

	PFIS – Práctica 2			 <div>DEPARTAMENTO DE TECNOLOGÍAS DE LA INFORMACIÓN</div> 
	Objetivos	Familiarizarse con la identificación, análisis y especificación de requisitos de un proyecto, utilizando para ello la herramienta REM (REquirements Management).		
	Sesiones totales	Sesión 1/4 (semana #7) Sesión (2-4)/4 (semanas #8 - #10)	Toma de contacto con la herramienta REM. Elaboración de un DRS por equipos de entre 4 ó 5 miembros.	
	Entrega	No se entrega el resultado de la presente sesión		
	Calificación	--		

Práctica 2 (Sesión 1/4). Toma de contacto con la herramienta REM.

1. Objetivos

Los objetivos de esta primera sesión son:

1. Elaboración de un Documento de Requisitos de Sistema (DRS).
2. Utilización de REM (*REquirements Manager*) como soporte del DRS.
3. Utilización de patrones lingüísticos.
4. Familiarización con el Diagrama de Casos de Uso, sus elementos y relaciones, y su correspondencia en la especificación en REM.

Junto con esta memoria se le entrega el siguiente material que necesitará para la elaboración guiada del Documento de Requisitos de Sistema, y que el alumno podrá descargar desde la plataforma Moodle:

- **DRS gestión de un videoclub.pdf**, que contiene un borrador de DRS, con las distintas tablas elaboradas siguiendo los patrones lingüísticos que propone la herramienta REM.
- **Imágenes.zip**, fichero con 4 imágenes, un diagrama de subsistemas y 3 diagramas de casos de uso correspondientes a los 3 subsistemas detectados, necesarias en el proceso de elaboración del documento.

2. Elaboración de Documentos de Requisitos en Asignaturas de Ingeniería del Software con REM

La relación entre la calidad de los requisitos y del producto final es hoy en día algo ampliamente asumido dentro de la Ingeniería del Software. Al contrario que hace unos años, en los que simplemente se hablaba del *análisis* de requisitos que se suponían proporcionados por el cliente, actualmente se reconoce como necesario un proceso mucho más complejo, la *Ingeniería de Requisitos*, en el que deben participar de forma activa tanto clientes y usuarios como desarrolladores de software.

Una de las actividades de la IR, es la **educación de requisitos**. Este subproceso, probablemente el más crítico y el más difícil de realizar, tiene como objetivos buscar, investigar y ayudar a los clientes y usuarios a documentar sus necesidades. La documentación de los requisitos deberá hacerse siempre usando el vocabulario de clientes y usuarios, de forma que éstos puedan entenderlos, siendo lo más habitual emplear lenguaje natural. Las técnicas más comunes son las entrevistas, reuniones en grupo, estudio *in situ*, etc.

Existen diversas propuestas para especificar los requisitos y nosotros optaremos por la Metodología de Durán y Bernárdez, que se ajusta a la herramienta REM.





Los casos de uso son una técnica sumamente conocida y utilizada en los métodos de desarrollo de software actuales. Esta técnica fue propuesta inicialmente por Ivar Jacobson y actualmente se encuentra incluida dentro de UML (*Unified Modeling Language*).

Para la correcta descripción de los casos de uso (que son una forma de expresar requisitos funcionales), los requisitos no funcionales, los requisitos de información, así como de otros requisitos del sistema, se recurre a la utilización de diversos tipos de plantillas, concretamente:

- *Objetivos del sistema.*
- *Requisitos de información.*
- *Requisitos de restricción (reglas de negocio)*
- *Actores.*
- *Casos de uso.*
- *Requisitos funcionales (expresados de forma tradicional, como texto libre).*
- *Requisitos no funcionales.*

Cada plantilla presenta los campos de información necesarios para especificar el concepto que está representando.

REM (*REquirements Manager*) es una herramienta de gestión de requisitos experimental desarrollada por uno de los autores como parte de su tesis doctoral. En REM, un proyecto de Ingeniería de Requisitos está compuesto por los cuatro documentos correspondientes a las cuatro vistas que pueden verse en la figura:

- *Documento de Requisitos del Sistema (Educación)* 
- *Documento de Análisis del Sistema (Análisis –modelos conceptuales-)* 
- *Registro de Conflictos y Defectos (Análisis, Verificación, Validación y Negociación)* 
- *Registro de Peticiones de Cambio (Gestión del cambio)* 

REM presenta un excelente generador de informes en HTML, para lo cual internamente maneja la información en XML, aplicándole una hoja de estilo XSLT a la hora de generar el correspondiente informe. Las hojas de estilo XSLT son configurables, por lo que permite la visualización de la información de los requisitos de acuerdo a diversos estándares, incluyendo la posibilidad de aplicar heurísticas de verificación automática.

3. El Documento de Requisitos del Sistema

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

La estructura del DRS puede verse en la siguiente figura:

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>

En las siguientes secciones se describe con detalle cada sección del DRS y cómo usar la herramienta REM en cada caso.

Portada

La portada del DRS debe tener el siguiente formato:

Proyecto *nombre del proyecto*

**Documento de
Requisitos del Sistema**

Versión *X.Y*
Fecha *fecha*

Realizado por *equipo de desarrollo*
Realizado para *cliente*

El *nombre del proyecto* coincidirá con el nombre del fichero **.rem**, de modo que en primer lugar debemos guardar el proyecto REM, nombrándolo tal y como queremos que aparezca en el documento. 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, 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.


El *equipo de desarrollo* y el *cliente* se podrán indicar cuando hayamos incluido las **Organizaciones** que participan en el proyecto.

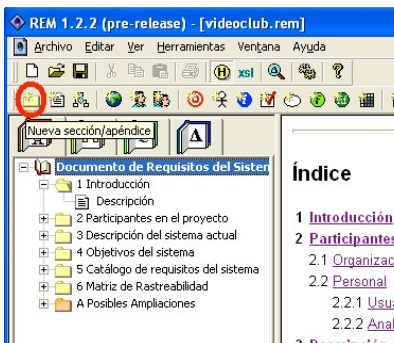
Lista de cambios

El documento debe incluir una lista de cambios en la que se especifique, para cada versión del documento, los cambios producidos en él. Para cada cambio realizado se debe incluir el número de orden, la fecha, una descripción y los autores. **Este apartado del DRS la vamos a obviar en las prácticas.**

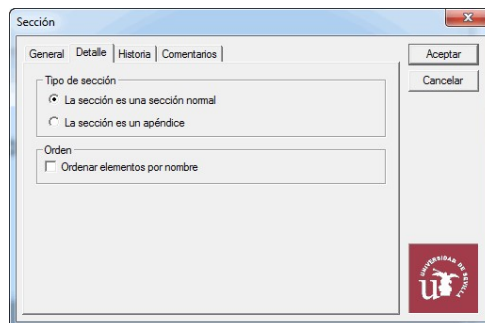
Índice

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.

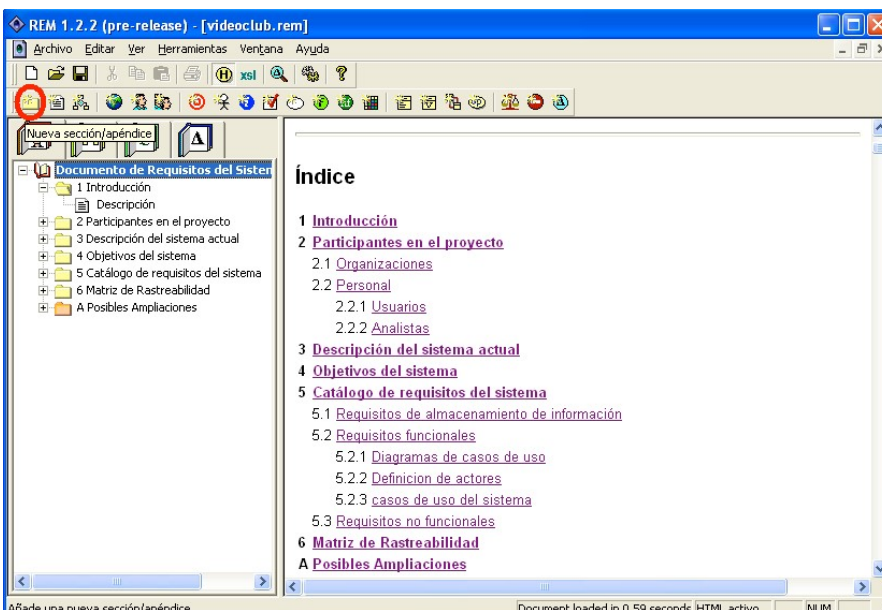
En REM usaremos la plantilla **“Nueva Sección/Apéndice”**  para ir estructurando el documento de acuerdo a la estructura indicada anteriormente. La herramienta genera dinámicamente en HTML las secciones que vamos añadiendo.



En la plantilla de Sección, pestaña **‘detalle’** es donde indicamos si el elemento es una sección o un nuevo apéndice (con lo que el icono tendría otro color, la numeración sería mediante letras y aparecerían al final del documento):



A continuación se muestra cuál debe ser el índice, para esta sesión práctica:



Como los diferentes **apartados** lo son del **Documento de Requisitos del Sistema**, cada vez que se añada uno hay que hacerlo teniendo seleccionado dicho documento.

Por lo tanto, **cuando queramos añadir un subapartado**, habrá que tener seleccionado el apartado al que corresponda.

Cada elemento del Índice tiene su menú contextual, desde el cual entre otras acciones podremos moverlo.

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.

Al ser un apartado que en REM hay que realizarlo manualmente, **lo vamos a obviar en las prácticas.**

Introducción

Esta sección debe contener:



- 1) Descripción breve de las principales características del sistema software que se va a desarrollar.
- 2) Situación actual que genera la necesidad del nuevo desarrollo.
- 3) Problemática que se acomete.
- 4) Cualquier otra consideración que sitúe al posible lector en el contexto oportuno para comprender el resto del documento.

En REM, tanto para la **Introducción** como para la **Descripción del Sistema Actual** se utiliza la plantilla '**Nuevo Párrafo/Elemento de Glosario**'  , seleccionando la pestaña **Detalle** para escribir el texto necesario.

Los alumnos deberán añadir el texto que consideren más adecuado para la Introducción al DRS.

Participantes en el proyecto

Esta sección debe contener una lista con todas las organizaciones y personas 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.

Para la sesión práctica, añadir dos Organizaciones  (la empresa que contrata y la que programa) y Personal  entre los que habrá varios analistas (los miembros del grupo) y varios usuarios finales (miembros de la empresa que contrata). **Habrà que prestar especial atención a la pestaña *Detalle*.**

Una vez añadidas las organizaciones podremos utilizarlas para **completar la portada del DRS**, para ello seleccionamos el DRS y desde su menú contextual escogemos la opción **Editar propiedades ...**, de manera que podremos rellenar los apartados **Preparado para** y **Preparado por** de la pestaña **General**, sin más que hacer clic en los correspondientes botones **Añadir organización**.

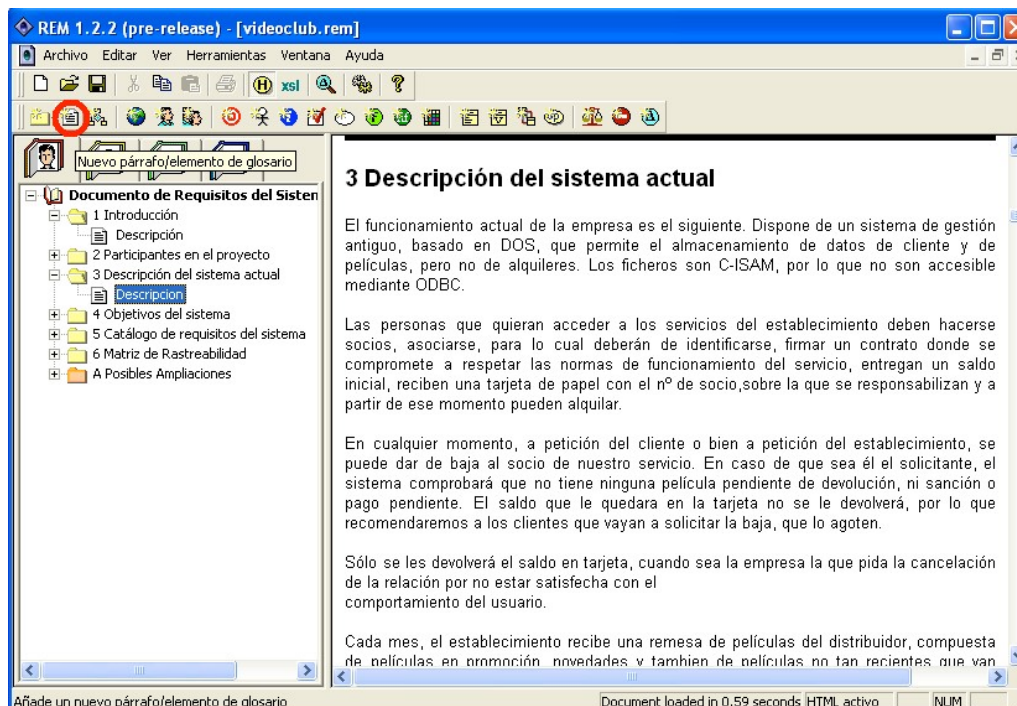
Descripción del sistema actual

Esta sección debe contener una descripción del sistema actual en el caso de que se haya acometido su estudio. Para ello puede utilizarse cualquier técnica que se considere oportuno (normalmente texto). 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.

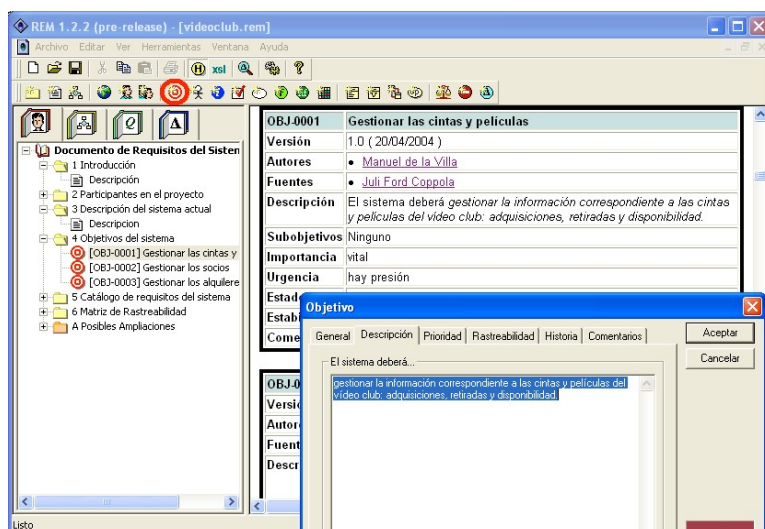
Para la realización de la sesión práctica, añadir un pequeño texto donde se describa cuál podría ser la situación actual de la empresa que nos contrata. En la siguiente imagen encontraréis un ejemplo:



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 .

Los objetivos del sistema pueden considerarse como requisitos de alto nivel, de forma que los requisitos propiamente dichos serían la forma de alcanzar los objetivos.



Los alumnos deberán tomar el documento aportado como material necesario para la realización de esta práctica, tanto para saber qué objetivos añadir, como para conocer el conjunto de requisitos de información, funcionales y no funcionales que se deberán ver reflejados en el DRS.

Subobjetivos: Se indican los subobjetivos, en el caso de que los haya, 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.

Importancia: Se 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**. En el caso de que no se haya establecido aún la importancia, se puede indicar que está por determinar (**PD**).

Urgencia: Indica la urgencia del cumplimiento del objetivo para los clientes y usuarios en el supuesto caso de un desarrollo incremental. Se puede asignar un valor numérico o una expresión enumerada como **inmediatamente, hay presión o puede esperar**, o **PD** en el caso de que aún no se haya determinado.

Estado: 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.
- **Pendiente de validación** si ya ha sido verificado y está a la espera de validación.
- **Validado** si ya ha sido validado por clientes y usuarios.

Estabilidad: 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.

Catálogo de requisitos del sistema

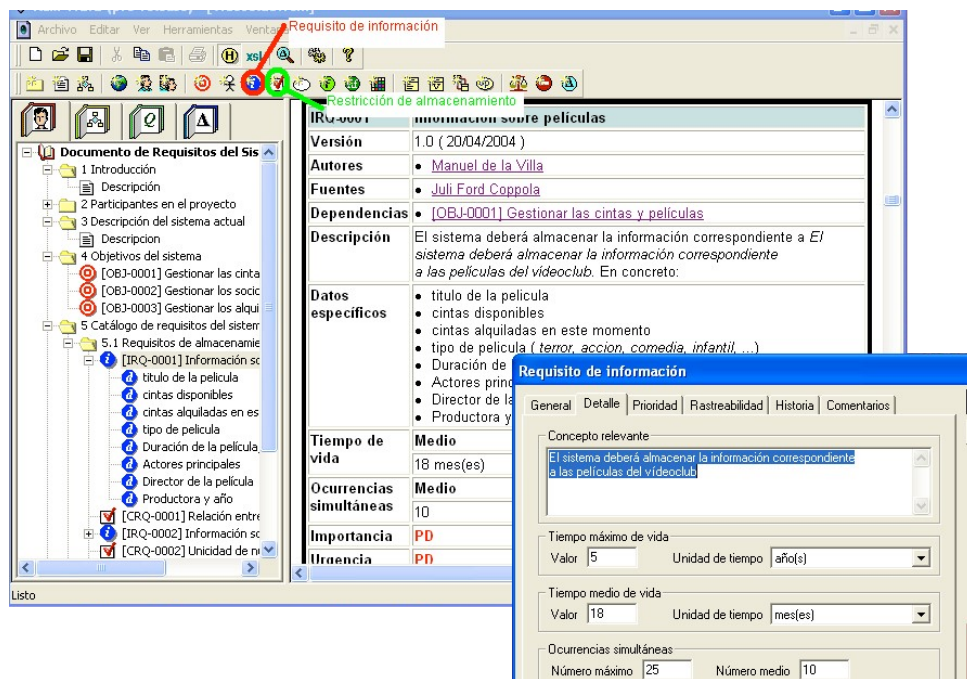
Esta sección se divide en las siguientes subsecciones en las que se describen los requisitos del sistema. Cada uno de los grandes grupos de requisitos, de información, funcionales y no funcionales, podrán dividirse para ayudar a la legibilidad del documento, por ejemplo dividiendo cada subsección en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

▪ 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 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.

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, 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?*".

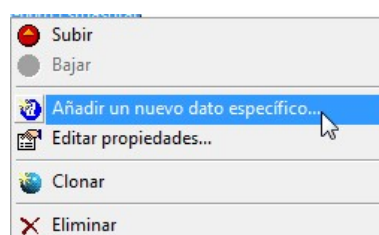


Dependencias (**Pestaña Rastreabilidad – Trazas desde este objeto-**):

- **Objetivos asociados:** Se indica la 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:** 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. Si queremos que los CU asociados aparezcan en este *atributo* y en la *Matriz de Trazabilidad* tendremos añadirlos también desde la *pestaña Rastreabilidad*.

Descripción (Pestaña Detalle – Concepto relevante-**):** 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.

Datos específicos (Menú contextual del Requisito de Información – Añadir un nuevo dato específico-**):** 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 (Pestaña Detalle): Indica el tiempo de vida medio y máximo que se espera para cada ocurrencia del concepto relevante.

Ocurrencias simultáneas (Pestaña Detalle): 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.

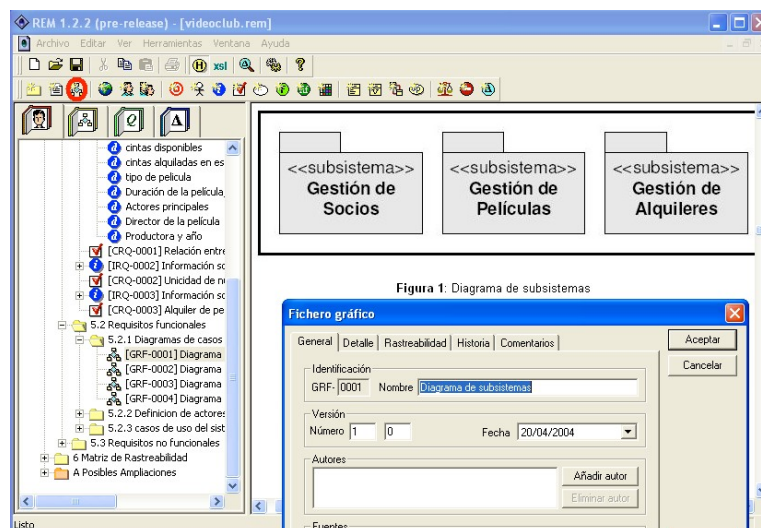
Importancia, Urgencia, Estado, Estabilidad, Comentarios (Pestaña Prioridad): Estos campos tienen el mismo significado que en la plantilla para objetivos, aunque referidos al requisito.

■ Requisitos funcionales

Esta subsección debe contener la lista de requisitos funcionales, expresados mediante Casos de Uso, que se hayan identificado, dividiéndose en los apartados que se describen a continuación:

1. Diagramas de casos de uso

Este apartado debe contener los diagramas de casos de uso del sistema que se hayan realizado. Para insertar un gráfico, usamos la plantilla señalada en la siguiente figura:



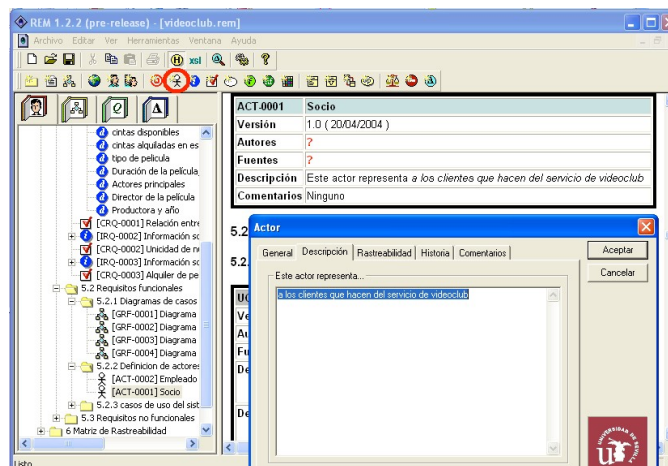
El nombre que le demos será el que aparezca en el documento a pie de ilustración. En “detalle”, indicamos la vía de acceso al gráfico.

Los alumnos deberán añadir el Diagrama de Subsistemas y los Diagramas de Casos de Uso aportados en el material necesario para la realización de la sesión práctica.

2. Definición de los actores

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.

Este apartado debe contener una lista con los actores que se hayan identificado, especificados mediante la plantilla para actores de casos de uso.



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.

3. Casos de uso del sistema

A continuación, se identificarán los casos de uso asociados a los actores, los pasos de cada uno de ellos y las posibles excepciones hasta definir todas las situaciones posibles.

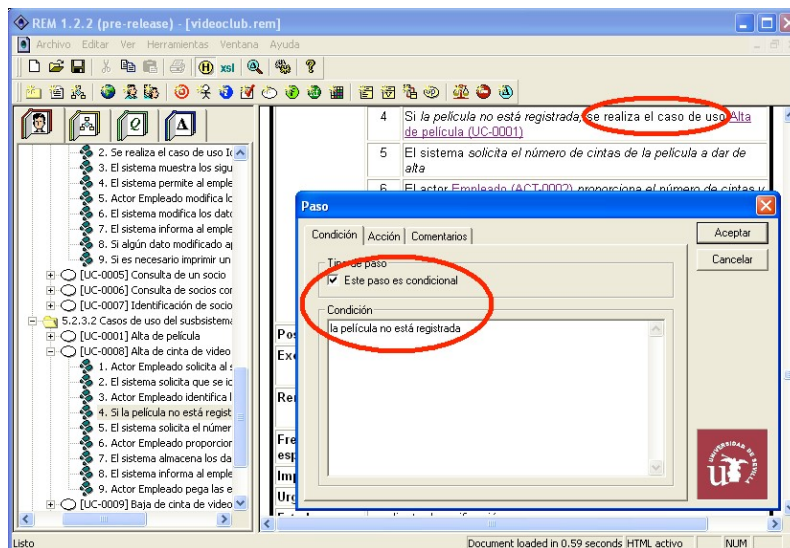
Los sistemas de información deben proporcionar unos servicios utilizando la información que en ellos almacenan. Las plantillas que ayudan a describir los casos de uso, facilitan a los clientes y usuarios la labor de responder a la pregunta “*¿qué debe hacer el sistema con la información almacenada para alcanzar los objetivos de su negocio?*”.

Este apartado debe contener los casos de uso que se hayan identificado, especificados mediante la **Plantilla de Casos de Uso**. 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 como casos de uso comienzan con **UC** y que 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*.

Descripción: 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**:

- 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. REM los añadirá automáticamente (**bastará con indicar que es abstracto**).
- 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.



Se asume que, después de realizar el último paso, el caso de uso termina.

En el caso de estructuras iterativas, pueden 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:

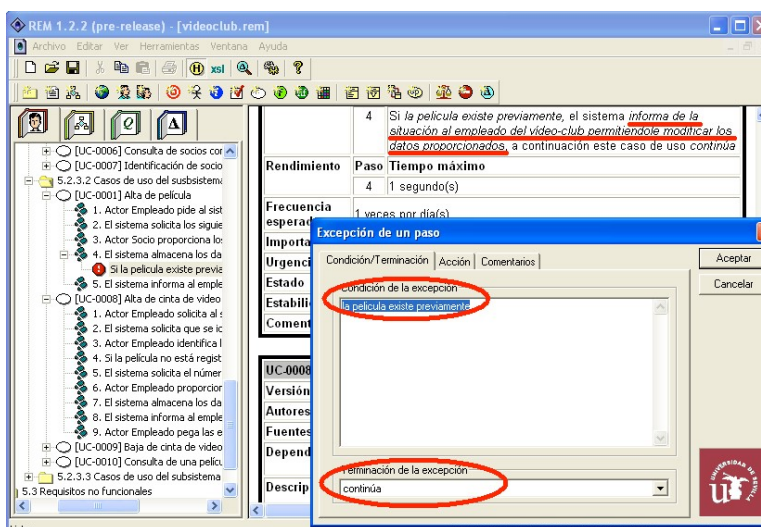
Repetir

Procesar artículo del pedido introducido por el usuario

Hasta que no haya más artículos

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, el caso de uso puede *continuar la secuencia 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, asumiendo una *semántica transaccional* del mismo.



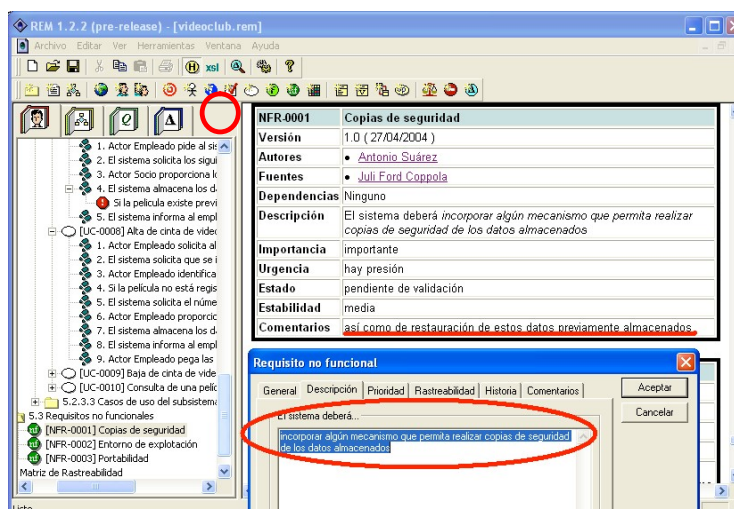
Para añadir una excepción a un paso, habrá que seleccionar dicho paso y desde su **menú contextual** seleccionar la opción **Añadir una nueva excepción ...**

Para cada excepción habrá que añadir la **condición**, la **terminación** y la **acción**.

Durante la sesión práctica se aconseja la realización sólo de los siguientes CU y en el orden indicado: **IDENTIFICACIÓN DE SOCIO, BAJA DE SOCIO, ALTA DE PELÍCULA, ALTA DE CINTA DE VIDEO.**

■ 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, normalmente de carácter técnico o legal. 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.



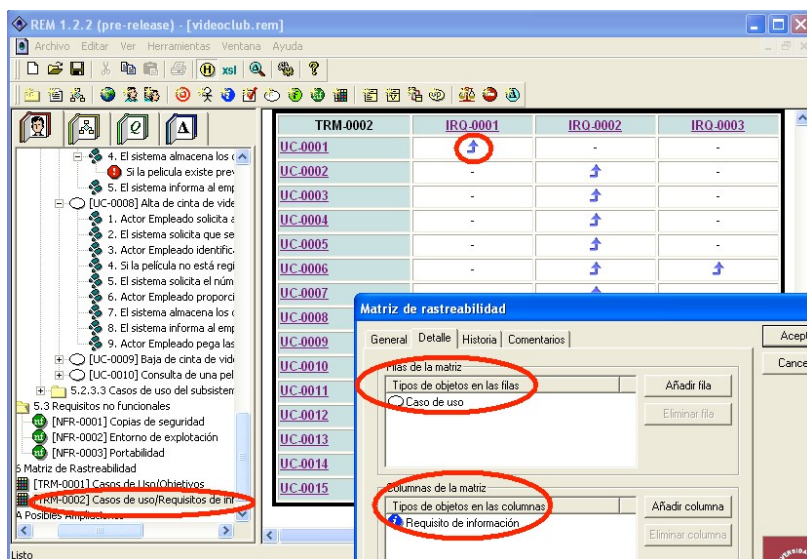
Algunos tipos de requisitos que se suelen incluir en esta sección son los siguientes:

- **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 8 como servidor y clientes Visual Basic .NET.*
- **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 Linux y Windows XP, siendo además posible el acceso al sistema a través de Internet usando cualquier navegador compatible con HTML 3.0.*

Matriz de rastreabilidad

Para la sesión práctica, esta sección deberá contener una **Matriz Requisito (de información y funcional)-Objetivo**, y una **Matriz Casos de Uso/Requisitos de Información**, de forma que para cada objetivo se pueda conocer con qué requisitos está asociado, y cada caso de uso de qué requisitos de información depende. Podríamos tener también una **Matriz CU/CU** para ver las dependencias entre los distintos CU, en cuyo caso habrá que haberlas indicado en la correspondiente plantilla de CU, en la **pestaña Rastreabilidad**.

El formato de la matriz de rastreabilidad puede verse en la figura.



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.

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.

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.

Referencias

- Durán, A. *Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información*. Tesis Doctoral, U. de Sevilla, 2000
- Durán, A. *REM Web Page*. U. de Sevilla. <http://klendathu.lsi.us.es/REM/>. 2002
- Durán, A., Bernárdez, B. *Metodología para la Elicitación de Requisitos de Sistemas Software (versión 2.3)*. Informe Técnico LSI-2000-10, U. de Sevilla. Abril 2002
- Durán, A., Bernárdez, B., Ruiz, A., Toro, M. *A Requirements Elicitation Approach Based on Templates and Patterns*. En *Proceedings of WER'99*, Buenos Aires, Argentina. 1999
- Durán, A., Ruiz, A., Corchuelo, R., Toro, M. *Supporting Requirements Verification Using XSLT*. En *Proceedings of the IEEE Joint International Conference on Requirements Engineering*. Essen, Alemania. 2002