

Metodología para la Elicitación de Requisitos de Sistemas Software

Versión 2.3

Amador Durán Toro

Beatriz Bernárdez Jiménez



**UNIVERSIDAD
de SEVILLA**

Informe Técnico LSI-2000-10 (revisado)

Departamento de Lenguajes y Sistemas Informáticos

Escuela Técnica Superior de Ingeniería Informática

Sevilla, abril de 2002

Este trabajo ha sido o está siendo financiado por los siguientes proyectos:

Proyecto CICYT "MENHIR"(TIC 97-0593-C05-03)

Proyecto CICYT "GEOZOCO"(TIC 2000-1106-C02-01)

Proyecto CYTED "WEST"(VII.18)

Lista de cambios

Núm.	Fecha	Descripción	Autor/es
0	13/10/1998	Versión 1.0	A. Durán y B. Bernárdez
1	04/11/1998	[pág. 18] En el apartado de importancia de la plantilla general de requisitos del sistema, donde aparecía " <i>En este apartado se indica la importancia que tiene el caso de uso para el cliente</i> " aparece ahora " <i>En este apartado se indica la importancia que tiene el requisito para el cliente</i> ".	A. Durán
2	04/11/1998	[pág. 18] En el apartado de urgencia de la plantilla general de requisitos del sistema, donde aparecía "... <i>incluyendo la funcionalidad expresada en el caso de uso</i> " aparece ahora "... <i>incluyendo las capacidades expresadas en el requisito</i> ".	A. Durán
3	04/11/1998	[pág. 23] En la descripción de la relación <i>extends</i> entre casos de uso, donde aparecía " <i>En cierta forma, B completa la funcionalidad de A</i> " aparece ahora " <i>En cierta forma, A completa la funcionalidad de B</i> ".	A. Durán
4	04/11/1998	[págs. 3, 6, 8, 15, 16, 17, 18, 19, 22, 23, 24, 26, 27, 30] Correcciones ortográficas diversas.	A. Durán
5	04/11/1998	Publicación de la versión 1.1.	A. Durán
6	18/10/1999	<p>En la versión 2.0 se han introducido numerosos cambios, los principales son los siguientes:</p> <p>El título cambia de <i>Norma para la Recolección de Requisitos de un Sistema Software</i> a <i>Metodología para la Elicitación de Requisitos de Sistemas Software</i>.</p> <p>La primera sección del documento pasa de denominarse <i>Objetivo y alcance</i> a <i>Objetivo de la metodología</i>.</p> <p>Se han eliminado de la tarea 1 las referencias relativas a la construcción de modelos de análisis del dominio.</p> <p>En la tarea 2 se habla de reuniones en lugar de hacerlo específicamente de entrevistas.</p> <p>Se ha añadido la tarea 3 <i>Identificar los objetivos del sistema</i> como tarea previa a la identificación de los requisitos y se ha diseñado una plantilla específica para los objetivos.</p> <p>Se han cambiado todas las apariciones de <i>otros requisitos</i> por <i>requisitos no funcionales</i>.</p> <p>Se ha cambiado el contenido de las secciones <i>Introducción</i> y <i>Objetivos del sistema</i> del DRS.</p> <p>Se han añadido al DRS las secciones <i>Participantes en el proyecto</i> y <i>Matriz de rastreabilidad</i>.</p> <p>Se ha contemplado la posibilidad de organizar los requisitos de forma que no sean una lista plana.</p> <p>Se han añadido las descripciones de las técnicas de JAD y brainstorming.</p> <p>Se ha reescrito la descripción de la técnica de los casos de uso.</p> <p>Se ha cambiado el nombre de la relación <i>uses</i> entre casos de uso a <i>includes</i> para adaptar la notación a UML.</p> <p>Se han añadido plantillas específicas para objetivos y actores.</p> <p>Se han añadido nuevos campos y patrones-L a todas las plantillas, introduciendo el concepto de patrón lingüístico.</p> <p>Se han eliminado los subpasos de la plantilla para requisitos funcionales (casos de uso).</p> <p>Se ha reescrito la mayor parte del ejemplo de aplicación de la metodología.</p>	A. Durán
7	18/10/1999	Versión 2.0.	A. Durán
8	18/10/2000	En la versión 2.1 se han introducido algunos cambios realizados durante la elaboración de la tesis doctoral de A. Durán [Durán 2000].	A. Durán
9	18/10/2000	Versión 2.1 (Informe Técnico LSI-2000-10)	A. Durán
10	26/10/2001	En la versión 2.2 se han introducido algunos cambios realizados durante la implementación de la herramienta CASE REM. Estos cambios afectan a algunos campos de algunas plantillas, aunque no de una forma significativa.	A. Durán
11	26/10/2001	Versión 2.2.	A. Durán
12	10/04/2002	Correcciones menores	A. Durán
13	10/04/2002	Versión 2.3.	A. Durán

Índice

1. Objetivo de la metodología	1
2. Tareas recomendadas	1
2.1. Tarea 1: Obtener información sobre el dominio del problema y el sistema actual	3
2.1.1. Objetivos	3
2.1.2. Descripción	3
2.1.3. Productos internos	3
2.1.4. Productos entregables	3
2.1.5. Técnicas recomendadas	4
2.2. Tarea 2: Preparar y realizar las sesiones de elicitación/negociación	4
2.2.1. Objetivos	4
2.2.2. Descripción	4
2.2.3. Productos internos	5
2.2.4. Productos entregables	5
2.2.5. Técnicas recomendadas	5
2.3. Tarea 3: Identificar/revisar los objetivos del sistema	5
2.3.1. Objetivos	5
2.3.2. Descripción	5
2.3.3. Productos internos	6
2.3.4. Productos entregables	6
2.3.5. Técnicas recomendadas	6
2.4. Tarea 4: Identificar/revisar los requisitos de información	6
2.4.1. Objetivos	6
2.4.2. Descripción	6
2.4.3. Productos internos	7
2.4.4. Productos entregables	7
2.4.5. Técnicas recomendadas	7

2.5.	Tarea 5: Identificar/revisar los requisitos funcionales	7
2.5.1.	Objetivos	7
2.5.2.	Descripción	8
2.5.3.	Productos internos	8
2.5.4.	Productos entregables	8
2.5.5.	Técnicas recomendadas	8
2.6.	Tarea 6: Identificar/revisar los requisitos no funcionales . . .	9
2.6.1.	Objetivos	9
2.6.2.	Descripción	9
2.6.3.	Productos internos	10
2.6.4.	Productos entregables	10
2.6.5.	Técnicas recomendadas	10
3.	Productos entregables	10
3.1.	Documento de requisitos del sistema	11
3.1.1.	Portada	11
3.1.2.	Lista de cambios	11
3.1.3.	Índice	14
3.1.4.	Listas de figuras y tablas	14
3.1.5.	Introducción	14
3.1.6.	Participantes en el proyecto	14
3.1.7.	Descripción del sistema actual	14
3.1.8.	Objetivos del sistema	15
3.1.9.	Catálogo de requisitos del sistema	15
3.1.10.	Requisitos de información	15
3.1.11.	Requisitos funcionales	15
3.1.12.	Diagramas de casos de uso	15
3.1.13.	Definición de los actores	15
3.1.14.	Casos de uso del sistema	16
3.1.15.	Requisitos no funcionales	16

3.1.16. Matriz de rastreabilidad objetivos/requisitos	16
3.1.17. Glosario de términos	17
3.1.18. Conflictos pendientes de resolución	17
3.1.19. Apéndices	17
4. Técnicas	17
4.1. Entrevistas	18
4.1.1. Preparación de entrevistas	18
4.1.2. Realización de entrevistas	19
4.1.3. Análisis de las entrevistas	20
4.2. Joint Application Development	21
4.2.1. Participantes del JAD	22
4.2.2. Fases del JAD	23
4.3. Brainstorming	25
4.3.1. Fases del brainstorming	25
4.4. Casos de uso	27
4.4.1. Diagramas de casos de uso	27
4.4.2. Relaciones entre casos de uso	28
4.4.3. Organización de casos de uso	30
5. Plantillas y patrones lingüísticos para elicitación de requisitos	30
5.1. Plantilla para los objetivos del sistema	32
5.2. Plantillas para requisitos de información	34
5.3. Plantilla para actores	36
5.4. Plantilla para requisitos funcionales	37
5.5. Plantilla para requisitos no funcionales	42
5.6. Plantilla para conflictos	42
A. Ejemplo: gestión de un vídeo-club	48
A.1. Objetivos del sistema	48
A.2. Requisitos de información	49

A.3. Requisitos funcionales	52
A.3.1. Diagramas de casos de uso	52
A.3.2. Definición de actores	55
A.3.3. Casos de uso del sistema	55
A.4. Requisitos no funcionales	71

Índice de figuras

1.	Tareas de elicitación de requisitos	2
2.	Estructura del Documento de Requisitos del Sistema	12
3.	Portada del Documento de Requisitos del Sistema	13
4.	Lista de cambios del Documento de Requisitos del Sistema	13
5.	Matriz de rastreabilidad del Documento de Requisitos del Sistema	16
6.	Diagrama de casos de uso	29
7.	Representación gráfica de las relaciones <i>includes</i> y <i>extends</i>	29
8.	Representación gráfica de los paquetes de casos de uso	30
9.	La plantilla como elemento de elicitación y negociación	31
10.	Plantilla y patrones-L para objetivos	32
11.	Plantilla y patrones-L para requisitos de almacenamiento de información	35
12.	Plantilla y patrones-L para requisitos de restricciones de información	35
13.	Plantilla y patrones-L para actores	37
14.	Plantilla y patrones-L para requisitos funcionales (casos de uso)	38
15.	Plantilla y patrones-L para requisitos funcionales (forma tradicional)	39
16.	Ejemplo de caso de uso de conexión de usuario (plantilla)	41
17.	Ejemplo de caso de uso de conexión de usuario (Coleman)	41
18.	Plantilla y patrones-L para requisitos no funcionales	43
19.	Plantilla para conflictos	44
20.	Diagrama de subsistemas	52
21.	Diagrama de casos de uso del subsistema Gestión de socios	52
22.	Diagrama de casos de uso del subsistema Gestión de películas	53
23.	Diagrama de casos de uso del subsistema Gestión de alquileres	54

1. Objetivo de la metodología

El objetivo de esta metodología es la definición de las tareas a realizar, los productos a obtener y las técnicas a emplear durante la actividad de *elicitación de requisitos* de la fase de *ingeniería de requisitos* del desarrollo de software.

En esta metodología se distinguen dos tipos de productos: los productos *entregables* y los productos *no entregables* o *internos*. Los productos entregables son aquellos que se entregan oficialmente al cliente como parte del desarrollo en fechas previamente acordadas, mientras que los no entregables son productos internos al desarrollo que no se entregan al cliente.

El único producto entregable definido en esta metodología es el *Documento de Requisitos del Sistema* (DRS), definido en la sección 3.1, pág. 11.

La estructura de este documento es la siguiente: en la sección 2 se describen las tareas recomendadas, en la sección 3 se definen los productos entregables, en este caso el DRS, y por último, en la sección 4 se describen algunas de las técnicas recomendadas para obtener los productos. También se incluye como apéndice un ejemplo de aplicación de esta metodología.

2. Tareas recomendadas

Las tareas recomendadas para obtener los productos descritos en esta metodología son las siguientes:

Tarea 1: Obtener información sobre el dominio del problema y el sistema actual.

Tarea 2: Preparar y realizar las reuniones de elicitación/negociación.

Tarea 3: Identificar/revisar los objetivos del sistema.

Tarea 4: Identificar/revisar los requisitos de información.

Tarea 5: Identificar/revisar los requisitos funcionales.

Tarea 6: Identificar/revisar los requisitos no funcionales.

Tarea 7: Priorizar objetivos y requisitos.

El orden recomendado de realización para estas tareas es: 1...7, aunque las tareas 4, 5, y 6 pueden realizarse simultáneamente o en cualquier orden que se considere oportuno (ver figura 1). La tarea 1 es opcional y depende del conocimiento previo que tenga el equipo de desarrollo sobre el dominio del problema y el sistema actual.

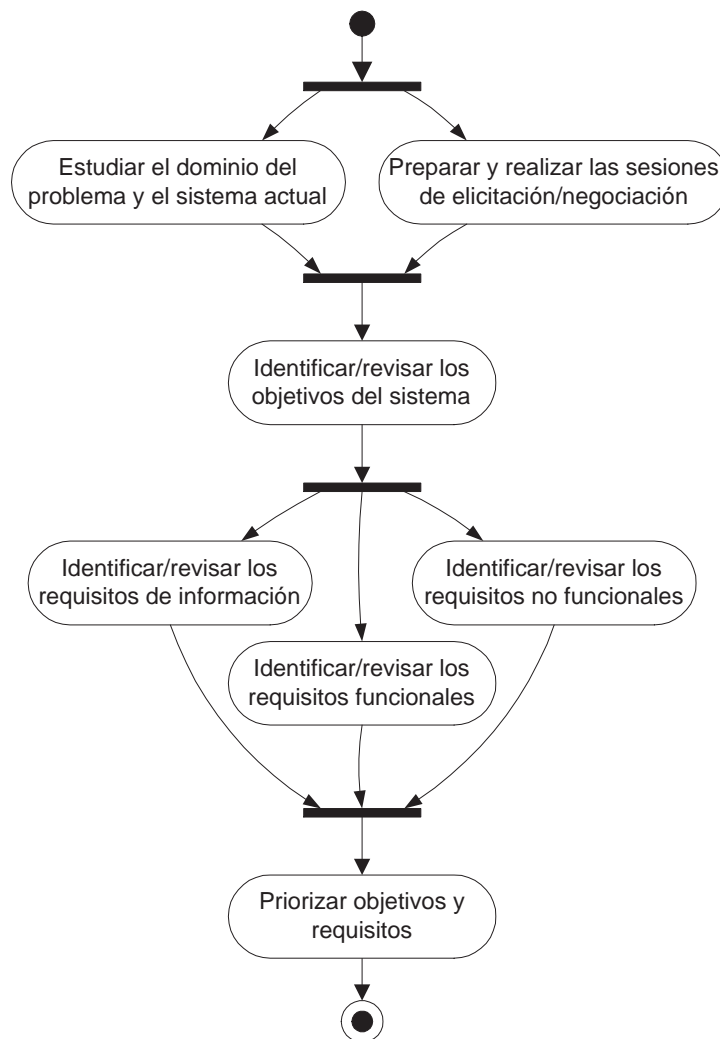


Figura 1: Tareas de elicitación de requisitos

En las siguientes secciones se describen cada una de las tareas mencionadas.

2.1. Tarea 1: Obtener información sobre el dominio del problema y el sistema actual

2.1.1. Objetivos

- Conocer el dominio del problema.
- Conocer la situación actual.

2.1.2. Descripción

Antes de mantener las reuniones con los clientes y usuarios e identificar los requisitos es fundamental conocer el dominio del problema y los contextos organizacional y operacional, es decir, la situación actual.

Enfrentarse a un desarrollo sin conocer las características principales ni el vocabulario propio de su dominio suele provocar que el producto final no sea el esperado por clientes ni usuarios.

Por otro lado, mantener reuniones con clientes y usuarios sin conocer las características de su actividad hará que probablemente no se entiendan sus necesidades y que su confianza inicial hacia el desarrollo se vea deteriorada enormemente.

Esta tarea es opcional, ya que puede que no sea necesario realizarla si el equipo de desarrollo tiene experiencia en el dominio del problema y el sistema actual es conocido.

2.1.3. Productos internos

- Información recopilada: libros, artículos, folletos comerciales, desarrollos previos sobre el mismo dominio, etc.
- Modelos del sistema actual.

2.1.4. Productos entregables

- Introducción, participantes en el proyecto, principalmente clientes y desarrolladores, descripción del sistema actual y glosario de términos como parte del DRS (ver secciones 3.1.5–3.1.7 y 3.1.17, págs. 14–14 y 17).

2.1.5. Técnicas recomendadas

- Obtener información de fuentes externas al negocio del cliente: folletos, informes sobre el sector, publicaciones, consultas con expertos, etc.

En el caso de que se trate de un dominio muy específico puede ser necesario recurrir a fuentes internas al propio negocio del cliente, en cuyo caso pueden utilizarse las técnicas auxiliares de elicitación de requisitos como el estudio de documentación, observación *in situ*, cuestionarios, inmersión o *aprendizaje*, etc. (ver secciones 4.1–4.3, págs. 18–25).

- Construcción de glosarios de términos [Leite *et al.* 1997].
- Modelado del sistema actual [Laguna *et al.* 1999, Ortín *et al.* 2001].

2.2. Tarea 2: Preparar y realizar las sesiones de elicitación/negociación

2.2.1. Objetivos

- Identificar a los usuarios participantes.
- Conocer las necesidades de clientes y usuarios.
- Resolver posibles conflictos.

2.2.2. Descripción

Teniendo en cuenta la información recopilada en la tarea anterior, en esta tarea se deben preparar y realizar las reuniones con los clientes y usuarios participantes con objeto de obtener sus necesidades y resolver posibles conflictos que se hayan detectado en iteraciones previas del proceso.

Esta tarea es especialmente crítica y ha de realizarse con especial cuidado, ya que generalmente el equipo de desarrollo no conoce los detalles específicos de la organización para la que se va a desarrollar el sistema y, por otra parte, los clientes y posibles usuarios no saben qué necesita saber el equipo de desarrollo para llevar a cabo su labor.

2.2.3. Productos internos

- Notas tomadas durante las reuniones, transcripciones o actas de reuniones, formularios, grabaciones en cinta o vídeo de las reuniones o cualquier otra documentación que se considere oportuna.

2.2.4. Productos entregables

- Participantes en el proyecto, en concreto los usuarios participantes, como parte del DRS (ver sección 3.1.6, pág. 14).
- Objetivos, requisitos o conflictos, que se hayan identificado claramente durante las sesiones de elicitación, como parte del DRS (ver secciones 3.1.8–3.1.9 y 3.1.18, págs. 15–15 y 17)

2.2.5. Técnicas recomendadas

- Técnicas de elicitación de requisitos (ver secciones 4.1–4.3, págs. 18–25), incluyendo las plantillas de objetivos, requisitos y conflictos descritas en la sección 5, pág. 30, que pueden usarse directamente durante las sesiones de elicitación.
- Técnicas de negociación como *WinWin* [Boehm *et al.* 1994].

2.3. Tarea 3: Identificar/revisar los objetivos del sistema

2.3.1. Objetivos

- Identificar los objetivos que se esperan alcanzar mediante el sistema software a desarrollar.
- Revisar, en el caso de que haya conflictos, los objetivos previamente identificados.

2.3.2. Descripción

A partir de la información obtenida en la tarea anterior, en esta tarea se deben identificar qué objetivos se esperan alcanzar una vez que el sistema software a desarrollar se encuentre en explotación o revisarlos en función

de los conflictos identificados. Puede que los objetivos hayan sido proporcionados antes de comenzar el desarrollo.

2.3.3. Productos internos

- No hay productos internos en esta tarea.

2.3.4. Productos entregables

- Objetivos del sistema como parte del DRS (ver sección 3.1.8, pág. 15).

2.3.5. Técnicas recomendadas

- Análisis de factores críticos de éxito [MAP 1995] o alguna técnica similar de identificación de objetivos.
- Plantilla para especificar los objetivos del sistema (ver sección 5.1, pág. 32).

2.4. Tarea 4: Identificar/revisar los requisitos de información

2.4.1. Objetivos

- Identificar los requisitos de almacenamiento de información que deberá cumplir el sistema software a desarrollar.
- Identificar los requisitos de restricciones de información o *reglas de negocio* que deberá cumplir el sistema software a desarrollar.
- Revisar, en el caso de que haya conflictos, los requisitos de almacenamiento y/o de restricciones de información previamente identificados.

2.4.2. Descripción

A partir de la información obtenida en la tareas 1 y 2, y teniendo en cuenta los objetivos identificados en la tarea 3 y el resto de los requisitos,

en esta tarea se debe identificar, o revisar si existen conflictos, qué información relevante para el cliente deberá gestionar y almacenar el sistema software a desarrollar así como qué restricciones o *reglas de negocio* debe cumplir dicha información.

Inicialmente se partirán de conceptos generales para posteriormente ir detallándolos hasta obtener todos los datos relevantes.

2.4.3. Productos internos

- No hay productos internos en esta tarea.

2.4.4. Productos entregables

- Requisitos de almacenamiento de información como parte del DRS (ver sección 3.1.10, pág. 15).

2.4.5. Técnicas recomendadas

- Plantilla para requisitos de almacenamiento de información (ver sección 5.2, pág. 34).
- Plantilla para requisitos de restricciones de información (ver sección 5.2, pág. 34).

2.5. Tarea 5: Identificar/revisar los requisitos funcionales

2.5.1. Objetivos

- Identificar los actores del sistema del sistema software a desarrollar.
- Identificar los requisitos funcionales, expresados de forma tradicional o como casos de uso, que deberá cumplir el sistema software a desarrollar.
- Revisar, en el caso de que haya conflictos, los requisitos funcionales previamente identificados.

2.5.2. Descripción

A partir de la información obtenida en las tareas 1 y 2, y teniendo en cuenta los objetivos identificados en la tarea 3 y el resto de los requisitos, en esta tarea se debe identificar, o revisar si existen conflictos, qué debe hacer el sistema a desarrollar con la información identificada en la tarea anterior.

Inicialmente se identificarán los actores que interactuarán con el sistema, es decir aquellas personas u otros sistemas que serán los orígenes o destinos de la información que consumirá o producirá el sistema a desarrollar y que forman su entorno.

A continuación se identificarán los casos de uso asociados a los actores, los pasos de cada caso de uso y posteriormente se detallarán los casos de uso con las posibles excepciones hasta definir todas las situaciones posibles.

En el caso de que se considere necesario, se podrá optar por expresar algunos o todos los requisitos funcionales de la forma tradicional, es decir, mediante un párrafo en lenguaje natural, en lugar de hacerlo mediante casos de uso.

2.5.3. Productos internos

- No hay productos internos en esta tarea.

2.5.4. Productos entregables

- Requisitos funcionales como parte del DRS (ver sección 3.1.11, pág. 15).

2.5.5. Técnicas recomendadas

- Casos de uso (ver sección 4.4, pág. 27).
- Plantilla para actores (ver sección 5.3, pág. 36).
- Plantilla para casos de uso (ver sección 5.4, pág. 37).
- Plantilla para requisitos funcionales (ver sección 5.4, pág. 37).

2.6. Tarea 6: Identificar/revisar los requisitos no funcionales

2.6.1. Objetivos

- Identificar los requisitos no funcionales del sistema software a desarrollar.
- Revisar, en el caso de que haya conflictos, los requisitos no funcionales previamente identificados.

2.6.2. Descripción

A partir de la información obtenida en las tareas 1 y 2, y teniendo en cuenta los objetivos identificados en la tarea 3 y el resto de los requisitos, en esta tarea se deben identificar, o revisar si existen conflictos, los requisitos no funcionales, normalmente de carácter técnico o legal.

Algunos tipos de requisitos que se suelen incluir en esta sección son los siguientes:

Requisitos de comunicaciones del sistema

Son requisitos de carácter técnico relativos a las comunicaciones que deberá soportar el sistema software a desarrollar. Por ejemplo: *el sistema deberá utilizar el protocolo TCP/IP para las comunicaciones con otros sistemas.*

Requisitos de interfaz de usuario

Este tipo de requisitos especifica las características que deberá tener el sistema en su comunicación con el usuario. Por ejemplo: *la interfaz de usuario del sistema deberá ser consistente con los estándares definidos en IBM's Common User Access.*

Se debe ser cuidadoso con este tipo de requisitos, ya que en esta fase de desarrollo todavía no se conocen bien las dificultades que pueden surgir a la hora de diseñar e implementar las interfaces, por esto no es conveniente entrar en detalles demasiado específicos.

Requisitos de fiabilidad

Los requisitos de fiabilidad deben establecer los factores que se requieren para la fiabilidad del software en tiempo de explotación. La

fiabilidad mide la probabilidad del sistema de producir una respuesta satisfactoria a las demandas del usuario. Por ejemplo: *la tasa de fallos del sistema no podrá ser superior a 2 fallos por semana.*

Requisitos de entorno de desarrollo

Este tipo de requisitos especifican si el sistema debe desarrollarse con un producto específico. Por ejemplo: *el sistema deberá desarrollarse con Oracle 7 como servidor y clientes Visual Basic 4.*

Requisitos de portabilidad

Los requisitos de portabilidad definen qué características deberá tener el software para que sea fácil utilizarlo en otra máquina o bajo otro sistema operativo. Por ejemplo: *el sistema deberá funcionar en los sistemas operativos Windows 95, Windows 98 y Windows NT 4.0, siendo además posible el acceso al sistema a través de Internet usando cualquier navegador compatible con HTML 3.0.*

2.6.3. Productos internos

- No hay productos internos en esta tarea.

2.6.4. Productos entregables

- Requisitos no funcionales del sistema como parte del DRS (ver sección 3.1.15, pág. 16).

2.6.5. Técnicas recomendadas

- Plantilla para requisitos no funcionales (ver sección 5.5, pág. 42).

3. Productos entregables

El único producto entregable que se contempla en esta metodología es el *Documento de Requisitos del Sistema* (DRS).

3.1. Documento de requisitos del sistema

La estructura del DRS puede verse en la figura 2. En las siguientes secciones se describe con detalle cada sección del DRS.

3.1.1. Portada

La portada del DRS debe tener el formato que puede verse en la figura 3. Los elementos que deben aparecer son los siguientes:

- **Nombre del proyecto:** el nombre del proyecto al que pertenece el DRS.
- **Versión:** la versión del DRS que se entrega al cliente. La versión se compone de dos números X e Y . El primero indica la versión, y se debe incrementar cada vez que se hace una nueva entrega formal al cliente. Cuando se incrementa el primer número, el segundo debe volver a comenzar en cero. El segundo número indica cambios dentro de la misma versión aún no entregada, y se debe incrementar cada vez que se publica una versión con cambios respecto a la última que se publicó y que no se vaya a entregar formalmente todavía. Este tipo de versiones pueden ser internas al equipo de desarrollo o ser entregadas al cliente a título orientativo.
- **Fecha:** fecha de la publicación de la versión.
- **Equipo de desarrollo:** nombre de la empresa o equipo de desarrollo.
- **Cliente:** nombre del cliente, normalmente otra empresa.

3.1.2. Lista de cambios

El documento debe incluir una lista de cambios en la que se especifiquen, para cada versión del documento, los cambios producidos en el mismo con un formato similar al que puede verse en la figura 4. Para cada cambio realizado se debe incluir el número de orden, la fecha, una descripción y los autores.

Portada
Lista de cambios
Índice
Lista de figuras
Lista de tablas
1. Introducción
2. Participantes en el proyecto
3. Descripción del sistema actual
4. Objetivos del sistema
5. Catálogo de requisitos del sistema
5.1 Requisitos de información
5.2 Requisitos funcionales
5.2.1 Diagramas de casos de uso
5.2.2 Definición de actores
5.2.3 Casos de uso del sistema
5.3 Requisitos no funcionales
6. Matriz de rastreabilidad objetivos/requisitos
7. Glosario de términos
8. Conflictos pendientes de resolución <i>[opcional, pueden ir en un documento aparte]</i>
Apéndices <i>[opcionales]</i>

Figura 2: Estructura del Documento de Requisitos del Sistema

Proyecto <i>nombre del proyecto</i>
Documento de Requisitos del Sistema
Versión <i>X.Y</i> Fecha <i>fecha</i>
Realizado por <i>equipo de desarrollo</i> Realizado para <i>cliente</i>

Figura 3: Portada del Documento de Requisitos del Sistema

Núm.	Fecha	Descripción	Autores
0	$fecha_0$	Versión x.y	$autor_0$
1	$fecha_1$	<i>descripción cambio</i> ₁	$autor_1$
⋮	⋮	⋮	⋮
n	$fecha_n$	<i>descripción cambio</i> _{n}	$autor_n$

Figura 4: Lista de cambios del Documento de Requisitos del Sistema

3.1.3. Índice

El índice del DRS debe indicar la página en la que comienza cada sección, subsección o apartado del documento. En la medida de lo posible, se sangrarán las entradas del índice para ayudar a comprender la estructura del documento.

3.1.4. Listas de figuras y tablas

El DRS deberá incluir listas de las figuras y tablas que aparezcan en el mismo. Dichas listas serán dos índices que indicarán el número, la descripción y la página en que aparece cada figura o tabla del DRS.

3.1.5. Introducción

Esta sección debe contener una descripción breve de las principales características del sistema software que se va a desarrollar, la situación actual que genera la necesidad del nuevo desarrollo, la problemática que se acomete, y cualquier otra consideración que sitúe al posible lector en el contexto oportuno para comprender el resto del documento.

3.1.6. Participantes en el proyecto

Esta sección debe contener una lista con todos los participantes en el proyecto, tanto desarrolladores como clientes y usuarios. Para cada participante se deberá indicar su nombre, el papel que desempeña en el proyecto, la organización a la que pertenece y cualquier otra información adicional que se considere oportuna.

3.1.7. Descripción del sistema actual

Esta sección debe contener una descripción del sistema actual en el caso de que se haya acometido su estudio. Para describir el sistema actual puede utilizarse cualquier técnica que se considere oportuno, por ejemplo las descritas en [Laguna *et al.* 1999] (*Diagrama Documentos–Tarea*, DDT) o en [Ortín *et al.* 2001] (*Diagramas de Actividad*, también descritos en [Booch *et al.* 1999]).

3.1.8. Objetivos del sistema

Esta sección debe contener una lista con los objetivos que se esperan alcanzar cuando el sistema software a desarrollar esté en explotación, especificados mediante la plantilla para objetivos descrita en la sección 5.1, pág. 32.

3.1.9. Catálogo de requisitos del sistema

Esta sección se divide en las siguientes subsecciones en las que se describen los requisitos del sistema. Cada uno de los grandes grupos de requisitos, de información, funcionales y no funcionales, podrán dividirse para ayudar a la legibilidad del documento, por ejemplo dividiendo cada subsección en requisitos asociados a un determinado objetivo, requisitos con características comunes, etc.

3.1.10. Requisitos de información

Esta subsección debe contener la lista de requisitos de almacenamiento y de restricciones de información que se hayan identificado, utilizando para especificarlos las plantillas para requisitos de información descritas en la sección 5.2, pág. 34.

3.1.11. Requisitos funcionales

Esta subsección debe contener la lista de requisitos funcionales, expresados de la forma tradicional o mediante casos de uso, que se hayan identificado, dividiéndose en los siguientes apartados que se describen a continuación.

3.1.12. Diagramas de casos de uso

Este apartado debe contener los diagramas de casos de uso del sistema que se hayan realizado.

3.1.13. Definición de los actores

Este apartado debe contener una lista con los actores que se hayan identificado, especificados mediante la plantilla para actores de casos de uso descrita en la sección 5.3, pág. 36.

3.1.14. Casos de uso del sistema

Este apartado debe contener los casos de uso que se hayan identificado, especificados mediante la plantilla para requisitos funcionales descrita en la sección 5.4, pág. 37. En el caso de que se considere oportuno, también se podrán expresar algunos o todos los requisitos funcionales de la forma tradicional, usando para ello la plantilla descrita en la sección indicada previamente.

3.1.15. Requisitos no funcionales

Esta subsección debe contener la lista los requisitos no funcionales del sistema que se hayan identificado, especificados mediante la plantilla para requisitos no funcionales descrita en la sección 5.5, pág. 42.

3.1.16. Matriz de rastreabilidad objetivos/requisitos

Esta sección debe contener una matriz *objetivo-requisito*, de forma que para cada objetivo se pueda conocer con qué requisitos está asociado. El formato de la matriz de rastreabilidad puede verse en la figura 5.

	OBJ-01	OBJ-02	...	OBJ- <i>n</i>
RI-01	•	•		
RI-02		•		
...				
RF-01	•			
RF-02	•	•		
...				
RNF-01				•
RNF-02		•		
...				

Figura 5: Matriz de rastreabilidad del Documento de Requisitos del Sistema

3.1.17. Glosario de términos

Esta sección, deberá contener una lista ordenada alfabéticamente de los términos específicos del dominio del problema, acrónimos y abreviaturas que aparezcan en el documento y que se considere que su significado deba ser aclarado. Cada término deberá acompañarse de su significado.

3.1.18. Conflictos pendientes de resolución

Esta sección, que se incluirá en el caso de que no se opte por registrar los conflictos en un documento aparte, deberá contener los conflictos identificados durante el proceso y que aún están pendientes de resolución, descritos mediante la plantilla para conflictos.

3.1.19. Apéndices

Los apéndices se usarán para proporcionar información adicional a la documentación obligatoria del documento. Sólo deben aparecer si se consideran oportunos y se identificarán con letras ordenadas alfabéticamente: A, B, C, etc.

4. Técnicas

A continuación, se describen algunas de las técnicas que se proponen en esta metodología para obtener los productos de las tareas que se han descrito.

Las técnicas más habituales en la elicitación de requisitos son las entrevistas, el *Joint Application Development* (JAD) o Desarrollo Conjunto de Aplicaciones, el *brainstorming* o tormenta de ideas y la utilización de *escenarios* [Weidenhaput *et al.* 1998, Rolland *et al.* 1998], más conocidos como *casos de uso* [Jacobson *et al.* 1993, Booch *et al.* 1999].

A estas técnicas, que se describen en los siguientes apartados, se las suele apoyar con otras técnicas complementarias como la observación *in situ*, el estudio de documentación, los cuestionarios, la *inmersión* en el negocio del cliente [Goguen y Linde 1993] o haciendo que los ingenieros de requisitos sean *aprendices* del cliente [Beyer y Holtzblatt 1995].

4.1. Entrevistas

Las entrevistas son la técnica de elicitación más utilizada, y de hecho son prácticamente inevitables en cualquier desarrollo ya que son una de las formas de comunicación más *naturales* entre personas.

En las entrevistas, y en otras técnicas de interacción, se pueden identificar tres fases: *preparación*, *realización* y *análisis* [Piattini *et al.* 1996].

4.1.1. Preparación de entrevistas

Las entrevistas no deben improvisarse, por lo que conviene realizar las siguientes tareas previas:

- **Estudiar el dominio del problema:** conocer las categorías y conceptos de la comunidad de clientes y usuarios es fundamental para poder entender las necesidades de dicha comunidad y su forma de expresarlas [Goguen y Linde 1993], y para generar en los clientes y usuarios la confianza de que el ingeniero de requisitos *entiende* sus problemas.

Para conocer el dominio del problema se puede recurrir a técnicas de estudio de documentación, a bibliografía sobre el tema, documentación de proyectos similares realizados anteriormente, la *inmersión* dentro de la organización para la que se va a desarrollar [Goguen y Linde 1993] o a periodos de *aprendizaje* por partes de los ingenieros de requisitos [Beyer y Holtzblatt 1995].

- **Seleccionar a las personas a las que se va a entrevistar:** se debe minimizar el número de entrevistas a realizar, por lo que es fundamental seleccionar a las personas a entrevistar. Normalmente se comienza por los directivos, que pueden ofrecer una visión global, y se continúa con los futuros usuarios, que pueden aportar información más detallada, y con el personal técnico, que aporta detalles sobre el entorno operacional de la organización.

Tal como se recomienda en [Piattini *et al.* 1996], conviene también estudiar el perfil de los entrevistados, buscando puntos en común con el entrevistador que ayuden a *romper el hielo*.

- **Determinar el objetivo y contenido de las entrevistas:** para minimizar el tiempo de la entrevista es fundamental fijar el objetivo que se pretende alcanzar y determinar previamente su contenido.

Previamente a su realización, se pueden enviar cuestionarios que los futuros entrevistados deben rellenar y devolver, y un pequeño documento de introducción al proyecto de desarrollo, de forma que el entrevistado conozca los temas que se van a tratar y el entrevistador recoja información para preparar la entrevista.

Es importante que los cuestionarios, si se usan, se preparen cuidadosamente teniendo en cuenta quién los va a responder y no incluir conceptos que se asuman conocidos cuando puedan no serlo.

- **Planificar las entrevistas:** la fecha, hora, lugar y duración de las entrevista deben fijarse teniendo en cuenta siempre la agenda del entrevistado.

En general, se deben buscar sitios agradables donde no se produzcan interrupciones y que resulten *naturales* a los entrevistados, tal como se describe en [Goguen y Linde 1993].

4.1.2. Realización de entrevistas

Dentro de la realización de las entrevistas se distinguen tres etapas, tal como se expone en [Piattini *et al.* 1996]:

1. **Apertura:** el entrevistador debe presentarse e informar al entrevistado sobre la razón de la entrevista, qué se espera conseguir, cómo se utilizará la información, la mecánica de las preguntas, etc.

Si se va a utilizar algún tipo de notación gráfica o matemática que el entrevistado no conozca debe explicarse antes de utilizarse. Es fundamental causar buena impresión en los primeros minutos.

2. **Desarrollo:** la entrevista en si no debería durar más de dos horas, distribuyendo el tiempo en un 20 % para el entrevistador y un 80 % para el entrevistado.

Se deben evitar los monólogos y mantener el control por parte del entrevistador, contemplando la posibilidad de que una tercera persona tome notas durante la entrevista o grabar la entrevista en cinta de vídeo o audio, siempre que el entrevistado esté de acuerdo [Robertson y Robertson 1999].

Durante esta fase se pueden emplear distintas técnicas:

- **Preguntas abiertas:** también denominadas *de libre contexto* [Gause y Weinberg 1989], estas preguntas no pueden responderse con un "sí" o un "no", permiten una mayor comunicación y evitan la sensación de interrogatorio. Por ejemplo, "*¿Qué se hace para registrar un pedido?*", "*Dígame qué se debe hacer cuando un cliente pide una factura*" o "*¿Cómo se rellena un albarán?*".

Estas preguntas se suelen utilizar al comienzo de la entrevista, pasando posteriormente a preguntas más concretas.

En general, se debe evitar la tendencia a anticipar una respuesta a las preguntas que se formulan [Raghavan *et al.* 1994]. En [Gause y Weinberg 1989, cap. 6] se exponen interesantes ejemplos de este tipo de preguntas y consejos para su utilización.

Una posibilidad es utilizar las plantillas descritas en la sección 5, pág. 30, como mecanismos tanto de obtención de información, ya que su estructura indica la información a buscar, como de registro de las respuestas a este tipo de preguntas.

- **Utilizar palabras apropiadas:** se deben evitar tecnicismos que no conozca el entrevistado y palabras o frases que puedan perturbar emocionalmente la comunicación [Goleman 1996, Goleman 1999].

- **Mostrar interés en todo momento:** es fundamental cuidar la *comunicación no verbal* [Davis 1985] durante la entrevista: tono de voz, movimiento, expresión facial, etc.

Por ejemplo, para animar a alguien a hablar puede asentirse con la cabeza, decir "*ya entiendo*", "*sí*", repetir algunas respuestas dadas, hacer pausas, poner una postura de atención, etc. Debe evitarse bostezar, reclinarse en el sillón, mirar hacia otro lado, etc.

3. **Terminación:** al terminar la entrevista se debe recapitular para confirmar que no ha habido confusiones en la información recogida, agradecer al entrevistado su colaboración y citarle para una nueva entrevista si fuera necesario, dejando siempre abierta la posibilidad de volver a contactar para aclarar dudas que surjan al estudiar la información o al contrastarla con otros entrevistados.

4.1.3. Análisis de las entrevistas

Una vez realizada la entrevista es necesario leer las notas tomadas, pasarlas a limpio, reorganizar la información, contrastarla con otras entrevistas o fuentes de información, etc.

Una vez elaborada la información, se puede enviar al entrevistado para confirmar los contenidos. También es importante evaluar la propia entrevista para determinar los aspectos mejorables.

4.2. Joint Application Development

La técnica denominada *JAD* (*Joint Application Development, Desarrollo Conjunto de Aplicaciones*), desarrollada por IBM en 1977, es una alternativa a las entrevistas individuales que se desarrolla a lo largo de un conjunto de reuniones en grupo durante un periodo de 2 a 4 días. En estas reuniones se ayuda a los clientes y usuarios a formular problemas y explorar posibles soluciones, involucrándolos y haciéndolos sentirse partícipes del desarrollo.

Esta técnica se base en cuatro principios [Raghavan *et al.* 1994]: dinámica de grupo, el uso de ayudas visuales para mejorar la comunicación (diagramas, transparencias, multimedia, herramientas CASE, etc.), mantener un proceso organizado y racional y una filosofía de documentación *WYSIWYG* (*What You See Is What You Get, lo que se ve es lo que se obtiene*), por la que durante las reuniones se trabaja directamente sobre los documentos a generar.

El JAD tiene dos grandes pasos, el *JAD/Plan* cuyo objetivo es elicitar y especificar requisitos, y el *JAD/Design*, en el que se aborda el diseño del software. En este documento sólo se verá con detalle el primero de ellos.

Debido a las necesidades de organización que requiere y a que no suele adaptarse bien a los horarios de trabajo de los clientes y usuarios, esta técnica no suele emplearse con frecuencia, aunque cuando se aplica suele tener buenos resultados, especialmente para elicitar requisitos en el campo de los sistemas de información [Raghavan *et al.* 1994].

En comparación con las entrevistas individuales, presenta las siguientes ventajas:

- Ahorra tiempo al evitar que las opiniones de los clientes se contrasten por separado.
- Todo el grupo, incluyendo los clientes y los futuros usuarios, revisa la documentación generada, no sólo los ingenieros de requisitos.
- Implica más a los clientes y usuarios en el desarrollo.

4.2.1. Participantes del JAD

Tal como se expone en [Raghavan *et al.* 1994], se pueden distinguir seis clases de participantes o roles en el JAD:

- **Jefe del JAD:** es el responsable de todo el proceso y asume el control durante las reuniones. Debe tener dotes de comunicación y liderazgo. Algunas habilidades importantes que debe tener son: entender y promover la dinámica de grupo, iniciar y centrar discusiones, reconocer cuándo la reunión se está desviando del tema y reconducirla, manejar las distintas personalidades y formas de ser de los participantes, evitar que decaiga la reunión aunque sea larga y difícil, etc.
- **Analista:** es el responsable de la producción de los documentos que se deben generar durante las sesiones JAD. Debe tener la habilidad de organizar bien las ideas y expresarlas claramente por escrito. En el caso de que se utilizan herramientas software durante las sesiones, debe ser capaz de manejarlas eficientemente.
- **Patrocinador ejecutivo:** es el que tiene la decisión final de que se lleve a cabo el desarrollo. Debe proporcionar a los demás participantes información sobre la necesidad del nuevo sistema y los beneficios que se espera obtener de él.
- **Representantes de los usuarios:** durante el *JAD/Plan*, suelen ser directivos con una visión global del sistema. Durante el *JAD/Design* suelen incorporarse futuros usuarios finales.
- **Representantes de sistemas de información:** son personas expertos en sistemas de información que deben ayudar a los usuarios a comprender qué es o no factible con la tecnología actual y el esfuerzo que implica.
- **Especialistas:** son personas que pueden proporcionar información detallada sobre aspectos muy concretos, tanto del punto de vista de los usuarios porque conocen muy bien el funcionamiento de una parte de la organización, como desde el punto de vista de los desarrolladores porque conocen perfectamente ciertos aspectos técnicos de la instalación hardware de la organización.

4.2.2. Fases del JAD

Dentro de la técnica del JAD se distinguen tres fases [Raghavan *et al.* 1994]:

1. **Adaptación:** es responsabilidad del jefe del JAD, ayudado por uno o dos analistas, adaptar la técnica del JAD para cada proyecto. La adaptación debe comenzar por definir el proyecto a alto nivel, para lo cual pueden ser necesarias entrevistas previas con algunos clientes y usuarios. También suele ser necesario recabar información sobre la organización para familiarizarse con el dominio del problema, por ejemplo utilizando técnicas complementarias como el estudio de documentación o la observación *in situ*.

Una vez obtenida una primera idea de los objetivos del proyecto, es necesario seleccionar a los participantes, citarles para las reuniones y proporcionarles una lista con los temas que se van a tratar en las reuniones para que las puedan preparar.

El jefe del JAD debe decidir la duración y el número de sesiones a celebrar, definir el formato de la documentación sobre la que se trabajará y preparar transparencias introductorias y todo el material audiovisual que considere oportuno.

2. **Celebración de las sesiones JAD:** durante las sesiones, los participantes exponen sus ideas y se discuten, analizan y refinan hasta alcanzar un acuerdo. Los pasos que se recomienda seguir para este proceso son los siguientes:
 - a) **Presentación:** se presenta y se da la bienvenida a todos los participantes por parte del patrocinador ejecutivo y del jefe del JAD. El patrocinador ejecutivo expone brevemente las necesidades que han llevado al desarrollo y los beneficios que se esperan obtener. El jefe del JAD explica la mecánica de las sesiones y la planificación prevista.
 - b) **Definir objetivos y requisitos:** el jefe del JAD promueve la discusión para elicitación los objetivos o requisitos de alto nivel mediante preguntas como: "*¿Por qué se construye el sistema?*", "*¿Qué beneficios se esperan del nuevo sistema?*", "*¿Cómo puede beneficiar a la organización en el futuro?*", "*¿Qué restricciones de recursos disponibles, normas o leyes afectan al proyecto?*", "*¿Es importante la seguridad de los datos?*", ...

A medida que se van elicitando requisitos, el analista los escribe en transparencias o en algún otro medio que permita que permanezcan visibles durante la discusión. Una posibilidad es utilizar para ello las plantillas descritas en la sección 5, pág. 30.

- c) **Delimitar el ámbito del sistema:** una vez obtenido un número importante de requisitos, es necesario organizarlos y llegar a un acuerdo sobre el ámbito del nuevo sistema.

En el caso de los sistemas de información, es útil identificar a los usuarios potenciales (*actores*) y determinar qué tareas les ayudará a realizar (*casos de uso*).

- d) **Documentar temas abiertos:** aquellas cuestiones que hayan surgido durante la sesión que no se han podido resolver, deben documentarse para las siguientes sesiones y ser asignadas a una persona responsable de su solución para una fecha determinada, para lo cual puede utilizarse la plantilla de conflictos descrita en la sección 5.6, pág. 42.

- e) **Concluir la sesión:** el jefe del JAD concluye la sesión revisando con los demás participantes la información elicitada y las decisiones tomadas. Se da la oportunidad a todos los participantes de expresar cualquier consideración adicional, fomentando por parte del jefe del JAD el sentimiento de propiedad y compromiso de todos los participantes sobre los requisitos elicitados.

- 3. **Conclusión:** una vez terminadas las sesiones es necesario transformar las transparencias, notas y demás documentación generada en documentos formales. Se distinguen tres pasos:

- a) **Completar la documentación:** los analistas recopilan la documentación generada durante las sesiones en documentos conformes a las normas o estándares vigentes en la organización para la que se desarrolla el proyecto.
- b) **Revisar la documentación:** la documentación generada se envía a todos los participantes para que la comenten. Si los comentarios son lo suficientemente importantes, se convoca otra reunión para discutirlos.
- c) **Validar la documentación:** una vez revisados todos los comentarios, el jefe del JAD envía el documento al patrocinador ejecutivo para su aprobación. Una vez aprobado el documento se envían copias definitivas a cada uno de los participantes.

4.3. Brainstorming

El *brainstorming* o tormenta de ideas es una técnica de reuniones en grupo cuyo objetivo es la generación de ideas en un ambiente libre de críticas o juicios [Gause y Weinberg 1989, Raghavan *et al.* 1994]. Las sesiones de brainstorming suelen estar formadas por un número de cuatro a diez participantes, uno de los cuales es el *jefe* de la sesión, encargado más de comenzar la sesión que de controlarla.

Como técnica de elicitación de requisitos, el brainstorming puede ayudar a generar una gran variedad de vistas del problema y a formularlo de diferentes formas, sobre todo al comienzo del proceso de elicitación, cuando los requisitos son todavía muy difusos.

Frente al JAD, el brainstorming tiene la ventaja de que es muy fácil de aprender y requiere poca organización, de hecho, hay propuestas de realización de brainstorming por vídeo-conferencia a través de Internet [Raghavan *et al.* 1994]. Por otro lado, al ser un proceso poco estructurado, puede no producir resultados con la misma calidad o nivel de detalle que otras técnicas.

4.3.1. Fases del brainstorming

En el brainstorming se distinguen las siguientes fases [Raghavan *et al.* 1994]:

1. **Preparación:** la preparación para una sesión de brainstorming requiere que se seleccione a los participantes y al jefe de la sesión, citarlos y preparar la sala donde se llevará a cabo la sesión. Los participantes en una sesión de brainstorming para elicitación de requisitos son normalmente clientes, usuarios, ingenieros de requisitos, desarrolladores y, si es necesario, algún experto en temas relevantes para el proyecto.
2. **Generación:** el jefe abre la sesión exponiendo un enunciado general del problema a tratar, que hace de *semilla* para que se vayan generando ideas. Los participantes aportan libremente nuevas ideas sobre el problema *semilla*, bien por un orden establecido por el jefe de la sesión, bien aleatoriamente. El jefe es siempre el responsable de dar la palabra a un participante. Este proceso continúa hasta que el jefe decide parar, bien porque no se están generando suficientes ideas, en cuyo caso la reunión se pospone, bien porque el número de ideas sea

suficiente para pasar a la siguiente fase. Durante esta fase se deben observar las siguientes reglas:

- Se prohíbe la crítica de ideas, de forma que los participantes se sientan libres de formular cualquier idea.
- Se fomentan las ideas más *avanzadas*, que aunque no sean factibles, estimulan a los demás participantes a explorar nuevas soluciones más creativas.
- Se debe generar un gran número de ideas, ya que cuantas más ideas se presenten más probable será que se generen mejores ideas.
- Se debe alentar a los participantes a combinar o completar las ideas de otros participantes. Para ello, es necesario, al igual que en la técnica del JAD, que todas las ideas generadas estén visibles para todos los participantes en todo momento.

Una posibilidad es utilizar como semilla objetivos del sistema e ir identificando requisitos. Si estos requisitos se recogen en las plantillas propuestas en la sección 5, pág. 30, pueden utilizarse dichas plantillas para que los participantes tengan visibles las ideas que se van generando.

3. **Consolidación:** en esta fase se deben organizar y evaluar las ideas generadas durante la fase anterior. Se suelen seguir tres pasos:
 - a) **Revisar ideas:** se revisan las ideas generadas para clarificarlas. Es habitual identificar ideas similares, en cuyo caso se unifican en un solo enunciado.
 - b) **Descartar ideas:** aquellas ideas que los participantes consideren excesivamente *avanzadas* se descartan.
 - c) **Priorizar ideas:** se priorizan las ideas restantes, identificando las absolutamente esenciales, las que estarían bien pero que no son esenciales y las que podrían ser apropiadas para una próxima versión del sistema a desarrollar.
4. **Documentación:** después de la sesión, el jefe produce la documentación oportuna conteniendo las ideas priorizadas y comentarios generados durante la consolidación.

4.4. Casos de uso

Los casos de uso son una técnica para la especificación de requisitos funcionales propuesta inicialmente en [Jacobson *et al.* 1993] y que actualmente forma parte de la propuesta de UML [Booch *et al.* 1999].

Un caso de uso es la descripción de una secuencia de interacciones entre el sistema y uno o más *actores* en la que se considera al sistema como una caja negra y en la que los actores obtienen resultados observables.

Los actores son personas u otros sistemas que interactúan con el sistema cuyos requisitos se están describiendo [Schneider y Winters 1998].

Los casos de uso presentan ciertas ventajas sobre la descripción meramente textual de los requisitos funcionales [Firesmith 1997], ya que facilitan la elicitación de requisitos y son fácilmente comprensibles por los clientes y usuarios. Además, pueden servir de base a las pruebas del sistema y a la documentación para los usuarios [Weidenhaput *et al.* 1998].

A pesar de ser una técnica ampliamente aceptada, existen múltiples propuestas para su utilización concreta [Cockburn 1997]. En esta metodología se propone la utilización de los casos de uso como técnica tanto de elicitación como de especificación de los requisitos funcionales del sistema. Para la descripción concreta de los casos de uso se proponen plantillas, en las que las interacciones se numeran siguiendo las propuestas de [Cockburn 1997], [Schneider y Winters 1998] y [Coleman 1998] y se describen usando lenguaje natural en forma de patrones lingüísticos (ver sección 5.4, pág. 37).

4.4.1. Diagramas de casos de uso

Los casos de uso tienen una representación gráfica en los denominados *diagramas de casos de uso* [Booch *et al.* 1999]. En estos diagramas, los actores se representan en forma de pequeños monigotes y los casos de uso se representan por elipses contenidas dentro de un rectángulo que representa al sistema. La participación de los actores en los casos de uso se indica por una flecha entre el actor y el caso de uso que apunta en la dirección en la que fluye la información. Un ejemplo de este tipo de diagramas puede verse en la figura 6.

Los diagramas de casos de uso sirven para proporcionar una visión global del conjunto de casos de uso de un sistema así como de los actores

y los casos de uso en los que éstos intervienen. Las interacciones concretas entre los actores y el sistema no se muestran en este tipo de diagramas.

4.4.2. Relaciones entre casos de uso

A veces conviene establecer relaciones entre distintos casos de uso para simplificar su descripción. Las dos relaciones posibles y sus semánticas según [Booch *et al.* 1999] son las siguientes, cuya representación gráfica puede verse en el ejemplo de la figura 7.

- **includes:** se dice que un caso de uso *A incluye* el caso de uso *B*, cuando *B* es una parte del caso de uso *A*, es decir, la secuencia de interacciones de *B* forma parte de la secuencia de interacciones de *A*.

El caso de uso *B* se realiza siempre dentro del caso de uso *A*. Además, siempre que ocurre *A* ocurre también *B*, por lo que se dice que *B* es un caso de uso *abstracto* [Jacobson *et al.* 1997, Firesmith 1997]. Un caso de uso es abstracto si no puede ser realizado por sí mismo, por lo que sólo tiene significado cuando se utiliza para describir alguna funcionalidad que es común a otros casos de uso. Por otra parte, un caso de uso será *concreto* si puede ser iniciado por un actor y realizado por sí mismo.

Se suele utilizar esta relación cuando se detectan subsecuencias de interacciones comunes a varios casos de uso. Dichas subsecuencias comunes se sacan "factor común" de los casos de uso que las contienen y se les da forma de casos de uso que son *incluidos* por los casos de uso de los que se han "extraído". De esta forma se evita repetir las mismas subsecuencias de interacciones una y otra vez en varios casos de uso.

- **extends:** un caso de uso *A extiende* a otro caso de uso *B* cuando *A* es una subsecuencia de interacciones de *B* que ocurre en una determinada circunstancia.

En cierta forma, *A* completa la funcionalidad de *B*. El caso de uso *A* puede realizarse o no cuando se realiza el caso de uso *B*, según se den las circunstancias. Por otro lado, el caso de uso *A* puede ser un caso de uso abstracto o concreto, en cuyo caso puede ocurrir sin necesidad de que ocurra el caso de uso *B*.

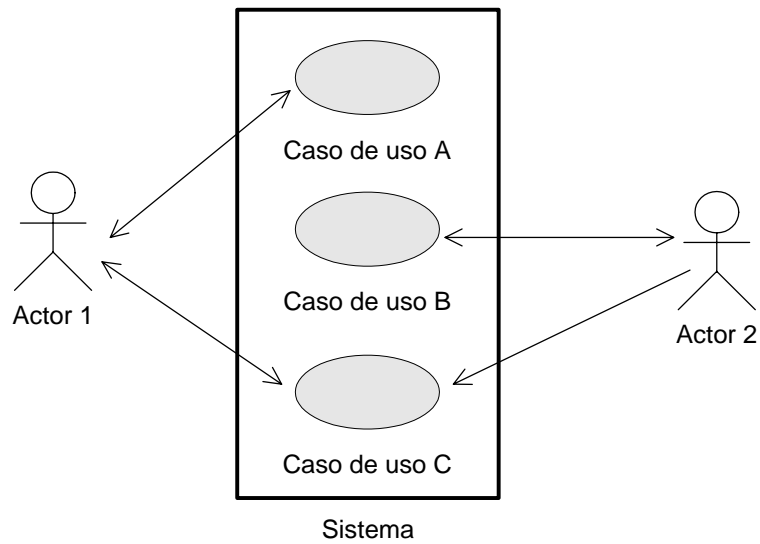
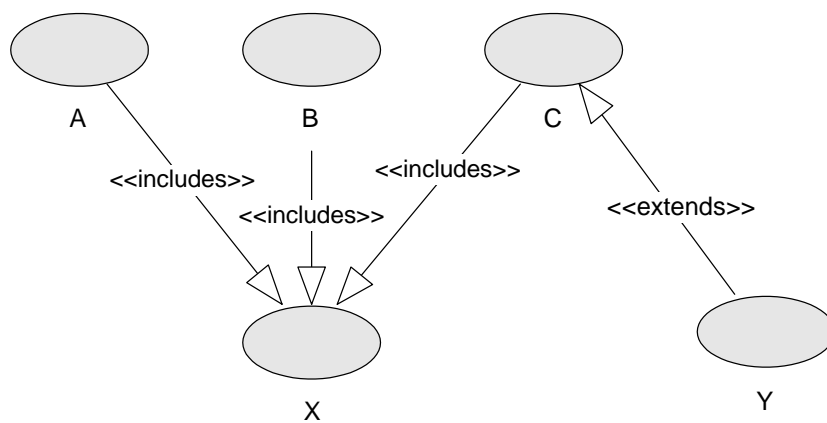


Figura 6: Diagrama de casos de uso

Figura 7: Representación gráfica de las relaciones *includes* y *extends*

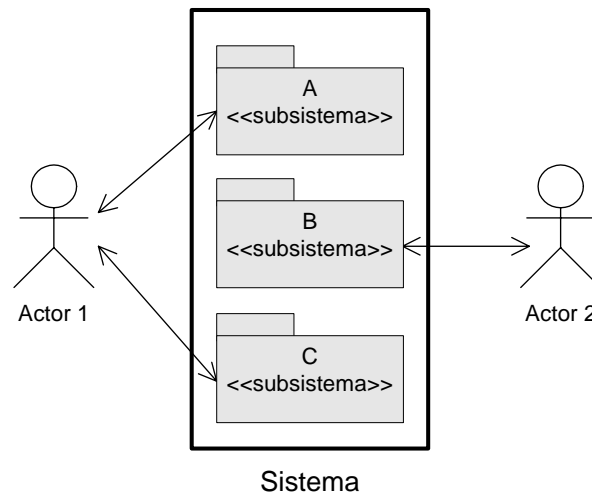


Figura 8: Representación gráfica de los paquetes de casos de uso

4.4.3. Organización de casos de uso

En la mayoría de sistemas, el número de casos de uso es lo suficientemente elevado como para que sea oportuno organizarlos de alguna forma en lugar de tener una *lista plana* por la que no es fácil navegar.

Una posible forma de organizar los casos de uso es recurrir a los *paquetes* descritos en la propuesta de UML [Booch *et al.* 1999]. De esta forma, los casos de uso pueden organizarse en niveles, facilitando así su comprensión. Cada paquete contiene a otros paquetes o a varios casos de uso.

En el caso de que los casos de uso se agrupen por criterios funcionales, los paquetes que los agrupan pueden estereotiparse como *subsistemas* [Schneider y Winters 1998], tal como puede verse en el ejemplo de la figura 8.

5. Plantillas y patrones lingüísticos para elicitación de requisitos

Las plantillas y patrones lingüísticos que se presentan en los siguientes apartados están pensados para utilizarse tanto durante las reuniones de elicitación con clientes y usuarios como para registrar y gestionar los requisitos [Durán 2000].

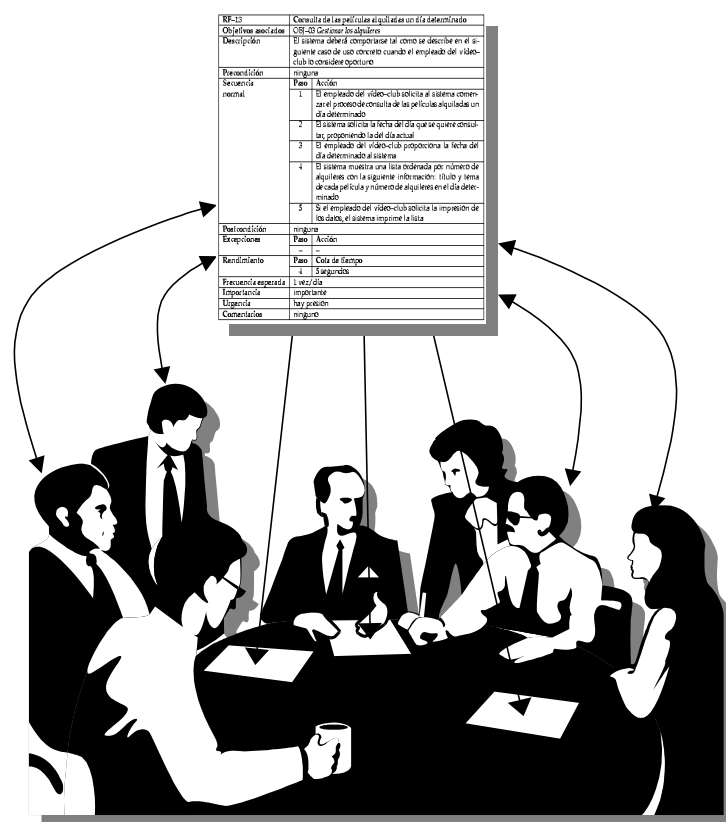


Figura 9: La plantilla como elemento de elicitación y negociación

Su objetivo es doble: por un lado intentar paliar la falta de propuestas concretas sobre la expresión de requisitos. Por otro lado, también pueden usarse como elementos de elicitación y negociación durante las reuniones con clientes y usuarios de forma similar a las conocidas tarjetas CRC (*Clase, Responsabilidad, Colaboración*) [Wirfs-Brock *et al.* 1990] (ver figura 9).

De esta forma se consigue que durante las sesiones de elicitación se trabaje con una filosofía WYSIWYG, tal como se propone en las técnicas de JAD o brainstorming, ya que los participantes manejan directamente la documentación final, favoreciéndose así su implicación en el proceso.

Como fruto de la experiencia de su utilización, para algunos campos de las plantillas se han identificado frases "estándar" que son habituales en las especificaciones de requisitos y que se han parametrizado. Estas frases, a las que hemos denominado *patrones lingüísticos*, o abreviadamente *patrones-L*, pueden usarse para rellenar los campos de las plantillas dándole valores a los parámetros con la información oportuna.

Ambos aspectos, la estructuración de la información en forma de plantilla y la propuesta de frases "estándar", facilita la redacción de los requisitos, permitiendo a los participantes en las actividades de elicitación centrarse en expresar sus necesidades y no en cómo expresarlas.

En la notación usada para describir los patrones-L, las palabras o frases entre < y > deben ser convenientemente reemplazadas, las palabras o frases que se encuentren entre { y } y separadas por comas representan opciones de las que se debe escoger una y las palabras entre [y] son opcionales, es decir, pueden aparecer o no.

En las siguientes secciones se describen las plantillas propuestas y los patrones-L identificados.

5.1. Plantilla para los objetivos del sistema

Los objetivos del sistema pueden considerarse como *requisitos de alto nivel* [Sawyer y Kontoya 1999], de forma que los requisitos propiamente dichos serían la forma de alcanzar los objetivos. La plantilla propuesta para los objetivos puede verse en la figura 10.

OBJ-<i><id></i>	<i><nombre descriptivo></i>
Versión	<i><nº de la versión actual></i> (<i><fecha de la versión actual></i>)
Autores	• <i><autor de la versión actual></i> (<i><organización del autor></i>) ...
Fuentes	• <i><fuente de la versión actual></i> (<i><organización de la fuente></i>) ...
Descripción	El sistema deberá <i><objetivo a cumplir por el sistema></i>
Subobjetivos	• OBJ-x <i><nombre del subobjetivo></i> • ...
Importancia	<i><importancia del objetivo></i>
Urgencia	<i><urgencia del objetivo></i>
Estado	<i><estado del objetivo></i>
Estabilidad	<i><estabilidad del objetivo></i>
Comentarios	<i><comentarios adicionales sobre el objetivo></i>

Figura 10: Plantilla y patrones-L para objetivos

El significado de los campos que la componen, cuya mayoría está presente también en las plantillas para los requisitos, es el siguiente:

- **Identificador y nombre descriptivo:** siguiendo la propuesta, entre otros, de [Sawyer *et al.* 1997], cada objetivo debe identificarse por un

código único y un nombre descriptivo. Con objeto de conseguir una rápida identificación, los identificadores de los objetivos comienzan con *OBJ*.

- **Versión:** para poder gestionar distintas versiones, este campo contiene el número y la fecha de la versión actual del objetivo.
- **Autores, Fuentes:** estos campos contienen el nombre y la organización de los autores (normalmente desarrolladores) y de las fuentes (clientes o usuarios), de la versión actual del objetivo, de forma que la rastreabilidad pueda llegar hasta las personas que propusieron la necesidad del objetivo.
- **Descripción:** este campo contiene un patrón-L que se debe completar con la descripción del objetivo.
- **Subobjetivos:** en este campo pueden indicarse los subobjetivos que dependen del objetivo que se está describiendo. En sistemas complejos puede ser necesario establecer una jerarquía de objetivos previa a la identificación de los requisitos. En caso de que esto no sea necesario, puede ignorarse este campo.
- **Importancia:** este campo indica la importancia del cumplimiento del objetivo para los clientes y usuarios. Se puede asignar un valor numérico o alguna expresión enumerada como *vital*, *importante* o *quedaría bien*, tal como se propone en [IBM OOTC 1997]. En el caso de que no se haya establecido aún la importancia, se puede indicar que está *por determinar* (*PD*), equivalente al TBD (*To Be Determined*) empleado en las especificaciones escritas en inglés.
- **Urgencia:** este campo indica la urgencia del cumplimiento del objetivo para los clientes y usuarios en el supuesto caso de un desarrollo incremental. Como en el caso anterior, se puede asignar un valor numérico o una expresión enumerada como *inmediatamente*, *hay presión* o *puede esperar* [IBM OOTC 1997], o *PD* en el caso de que aún no se haya determinado.
- **Estado:** este campo indica el estado del objetivo desde el punto de vista de su desarrollo. El objetivo puede estar *en construcción* si se está elaborando, *pendiente de negociación* si tiene algún conflicto asociado pendiente de solución, *pendiente de verificación* si no tiene ningún conflicto pendiente y está a la espera de verificación o, *pendiente de validación* si ya ha sido verificado y está a la espera de validación o

por último, puede estar *validado* si ya ha sido validado por clientes y usuarios.

- **Estabilidad:** este campo indica la estabilidad del objetivo, es decir una estimación de la probabilidad de que pueda sufrir cambios en el futuro. Esta estabilidad puede indicarse mediante un valor numérico o mediante una expresión enumerada como *alta*, *media* o *baja* o *PD* en el caso de que aún no se haya determinado.

La información sobre la estabilidad, bien a nivel de objetivos como en este caso, bien a nivel de requisitos, ayuda a los diseñadores a diseñar software que prevea de antemano la necesidad de posibles cambios futuros en aquellos aspectos relacionados con los elementos identificados como inestables durante la fase de ingeniería de requisitos, favoreciendo así el mantenimiento y la evolución del software [Brackett 1990].

- **Comentarios:** cualquier otra información sobre el objetivo que no encaje en los campos anteriores puede recogerse en este apartado.

5.2. Plantillas para requisitos de información

Lo más importante en los sistemas de información es precisamente la información que gestionan. Las plantillas para requisitos de almacenamiento y de restricciones de información, que pueden verse en las figuras 11 y 12, ayudan a los clientes y usuarios a responder a las preguntas "*¿qué información, relevante para los objetivos de su negocio, debe ser almacenada por el sistema?*" "*¿qué restricciones o reglas de negocio debe cumplir dicha información?*".

El significado de los campos de las plantillas es el siguiente:

- **Identificador y nombre descriptivo:** siguiendo las recomendaciones, entre otros, de [IEEE 1993] y [Sawyer *et al.* 1997], cada requisito se debe identificar por un código único y un nombre descriptivo. Con objeto de conseguir una rápida identificación, los identificadores de los requisitos de almacenamiento de información comienzan con *IRQ* y los de requisitos de restricciones de información con *CRQ*.
- **Versión, Autores, Fuentes:** estos campos tienen el mismo significado que en la plantilla para objetivos aunque referidos al requisito.

IRQ-<i><id></i>	<i><nombre descriptivo></i>	
Versión	<i><nº de la versión actual></i> (<i><fecha de la versión actual></i>)	
Autores	<ul style="list-style-type: none"> <i><autor de la versión actual></i> (<i><organización del autor></i>) ... 	
Fuentes	<ul style="list-style-type: none"> <i><fuente de la versión actual></i> (<i><organización de la fuente></i>) ... 	
Objetivos asociados	<ul style="list-style-type: none"> OBJ-x <i><nombre del objetivo></i> ... 	
Requisitos asociados	<ul style="list-style-type: none"> Rx-y <i><nombre del requisito></i> ... 	
Descripción	El sistema deberá almacenar la información correspondiente a <i><concepto relevante></i> . En concreto:	
Datos específicos	<ul style="list-style-type: none"> <i><datos específicos sobre el concepto relevante></i> ... 	
Tiempo de vida	Medio	Máximo
	<i><tiempo medio de vida></i>	<i><tiempo máximo de vida></i>
Ocurrencias simult.	Medio	Máximo
	<i><nº medio de ocurr. simult.></i>	<i><nº máximo de ocurr. simult.></i>
Importancia	<i><importancia del requisito></i>	
Urgencia	<i><urgencia del requisito></i>	
Estado	<i><estado del requisito></i>	
Estabilidad	<i><estabilidad del requisito></i>	
Comentarios	<i><comentarios adicionales sobre el requisito></i>	

Figura 11: Plantilla y patrones-L para requisitos de almacenamiento de información

CRQ-<i><id></i>	<i><nombre descriptivo></i>	
Versión	<i><nº de la versión actual></i> (<i><fecha de la versión actual></i>)	
Autores	<ul style="list-style-type: none"> <i><autor de la versión actual></i> (<i><organización del autor></i>) ... 	
Fuentes	<ul style="list-style-type: none"> <i><fuente de la versión actual></i> (<i><organización de la fuente></i>) ... 	
Objetivos asociados	<ul style="list-style-type: none"> OBJ-x <i><nombre del objetivo></i> ... 	
Requisitos asociados	<ul style="list-style-type: none"> Rx-y <i><nombre del requisito></i> ... 	
Descripción	La información almacenada por el sistema deberá satisfacer la siguiente restricción: <i><restricción o regla de negocio></i> .	
Importancia	<i><importancia del requisito></i>	
Urgencia	<i><urgencia del requisito></i>	
Estado	<i><estado del requisito></i>	
Estabilidad	<i><estabilidad del requisito></i>	
Comentarios	<i><comentarios adicionales sobre el requisito></i>	

Figura 12: Plantilla y patrones-L para requisitos de restricciones de información

- **Objetivos asociados:** este campo debe contener una lista con los objetivos a los que está asociado el requisito, es decir de los objetivos de los que depende. Esto permite conocer qué requisitos harán que el sistema a desarrollar alcance los objetivos propuestos y justifican de esta forma la existencia o propósito del requisito.
- **Requisitos asociados:** en este campo se indican otros requisitos que estén asociados por algún motivo con el requisito que se está describiendo, es decir de los requisitos de los que depende. De esta forma se posibilita tener una rastreabilidad *horizontal*, similar a las relaciones entre *assets* del mismo nivel descritas en [García 2000].
- **Descripción:** para los requisitos de almacenamiento de información este campo usa un patrón-L que se debe completar con el concepto relevante sobre el que se debe almacenar información. En el caso de los requisitos de restricciones de información, este campo usa un patrón-L que se debe completar con la restricción o regla de negocio que debe cumplir la información almacenada por el sistema.
- **Datos específicos:** este campo contiene una lista de los datos específicos asociados al concepto relevante, de los que pueden indicarse todos aquellos aspectos que se considere oportunos (descripción, restricciones, ejemplos, etc.).
- **Tiempo de vida:** este campo indica el tiempo de vida medio y máximo que se espera para cada ocurrencia del concepto relevante.
- **Ocurrencias simultáneas:** este campo indica el número medio y máximo de ocurrencias simultáneas del concepto relevante. Tanto este campo como el anterior permiten a los diseñadores prever determinadas necesidades del sistema a desarrollar en lo relativo a las necesidades de almacenamiento de información.
- **Importancia, Urgencia, Estado, Estabilidad, Comentarios:** estos campos tienen el mismo significado que en la plantilla para objetivos aunque referidos al requisito.

5.3. Plantilla para actores

Aunque, estrictamente hablando, los actores de los casos de uso no son requisitos, por homogeneidad con el estilo de definición del resto de

los elementos que componen el catálogo de requisitos se ha descrito la plantilla para definirlos que puede verse en la figura 13.

ACT-<i><id></i>	<i><nombre descriptivo></i>
Versión	<i><nº de la versión actual> (<fecha de la versión actual>)</i>
Autores	• <i><autor de la versión actual> (<organización del autor>)</i> ...
Fuentes	• <i><fuente de la versión actual> (<organización de la fuente>)</i> ...
Descripción	Este actor representa a <i><rol que representa el actor></i>
Comentarios	<i><comentarios adicionales sobre el actor></i>

Figura 13: Plantilla y patrones-L para actores

El único campo específico de esta plantilla es la descripción, en la que se usa un patrón-L que debe completarse con la descripción del rol o papel que representa el actor respecto al sistema. El significado del resto de los campos es el mismo que para las plantillas anteriores.

5.4. Plantilla para requisitos funcionales

Los sistemas de información no sólo almacenan información, también deben proporcionar servicios usando la información que almacenan. La plantillas de requisitos funcionales, que puede verse en las figuras 14 y 15, describen casos de uso y requisitos funcionales de la forma tradicional, y ayudan a los clientes y usuarios a responder a la pregunta "*¿qué debe hacer el sistema con la información almacenada para alcanzar los objetivos de su negocio?*".

El significado de los campos específicos de estas plantillas es el siguiente (los campos comunes con las plantillas para requisitos de información tienen el mismo significado):

- **Identificador y nombre descriptivo:** igual que en las plantillas anteriores, excepto que los identificadores de los requisitos funcionales empiezan con *UC* para los casos de uso y con *FRQ* para los requisitos funcionales expresados de la forma tradicional, y que para los casos de uso, el nombre descriptivo suele coincidir con el objetivo que los actores esperan alcanzar al realizarlo. No se debe confundir este objetivo con los objetivos del sistema. El objetivo que los actores esperan alcanzar al realizar un caso de uso es de más bajo nivel, por ejemplo *registrar un nuevo socio* o *consultar los pedidos pendientes*.

UC-<i><id></i>	<i><nombre descriptivo></i>	
Versión	<i><nº de la versión actual></i> (<i><fecha de la versión actual></i>)	
Autores	<ul style="list-style-type: none"> <i><autor de la versión actual></i> (<i><organización del autor></i>) ... 	
Fuentes	<ul style="list-style-type: none"> <i><fuente de la versión actual></i> (<i><organización de la fuente></i>) ... 	
Objetivos asociados	<ul style="list-style-type: none"> OBJ-<i>x</i> <i><nombre del objetivo></i> ... 	
Requisitos asociados	<ul style="list-style-type: none"> R<i>x</i>-<i>y</i> <i><nombre del requisito></i> ... 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso { abstracto durante la realización de los siguientes casos de uso: <i><lista de casos de uso></i> , cuando <i><evento de activación></i> [o durante la realización de los siguientes casos de uso: <i><lista de casos de uso></i>]}	
Precondición	<i><precondición del caso de uso></i>	
Secuencia normal	Paso	Acción
	<i>p</i> ₁	{El actor <i><actor></i> , El sistema} <i><acción/es realizada/s por actor/sistema></i>
	<i>p</i> ₂	Se realiza el caso de uso <i><caso de uso (RF-x)></i>
	<i>p</i> ₃	Si <i><condición></i> , {el actor <i><actor></i> , el sistema} <i><acción/es realizada/s por actor/sistema></i>
	<i>p</i> ₄	Si <i><condición></i> , se realiza el caso de uso <i><caso de uso (RF-x)></i>

Postcondición	<i><postcondición del caso de uso></i>	
Excepciones	Paso	Acción
	<i>p</i> _{<i>i</i>}	Si <i><condición de excepción></i> , {el actor <i><actor></i> , el sistema} <i><acción/es realizada/s por actor/sistema></i> , a continuación este caso de uso {continúa, queda sin efecto}
	<i>p</i> _{<i>j</i>}	Si <i><condición de excepción></i> , se realiza el caso de uso <i><caso de uso (RF-x)></i> , a continuación este caso de uso {continúa, queda sin efecto}

Rendimiento	Paso	Cota de tiempo
	<i>q</i>	<i>m</i> <i><unidad de tiempo></i>

Frecuencia	<i><nº de veces></i> veces / <i><unidad de tiempo></i>	
Importancia	<i><importancia del requisito></i>	
Urgencia	<i><urgencia del requisito></i>	
Estado	<i><estado del requisito></i>	
Estabilidad	<i><estabilidad del requisito></i>	
Comentarios	<i><comentarios adicionales sobre el requisito></i>	

Figura 14: Plantilla y patrones-L para requisitos funcionales (casos de uso)

FRQ-<i><id></i>	<i><nombre descriptivo></i>
Versión	<i><nº de la versión actual> (<fecha de la versión actual>)</i>
Autores	• <i><autor de la versión actual> (<organización del autor>)</i> ...
Fuentes	• <i><fuente de la versión actual> (<organización de la fuente>)</i> ...
Objetivos asociados	• <i>OBJ-x <nombre del objetivo></i> ...
Requisitos asociados	• <i>Rx-y <nombre del requisito></i> ...
Descripción	El sistema deberá <i><capacidad del sistema></i>
Importancia	<i><importancia del requisito></i>
Urgencia	<i><urgencia del requisito></i>
Estado	<i><estado del requisito></i>
Estabilidad	<i><estabilidad del requisito></i>
Comentarios	<i><comentarios adicionales sobre el requisito></i>

Figura 15: Plantilla y patrones-L para requisitos funcionales (forma tradicional)

- **Descripción:** para los requisitos funcionales expresados de la forma tradicional, este campo contiene un patrón-L que debe completarse con la capacidad o funcionalidad que debe presentar el sistema a desarrollar.

Para los requisitos funcionales expresados como casos de uso, este campo contiene un patrón-L que debe completarse de forma distinta en función de que el caso de uso sea abstracto o concreto (ver sección 4.4.2, pág. 28).

Si el caso de uso es abstracto, deben indicarse los casos de uso en los que se debe realizar, es decir, aquellos desde los que es *incluido* o a los que *extiende*. Si, por el contrario, se trata de un caso de uso concreto, se debe indicar el *evento de activación* que provoca su realización, y en el caso de que sea *incluido* desde, o *extienda* a, otros casos de uso, se deberán indicar dichos casos de uso.

- **Precondición:** en este campo se expresan en lenguaje natural las condiciones necesarias para que se pueda realizar el caso de uso. Estas condiciones se establecen bien sobre el entorno en el que opera el sistema, y que por lo tanto quedarán fuera de su control, bien sobre el estado del propio sistema.
- **Secuencia normal:** este campo contiene la secuencia normal de interacciones del caso de uso. En cada paso, un actor o el sistema realiza

una o más acciones, o se realiza (se *incluye*) otro caso de uso. Un paso puede tener una condición de realización, en cuyo caso si se realizara otro caso de uso se tendría una relación de *extensión*. Se asume que, después de realizar el último paso, el caso de uso termina.

Otras propuestas similares, por ejemplo [Coleman 1998], proponen utilizar estructuras similares al pseudocódigo para expresar las interacciones de los casos de uso. En nuestra opinión, esto puede llevar a que dichas descripciones sean excesivamente complejas de entender para los participantes sin conocimientos de programación y se corre el peligro de especificar los casos de uso con un estilo cercano a la programación.

Para representar estructuras condicionales complejas se puede recurrir a añadir información aparte, por ejemplo una tabla de decisión, y referenciarla desde el paso o los pasos oportunos.

En el caso de estructuras iterativas, su uso puede evitarse con un uso cuidadoso del lenguaje natural. Por ejemplo, para indicar que se procesan todos los artículos de un pedido se puede optar por frases como "*el sistema procesa todos los artículos del pedido introducidos por el usuario*", en lugar de estructuras como:

REPETIR

 procesar artículo del pedido introducido por el usuario
HASTA que no haya más artículos

Otro ejemplo puede ser especificar que el usuario puede intentar conectarse al sistema un máximo de tres veces. Una posible especificación sería la que puede verse en la figura 16, bastante más *natural* y fácil de entender que la que puede verse en la figura 17 utilizando la propuesta descrita en [Coleman 1998].

- **Postcondición:** en este campo se expresan en lenguaje natural las condiciones que se deben cumplir después de la terminación normal del caso de uso. Al igual que en el caso de las precondiciones, las postcondiciones se pueden establecer tanto sobre el entorno del sistema como sobre el estado del propio sistema.
- **Excepciones:** este campo especifica el comportamiento del sistema en el caso de que se produzca alguna situación excepcional durante la realización de un paso determinado.

Después de realizar las acciones o el caso de uso asociados a la excepción (una *extensión*), el caso de uso puede continuar la secuencia

Secuencia normal	Paso	Acción
	1	El sistema solicita al usuario su nombre de usuario y su clave de acceso
	2	El usuario proporciona el sistema su nombre y su clave de acceso
	3	El sistema comprueba si el nombre de usuario y la clave de acceso son correctas
	4	Si el nombre de usuario y la clave no son correctas, el sistema permite al usuario repetir el intento (pasos 1-3) hasta un máximo de tres veces
	5	Si el nombre de usuario y la clave son correctas, el sistema permite el acceso al usuario
Excepciones	Paso	Acción
	4	Si el usuario ha intentado tres veces acceder sin éxito, el sistema rechaza el acceso del usuario, a continuación este caso de uso termina

Figura 16: Ejemplo de caso de uso de conexión de usuario (plantilla)

1. El sistema inicializa intentos a 0
2. REPETIR
 - 2.1 El sistema solicita al usuario su nombre de usuario
 - 2.2 El usuario proporciona al sistema su nombre de usuario
 - 2.3 El sistema solicita al usuario su clave
 - 2.4 El usuario proporciona al sistema su clave
 - 2.5 El sistema comprueba si el nombre de usuario y la clave son correctas
 - 2.6 El sistema incrementa intentos
 HASTA QUE el nombre de usuario y la clave sean correctas o intentos = 3
3. SI el nombre de usuario y la clave son correctas
 - 3.1 El sistema permite el acceso al usuario
 SINO
 - 3.2 El sistema rechaza el acceso del usuario
 FINSI

Figura 17: Ejemplo de caso de uso de conexión de usuario (Coleman)

normal o quedar sin efecto, en cuyo caso se cancelan todas las acciones realizadas en el caso de uso dejando al sistema en el mismo estado que antes de comenzar el caso de uso, asumiendo una semántica *transaccional* del mismo, tal como se describe en [Jacobson *et al.* 1997].

Inicialmente, la expresión utilizada para indicar una terminación anormal del caso de uso como resultado de una excepción era "*este caso de uso aborta*". La experiencia durante su aplicación nos llevó a la conclusión de que el termino *abortar* resultaba *emocionalmente* molesto para algunos participantes [Goleman 1996], por lo que se cambió por "*este caso de uso queda sin efecto*" con el significado comentado anteriormente.

- **Rendimiento:** en este campo puede especificarse el tiempo máximo para cada paso en el que el sistema realice un acción.
- **Frecuencia esperada:** en este campo se indica la frecuencia esperada de realización del caso de uso, que aunque no es realmente un requisito, es una información interesante para los desarrolladores.

5.5. Plantilla para requisitos no funcionales

Los requisitos no funcionales del sistema se pueden expresar usando la plantilla que puede verse en la figura 18. El único campo específico de esta plantilla es la descripción, en la que se usa un patrón-L que debe completarse con la capacidad que deberá presentar el sistema, el significado del resto de los campos es el mismo que para las plantillas anteriores.

5.6. Plantilla para conflictos

Como ya se ha comentado, durante las sesiones de elicitación puede ser necesario resolver mediante algún tipo de negociación posibles conflictos en los requisitos-C elicitados en iteraciones previas del proceso. Para documentar dichos conflictos, y las soluciones adoptadas, se propone la plantilla que puede verse en la figura 19.

El significado de los campos de la plantilla es el siguiente:

- **Identificador y nombre descriptivo:** al igual que el resto de la información correspondiente a los requisitos-C, cada conflicto debe po-

NFR-<i><id></i>	<i><nombre descriptivo></i>
Versión	<i><nº de la versión actual> (<fecha de la versión actual>)</i>
Autores	• <i><autor de la versión actual> (<organización del autor>)</i> ...
Fuentes	• <i><fuente de la versión actual> (<organización de la fuente>)</i> ...
Objetivos asociados	• <i>OBJ-x <nombre del objetivo></i> ...
Requisitos asociados	• <i>Rx-y <nombre del requisito></i> ...
Descripción	El sistema deberá <i><capacidad del sistema></i>
Importancia	<i><importancia del requisito></i>
Urgencia	<i><urgencia del requisito></i>
Estado	<i><estado del requisito></i>
Estabilidad	<i><estabilidad del requisito></i>
Comentarios	<i><comentarios adicionales sobre el requisito></i>

Figura 18: Plantilla y patrones-L para requisitos no funcionales

derse identificar de forma única y tener un nombre descriptivo. El prefijo propuesto para lograr una rápida identificación es *CFL*.

- **Versión, Autores, Fuentes:** estos campos tienen el mismo significado que en las plantillas para objetivos y requisitos, aunque referidos al conflicto. En este caso especial, las fuentes son los participantes que deben participar en las posibles negociaciones necesarias para su resolución.
- **Objetivos y requisitos en conflicto:** este campo debe contener una lista con los objetivos y/o requisitos afectados por el conflicto.
- **Descripción:** este campo debe contener la descripción del conflicto.
- **Alternativas:** este campo debe contener una lista con las posibles alternativas de solución que se hayan identificado para solucionar el conflicto así como los autores de dicha alternativas.
- **Solución:** este campo debe contener la descripción de la solución negociada del conflicto, una vez que se haya acordado.
- **Importancia, Urgencia:** estos campos indican respectivamente la importancia y la urgencia de la resolución del conflicto.
- **Estado:** este campo indica el estado de resolución del conflicto, que podrá estar *no resuelto*, *en negociación* o bien *resuelto*.

CFL-<i><id></i>	<i><nombre descriptivo></i>
Versión	<i><nº de la versión actual> (<fecha de la versión actual>)</i>
Autores	• <i><autor de la versión actual> (<organización del autor>)</i> ...
Fuentes	• <i><fuente de la versión actual> (<organización de la fuente>)</i> ...
Objs./Reqs. en conflicto	• <i>OBJ/Ryy--x <nombre del objetivo o requisito en conflicto></i> ...
Descripción	<i><descripción del conflicto></i>
Alternativas	• <i><descripción alternativa de solución> (<autores alternativa>)</i> ...
Solución	<i><descripción de la solución adoptada (si se ha acordado)></i>
Importancia	<i><importancia de la resolución del conflicto></i>
Urgencia	<i><urgencia de la resolución del conflicto></i>
Estado	<i><estado del resolución del conflicto></i>
Comentarios	<i><comentarios adicionales sobre el conflicto></i>

Figura 19: Plantilla para conflictos

- **Comentarios:** este campo tienen el mismo significado que en las plantillas descritas previamente.

Referencias

- [Beyer y Holtzblatt 1995] H. R. Beyer y K. Holtzblatt. Apprenticing with the Customer. *Communications of the ACM*, 38(5), Mayo 1995.
- [Boehm et al. 1994] B. W. Boehm, P. Bose, E. Horowitz, y M.-J. Lee. Software Requirements as Negotiated Win Conditions. En *Proceedings of the First International Conference on Requirements Engineering*, 1994. Disponible en <http://sunset.usc.edu/TechRpts/Papers/NGPM-Requirements93.ps>.
- [Booch et al. 1999] G. Booch, J. Rumbaugh, y I. Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, 1999.
- [Brackett 1990] J. W. Brackett. Software Requirements. Curriculum Module SEI-CM-19-1.2, Software Engineering Institute, Carnegie Mellon University, 1990. Disponible en <http://www.sei.cmu.edu>.
- [Cockburn 1997] A. Cockburn. Structuring Use Cases with Goals. *Journal of Object-Oriented Programming*, Sept. y Nov./Dic. 1997. Disponible en <http://members.aol.com/acockburn/papers/usecases.htm>.

- [Coleman 1998] D. Coleman. A Use Case Template: Draft for Discussion. *Fusion Newsletter*, Abril 1998. Disponible en http://www.hpl.hp.com/fusion/md_newletters.html.
- [Davis 1985] F. Davis. *La comunicación no verbal*, volumen 616 de *El Libro de Bolsillo*. Alianza Editorial, 1985.
- [Durán 2000] A. Durán. *Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información*. Tesis doctoral, Universidad de Sevilla, 2000.
- [Firesmith 1997] D. G. Firesmith. Uses Cases: the Pros and Cons, 1997. Disponible en <http://www.ksscary.com/usecjrn1.html>.
- [García 2000] F. J. García. *Modelo de Reutilización Soportado por Estructuras Complejas de Reutilización Denominadas Mecanos*. Tesis doctoral, Universidad de Salamanca, 2000.
- [Gause y Weinberg 1989] D. C. Gause y G. M. Weinberg. *Exploring Requirements: Quality Before Design*. Dorset House, 1989.
- [Goguen y Linde 1993] J. A. Goguen y C. Linde. Techniques for Requirements Elicitation. En *Proceedings of the First International Symposium on Requirements Engineering*, 1993. También aparece en [Thayer y Dorfman 1997]. Disponible en <http://www.cse.ucsd.edu/~goguen>.
- [Goleman 1996] D. Goleman. *La Inteligencia Emocional*. Kairós, 1996.
- [Goleman 1999] D. Goleman. *La Práctica de la Inteligencia Emocional*. Kairós, 1999.
- [IBM OOTC 1997] IBM OOTC. *Developing Object-Oriented Software*. IBM Object-Oriented Technology Center. Prentice-Hall, 1997.
- [IEEE 1993] IEEE. IEEE Recommended Practice for Software Requirements Specifications. IEEE/ANSI Standard 830-1993, Institute of Electrical and Electronics Engineers, 1993.
- [Jacobson et al. 1993] I. Jacobson, M. Christerson, P. Jonsson, y G. Övergaard. *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley, 4ª edición, 1993.
- [Jacobson et al. 1997] I. Jacobson, M. Griss, y P. Jonsson. *Software Reuse: Architecture, Process and Organization for Business Success*. Addison-Wesley, 1997.

- [Laguna *et al.* 1999] M. A. Laguna, J. M. Marqués, y F. J. García. Una Herramienta para la Captura de Requisitos de Usuario. En *Actas de las JISBD'99*, Cáceres, 1999.
- [Leite *et al.* 1997] J. C. S. P. Leite, G. Rossi, F. Balaguer, V. Maiorana, G. Kaplan, G. Hadad, y A. Oliveros. Enhancing a Requirements Baseline with Scenarios. En *Proceedings of the 3rd IEEE International Symposium on Requirements Engineering (RE'97)*, 1997.
- [MAP 1995] MAP. *Metodología de Planificación y Desarrollo de Sistemas de Información. MÉTRICA Versión 2.1*. Tecnos/Ministerio para las Administraciones Públicas, 1995.
- [Ortín *et al.* 2001] M. J. Ortín, J. García, B. Moros, y J. Nicolás. El Modelo de Negocio como Base del Modelo de Requisitos. En *Actas de las Jornadas de Ingeniería de Requisitos Aplicada*, Sevilla, 2001.
- [Piattini *et al.* 1996] M. G. Piattini, J. A. Calvo-Manzano, J. Cervera, y L. Fernández. *Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión*. ra-ma, 1996.
- [Raghavan *et al.* 1994] S. Raghavan, G. Zelesnik, y G. Ford. Lecture Notes on Requirements Elicitation. Educational Materials CMU/SEI-94-EM-10, Software Engineering Institute, Carnegie Mellon University, 1994. Disponible en <http://www.sei.cmu.edu>.
- [Robertson y Robertson 1999] S. Robertson y J. Robertson. *Mastering the Requirement Process*. Addison-Wesley, 1999.
- [Rolland *et al.* 1998] C. Rolland, C. Ben Achour, C. Cauvet, J. Ralyté, A. Sutcliffe, N. A. M. Maiden, M. Jarke, P. Haumer, K. Pohl, E. Dubois, y P. Heymans. A Proposal for a Scenario Classification Framework. *Requirements Engineering Journal*, 3(1), 1998. Disponible en <http://sunsite.informatik.rwth-aachen.de/CREWS/reports98.htm>.
- [Sawyer *et al.* 1997] P. Sawyer, I. Sommerville, y S. Viller. Requirements Process Improvement through The Phased Introduction of Good Practice. *Software Process – Improvement and Practice*, 3(1), 1997. Disponible en <http://www.comp.lancs.ac.uk/computing/research/cseg/reaims/publications.html>.
- [Sawyer y Kontoya 1999] P. Sawyer y G. Kontoya. SWEBOK: Software Requirements Engineering Knowledge Area Description. Informe Técnico Versión 0.5, SWEBOK Project, 1999. Disponible en <http://www.swebok.org>.

-
- [Schneider y Winters 1998] G. Schneider y J. P. Winters. *Applying Use Cases: a Practical Guide*. Addison-Wesley, 1998.
- [Thayer y Dorfman 1990] R. H. Thayer y M. Dorfman, editores. *System and Software Requirements Engineering*. IEEE Computer Society Press, 1990.
- [Thayer y Dorfman 1997] R. H. Thayer y M. Dorfman, editores. *Software Requirements Engineering*. IEEE Computer Society Press, 2ª edición, 1997. Es la 2ª edición de [Thayer y Dorfman 1990].
- [Weidenhaput *et al.* 1998] K. Weidenhaput, K. Pohl, M. Jarke, y P. Haumer. Scenarios in System Development: Current Practice. *IEEE Software*, 15(2):34-45, Marzo/Abril 1998. Este artículo aparece también en las actas del ICRE'98 y está disponible en <http://sunsite.informatik.rwth-aachen.de/CREWS/reports97.htm>.
- [Wirfs-Brock *et al.* 1990] R. Wirfs-Brock, B. Wilkerson, y L. Wiener. *Designing Object-Oriented Software*. Prentice-Hall, 1990.

A. Ejemplo: gestión de un vídeo-club

En este apéndice se ofrecen algunos ejemplos de aplicación de las técnicas propuestas en esta metodología suponiendo el caso de la gestión de pequeño vídeo-club.

Dado que se trata de un ejemplo ficticio se han simplificado las plantillas eliminando los campos relativos a versión, autores, fuentes, importancia, urgencia y estado de desarrollo.

El ejemplo no es una especificación de requisitos completa, se incluye sólo a modo de ejemplo.

A.1. Objetivos del sistema

OBJ-01	Gestionar las cintas y películas
Descripción	El sistema deberá <i>gestionar la información correspondiente a las cintas y películas del vídeo club: adquisiciones, retiradas y disponibilidad.</i>
Estabilidad	alta
Comentarios	ninguno

OBJ-02	Gestionar los socios
Descripción	El sistema deberá <i>gestionar la información correspondiente a los socios del vídeo-club: altas, bajas, modificaciones de datos, sanciones, personas autorizadas y cuentas.</i>
Estabilidad	alta
Comentarios	ninguno

OBJ-02	Gestionar los alquileres
Descripción	El sistema deberá <i>gestionar la información correspondiente a los alquileres de cintas: entregas, devoluciones, devoluciones tardías, reclamaciones y disponibilidad.</i>
Estabilidad	alta
Comentarios	ninguno

A.2. Requisitos de información

IRQ-01	Información sobre películas	
Objetivos asociados	• OBJ-01 <i>Gestionar las películas y cintas</i>	
Requisitos asociados	<ul style="list-style-type: none"> • UC-04 <i>Alta de película</i> • UC-05 <i>Alta de cinta de vídeo</i> • UC-08 <i>Baja de cinta de vídeo</i> • UC-10 <i>Consulta de película</i> • UC-13 <i>Consulta de películas alquiladas un día determinado</i> 	
Descripción	El sistema deberá almacenar la información correspondiente a <i>las películas del vídeo-club</i> . En concreto:	
Datos específicos	<ul style="list-style-type: none"> • Título de la película • Cintas de la película alquiladas en cada momento • Cintas de la película disponibles para ser alquiladas en cada momento • Tipo de la película: infantil, acción, ciencia-ficción o adultos • Duración de la película, en horas y minutos • Actores principales de la película • Director de la película • Productora de la película • Año de producción de la película 	
Tiempo de vida	Medio	Máximo
	1.5 años	5 años
Ocurrencias simult.	Medio	Máximo
	10	25
Estabilidad	alta	
Comentarios	En este requisito se han mezclado deliberadamente dos conceptos, el de película y el de cinta de vídeo que contiene una película. En el documento de la metodología de análisis se continúa este ejemplo, se descubre el conflicto y se propone su separación en dos requisitos distintos.	

CRQ-01	Relación entre películas y cintas
Objetivos asociados	• OBJ-01 <i>Gestionar las películas y cintas</i>
Requisitos asociados	• IRQ-01 <i>Información sobre películas</i>
Descripción	La información almacenada por el sistema deberá satisfacer la siguiente restricción: <i>una cinta de vídeo contiene como mucho una película, sin embargo es posible que películas de muy larga duración se presenten en formato de dos o más cintas.</i>
Estabilidad	alta
Comentarios	ninguno

IRQ-02	Información sobre socios
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-02 <i>Gestionar los socios</i>
Requisitos asociados	<ul style="list-style-type: none"> • UC-01 <i>Alta de socio</i> • UC-02 <i>Baja de socio</i> • UC-03 <i>Modificación de datos de un socio</i> • UC-11 <i>Consulta de un socio</i> • UC-12 <i>Consulta de socios con pagos pendientes</i> • UC-12 <i>Consulta de los socios más rentables</i> • UC-15 <i>Identificación de socio</i>
Descripción	El sistema deberá almacenar la información correspondiente a <i>los socios del vídeo-club</i> . En concreto:
Datos específicos	<ul style="list-style-type: none"> • Número de socio • Número del documento nacional de identidad • Nombre y apellidos • Fecha de nacimiento • Sexo • Fecha de alta como socio • Dirección • Teléfonos • Películas alquiladas en un momento dado
Estabilidad	alta
Comentarios	ninguno

CRQ-02	Unicidad de números de socio
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-02 <i>Gestionar los socios</i>
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-02 <i>Información sobre socios</i> • UC-01 <i>Alta de socio</i>
Descripción	La información almacenada por el sistema deberá satisfacer la siguiente restricción: <i>los números de socio deberán ser únicos para cada socio, es decir, no puede haber dos socios distintos con el mismo número.</i>
Estabilidad	alta
Comentarios	ninguno

IRQ-03	Información sobre cuentas de socios
Objetivos asociados	• OBJ-02 <i>Gestionar los socios</i>
Requisitos asociados	<ul style="list-style-type: none"> • UC-01 <i>Alta de socio</i> • UC-02 <i>Baja de socio</i> • UC-05 <i>Alquiler de cinta de vídeo</i> • UC-08 <i>Devolución de cintas de vídeo</i> • UC-09 <i>Ingreso a cuenta</i> • UC-11 <i>Consulta de un socio</i> • UC-12 <i>Consulta de socios con pagos pendientes</i>
Descripción	El sistema deberá almacenar la información correspondiente a <i>las cuentas de los socios del vídeo-club</i> . En concreto:
Datos específicos	<ul style="list-style-type: none"> • Saldo de la cuenta en cada momento • Ingresos realizados en la cuenta, indicando fecha y cantidad • Cargos realizados en la cuenta, indicando fecha, motivo y cantidad • Pagos pendientes, indicando motivo que podrá ser alquiler no pagado o multa; en el caso de alquiler no pagado se debe indicar también la película alquilada y la fecha del alquiler
Estabilidad	alta
Comentarios	Un socio puede hacer ingresos a cuenta, por ejemplo para enviar a sus hijos por películas sin que éstos tengan que llevar dinero

CRQ-03	Alquiler de películas de más de una cinta
Objetivos asociados	• OBJ-03 <i>Gestionar los alquileres</i>
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-02 <i>Información sobre socios</i> • UC-06 <i>Alquiler de cintas de vídeo</i>
Descripción	La información almacenada por el sistema deberá satisfacer la siguiente restricción: <i>los alquileres de películas que se presentan en varias cintas se consideran como un único alquiler aunque el socio se lleve más de una cinta.</i>
Estabilidad	alta
Comentarios	ninguno

A.3. Requisitos funcionales

A.3.1. Diagramas de casos de uso

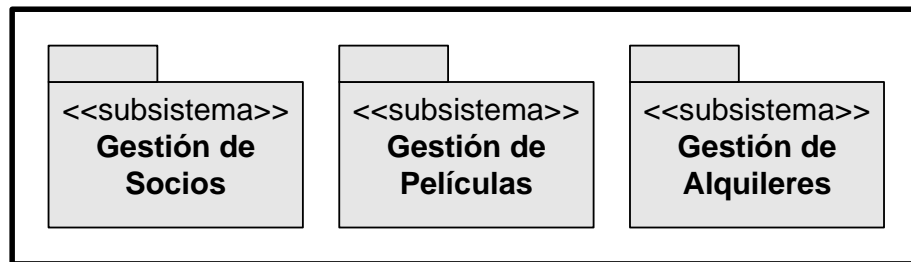


Figura 20: Diagrama de subsistemas

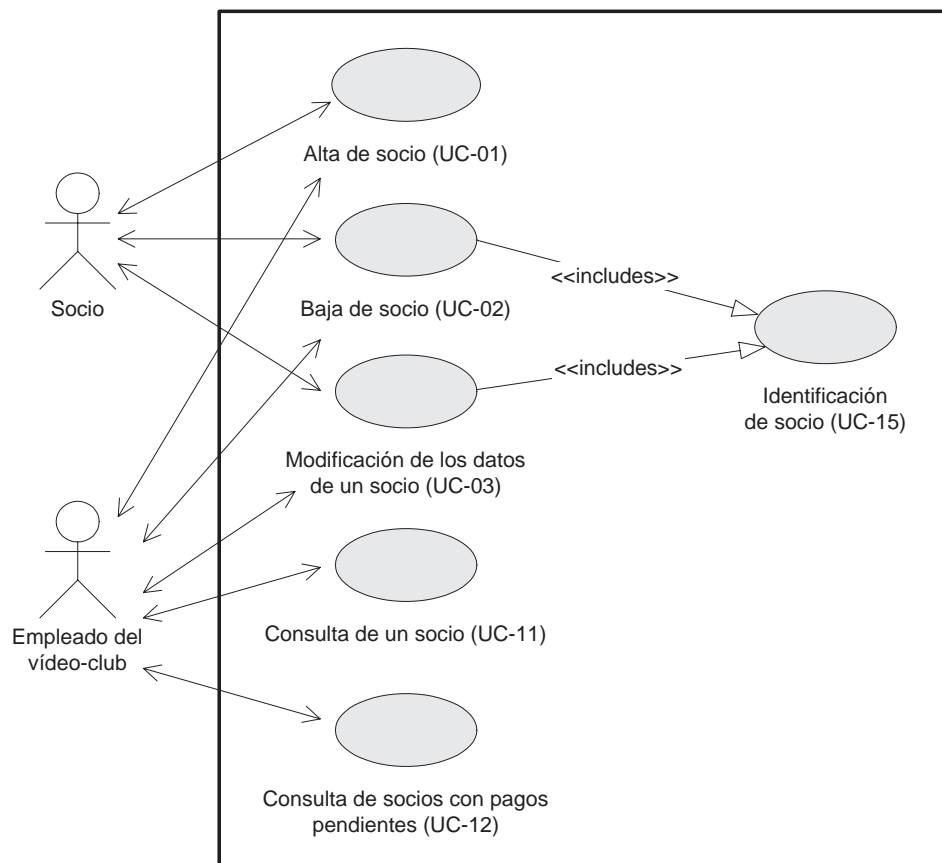


Figura 21: Diagrama de casos de uso del subsistema Gestión de socios

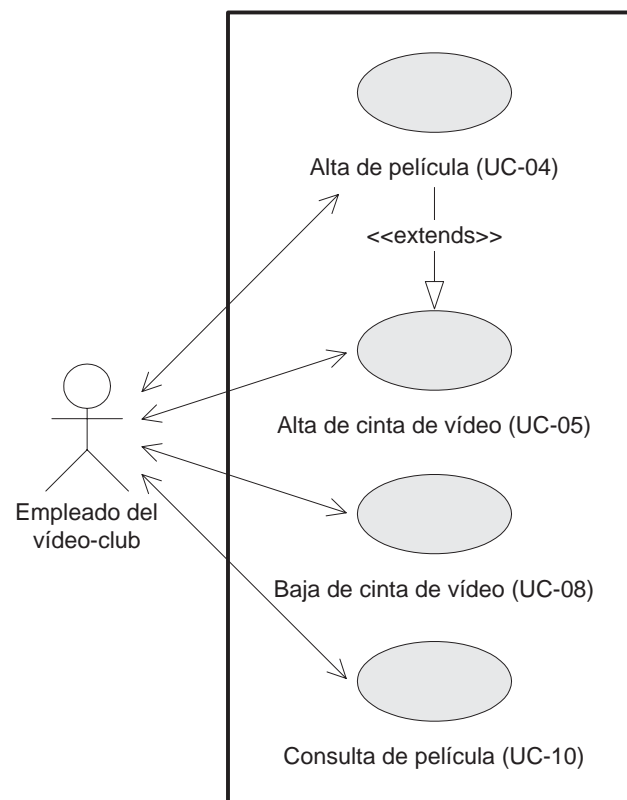


Figura 22: Diagrama de casos de uso del subsistema Gestión de películas

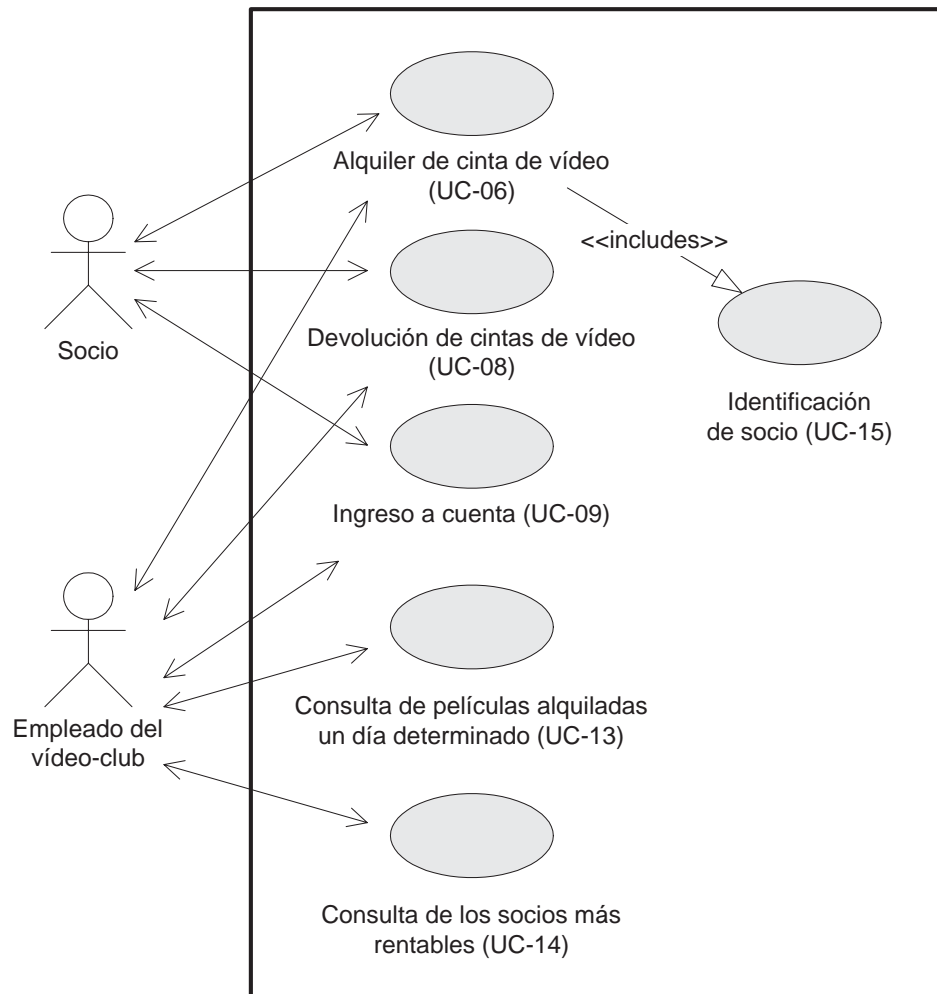


Figura 23: Diagrama de casos de uso del subsistema Gestión de alquileres

A.3.2. Definición de actores

ACT-01	Socio
Descripción	Este actor representa a los socios del vídeo-club
Comentarios	ninguno

ACT-02	Empleado del vídeo-club
Descripción	Este actor representa a los empleados del vídeo-club
Comentarios	ninguno

A.3.3. Casos de uso del sistema

(ver páginas siguientes)

A.3.3.1. Casos de uso del subsistema Gestión de socios

UC-01	Alta de socio	
Objetivos asociados	• OBJ-02 <i>Gestionar las socios</i>	
Requisitos asociados	• IRQ-02 <i>Información sobre socios</i>	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>alguna persona solicite su ingreso como socio del vídeo-club</i>	
Precondición	El solicitante no es un socio del vídeo-club y tiene su documentación disponible	
Secuencia normal	Paso	Acción
	1	El empleado del vídeo-club solicita al sistema comenzar el proceso de alta de un nuevo socio
	2	El sistema solicita los siguientes datos del nuevo socio: nº del DNI, nombre, apellidos, fecha de nacimiento, sexo, dirección y teléfonos de contacto
	3	El empleado del vídeo-club solicita los datos requeridos y la documentación al nuevo socio
	4	El empleado del vídeo-club comprueba que los datos del nuevo socio coinciden con los de la documentación aportada
	5	El empleado del vídeo-club proporciona los datos requeridos y solicita al sistema que los almacene
	6	El sistema almacena los datos proporcionados e imprime el carnet de socio
	7	El sistema informa al empleado del vídeo-club que el proceso ha terminado con éxito
	8	El empleado del vídeo-club entrega el carnet al nuevo socio
Postcondición	El solicitante es socio del vídeo-club y el su cuenta no tiene ningún movimiento	
Excepciones	Paso	Acción
	4	Si la documentación aportada no es correcta, el empleado del vídeo-club cancela la operación, a continuación este caso de uso queda sin efecto
	5	Si el sistema detecta que el nuevo socio ya es socio del vídeo-club, el sistema informa de la situación al empleado del vídeo-club, permitiéndole modificar los datos proporcionados, a continuación este caso de uso continúa
Rendimiento	Paso	Cota de tiempo
	6	5 segundos
Frecuencia	10 veces/día	
Estabilidad	alta	
Comentarios	La frecuencia será mucho mayor durante los dos primeros meses, probablemente hasta 100 veces/día	

UC-02	Baja de socio	
Objetivos asociados	• OBJ-02 <i>Gestionar las socios</i>	
Requisitos asociados	• IRQ-02 <i>Información sobre socios</i>	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>un socio solicite su baja</i>	
Precondición	El solicitante es un socio del vídeo-club, no tiene ningún pago pendiente y tiene su documentación disponible	
Secuencia normal	Paso	Acción
	1	El empleado del vídeo-club solicita al sistema comenzar el proceso de baja de un socio
	2	Se realiza el caso de uso UC-15 (Identificación de socio)
	3	El empleado del vídeo-club solicita al sistema que elimine la información correspondiente al socio
	4	El sistema elimina los datos correspondientes al socio
	5	El sistema informa al empleado del vídeo-club que el proceso ha terminado con éxito
	6	El empleado del vídeo-club inhabilita el carnet al socio que se acaba de dar de baja
Postcondición	El solicitante no es socio del vídeo-club	
Excepciones	Paso	Acción
	3	Si el socio tiene pagos pendientes, el sistema comunica la situación al empleado del vídeo-club, a continuación este caso de uso queda sin efecto
Rendimiento	Paso	Cota de tiempo
	4	1 segundo
Frecuencia	1 vez/ mes	
Estabilidad	alta	
Comentarios	Si el socio que desea darse de baja tiene un pago pendiente, puede hacer un ingreso por su importe y repetir el proceso de darse de baja	

UC-03	Modificación de los datos de un socio	
Objetivos asociados	• OBJ-02 <i>Gestionar las socios</i>	
Requisitos asociados	• IRQ-02 <i>Información sobre socios</i>	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>un socio solicite la modificación de sus datos</i>	
Precondición	El solicitante es un socio del vídeo-club y tiene su documentación disponible	
Secuencia normal	Paso	Acción
	1	El empleado del vídeo-club solicita al sistema comenzar el proceso de modificación de los datos de un socio
	2	Se realiza el caso de uso UC-15 (Identificación de socio)
	3	El sistema muestra los siguientes datos correspondientes al socio a modificar: nº del DNI, nombre, apellidos, fecha de nacimiento, sexo, dirección y teléfonos de contacto
	4	El sistema permite al empleado del vídeo-club modificar los siguientes datos: dirección y teléfonos de contacto
	5	El empleado del vídeo-club modifica los datos que el sistema le permite y solicita al sistema que los almacene
	6	El sistema modifica los datos correspondientes al socio
	7	El sistema informa al empleado del vídeo-club que el proceso ha terminado con éxito
	8	Si algún dato modificado aparece en el carnet de socio, el sistema imprime un nuevo carnet de socio
	9	Si fue necesario imprimir un nuevo carnet de socio, el empleado del vídeo-club entrega el nuevo carnet al socio e inhabilita el antiguo
Postcondición	La información del socio está actualizada	
Excepciones	Paso	Acción
	–	–
Rendimiento	Paso	Cota de tiempo
	6	1 segundo
Frecuencia	1 vez/mes	
Comentarios	ninguno	

UC-11	Consulta de un socio	
Objetivos asociados	• OBJ-02 <i>Gestionar las socios</i>	
Requisitos asociados	• IRQ-02 <i>Información sobre socios</i>	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el empleado del vídeo-club lo considere oportuno</i>	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El empleado del vídeo-club solicita al sistema comenzar el proceso de consulta de los datos de un socio
	2	El sistema solicita que se identifique al socio
	3	El empleado del vídeo-club proporciona los datos de identificación al sistema
	4	El sistema muestra la siguiente información asociada al socio: nombre, apellidos, dirección, números de teléfono, alquileres pendientes y saldo de su cuenta
	5	Si el empleado del vídeo-club solicita la impresión de los datos, el sistema imprime los datos del socio
Postcondición	Ninguna	
Excepciones	Paso	Acción
	5	Si el sistema no tiene registrado ningún socio con la identificación proporcionada, el sistema comunica al empleado del vídeo-club la situación, a continuación este caso de uso queda sin efecto
Rendimiento	Paso	Cota de tiempo
	4	1 segundo
Frecuencia	5 veces/día	
Comentarios	El formato de visualización de los datos está pendiente de definición	

UC-12	Consulta de socios con pagos pendientes	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-02 <i>Gestionar las socios</i> 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-02 <i>Información sobre socios</i> • IRQ-03 <i>Información sobre cuentas de socios</i> 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el empleado del vídeo-club lo considere oportuno</i>	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El empleado del vídeo-club solicita al sistema comenzar el proceso de consulta de los socios con pagos pendientes
	2	El sistema muestra una lista ordenada por cantidad pendiente con la siguiente información por cada socio: nombre, apellidos, cantidad total pendiente y detalle de las cantidades pendientes
	3	Si el empleado del vídeo-club solicita la impresión de los datos, el sistema imprime la lista
Postcondición	Ninguna	
Excepciones	Paso	Acción
	–	–
Rendimiento	Paso	Cota de tiempo
	2	5 segundos
Frecuencia	1 vez/semana	
Comentarios	ninguno	

UC-15	Identificación de socio	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-02 <i>Gestionar las socios</i> 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-02 <i>Información sobre socios</i> 	
Descripción	<p>El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización de los casos de uso:</p> <ul style="list-style-type: none"> • UC-02 <i>Baja de socio</i> • UC-03 <i>Modificación de datos de un socio</i> • UC-06 <i>Alquiler de cintas de vídeo</i> 	
Precondición	El socio tiene su documentación disponible	
Secuencia normal	Paso	Acción
	1	El sistema solicita que se identifique al socio
	2	El empleado del vídeo-club solicita el carnet de socio
	3	El empleado del vídeo-club proporciona los datos de identificación al sistema
	4	El sistema muestra los números de teléfonos que el socio proporcionó cuando se dio de alta
	5	El empleado del vídeo-club solicita al socio que le confirme alguno de los números de teléfono registrados en el sistema
	6	El empleado del vídeo-club confirma la identidad del socio al sistema
Postcondición	Ninguna	
Excepciones	Paso	Acción
	3	Si el sistema detecta que el supuesto socio no es socio del vídeo-club, el sistema comunica al empleado del vídeo-club la situación, a continuación este caso de uso queda sin efecto
	5	Si el socio no conoce ningún número de teléfono registrado en el sistema y no puede demostrar su identidad, el empleado del vídeo-club retiene el carnet de socio y cancela la operación, a continuación este caso de uso queda sin efecto
	5	Si el socio no conoce ningún número de teléfono registrado pero puede demostrar su identidad por otros medios, el empleado del vídeo-club le recuerda los números de teléfonos que proporcionó cuando se dio de alta, a continuación este caso de uso continúa
Rendimiento	Paso	Cota de tiempo
	–	–
Frecuencia	50 veces/día	
Comentarios	ninguno	

A.3.3.2. Casos de uso del subsistema Gestión de películas

UC-04	Alta de película	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-01 <i>Gestionar las cintas y películas</i> 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-01 <i>Información sobre películas</i> • IRQ-05 <i>Alta de cinta de vídeo</i> 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>se adquiera una cinta de una película nueva</i>	
Precondición	La película no está registrada en el sistema	
Secuencia normal	Paso	Acción
	1	El empleado del vídeo-club solicita al sistema comenzar el proceso de alta de película
	2	El sistema solicita los siguientes datos de la nueva película: título, tipo de película, duración, actores principales, director, productora y año de producción
	3	El empleado del vídeo-club proporciona los datos requeridos y solicita al sistema que los almacene
	4	El sistema almacena los datos proporcionados
	5	El sistema informa al empleado del vídeo-club que el proceso ha terminado con éxito
Postcondición	El sistema ha almacenado la información correspondiente a la nueva película	
Excepciones	Paso	Acción
	4	Si el sistema detecta que la película ya está registrada, el sistema informa de la situación al empleado del vídeo-club permitiéndole modificar los datos proporcionados, a continuación este caso de uso continúa
Rendimiento	Paso	Cota de tiempo
	4	1 segundo
Frecuencia	1 vez/día	
Comentarios	ninguno	

UC-05	Alta de cinta de vídeo	
Objetivos asociados	• OBJ-01 <i>Gestionar las cintas y películas</i>	
Requisitos asociados	• IRQ-01 <i>Información sobre películas</i>	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>se adquieran nuevas cintas de una película</i>	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El empleado del vídeo-club solicita al sistema comenzar el proceso de alta de cinta
	2	El sistema solicita que se identifique la película que contiene la cinta
	3	El empleado del vídeo-club identifica la película
	4	Si la película no está registrada, se realiza el caso de uso UC-04 (Alta de película)
	5	El sistema solicita el número de cintas de la película a dar de alta
	6	El empleado del vídeo-club proporciona el número de cintas y solicita al sistema que almacene la información
	7	El sistema almacena los datos proporcionados e imprime la etiquetas de identificación de cintas autoadhesivas
	8	El sistema informa al empleado del vídeo-club que el proceso ha terminado con éxito
	9	El empleado del vídeo-club pega las etiquetas en las cintas y las coloca en las estanterías
Postcondición	Las cintas están registradas en el sistema	
Excepciones	Paso	Acción
	–	–
Rendimiento	Paso	Cota de tiempo
	7	1 segundo
Frecuencia	1 vez/día	
Comentarios	ninguno	

UC-08	Baja de cinta de vídeo	
Objetivos asociados	• OBJ-01 <i>Gestionar las cintas y películas</i>	
Requisitos asociados	• IRQ-01 <i>Información sobre películas</i>	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el empleado del vídeo-club lo considere oportuno</i>	
Precondición	La cinta está registrada en el sistema	
Secuencia normal	Paso	Acción
	1	El empleado del vídeo-club solicita al sistema comenzar el proceso de baja de cinta de vídeo
	2	El sistema solicita que se identifique la cinta a dar de baja
	3	El empleado del vídeo-club identifica la cinta a eliminar y solicita al sistema que la dé de baja
	4	El sistema registra la baja de la cinta
	5	El sistema informa al empleado del vídeo-club que el proceso ha terminado con éxito
	6	El empleado del vídeo-club elimina la cinta de las estanterías
Postcondición	La cinta no está registrada en el sistema	
Excepciones	Paso	Acción
	3	Si el sistema no tiene registrada ninguna cinta con la identificación proporcionada, el sistema comunica al empleado del vídeo-club la situación, a continuación este caso de uso queda sin efecto
Rendimiento	Paso	Cota de tiempo
	4	1 segundo
Frecuencia	1 vez/ mes	
Comentarios	ninguno	

UC-10	Consulta de una película	
Objetivos asociados	• OBJ-01 <i>Gestionar las cintas y películas</i>	
Requisitos asociados	• IRQ-01 <i>Información sobre películas</i>	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el empleado del vídeo-club lo considere oportuno</i>	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El empleado del vídeo-club solicita al sistema comenzar el proceso de consulta de los datos de una película
	2	El sistema solicita que se identifique la película a consultar
	3	El empleado del vídeo-club identifica la película a consultar
	4	El sistema muestra los siguientes datos correspondientes a la película: título, tema, año de producción, actores principales, nombre de la productora y número de cintas disponibles
	5	Si el empleado del vídeo-club solicita la impresión de los datos, el sistema imprime los datos de la película
Postcondición	La información correspondiente a la película consultada no ha cambiado	
Excepciones	Paso	Acción
	–	–
Rendimiento	Paso	Cota de tiempo
	4	1 segundo
Frecuencia	1 vez/día	
Comentarios	ninguno	

A.3.3.3. Casos de uso del subsistema Gestión de alquileres

UC-06	Alquiler de cintas de vídeo	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-03 <i>Gestionar los alquileres</i> 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-02 <i>Información sobre socios</i> • IRQ-03 <i>Información sobre cuentas de socios</i> 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>un socio solicite alquilar una o más cintas de vídeo</i>	
Precondición	Ninguna de las cintas a alquilar está registradas como alquiladas	
Secuencia normal	Paso	Acción
	1	El empleado del vídeo-club solicita al sistema comenzar el proceso de alquiler de cintas de vídeo
	2	Se realiza el caso de uso UC-15 (Identificación de socio)
	2	El sistema solicita que se identifiquen las cintas que desean alquilar
	3	El empleado del vídeo-club identifica las cintas y solicita al sistema que registre el alquiler
	4	El sistema almacena la información de los alquileres
	5	Si el socio decide pagar al contado, el sistema imprime el ticket con el importe correspondiente y registra el pago como un ingreso en la cuenta del socio
	6	Si el socio decide pagar a cuenta, el sistema registra el cargo en la cuenta del socio
	7	El sistema comunica al empleado del vídeo-club que el proceso de registro ha terminado con éxito
Postcondición	Las cintas a alquilar están registradas como alquiladas y la cuenta del socio está actualizada	
Excepciones	Paso	Acción
	3	Si alguna de las cintas está registrada como alquilada, el sistema comunicar la situación al empleado del vídeo-club y excluir la cinta del alquiler, a continuación este caso de uso continúa
Rendimiento	Paso	Cota de tiempo
	4	1 segundo
Frecuencia	50 veces/día	
Comentarios	ninguno	

UC-07	Devolución de cintas de vídeo	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-03 <i>Gestionar los alquileres</i> 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-02 <i>Información sobre socios</i> • IRQ-03 <i>Información sobre cuentas de socios</i> 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>un socio solicite devolver una o más cintas de vídeo</i>	
Precondición	Todas las cintas a devolver están registradas como alquiladas	
Secuencia normal	Paso	Acción
	1	El empleado del vídeo-club solicita al sistema comenzar el proceso de devolución de cintas de vídeo
	2	El sistema solicita que se identifiquen las cintas que se desean devolver
	3	El empleado del vídeo-club identifica las cintas y solicita al sistema que registre su devolución
	4	El sistema registra los devoluciones
	5	Si alguna cinta ha sido devuelta fuera de plazo, el sistema registra la multa correspondiente como un cargo en la cuenta del socio
	6	Si el socio decide pagar al contado, el sistema imprime el ticket con el importe correspondiente y registra el pago como un ingreso en la cuenta del socio
	7	Si el socio decide pagar a cuenta, el sistema registra el cargo en la cuenta del socio
Postcondición	Las cintas a alquilar están registradas como alquiladas y la cuenta del socio está actualizada	
Excepciones	Paso	Acción
	4	Si alguna de las cintas está registrada como alquilada, el sistema comunica la situación al empleado del vídeo-club y excluye la cinta del alquiler, a continuación este caso de uso continúa
Rendimiento	Paso	Cota de tiempo
	4	1 segundo
Frecuencia	50 veces/día	
Comentarios	ninguno	

UC-09	Ingreso a cuenta	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-03 <i>Gestionar los alquileres</i> 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-02 <i>Información sobre socios</i> • IRQ-03 <i>Información sobre cuentas de socios</i> 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>un socio solicite hacer un ingreso en su cuenta</i>	
Precondición	El socio tiene disponible su carnet	
Secuencia normal	Paso	Acción
	1	El empleado del vídeo-club solicita al sistema comenzar el proceso de ingreso en cuenta
	2	El sistema solicita que se identifique al socio y se indique la cantidad a ingresar
	3	El empleado del vídeo-club proporciona al sistema la identificación del socio y la cantidad a ingresar
	4	El sistema registra el ingreso e informa del nuevo saldo
	3	El empleado del vídeo-club comunica al socio su nuevo saldo
Postcondición	El saldo de la cuenta del socio está actualizado	
Excepciones	Paso	Acción
	–	–
Rendimiento	Paso	Cota de tiempo
	4	1 segundo
Frecuencia	5 veces/día	
Comentarios	Mientras no se implemente se puede hacer que todos los pagos sean al contado	

UC-13	Consulta de las películas alquiladas un día determinado	
Objetivos asociados	• OBJ-03 <i>Gestionar los alquileres</i>	
Requisitos asociados	• IRQ-01 <i>Información sobre películas</i>	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el empleado del vídeo-club lo considere oportuno</i>	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El empleado del vídeo-club solicita al sistema comenzar el proceso de consulta de las películas alquiladas un día determinado
	2	El sistema solicita la fecha del día que se quiere consultar, proponiendo la del día actual
	3	El empleado del vídeo-club proporciona la fecha del día determinado al sistema
	4	El sistema muestra una lista ordenada por número de alquileres con la siguiente información: título y tema de cada película y número de alquileres en el día determinado
	5	Si el empleado del vídeo-club solicita la impresión de los datos, el sistema imprime la lista
Postcondición	La información sobre las películas no ha cambiado	
Excepciones	Paso	Acción
	–	–
Rendimiento	Paso	Cota de tiempo
	4	5 segundos
Frecuencia	1 vez/día	
Importancia	importante	
Urgencia	hay presión	
Comentarios	ninguno	

UC-14	Consulta de los socios más rentables	
Objetivos asociados	• OBJ-03 <i>Gestionar los alquileres</i>	
Requisitos asociados	• IRQ-01 <i>Información sobre películas</i>	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el empleado del vídeo-club lo considere oportuno</i>	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El empleado del vídeo-club solicita al sistema comenzar el proceso de consulta de los socios más rentables
	2	El sistema solicita el periodo de selección: última semana, último mes, último año o siempre
	3	El empleado del vídeo-club proporciona el periodo de selección al sistema
	4	El sistema muestra una lista ordenada por cantidad de alquileres realizados con la siguiente información: número de socio, nombre, apellidos, teléfono y número de alquileres realizados en el periodo indicado
	5	Si el empleado del vídeo-club solicita la impresión de los datos, el sistema imprime la lista
Postcondición	La información sobre los socios no ha cambiado	
Excepciones	Paso	Acción
	–	–
Rendimiento	Paso	Cota de tiempo
	4	5 segundos
Frecuencia	1 vez/día	
Comentarios	Si el periodo es <i>siempre</i> , el tiempo de respuesta puede ser muy alto	

A.4. Requisitos no funcionales

NFR-01	Copias de seguridad
Objetivos asociados	–
Requisitos asociados	–
Descripción	El sistema deberá incorporar algún mecanismo que permita realizar copias de seguridad de los datos almacenados
Comentarios	ninguno

NFR-02	Entorno de explotación
Objetivos asociados	–
Requisitos asociados	–
Descripción	El sistema deberá funcionar en un entorno de 2 PC's Pentium con 16 Mbytes de RAM y 2 GBytes de disco duro conectados en red con sistema operativo Microsoft Windows 98
Comentarios	ninguno

NFR-03	Portabilidad
Objetivos asociados	–
Requisitos asociados	–
Descripción	El sistema deberá ser fácilmente portable al sistema operativo Microsoft Windows 2000
Comentarios	ninguno

Esta página está intencionalmente en blanco