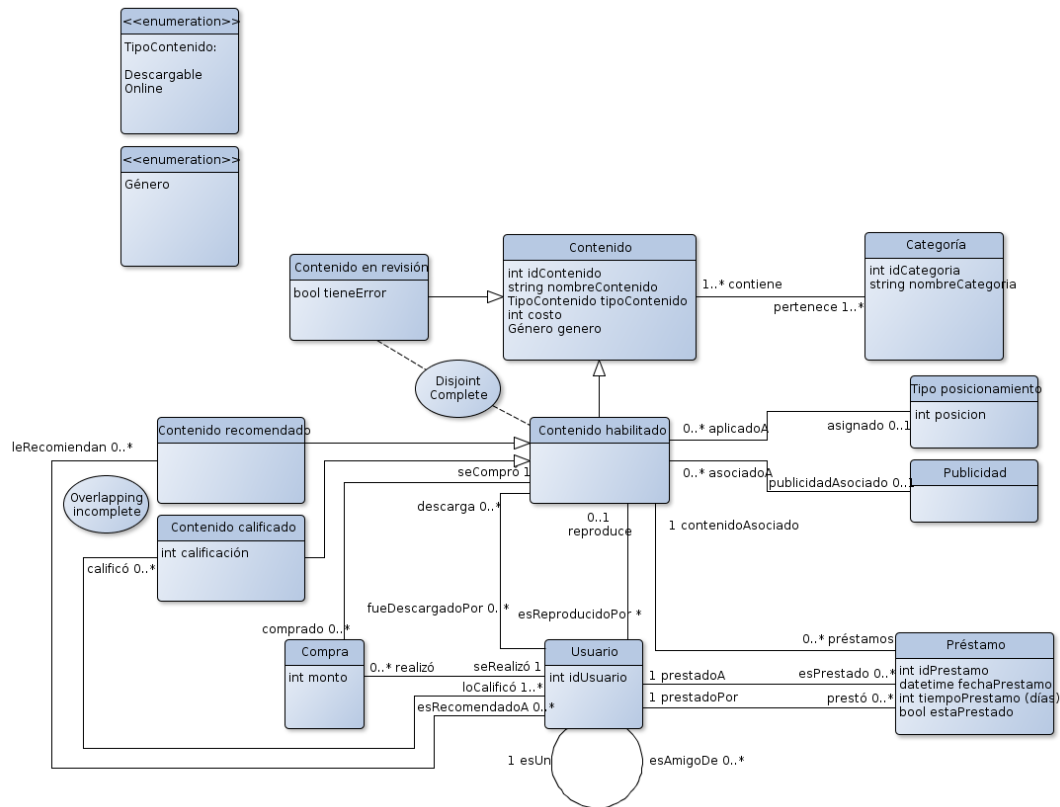
**2.3.6. Validar si puede enviarse contenido**



### 2.3.7. Realizar operación económica



## 2.4. Modelo Conceptual



### 2.4.1. Object Constraint Language (OCL)

Para complementar el *Modelo Conceptual* agregaremos una serie de limitaciones que nos permiten modelar reglas para el sistema. Este lenguaje nos permite especificar formalmente restricciones a las clases definidas.

— OCL —

context Prestamos (prestamos valido) 1) Los que se prestan son amigos (el id del usuario que recibe el préstamo esta entre los ids de los amigos del usuario que presta) inv: self.prestadoA.esAmigoDe → size() > 0 and

self.prestadoA.esAmigoDe → includes( self.prestadoPor )

2) - No se prestan contenidos de valor cero (contenidos gratis)

Context: Préstamo

inv: self.contenidoAsociado.monto > 0

2.0) Si un usuario presta un contenido, debió haberlo comprado

Context Prestamo

p.prestadoPor.realizó → size() > 0 and p.prestadoPor.realizó → collect(seCompro) → includes(p.ContenidoAsociado)

21) Los ids de los prestamos son diferentes

inv: Prestamo.allInstances() → forAll(i,j | i ≠ j ⇒ i.id ≠ j.id)

4) - Sólo se puede prestar contenido de tipo Online

Context Prestamo

inv : self.contenidoAsociado.TipoContenido = online

5) - Ver q no haya coliciones de prestamos

Prestamo.allInstances() $\rightarrow$ forall( i , j | i  $\neq$  j and i.prestadoPor = j.prestadoPor  $\Rightarrow$  ((i.contenidoAsociado = j.contenidoAsociado) $\Rightarrow$  i.fechaPrestamo < j.fechaPrestamo + j.tiempoPrestamo or i.fechaPrestamo + i.tiempoPrestamo < j.fechaPrestamo or j.fechaPrestamo < i.fechaPrestamo + i.tiempoPrestamo or j.fechaPrestamo + j.tiempoPrestamo < i.fechaPrestamo )

5.1) No puede pasar que si tengo un contenido prestado más de una vez por la misma persona, que más de un préstamo figure como actualmente prestado, ni que hay un préstamo con fecha de préstamo posterior al del préstamo que figure como actualmente prestado

Prestamo.allInstances() $\rightarrow$ select(p | p.estaPrestado) $\rightarrow$ forall(p|  $\neg$ Prestamo.allInstances() $\rightarrow$ Exists(p2 | p2 $\neq$ p and (p.prestadoPor = p2.prestadoPor and p.contenidoAsociado = p2.contenidoAsociado and p2.estaPrestado or p.prestadoPor = p2.prestadoPor and p.contenidoAsociado = p2.contenidoAsociado and p2.fechaPrestamo > p.fechaPrestamo)) )

6) - Un usuario no puede ser amigo de si mismo Context Usuario inv: self.esAmigoDe.size() $\leq$ 0  $\Rightarrow$  self.esAmigoDe() $\rightarrow$ forall( x | x.usuarioId  $\neq$  self.usuarioId)

7) -El usuario tiene contenido recomendado sii calificó al menos un contenido *nota: not+xor equivale a si y sólo si* Context Usuario

inv: not( self.califico.isEmpty() xor self.leRecomiendan.isEmpty() )

8)El usuario sólo calificó contenido que compró (para todo contenido calificado, existe alguna compra hecha con ese contenido) Context Usuario inv: self.califico $\rightarrow$ forall( u | self.realizo $\rightarrow$ exists( c | c.seCompro.idContenido = u.idContenido ))

9) - Todos los contenidos tienen diferente id

inv: Contenido.allIntances() $\rightarrow$ forall(c1,c2 | c1  $\neq$  c2  $\Rightarrow$  c1.idContenido  $\neq$  c2.idContenido)

10) -La cantidad de contenidos recomendados a un usuario es igual a cero (si nunca calificó) o diez (cuando ya calificó)

Context Usuario

inv: self.leRecomiendan $\rightarrow$ size() = 0 or self.leRecomiendan $\rightarrow$ size()= 10

11) La aparición de cada género en el contenido recomendado a un usuario es proporcional a la suma de las calificaciones que el usuario le dió a contenido de ese género, sobre la suma de calificaciones totales.

12) - No se recomienda contenido ya comprado ni contenido obtenido de préstamos

Context Usuario

self.leRecomiendan $\rightarrow$ forall(c | not c.comprado(isEmpty()))  $\Rightarrow$  c.comprado $\rightarrow$ forall(c2| c2.seRealizó.idUsuario  $\neq$  self.seRealizó.idUsuario)

13) - La publicidad sólo se aplica a contenido de precio igual a cero (contenido gratuito)

Context Publicidad

self.asociadoA $\rightarrow$ size()  $\leq$ 0  $\Rightarrow$  (self.asociadoA $\rightarrow$ forall(c | c.costo  $\leq$ 0))

14) - Se puede reproducir sólo la reproduccion online

Context Usuario

`self.reproduce.size() = 1  $\Rightarrow$  self.reproduce.TipoContenido = Online`

15) - Se puede reproducir sólo lo que compro y no prestó, o pidió prestado

Context usuario

`self.Reproduce.size() = 1  $\Rightarrow$  self.realizo $\rightarrow$ collect(seCompró) $\rightarrow$ includes(self.Reproduce) and  
self.Reproduce.préstamos $\rightarrow$ forall(p | p.esPrestadoPor = self  $\Rightarrow$  not p.estaPrestado))) or  
self.Reproduce.préstamos $\rightarrow$ exists(p2 | p2.esPrestadoA = self  $\Rightarrow$  p2.estaPrestado)))`

16) - No hay dos usuarios iguales. Todos los usuarios tienen diferente id

Context Usuario

`inv: Usuario.allInstances() $\rightarrow$ forall(u1, u2 | u1 $\neq$ u2  $\Rightarrow$  u1.idUsuario $\neq$ u2.idUsuario)`

17) - El contenido se califica de 0 a 10.

Context Contenido Calificado

`inv: 0  $\leq$  self.calificacion and self.calificacion = 10`

18) El monto de la compra es igual al costo del contenido

Context Compra

`inv: self.monto = self.seCompro.costeo`

— OCL —

### **3. Trazabilidad**

Trazabilidad de los Objetivos del Tp1 con los diagramas antes mencionados