

3.4 Técnicas de elicitación de requisitos

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].

Aunque la construcción de prototipos puede considerarse también como una técnica de elicitación, en esta tesis se le prestará especial atención como herramienta de validación de requisitos, sin dejar de reconocer que durante la validación mediante casos de uso y prototipos pueden elicitar nuevos requisitos (ver capítulo 5).

3.4.1 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 se pueden identificar tres fases: *preparación*, *realización* y *análisis* [Piattini *et al.* 1996].

3.4.1.1 Preparación de entrevistas

Las entrevistas no deben improvisarse, por lo que conviene realizar las siguiente 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 serlo⁷.

- **Planificar las entrevistas:** la fecha, hora, lugar y duración de las entrevistas 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].

⁷En [Goguen y Linde 1993] aparecen dos ejemplos de preguntas de cuestionario cuya respuesta no se previó de antemano por parte del autor del cuestionario dando lugar a respuestas, cuanto menos, curiosas.

3.4.1.2 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 3.6 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.

3.4.1.3 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.

3.4.2 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.

3.4.2.1 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.

3.4.2.2 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:

2.1 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.

2.2 Definir objetivos y requisitos: el jefe del JAD promueve la discusión para elicitar 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 3.6.

2.3 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*).

2.4 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 3.6.6.

2.5 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:

3.1 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.

3.2 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.

3.3 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.

3.4.3 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.

3.4.3.1 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 3.6, 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:

3.1 Revisar ideas: se revisan las ideas generadas para clarificarlas. Es habitual identificar ideas similares, en cuyo caso se unifican en un solo enunciado.

3.2 Descartar ideas: aquellas ideas que los participantes consideren excesivamente *avanzadas* se descartan.

3.3 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.

3.4.4 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 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 este trabajo

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 3.6.4, pág. 90).

3.4.4.1 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 3.1.

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.

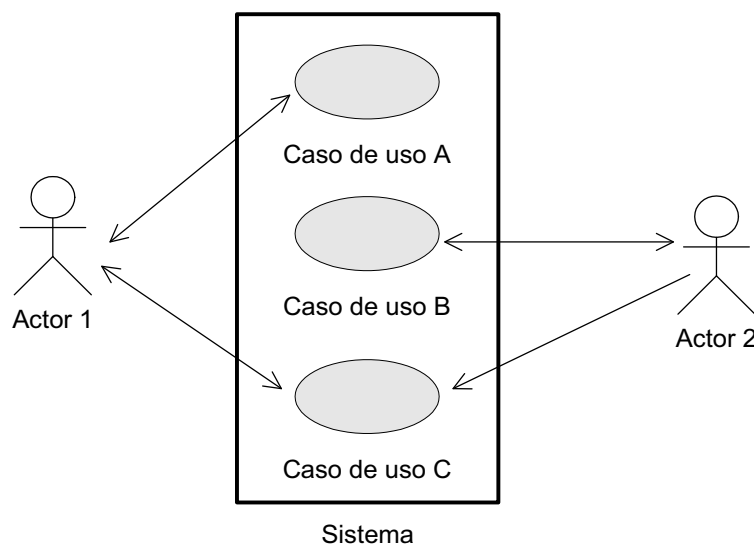


Figura 3.1: Diagrama de casos de uso

3.4.4.2 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 3.2.

- ***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.

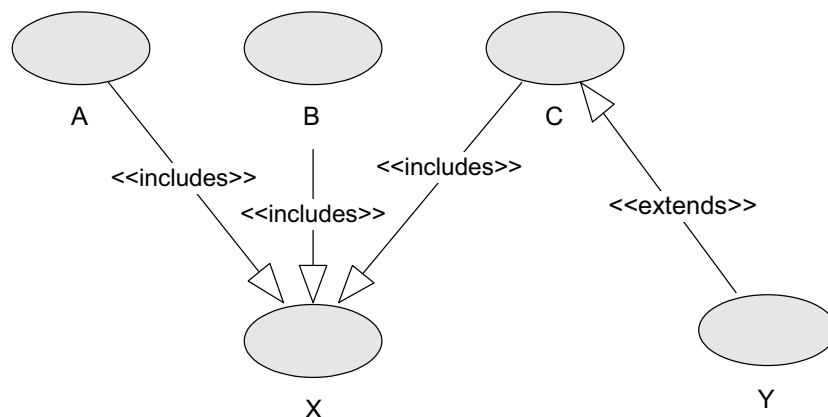


Figura 3.2: Representación gráfica de las relaciones *includes* y *extends*

- **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*.

3.4.4.3 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 3.3.

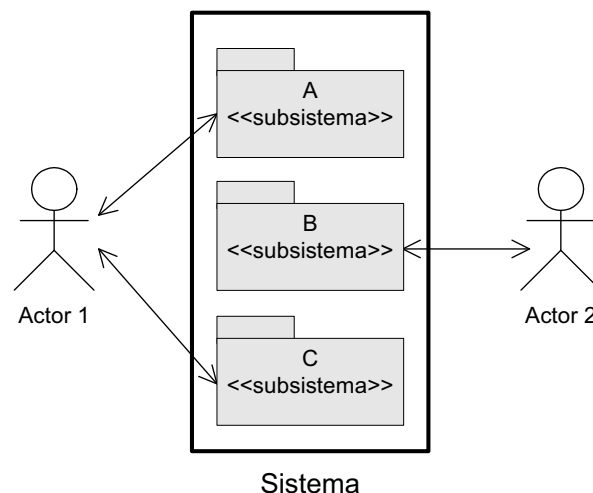


Figura 3.3: Representación gráfica de los paquetes de casos de uso