



CURSO 2012 - 2013

ACTIVIDAD EVALUABLE Y CALIFICABLE

1. Portada con:

Asignatura: 71013035 – Diseño de Software

Año de la práctica: 2012-2013

Centro Asociado al que pertenece: todos

Tutor que le da asistencia: Grupo de tutoría 1 y Grupo de tutoría 2

Datos personales: Nombre y apellidos:

DNI o número de expediente:

Contacto (teléfono o correo electrónico):

Localidad de residencia:

Datos de coste: Horas de dedicación al estudio de los contenidos:

Nº de actividades no evaluables realizadas y horas de dedicación:

Horas de dedicación para realizar esta actividad:

2. El enunciado y planteamiento del caso de estudio.

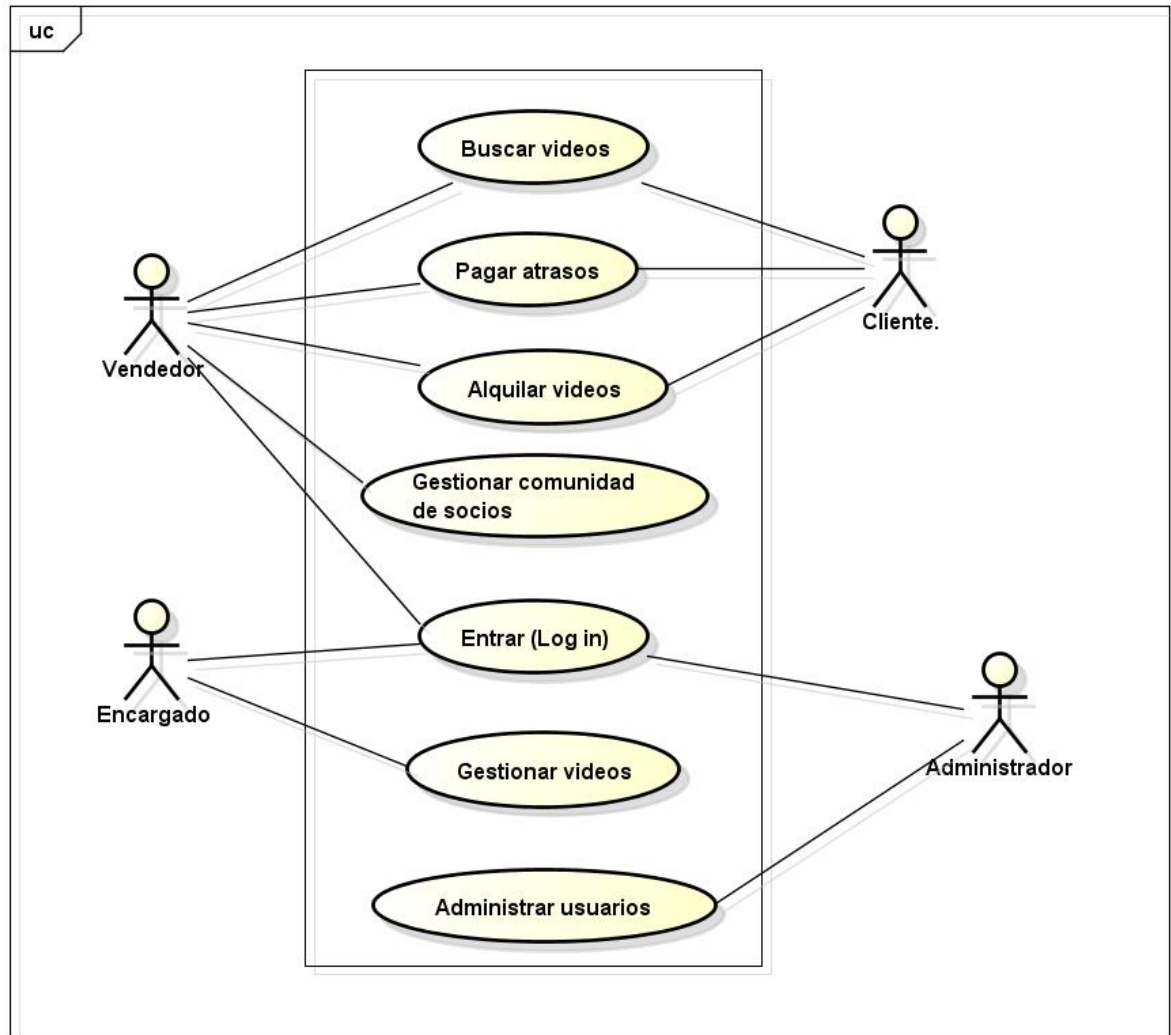
El dominio del problema es un **sistema de información para un establecimiento de alquiler de vídeo** (TAlquilerVideo). Los detalles y simplificaciones admitidas son:

- Es un establecimiento aislado y no forma parte de una organización mayor.
- Sólo alquila vídeos, no videojuegos ni vende consumibles u otros elementos.
- Un “vídeo” puede estar en alguno de estos soportes: cinta magnética o disco digital (CD o DVD, con cualquier formato).
- El precio del alquiler varía según el soporte. Por ejemplo, el alquiler de un DVD es más caro que el de una cinta.
- El establecimiento no vende nada. Por ejemplo, no hay ofertas de venta de vídeos o comida.
- Todas las transacciones son de alquiler.
- Carecen de importancia los medios por los que se suscribe a los clientes o se captura el alquiler de un vídeo.
- Sólo se admite pago en metálico.
- Al finalizar un alquiler, el cliente recibe una notificación de la transacción con información ‘típica’ —determine dicha información según su criterio—
- Cada empleado que alquila tiene una comunidad de socios propia.

3. El enunciado de cada cuestión y las respuestas. Para cada cuestión, incluirá los desarrollos, listados, diagramas y las argumentaciones que estime necesarios.

Evaluación de *Casos de Uso*

1. (0'5 puntos) Identifique al menos 4 casos de uso primarios y sus actores correspondientes. Represente los resultados en un diagrama de casos de uso de UML.



powered by Astah

En este punto, es necesario hacer algunas reflexiones. Según se desprende del enunciado, el Cliente es un actor pasivo puesto que no tiene ninguna interacción directa con el sistema. Su interacción se realiza a través del Vendedor y, en ese caso, no debería aparecer en el diagrama. Sin embargo, cabría admitir que el sistema tenga algún punto de acceso que permita al cliente, personalmente, realizar búsquedas de vídeos, seleccionarlos, pagarlos y devolverlos. Por ejemplo, un 'cajero' que funcione las 24 horas.

De cualquier forma, es el Vendedor el que gestiona su propia comunidad de socios cuando da de alta a un nuevo cliente. La comunidad a la que pertenece es una característica inherente

al cliente, junto con el resto de sus datos, e identifica al Vendedor que le atiende.

2. (1 punto) De los casos de uso anteriores, escriba el que le parezca principal <<Alquilar Vídeos>> en un estilo y formato completo, esencial y breve. Incluya tanto el escenario principal de éxito (flujo básico) como 2 extensiones o flujos alternativos que pudieran ser frecuentes. No escriba un encabezamiento demasiado elaborado del caso de uso (es decir, omita *propósito, resumen...*); en su lugar, afronte directamente el transcurso típico de los acontecimientos.

Caso de uso: **Alquilar Vídeos**

Estilo informal, extendido y esencial.

Evolución típica de los acontecimientos

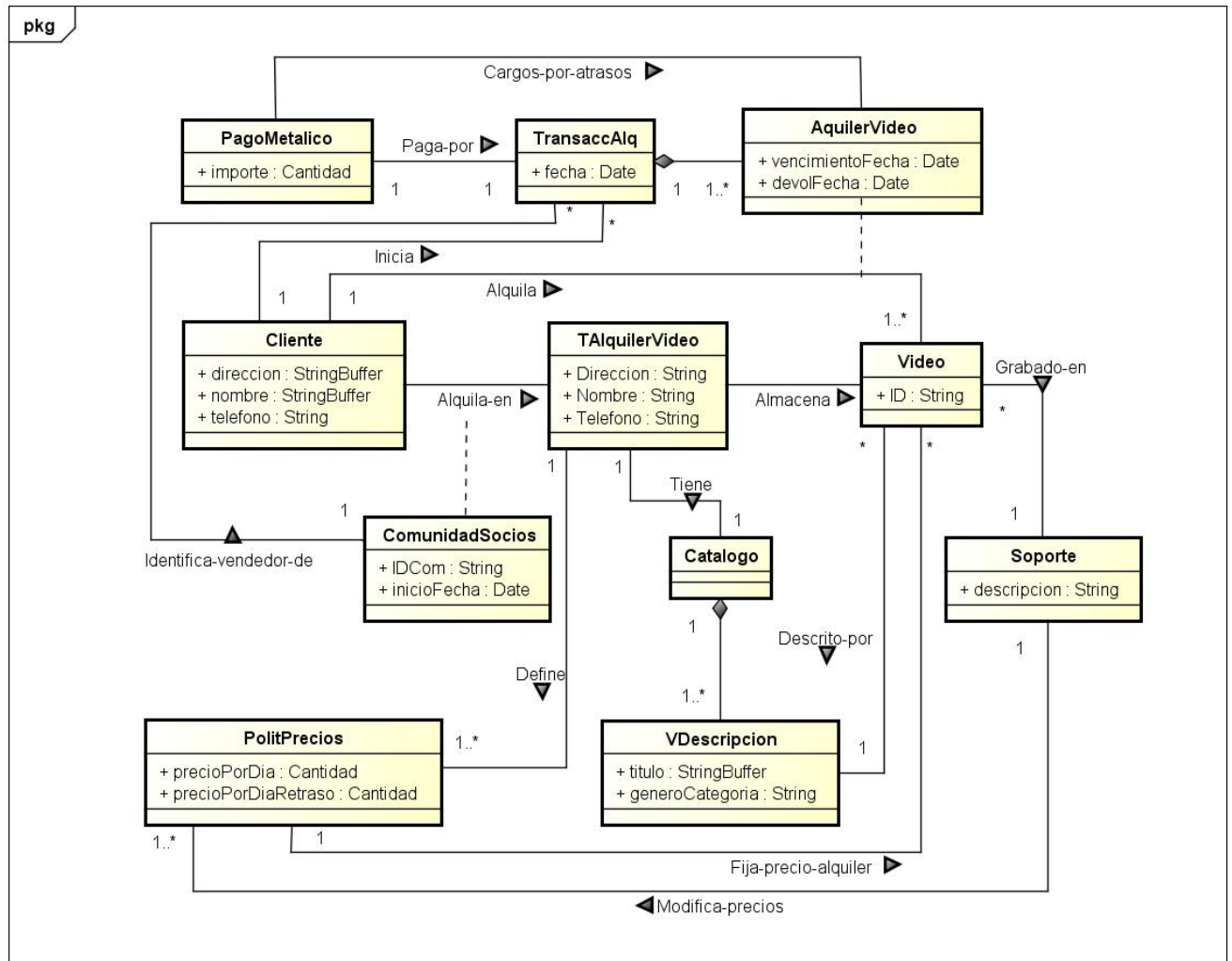
Acciones del actor	Respuesta del sistema
1. El caso de uso comienza cuando un cliente se acerca al mostrador con videos para alquilar.	
2. El cliente presenta su identificación de comunidad de cliente al vendedor, que la introduce en el sistema.	3. Presenta la información en la comunidad y la situación de los préstamos (normalmente sin préstamos ni pagos pendientes).
4. Para cada vídeo, el vendedor registra la identificación del artículo en el sistema.	5. Presenta la lista acumulada con los títulos de los videos alquilados, fechas de devolución y coste total.
6. El vendedor informa del coste total y solicita el pago al cliente.	
7. El cliente paga en metálico al vendedor.	
8. El vendedor registra el pago en el sistema.	9. Se genera el recibo e informe del préstamo.
10. El vendedor da al cliente el recibo y el informe del préstamo, que se va con los artículos alquilados.	

Alternativas

- Paso 3: El cliente tiene cuentas sin pagar. Debe pagarlas antes de proseguir el préstamo.
- Paso 6. El cliente no tiene suficiente dinero en metálico. O se cancela la transacción o se deducen artículos hasta que pueda pagarlo.

Evaluación del **Modelado Conceptual**

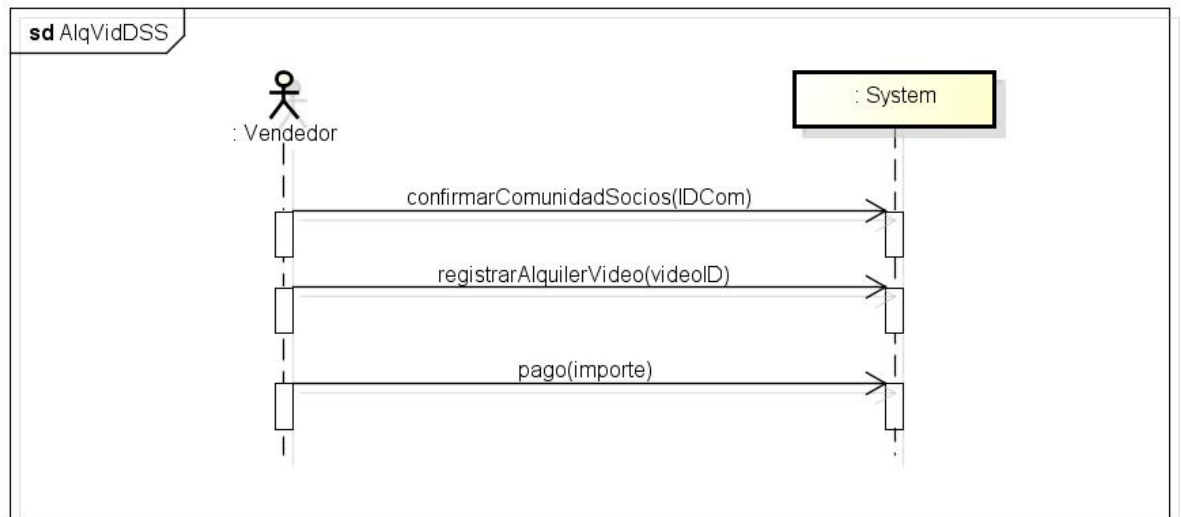
3. (2 puntos) En relación al caso de uso anterior <<Alquilar Videos>>, construya un Modelo de Dominio y representelo en notación UML. Represente los objetos conceptuales, asociaciones y atributos.



powered by Astah

Evaluación de los **Eventos del Sistema**

4. (0'5 puntos) Circunscrito al caso de uso anterior <<Alquilar Videos>>, construya un Diagrama de Secuencia del Sistema (DSS) en UML. Represente los actores y los eventos del sistema. SA partir de este DSS, especifique los contratos de dos operaciones principales: <<confirmarComunidadSocios>> y <<registrarAlquilerVideo>>.



powered by Astah

Operación *confirmarComunidad*

Contrato

Nombre: confirmarComunidadSocios (IDCom)

Responsabilidades: Confirma la validez de la pertenencia a la Comunidad —es decir, que es válida y que no ha sido anulada—. Devuelve una confirmación e informa de que no haya pagos o deudas pendientes.

Referencias cruzadas: Caso de Uso: Alquilar Videos

Precondiciones: El sistema no tiene en proceso otras transacciones de alquiler.

Postcondiciones:

Si pertenece a la Comunidad de Socios:

- Se crea e inicializa una *TransaccAlq* llamada *txn*.
- Se asocia la *ComunidadSocios* (correspondiente a *IDCom*) a la transacción *txn*.

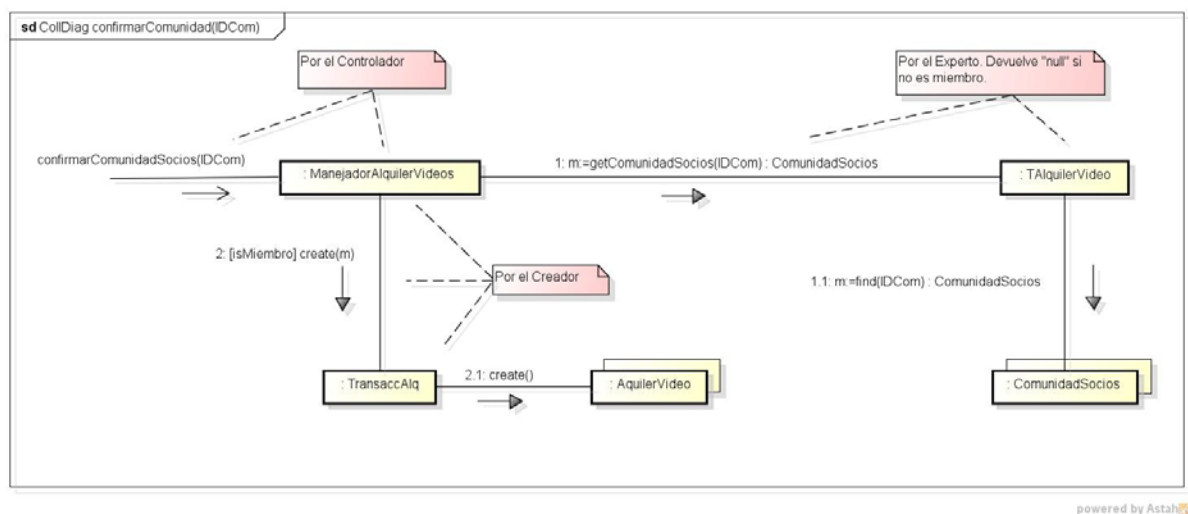
Operación registrarAlquilerVideo

Contrato

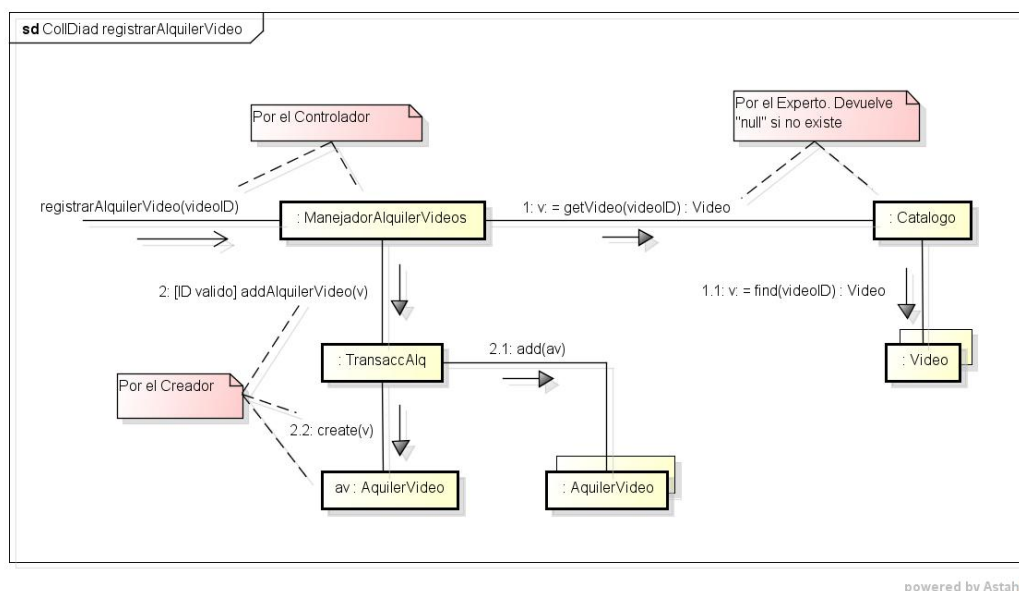
Nombre:	registrarAlquilerVideo (videoID)
Responsabilidades:	Registra el alquiler de un video y lo añade a la transacción de alquiler en curso. Presenta el título del elemento, precio y fecha de devolución.
Referencias Cruzadas:	Caso de Uso: Alquiler Videos
Precondiciones:	videoID es un dato conocido para el sistema. Hay una <i>TransaccAlq</i> que ya se ha creado y está en curso.
Postcondiciones:	<ul style="list-style-type: none">▪ Se crea e inicializa una instancia, <i>av</i>, de <i>AlquilerVideo</i>.▪ <i>av</i> se asocia a <i>Video</i>, en virtud de su correspondencia con <i>videoID</i>.▪ <i>av</i> se asocia a la <i>TransaccAlq</i> que está en curso.

Evaluación de la **Asignación de Responsabilidades** y **Diseño de Colaboraciones**

5. (2'5 puntos) A partir del contrato de la operación <<[confirmarComunidadSocios](#)>> del punto 4, complete el diagrama de colaboración en UML. Consigne cada mensaje con los patrones GRASP (Experto, Creador, etc.) o cualquier otro que lo justifique. Si añade responsabilidades no explicitadas en el contrato (porque crea que es importante señalarlas), explíquelas brevemente.

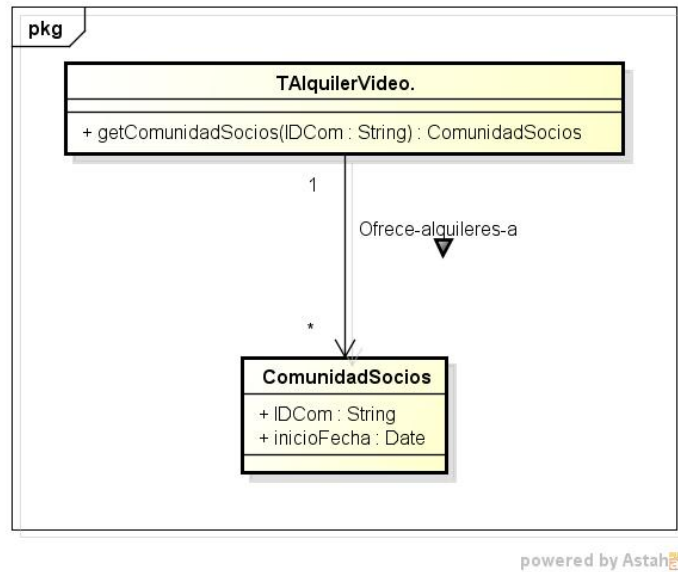


6. (2'5 puntos) A partir del contrato de la operación <<[registrarAlquilerVideo](#)>> del punto 4, complete el diagrama de colaboración en UML. Consigne cada mensaje con los patrones GRASP (Experto, Creador, etc.) o cualquier otro que lo justifique. Si añade responsabilidades no explicitadas en el contrato (porque crea que es importante señalarlas), explíquelas brevemente.



Evaluación de los **Diagramas de Clases** de diseño

7. (0'5 puntos) Elabore un diagrama de clases parcial: sólo para las clases <<TAquilerVideo>> y <<ComunidadSocios>>, analizadas en el Modelo de Dominio. Represente los nombres de todos sus atributos, asociaciones (con la navegabilidad) y métodos.



Evaluación de la ***Transformación del Diseño en Código***

8. (0'5 puntos) A partir de los anteriores diagramas de clases y colaboraciones, elabore y defina la clase <<*TAlquilerVideo*>>. Incluya las definiciones de todas variables que la componen (miembros), pero escriba solamente la definición completa del cuerpo para el método (función miembro): <<*getComunidadSocios*>>. Ignore los pequeños detalles de sintaxis —el objetivo es evaluar la capacidad fundamental para transformar el diseño en código—. Utilice la sintaxis de Java.

```
// Java
```

```
public class TAlquilerVideo
{

    private Hashtable comunidades = new Hashtable();

    public ComunidadSocios getComunidadSocios( String IDCom )
    {
        return comunidades.get( IDCom );
    }

} // fin de clase
```