

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

Ingeniería de Software I
Primer Cuatrimestre de 2012

Reentrega 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. Correcciones	3
1.1. Puntos a corregir	3
2. Aclaración sobre las cuentas	4
3. Amigo vs Usuario	4
4. Contenido Calificado	4
4.1. Cambios en el OCL	5
5. Modelo de FSM	7

1. Correcciones

1.1. Puntos a corregir

A continuación listamos los puntos de corrección del Trabajo práctico 2:

- *Aclaración sobre las cuentas*: Aclarar que el resto de las cuentas (salvo la del usuario) serán creadas previamente por el administrador del sistema.
- *Amigo vs Usuario*: Justificar mejor lo referente a esta interacción Amigo-Usuario o cambiar el diseño.
- *Contenido Calificado*: Corregir Contenido Calificado y Contenido Recomendado, ambas podrian ser clases en si mismas, ajustar el OCL de forma adecuada.
- *Modelo de FSM*: Modelar con FSM la interacción entre los diversos dispositivos de un mismo usuario.

Procederemos entonces a enumerar las correcciones al Tp2, destacando en **negrita** los cambios que sean pequeños.

2. Aclaración sobre las cuentas

En el modelado a continuación asumiremos que todos los actores que interactúen con el sistema (**salvo el usuario, el cual deberá crearla el mismo**) ya tienen creada una cuenta previa para poder loguearse y realizar sus acciones.

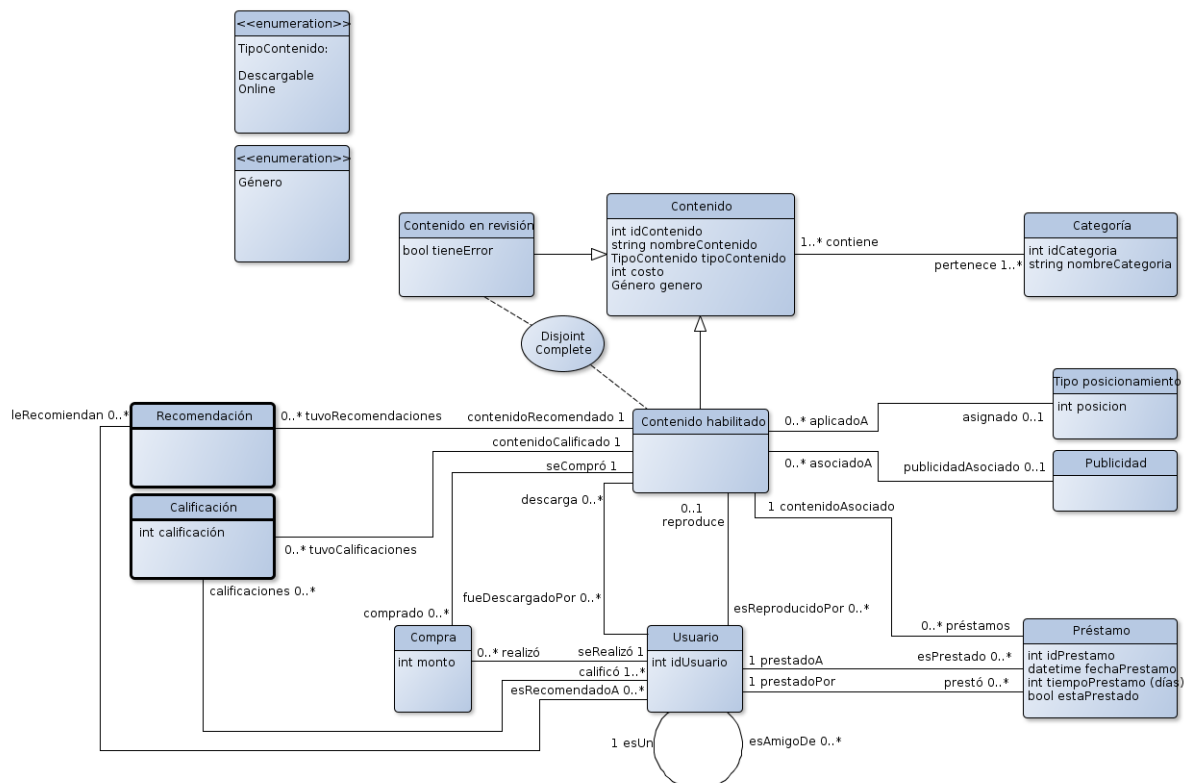
3. Amigo vs Usuario

Se decidió mantener el rol de *Amigo* en los casos de uso ya que el préstamo de contenido sólo tiene sentido entre personas que se conozcan entre sí y posean una relación de amistad. Un usuario hasta no tener al menos un amigo, no puede realizar ningún préstamo.

El sistema permite la creación de nuevas amistades, así como la terminación de estas. Siempre que la cantidad de amigos de un usuario sea mayor estricto que cero, él podrá acceder a las funcionalidades de los préstamos.

No es necesario que el usuario conozca esta diferencia de roles alcanza con que el sistema habilite o inhabilite la funcionalidad de préstamos dependiendo de la cantidad de amigos asociados a un usuario.

4. Contenido Calificado



En negrita figuran las clases modificadas.

Se cambió la clase *Contenido Calificado* que heredaba de *Contenido habilitado* por *Calificación* que se relaciona con Contenido habilitado.

Se cambió la clase *Contenido Recomendado* que heredaba de *Contenido habilitado* por *Recomendación* que se relaciona con Contenido habilitado.

4.1. Cambios en el OCL

La modificación de clases anterior genera los siguientes cambios en el OCL:

Context Usuario

El usuario tiene contenido recomendado sii calificó al menos un contenido.

inv: self.calificaciones→isEmpty() \iff self.leRecomiendan→isEmpty()

El usuario sólo calificó contenido que compró (para todo contenido calificado, existe alguna compra hecha con ese contenido)

inv: self.calificaciones→notEmpty() \implies self.realizo→notEmpty() \wedge
 self.calificaciones→collect(contenidoCalificado)→
 forAll(u | self.realizo→exists(c | c.seCompro.idContenido == u.idContenido))

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

inv: (self.calificaciones→isEmpty() \implies self.leRecomiendan→isEmpty()) \wedge
 (self.calificaciones→notEmpty() \implies (self.leRecomiendan→size() == 10))

No se recomienda contenido ya comprado ni contenido obtenido de préstamos.

inv: self.leRecomiendan→collect(contenidoRecomendado)→forall(c | c.comprado→notEmpty()
 \implies c.comprado→forall(c2 | c2.seRealizó.idUsuario \neq self.seRealizó.idUsuario))

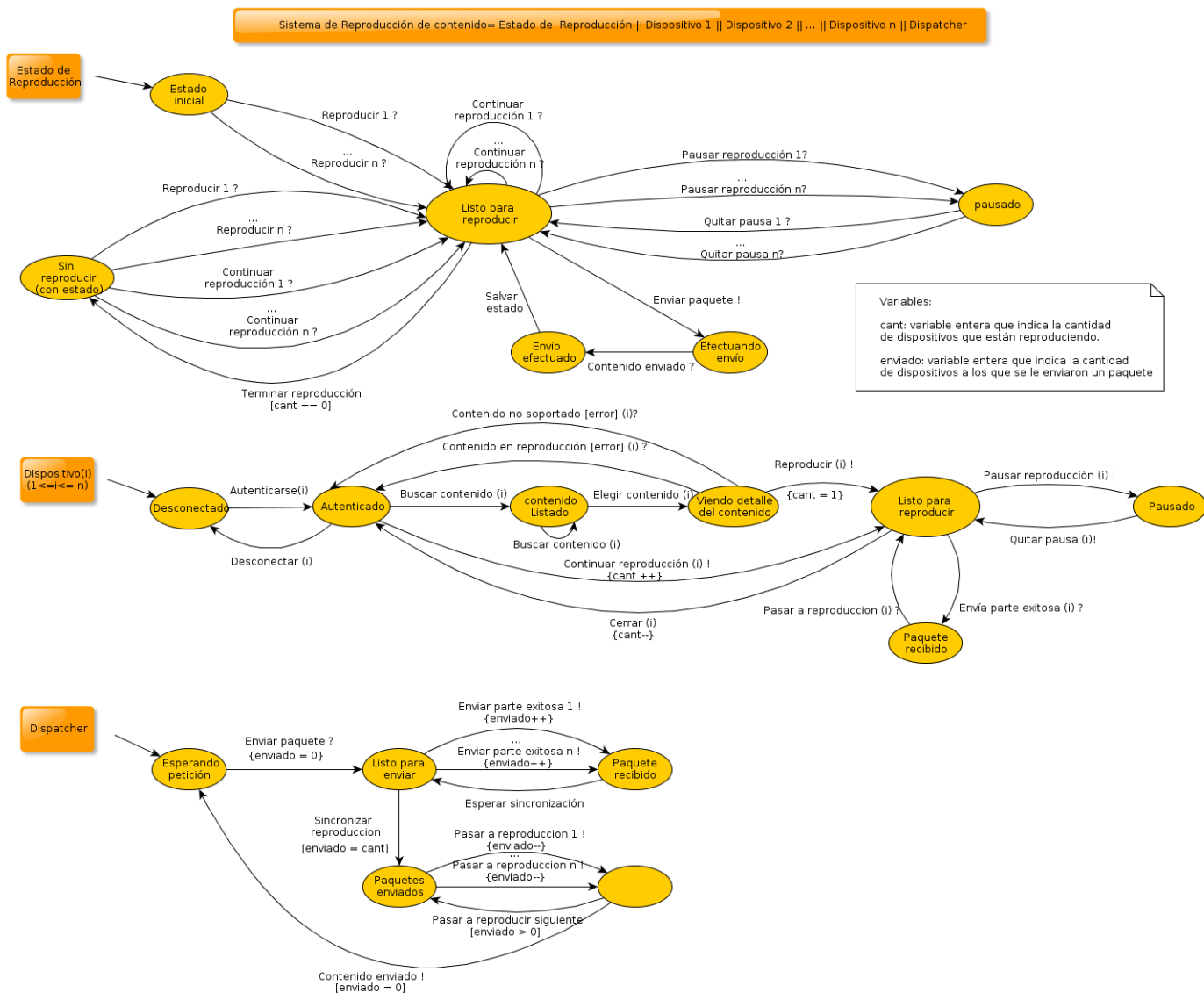
Context Calificación

Las calificaciones son de 0 a 10.

inv: $0 \leq \text{self.calificación} \wedge \text{self.calificación} \leq 10$

5. Modelo de FSM

A continuación modelaremos la interacción de los diversos dispositivos de una sola cuenta de usuario para la reproducción de contenido.



Cosas que se pueden apreciar en el modelo:

- Sólo se puede estar reproduciendo un solo contenido para los diversos dispositivos de un usuario.
- Todos los dispositivos pueden estar reproduciendo contenido al mismo tiempo, pero éste debe ser siempre el mismo (para la misma cuenta de usuario).
- Si un contenido en reproducción está siendo visto por varios dispositivos y uno de ellos pausa la reproducción, el resto también será pausado.
- Se guarda el estado del último paquete enviado exitosamente.