



CURSO 2019 - 2020

SOLUCIONES ACTIVIDAD EVALUABLE Y CALIFICABLE

1. El enunciado y planteamiento del caso de estudio.

El escenario en el que se situará el funcionamiento del caso de uso (pregunta 2), consiste en una aplicación para la **gestión de los recursos hídricos** de un territorio (aplicación llamada **HidroGest**).

Los recursos hídricos se refieren al agua de la superficie terrestre expuesta a la atmósfera y a lo relacionado con su tránsito, utilización y aprovechamiento.

Para su administración territorial, los recursos se agrupan en **Regiones**, que contienen **Cuencas** en las que cada una, a su vez, está constituida por todas las corrientes que drenan, finalmente, en un único lugar: el mar o un lago endorreico.

Para la regulación y el aprovechamiento del agua se utilizan embalses o **Presas**: acumulaciones *forzadas* situadas en los cauces de las **cuencas**.

El objetivo principal de la gestión hidrológica es poder realizar el seguimiento de la disponibilidad del agua y otros factores de su uso. Para ello, la **Medida** es el elemento fundamental que lo permite. En este escenario, simplificado, las **medidas** son volumétricas (hm^3) y se recogen, mediante el **sensor** correspondiente, en cada **Presa**. Es decir, se refieren al volumen de llenado de la **Presa**. Por consiguiente, la información más notable de una **Medida** (asociada a una **Presa**) es su valor (cantidad de agua acumulada, en hm^3) y la fecha en que se ha tomado.

Con esta red de sensores, el sistema **HidroGest** monitoriza las medidas recibidas durante una ventana temporal (tanto la frecuencia de muestreo como la amplitud de la ventana son configurables) mediante un conjunto de **Reglas** de supervisión y con la colaboración de un motor de razonamiento o **Sistema Experto** externo. Al final de cada día se almacena, de forma persistente, la última medida recogida, durante ese día, en cada **Presa**.

En definitiva, la funcionalidad proyectada para **HidroGest** se basa en:

- Gestión (CRUD) de los recursos hídricos. Se refiere a todos los datos e información asociada a los recursos que maneja el sistema para su funcionamiento (**Regiones, Cuencas, Presas, Medidas, Sensores**, etc.). Dicha información se almacena, de forma persistente, **en un sistema externo a HidroGest**.
- Configuración y mantenimiento de los parámetros del funcionamiento de la aplicación: reglas de supervisión, frecuencias de muestreo de los sensores, ventana de monitorización, zonas de enfoque, estrategia de almacenamiento de medidas, etc.
- Consultas de las medidas almacenadas. En general, una consulta se podría plantear con estos 3 aspectos:
 - Objetivo o qué se busca: puede ser las medidas o sus variaciones.
 - Campo o dominio de búsqueda: se puede referir a una **Presa**, a una **Cuenca** (en cuyo caso significa obtener el acumulado de todas sus

Presas) o a una Región (en cuyo caso significa obtener el acumulado de todas sus Cuencas y, así, recursivamente).

- Restricciones. Se refiere a cualquier grupo de condiciones que deban cumplir el objetivo o el dominio de búsqueda. Por ejemplo, '*entre 2 fechas*' o '*cuyo valor supere una cantidad*'.

Cualquiera de estos términos determina el tipo de la consulta, cómo se obtiene y el formato del resultado.

- Monitorización de la red y gestión de alarmas. El sistema permite realizar el seguimiento del estado de los recursos de la red, presenta alarmas o enfoca la atención en una zona de interés para tomar decisiones al respecto.

Detalles y simplificaciones admitidas:

- Como en el resto de la asignatura, la atención del estudio se dirige a la capa de la lógica de la aplicación, a los objetos del dominio del negocio y los mínimos servicios técnicos de apoyo que permitan interactuar con el acceso a los datos u otros sistemas considerados externos. Por tanto, la capa de presentación se considerará *transparente* y la interacción entre los actores humanos y la lógica del negocio será directa (como si se tratara de una comunicación mediante lenguaje de invocación de funciones a través de comandos). Igualmente, la interacción entre la lógica del negocio y los sistemas de apoyo externos se realizará a través de adaptadores.
- Todos los datos e información que requiere el funcionamiento de HidroGest están almacenados, de forma persistente, **en un sistema externo** de cuyo manejo *directo no debe* ocuparse la aplicación.
- La capacidad de una Presa es un dato inherente a ella. Sin embargo, la de una Cuenca depende de las presas que tenga registradas y se calcula como la suma de sus capacidades. De igual forma, la capacidad de una Región se calcula como la suma de las capacidades de las Cuencas que tenga registradas. Lo mismo ocurre con la ocupación de estos recursos: se calculan como la suma de los niveles de llenado del grupo de recursos subordinado.
- Dada la correlación existente entre los datos almacenados (que debería mantenerse en el funcionamiento de la aplicación), como simplificación, se considerará que la selección de un recurso en la IU lleva implícita la del recurso al que está subordinado, en el funcionamiento de la aplicación. Por ejemplo, la selección de una Presa implica, también, la selección de la Cuenca y la Región a la que pertenece.



Material recomendado para la realización de esta prueba:

- Libro de texto de la asignatura.
- Manuales y documentación adicional para el enfoque y comprensión de los contenidos del libro y de la asignatura. Referenciados en las indicaciones para el estudio y realización de las actividades en la guía de la asignatura, en su página web informativa y en el curso virtual.
- Soluciones propuestas para las PEC de los cursos 2016, 2017 y 2019.

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

Sección 1. Evaluación de los **Casos de Uso**

1. (0'5 puntos) Represente, en un diagrama UML de casos de uso, los casos de uso primarios más importantes, sus actores principales y de apoyo y las interacciones correspondientes, para la aplicación **HidroGest**.

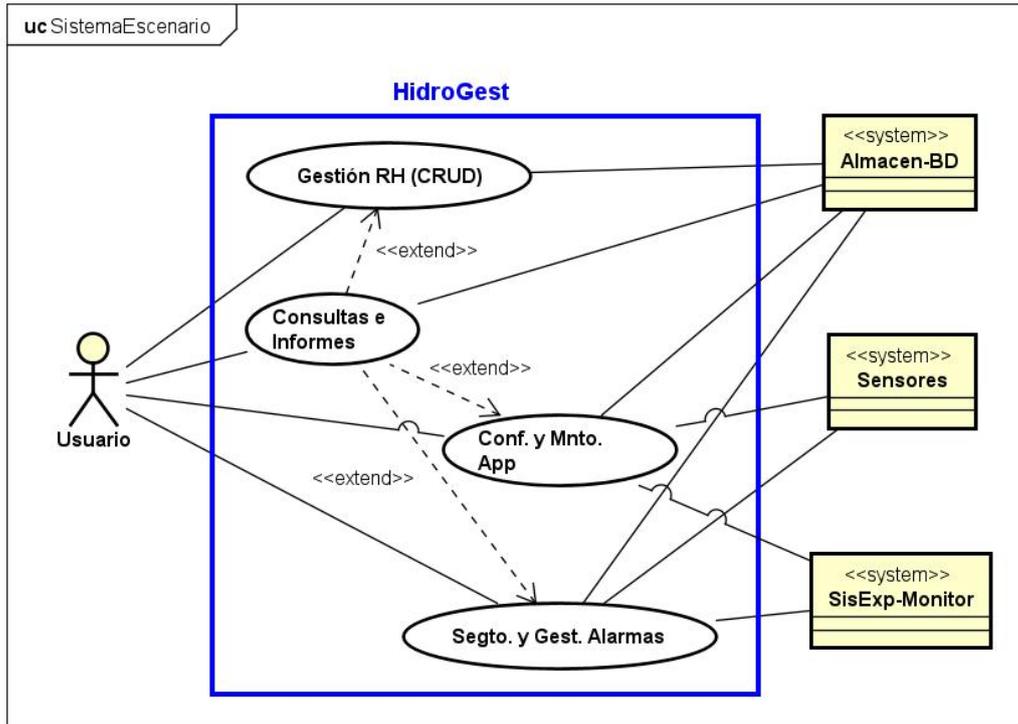
Notas:

Los Casos de Uso **no son orientados a objetos**. En la construcción progresiva del Modelo de Casos de Uso inicialmente se utilizan elementos que no son software (ni código, ni clases ni sus relaciones), sino que se refieren a cómo se usa la aplicación y qué información se necesita para ello.

No es hasta la implementación del funcionamiento de cada caso de uso (en el diseño, en la pregunta 4.1 y siguientes), cuando aparecerán las clases y cómo colaboran sus instancias para describir ese funcionamiento.

En este diagrama, la aplicación **HidroGest** delimita la principal funcionalidad que contiene (sus casos de uso primarios). Fuera de ella, se representan los actores primarios, su interacción (mediante la que los actores manejan determinados casos de uso) y los actores de apoyo que, mediante su interacción correspondiente, necesita cada caso de uso para cumplir con su funcionalidad.

Solución propuesta:



Nota: En el diagrama se han incluido relaciones <<extend>> (relación condicional) para facilitar la comprensión de la función que cumplen los casos de uso y en qué consiste cada una. **En un examen, no se recomienda que se entre en esos detalles, sino que se identifique con precisión dónde se sitúa en caso de uso de trabajo** (del enunciado en la pregunta 2) **y con qué actores interacciona.**

En este ejercicio, el caso de uso de trabajo es una de las opciones CRUD de 'Gestión de Recursos Hídricos', la opción *Create*.

Se recuerda: de aquí en adelante, todas las preguntas se refieren a las especificaciones definidas en la siguiente pregunta y para ese caso de uso. El objetivo es que realice el diseño para que también admita las otras funcionalidades y opciones de la aplicación.

- (1 punto) Escriba el caso de uso <<AgregarNuevaPresas>> en un formato completo (se recomienda la variante 'en dos columnas') y estilo esencial. Incluya tanto el escenario principal de éxito como 2 extensiones o flujos alternativos que pudieran ser frecuentes.

Consiste en agregar toda la información necesaria relativa a una presa nueva: nombre, capacidad, cuenca a la que pertenece y sensor asociado que realiza las medidas.

Nótese que cada Presa tiene asociado un Sensor (o grupo de ellos) que provea sus medidas. Se considera, por tanto, que el alta de una Presa requiere el alta del Sensor correspondiente (por conveniencia, en otro caso, podría no ser así); cuya información es: fabricante, marca, modelo, características de funcionamiento y medida, estado y número de serie.

No escriba un encabezamiento demasiado elaborado del caso de uso (es decir, omita *propósito, resumen, antecedentes...*); en su lugar, afronte directamente el transcurso típico de los acontecimientos.

Notas aclaratorias:

Recuerde que la escritura del caso de uso es un relato de la secuencia ordenada de '*lo que hace el actor*' y '*la respuesta que obtiene*' (la que ve él y, como mucho, algún hito importante en el cumplimiento de su objetivo de uso) al utilizar la aplicación.

Tenga en cuenta que el actor, aunque tenga la voluntad de conseguir su objetivo, no va a realizar ninguna acción a no ser que el sistema la presente como la única alternativa para conseguir ese objetivo. Por tanto, es el sistema (sin voluntad, pero con una programación determinista en su comportamiento) el que guía las acciones del actor.

En el relato de esta secuencia, olvídense de lo que hace la IU, la capa del negocio, los actores de apoyo externos o su repercusión en ellos. Considere *lo que no es el actor principal* como una *caja negra* y límitese a describir la secuencia de lo que pasa entre él y dicha caja negra.

Solución propuesta:

Caso de uso: **AgregarNuevaPresa**

Formato completo (variante 'a dos columnas'), estilo esencial.

Evolución típica de los acontecimientos

Acciones del actor (el Usuario de la aplicación)	Respuesta del sistema
1. El caso de uso comienza cuando el usuario solicita el registro de una nueva presa en los recursos hídricos que maneja la aplicación.	2. El sistema solicita que se indique a qué cuenca se asigna la nueva presa.
3. El usuario selecciona una cuenca existente.	4. El sistema solicita los datos que caracterizan la presa (nombre y capacidad máxima –en hm ³ —).
5. El usuario aporta los datos solicitados.	6. El sistema muestra un resumen de los datos de la nueva presa, incluyendo su identificador, que señala su registro persistente.
7. El usuario confirma los datos registrados.	

	(Opcional)
	8. El sistema ofrece la posibilidad de asignar un sensor a la nueva presa y, si no existe, su creación (otros 2 casos de uso).
9. El usuario acepta realizar un nuevo registro, esta vez, de un nuevo sensor.	10. El sistema solicita los datos que caracterizan el nuevo sensor (fabricante, marca, modelo, características de funcionamiento y medida, estado y número de serie).
11. El usuario aporta los datos solicitados.	12. El sistema muestra un resumen de los datos del nuevo sensor, incluyendo su identificador, que señala su registro persistente.
13. El usuario confirma los datos registrados.	(Opcional, por simetría con la presa. Ficticio)
(Ficticio. La presa es la recién creada)	14. El sistema solicita que se indique a qué presa se asigna el nuevo sensor.
15. El usuario selecciona una presa existente.	(Ficticio. El sensor ya se ha asignado a la presa recién creada)
(Ficticio. El sensor ya se ha asignado a la presa recién creada)	16. El sistema muestra el nombre de la presa a la que se ha asignado el sensor.
17. El usuario acepta los datos de la asignación.	

Notas aclaratorias:

La descripción y el contenido del caso de uso son *muy artificiales*. Tras reflexionar sobre sus necesidades procedimentales, se podrían identificar 3 usos que, para el manejo general del caso de uso primario CRUD de la aplicación, deberían ser independientes y cuyos límites están marcados con una línea en el paso correspondiente:

- Alta de una presa. Incluye su asignación a una cuenca. Pasos 1 a 7.
- Alta de un sensor. Pasos 9 a 13.
- Asignación de un sensor a una presa. Pasos 14 a 17. En este caso, esta operación está implícita en las otras 2.

Solución propuesta para los flujos alternativos:

Alternativas

- 4, 6 y 7 En 3) se produce un error en la selección de la cuenca. En 4) debería insertarse una comprobación que permitiera la corrección del error antes de registrar la presa de forma persistente. El error también se puede producir en 5) y, de la misma manera, debería insertarse una comprobación, de todos los datos, que permitiera su modificación antes del registro persistente, entre 5) y 6). Si, aun así, el error no se detecta antes del registro persistente, debe incluirse la alternativa 7a) para iniciar un procedimiento de modificación de ese registro.
- 12, 13 y 17 Los errores anteriores también se pueden producir en 11) o 15), por lo que deberían insertarse las comprobaciones análogas en 12) –17) ya lo es—, antes de que los datos se registren persistentemente. Si no se detectan, debe incluirse la alternativa 13a) para iniciar un procedimiento de modificación de ese registro.

- 8 Tal como se describe la marcha del caso de uso, este paso está implícito por lo que debería considerarse como un flujo alternativo.
- 14 a 17 Lo mismo ocurre con los pasos 14) a 17), que no tienen razón de ser según se enuncia el caso de uso de trabajo.

Notas aclaratorias:

La escritura del caso de uso es el primer paso para diseñar su funcionamiento mediante el código. Su objetivo es obtener una secuencia de operaciones, o acciones, *visibles* entre el actor y el sistema. Sin embargo, dicho objetivo está supeditado a un *horizonte final* que consiste en que el código funcione como se pide en la descripción y, además, sea funcionalmente independiente.

Es decir, la escritura del caso de uso es un elemento constructivo inicial del diseño y su utilidad está en hacernos reflexionar y comprender los requisitos funcionales del caso de uso.

Con ese análisis, y el objetivo final de la independencia funcional, se identifican 3 operaciones que claramente deberían diferenciarse e independizarse, en aras de la coherencia en el funcionamiento del caso de uso primario (CRUD) y de la aplicación en general, porque:

- Dada la significación y los datos que contienen las entidades que se están manejando (objetos conceptuales), no es lo mismo crear una presa que crear una región o que crear un sensor.
- Ya que hay distintas entidades implicadas en el caso de uso de trabajo, la vinculación entre ellas también está diferenciada y afecta al funcionamiento: en el caso de una presa, está implícita la necesidad de que esté vinculada a una cuenca determinada; mientras que, en el caso de un sensor, la vinculación a una presa es opcional.

Por tanto, la conclusión de que hay 3 operaciones diferenciadas en la descripción del caso de uso y de la conveniencia de independizarlas entre sí, nos lleva a plantearlo así desde el principio, es decir, como operaciones independientes en la escritura del caso de uso. Si se elabora la secuencia de acciones mediante una interpretación *literal* de la descripción del caso de uso, y se elabora el diseño basándose en ello, es prácticamente imposible obtener la independencia funcional y la flexibilidad que se están buscando o, para remediarlo, requeriría una reingeniería y refactorización muy costosas.

Éste es el motivo por el que se ha planteado el flujo principal del caso de uso como una secuencia de 3 subfunciones independientes. Así el comportamiento pedido se puede obtener, con facilidad, mediante la labor de un controlador que dirija el flujo de trabajo enlazando adecuadamente las 3 subfunciones.

Sección 2. Evaluación del **Modelado Conceptual**

3. (2 puntos) En relación con el caso de uso anterior, <<AgregarNuevaPresa>>, construya un Modelo de Dominio y represéntelo en notación UML. Represente los objetos conceptuales, las relaciones relevantes entre ellos, su cardinalidad y los atributos *candidatos* de los objetos.

Notas aclaratorias:

Con este diagrama se diseña la lógica del funcionamiento del caso de uso. La representación de esa lógica se fundamenta en que un objeto sólo puede '*hacer cosas*' con sus atributos. Por consiguiente, **la manera de expresar lo que hace un objeto, su funcionamiento, es asignarle los atributos o los componentes que necesita para hacerlo.**

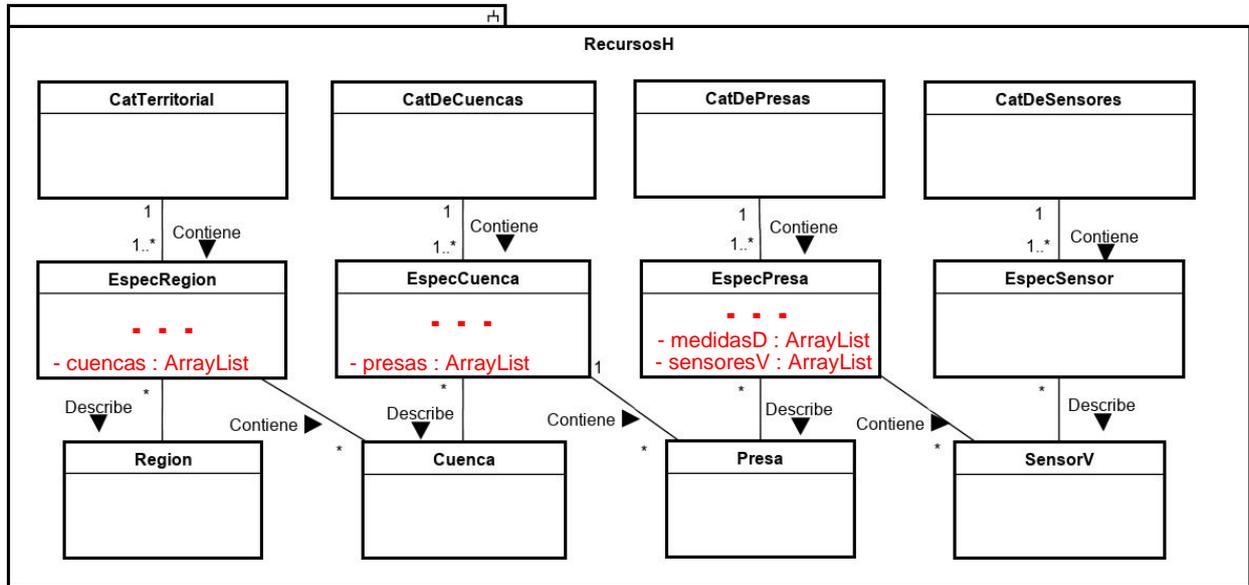
De todas las indicaciones que aparecen en el libro, sobre cómo representar los objetos conceptuales, sus atributos y las relaciones entre ellos, se resalta la que se refiere a la conveniencia de especificar, como otros objetos separados y relacionados con el anfitrión, aquellos atributos de tipo no primitivo con los que se va a hacer alguna operación o procesamiento en el funcionamiento del caso de uso. Así, por ejemplo, el contenido de *LineaDeVenta*, en el ejemplo PdV del libro, debe ser la cantidad de artículos, la descripción del producto y su precio. Sin embargo, los 2 últimos son atributos que describen un objeto del modelo de datos, *EspecificacionDelProducto*, mientras que el único atributo que se representa en el objeto *LineaDeVenta*, en este diagrama, es la información que aporta el actor: la cantidad de artículos, y no la que finalmente va a contener, con *EspecificacionDelProducto*.

Según las indicaciones de la Guía de la Asignatura, para los elementos con los que se construyen estos diagramas:

1. Se recomienda empezar por el objeto que identifica la función del caso de uso o su **objetivo**. En este ejercicio sería crear el objeto *AltaPresa* y, adicionalmente, la creación (vinculada) del otro elemento: *AltaSensor*. Ambos son el objetivo del uso del actor principal, y se refleja esa intencionalidad expresándola mediante la interacción 'Inicia'. Para deducir el contenido de estos objetos, en principio, podría pensarse en qué información aporta el actor. Información que, junto con la existente en el modelo de datos, permite al sistema realizar las operaciones necesarias para conseguir el objetivo; por lo que conviene simultanearlo con el siguiente paso.
2. Se continua con la representación del **modelo de datos**. Está constituido por los objetos que contienen los datos necesarios para que, junto con la información aportada por el actor, el sistema pueda alcanzar su objetivo. En este caso, esos datos se refieren a un conjunto de recursos hídricos. Tanto para construir este modelo de dominio como para elaborar todo el diseño posterior, es imprescindible organizar esos datos en estructuras (objetos):

- a. Que reflejen con fidelidad las relaciones que existen entre los datos, pero sólo aquellas que se necesiten para el funcionamiento del caso de uso. Esto es importante. Hay casos en los que existe una gran variedad de categorías en los datos que se manejan y, sin embargo, esas diferencias son irrelevantes para el funcionamiento del caso de uso. En esas situaciones, es mejor utilizar una colección de elementos única, lineal.
- b. Es fundamental que esas estructuras sean independientes, que estén desacopladas.

Con esas condiciones, la estructura propuesta, tanto para las operaciones CRUD como para las consultas, es esta:



Para el alta de los elementos que se piden en el caso de uso, sólo se requieren las colecciones de presas, de sensores y de cuencas.

3. El siguiente paso en la comprensión del funcionamiento del caso de uso es analizar qué datos aporta el actor y cómo se necesita combinarlos con los del modelo de datos. Si se asume que el caso de uso consiste en crear un elemento con la información que describe la presa y otro con la información que describe un sensor, se obtienen **las relaciones** necesarias entre los objetos del modelo de datos y los objetivos del caso de uso **AltaPresa** y **AltaSensor**.

Una cuestión muy importante en este análisis es comprender que el caso de uso consiste en crear un elemento nuevo de esas colecciones. Por tanto, la responsabilidad de realizar esa acción debe residir en el objeto que se ha concebido para manejar las colecciones y sus elementos: el correspondiente catálogo.

Es decir, el funcionamiento del caso de uso requiere que *algún objeto* transfiera los datos de **AltaPresa**, o **AltaSensor**, aportados por el actor, al catálogo que corresponda. Ahora bien, ni la estructura ni el contenido de **AltaPresa** **son los mismos** que los que se manejan

en la colección del catálogo, los elementos **EspecPres**a, por lo que *ese objeto (el controlador)* tendrá que *transformar* los datos aportados por el actor (**AltaPres**a –o **AltaSensor**–) para que el catálogo pueda realizar su cometido.

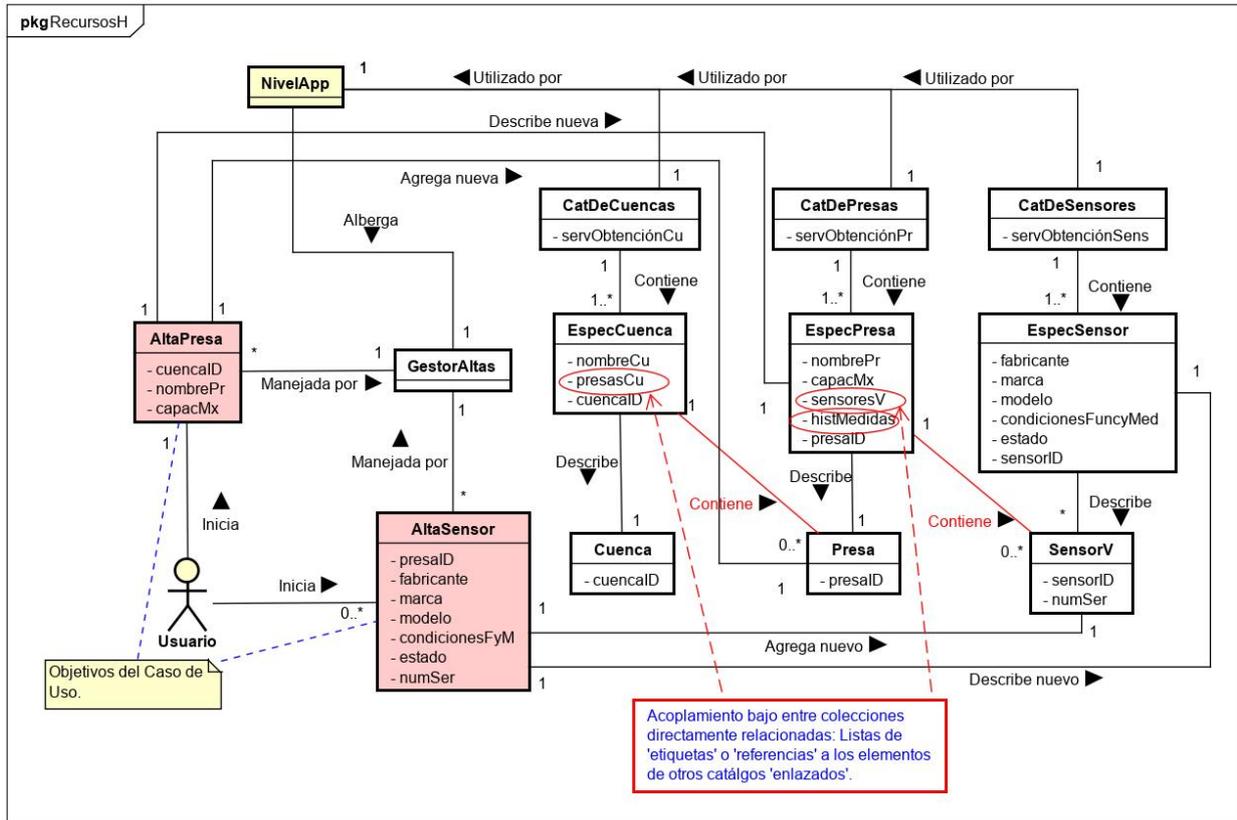
4. Para completar del diagrama del modelo de dominio sólo falta *dar forma* a la lógica del funcionamiento con unos objetos adicionales. **El controlador**, que ya se ha mencionado, es el encargado de organizar todo el funcionamiento y de establecer su *flujo de trabajo*. Como se ha descrito, va a recoger la información que aporta el actor, la va a manipular para transferirla al catálogo y que éste la procese. Por consiguiente, el controlador siempre tiene una estrecha relación con el objetivo del caso de uso.

Por el contrario, esto no es así respecto al modelo de datos y sus elementos. El contenido de dicho modelo de datos está en perfecta sintonía, aunque totalmente desacoplado, con los elementos externos (el sistema de información y de datos externo u otros servicios de apoyo). Esta vinculación se implementa mediante artificios de código (por eso no aparecen en el diagrama conceptual del modelo de dominio) cuyo manejo, como el del propio modelo de datos, se realiza globalmente a través de un controlador de nivel superior al del caso de uso o en el nivel del gobierno de la aplicación (nivel de aplicación). En el ámbito conceptual del modelo de dominio del caso de uso, es este controlador del nivel de aplicación el que contiene y maneja los elementos del modelo de datos, así como también contiene al controlador del caso de uso. Esas son las relaciones que se representan (las del controlador del nivel de aplicación con el modelo de datos y con el controlador del caso de uso) y no la relación entre el controlador del caso de uso (se ha denominado **GestorAl**tas) y el modelo de datos.

Por último, el **actor principal** del caso de uso, que no es un objeto, sino el origen de los eventos y de su interacción con el sistema para alcanzar su objetivo. Aunque esa interacción se implementa, mediante el código, canalizándola únicamente a través del controlador del caso de uso (**GestorAl**tas), el actor *no conoce* su existencia ni la de ningún otro componente software por lo que, en este nivel conceptual de la representación de esa interacción, las únicas relaciones que aparecen en el diagrama son las que se refieren a los objetivos de las acciones o eventos del actor (con **AltaPres**a y **AltaSensor**).

Advertencia: Los atributos de los catálogos, **servicioObtenciónXX**, son de la clase **AdaptadorBD-RecuH**. Su misión consiste en *conectar* los catálogos, y su contenido, con el servicio de almacén de datos externo. Por tanto, se corresponden con objetos puramente software que no deben aparecer en el diagrama del modelo de dominio.

El modelo podría ser similar a éste:

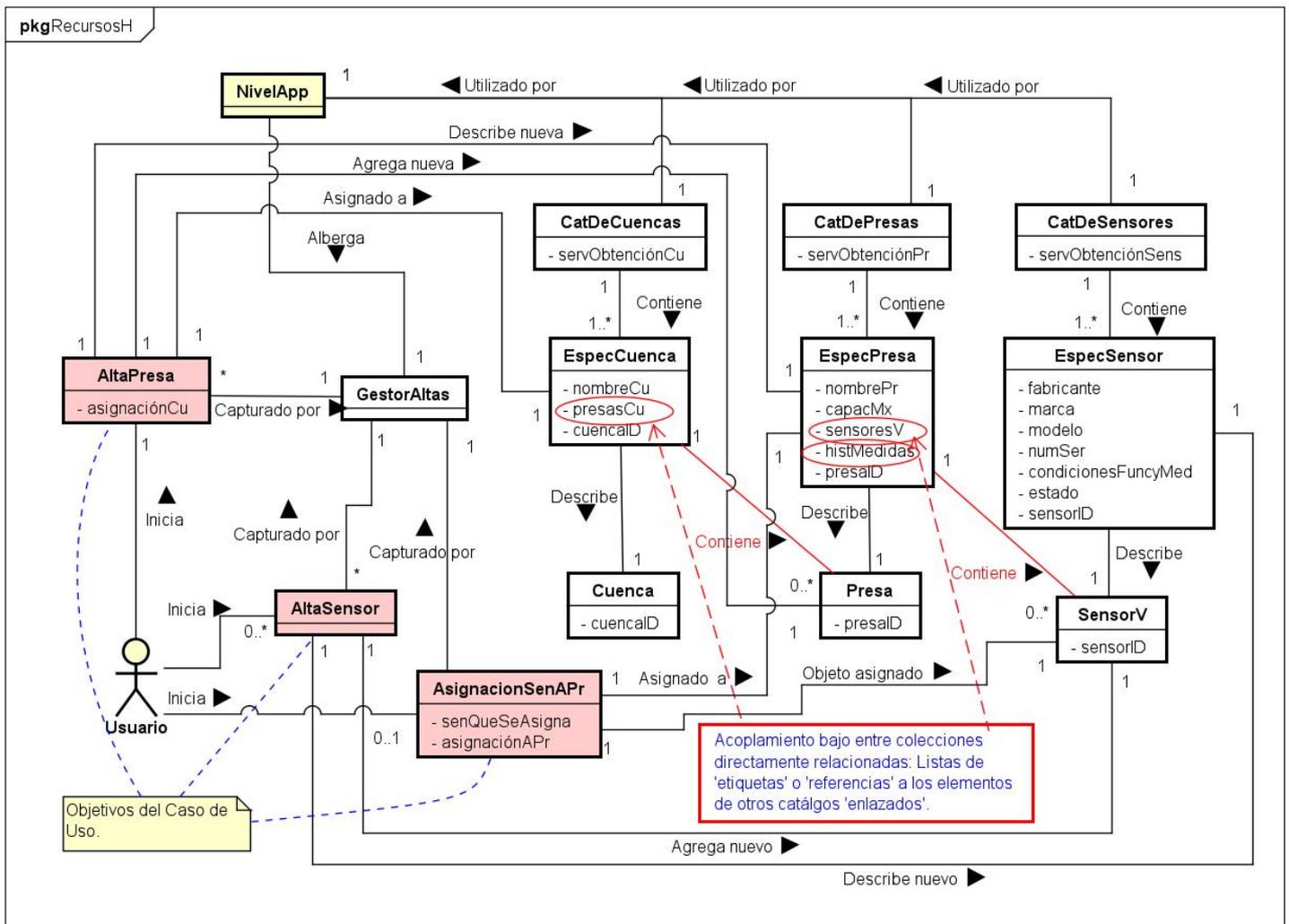


Conviene tener en cuenta unas cuestiones adicionales:

- Aunque el contenido de **AltaPresas** no coincide con el del nuevo elemento que se pretende crear (**EspecPresas**), a excepción de la información aportada para asignar la presa a una cuenca concreta (asignaciónCu:Cuenca), el resto, aunque incompleto, sí puede coincidir con la estructura del objeto **EspecPresas**. Es decir, ese atributo debería representarse como lo que es: un objeto separado con la estructura de **EspecPresas**, aunque esté incompleto.
- Según se ha concluido en el análisis del caso de uso y en la independencia de sus objetivos, **AltaSensor** se debe descomponer en otra subfunción para la asignación del sensor a una presa (**AsignacionSenAPr**).
- Una vez dividido el objeto, la estructura de **AltaSensor** sí coincide con la del elemento que se quiere crear, **EspecSensor**, aunque los datos aportados por el actor no estén completos.

En conclusión: aunque aquí aún no se está pensando en la implementación del funcionamiento del código (diseño detallado), la principal línea de razonamiento que debe primar al construir el modelo de dominio (diseño lógico del funcionamiento deseado para el caso de uso) es el principio de Experto en Información.

Solución propuesta:



Sección 3. (Diseño) Evaluación de los **Eventos del Caso de Uso** y **Asignación de Responsabilidades**.

4. **Eventos, Responsabilidades y Contratos.**

4.1. (2 puntos) Circunscrito al caso de uso anterior <<AgregarNuevaPresas>>, construya un Diagrama de Secuencia detallado (diagrama de interacción DS) en UML. Represente el actor, sus eventos y el paso de mensajes con las clases software del sistema para este caso de uso (en principio, las instancias de los objetos conceptuales que ha usado en el modelo de dominio).

¡ATENCIÓN!: se pide un diagrama de secuencia en el que represente el paso de mensajes entre el actor, el objeto que los recibe en el sistema y cómo se reparten entre los distintos objetos del modelo, y **NO** globalmente entre el actor y el sistema (el denominado Diagrama de Secuencia del Sistema –DSS–). Por tanto, **represente las líneas de tiempo de los objetos identificados en el modelo, NO las interacciones entre los actores y una única línea temporal correspondiente al objeto sistema global.**

Notas aclaratorias:

Siguiendo las conclusiones del análisis, se va a implementar el funcionamiento de 3 operaciones independientes:

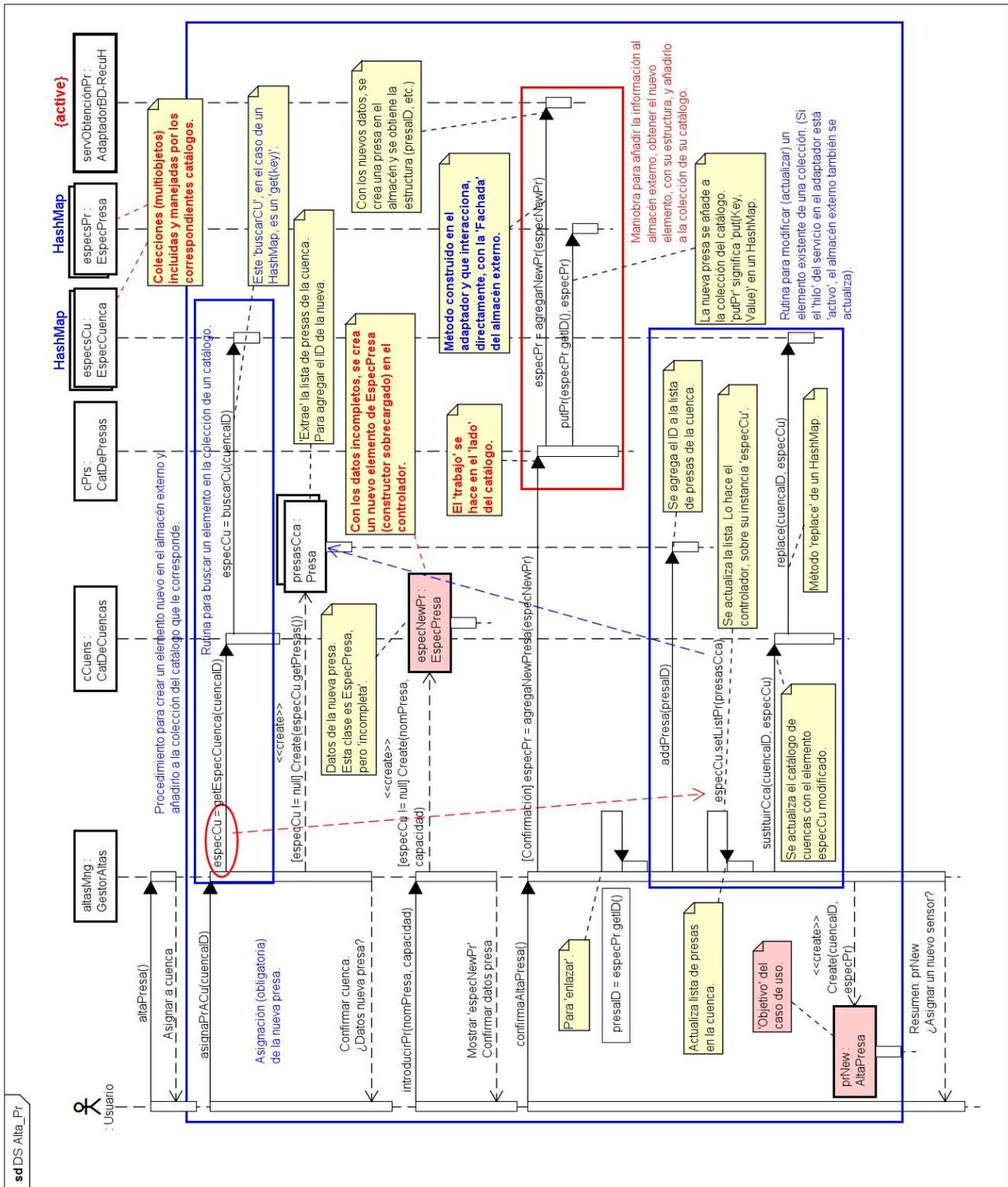
- Creación de una presa nueva. Incluye su asignación a una cuenca existente.
- Creación de un sensor nuevo.
- Asignación de un sensor a una presa.

Con esta premisa, el diseño obtenido es totalmente compatible con el comportamiento pedido para el caso de uso ya que, al haberlo construido evitando el acoplamiento (tanto en los datos como en las funciones) y con esa independencia funcional, cada bloque se puede concatenar con el otro sin más modificación que la del gobierno adecuado del controlador al invocar las funciones que los componen.

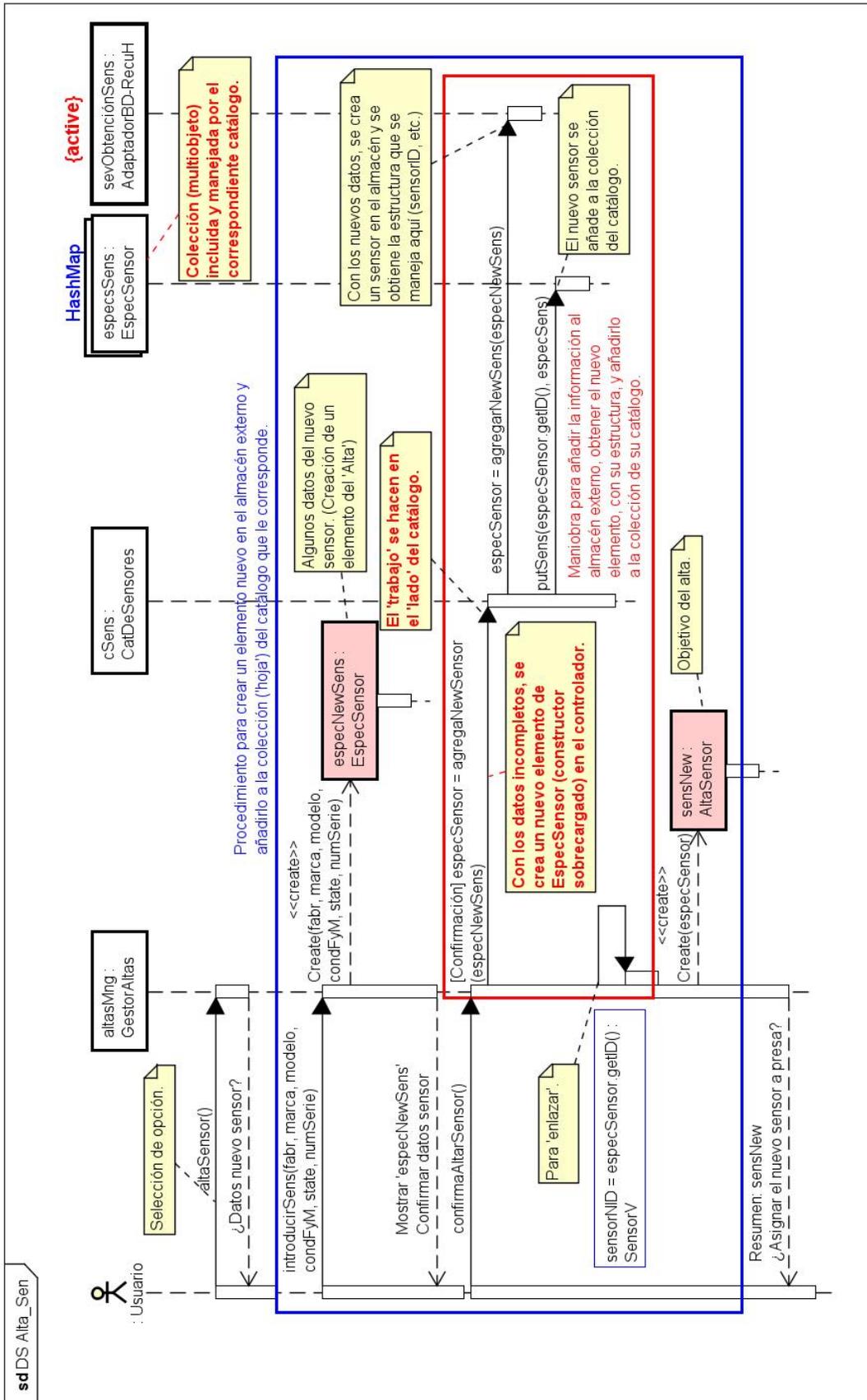
- Nótese que hay un gran número de procedimientos o maniobras, que se repiten constantemente (búsqueda de un elemento, añadirlo a una lista o a una colección, etc.). Su repetición también persiste en todos los ejercicios y exámenes, por lo que **se recomienda encarecidamente que se entiendan y se aprendan: son los elementos constructivos de todos los diseños.**
- A excepción de la asignación a una cuenca, el esquema de los procedimientos en la operación de **AltaSensor** tiene una acusada simetría con el de **AltaPresa**.
- Es importante darse cuenta de que, entre **AltaPresa** y el elemento del catálogo correspondiente, se produce una asociación; pero no porque el controlador **RegistroVentas inyecte**, en **LíneaDeVenta**, el elemento **EspecificaciónDelProducto** encontrado, como en la **Venta de PdV**, sino que (en el sentido contrario) son los datos que aporta el actor (**AltaPresa**) los que el controlador **inyecta** en el catálogo para que éste pueda realizar su labor. Esa asociación entre **AltaPresa** y **EspecPresa** ya se había representado en el modelo de dominio. Pero el hallazgo del diseño está en la asociación entre el controlador, **GestorAltas**, y ese elemento **EspecPresa**. El acoplamiento es tolerable porque se limita a un elemento **EspecPresa** y queda restringido al controlador **GestorAltas** (no tiene propagación u otra repercusión en otros objetos).
- Así mismo, el aislamiento de la aplicación, respecto a los datos y otros sistemas externos, la preserva de tener que ocuparse de manipularlos directamente, de gestionar la organización o mantener la consistencia de esos datos. Por la misma razón, también está incapacitada para crear, ahí, elementos persistentes por sí sola y asignarles un identificador que los distinga de los demás. De eso se ocupa el sistema externo, que gestiona el almacenamiento de esos datos e interactúa con la aplicación a través de su fachada (*del lado del sistema externo*) y del adaptador del correspondiente subsistema de persistencia (en la capa de acceso a los datos, *del lado de la aplicación*).

Ese adaptador se instala como un componente del catálogo al que proporciona sus servicios y, por tanto, sólo debería manejar las clases que él maneja. De esta forma, la petición al adaptador para que, a partir de la información incompleta de un elemento, solicite a la fachada del sistema de gestión del almacén de datos, a su vez, la creación de ese elemento (con su información completa), se debe hacer con *EspecPres* y con *EspecSensor*, respectivamente.

Solución propuesta para la creación de una nueva presa:



Solución propuesta para la creación de un nuevo sensor:



4.2. (1 punto) A partir de este DS, escriba y desarrolle los contratos de las operaciones 'AltaPresa' y 'AltaSensor'. Recuerde que es imprescindible utilizar una sintaxis específica en estas descripciones.

Solución propuesta:

Contrato CO1: **altaPresa**

Operación:	altaPresa(), que incluye: asignarPrACu(cuencaID), introducirPr(nomPresa, capacidad) y confirmarAltaPr().
Referencias cruzadas:	Caso de Uso: AgregarNuevaPresa.
Precondiciones:	<ul style="list-style-type: none"> - Hay una sesión activa, en la que se ha llegado a la opción CRUD para el 'Alta' de una presa en la colección <i>cPrs</i>, de CatDePresas. Para ello, hay una instancia de GestorAltas, <i>altasMng</i>. - Hay una instancia del catálogo CatDeCuencas, <i>cCuens</i>, inicializada y asociada a <i>altasMng</i>, de tipo GestorAltas. - Hay una colección HashMap (multiobjeto) de EspecCuenca, <i>especsCu</i>, asociada al catálogo <i>cCuens</i> de CatDeCuencas. - Hay una instancia del catálogo CatDePresas, <i>cPrs</i>, inicializada y asociada a <i>altasMng</i>, de tipo GestorAltas. - Hay una colección HashMap (multiobjeto) de EspecPresa, <i>especsPr</i>, asociada al catálogo <i>cPrs</i> de CatDePresas. - Hay una instancia del adaptador AdaptadorBD-RecuH, <i>servicioObtenciónPr</i>, inicializada y asociada al catálogo <i>cPrs</i>.
Postcondiciones:	<ul style="list-style-type: none"> - Asociada a <i>altasMng</i>, se ha creado una instancia de la clase EspecCuenca, <i>especCu</i>, e inicializado con los valores del elemento encontrado en el HashMap del catálogo <i>cCuens</i>, <i>especsCu</i>, en virtud del valor <i>cuencaID</i>, de Cuenca, aportado por el actor. - Asociada a <i>altasMng</i>, se ha creado una instancia ArrayList (multiobjeto), <i>presasCca</i>, e inicializado con los elementos del atributo <i>presasCu</i>, también ArrayList, de la instancia <i>especCu</i>, cuya clase es EspecCuenca. - Se ha creado la instancia <i>especNewPr</i>, de EspecPresa, e inicializado parcialmente con los valores aportados por el actor. Dicha instancia se ha asociado a <i>altasMng</i>, de GestorAltas. - <i>especNewPr</i>, de EspecPr, se ha asociado al catálogo <i>cPrs</i>. - <i>especNewPr</i>, de EspecPr, se ha asociado a <i>servicioObtenciónPr</i>, de AdaptadorBD-RecuH. - Asociada a <i>servicioObtenciónPr</i>, de AdaptadorBD-RecuH, se ha creado la instancia <i>especPr</i>, de clase EspecPresa, e inicializado completamente con los valores de <i>especNewPr</i>, de EspecPresa, junto con los obtenidos de la base de datos. El contenido de <i>especPr</i> ha quedado almacenado de manera persistente. - La instancia <i>especPr</i>, de EspecPresa, se ha asociado al catálogo <i>cPrs</i>, de CatDePresas. - Se ha agregado la instancia <i>especPr</i>, de clase EspecPresa, al HashMap <i>especsPr</i> (multiobjeto), componente del catálogo CatDePresas, <i>cPrs</i>. - La instancia del catálogo <i>cPrs</i>, <i>especPr</i>, de EspecPresa, se ha asociado a <i>altasMng</i>, de GestorAltas. - Se ha creado la instancia <i>presaID</i>, de Presa, inicializada con el valor del atributo <i>presaID</i>, de clase Presa, de la instancia EspecPresa, <i>especPr</i>. La nueva instancia <i>presaID</i> ha quedado asociada a <i>altasMng</i>, de GestorAltas. - Se ha agregado la instancia <i>presaID</i> al ArrayList de Presa (multiobjeto) <i>presasCca</i>.

- Se ha modificado el valor de *presasCu*, ArrayList de Presa, atributo de la instancia *especCu*, de la clase *EspecCuenca*, con el nuevo valor obtenido para el ArrayList *presasCca*.
- Se ha modificado el valor del elemento del HashMap *especsCu*, cuya clave es *cuencaID*, con el nuevo valor de *especCu*, de tipo *EspecCuenca*, en el catálogo *cCuens*.

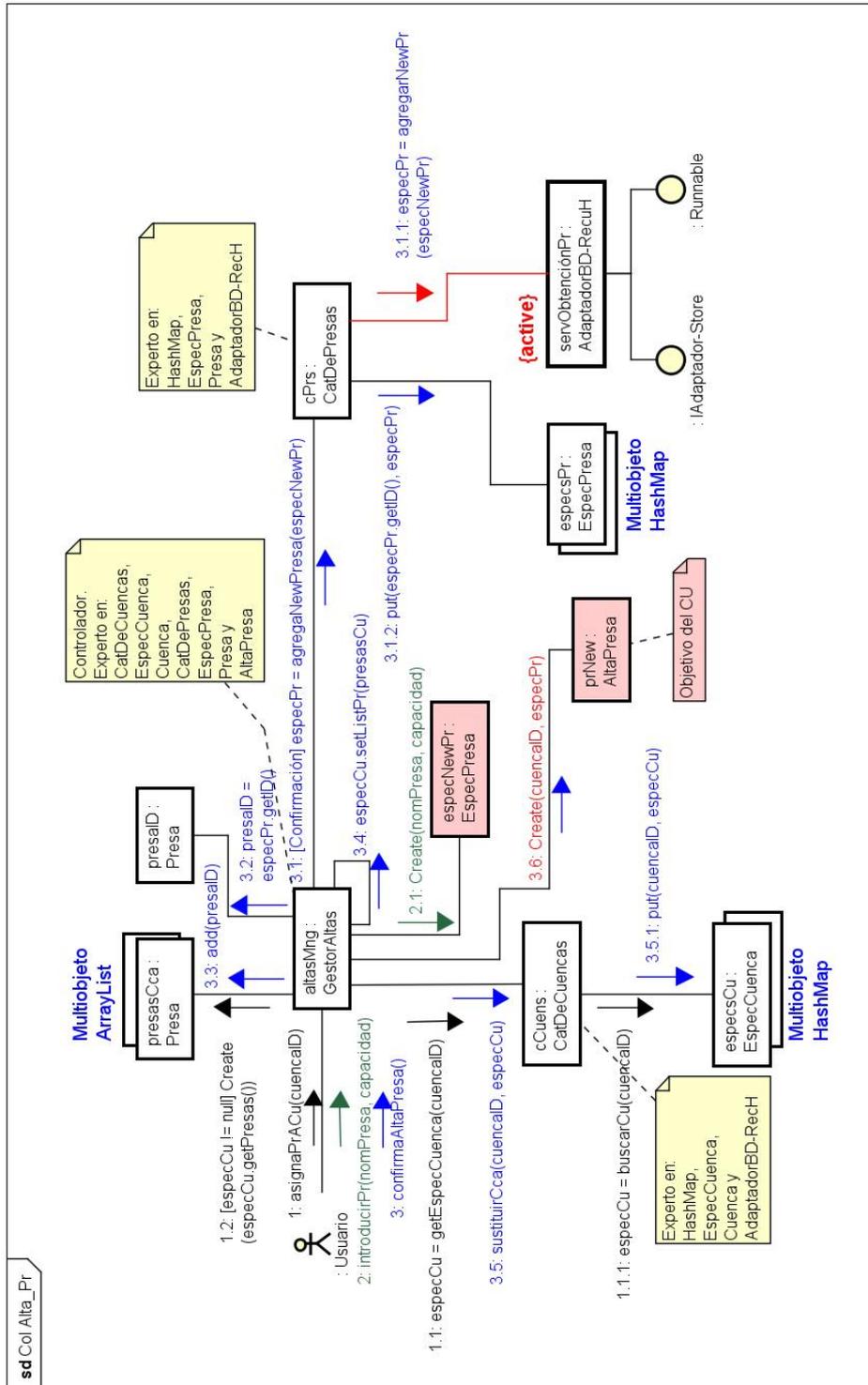
Contrato CO2: *altaSensor*

Operación:	<i>altaSensor()</i> , que incluye: <i>introducirSens(fabr, marca, modelo, condFyM, state, numSerie)</i> y <i>confirmarAltaSensor()</i> .
Referencias cruzadas:	Caso de Uso: <i>AgregarNuevaPresa</i> .
Precondiciones:	<ul style="list-style-type: none"> - Hay una sesión activa, en la que se ha llegado a la opción CRUD para el '<i>Alta</i>' de un sensor en la colección <i>cSens</i>, de <i>CatDeSensores</i>. Para ello, hay una instancia de <i>GestorAltas</i>, <i>altasMng</i>. - Hay una instancia del catálogo <i>CatDeSensores</i>, <i>cSens</i>, inicializada y asociada a <i>altasMng</i>, de tipo <i>GestorAltas</i>. - Hay una colección HashMap (multiobjeto) de <i>EspecSensor</i>, <i>especsSens</i>, asociada al catálogo <i>cSens</i> de <i>CatDeSensores</i>. - Hay una instancia del adaptador <i>AdaptadorBD-RecuH</i>, <i>servicioObtenciónSen</i>, inicializada y asociada al catálogo <i>cSens</i>.
Postcondiciones:	<ul style="list-style-type: none"> - Asociada a <i>altasMng</i>, de <i>GestorAltas</i>, se ha creado la instancia <i>especNewSens</i>, de <i>EspecSensor</i>, e inicializado parcialmente con los valores aportados por el actor: <i>fabr</i>, <i>marca</i>, <i>modelo</i>, <i>condFyM</i>, <i>state</i> y <i>numSerie</i>. - <i>especNewSens</i>, de <i>EspecSens</i>, se ha asociado al catálogo <i>cSens</i>. - <i>especNewSens</i>, de <i>EspecSens</i>, se ha asociado a <i>servicioObtenciónSen</i>, de <i>AdaptadorBD-RecuH</i>. - Asociada a <i>servicioObtenciónSen</i>, de <i>AdaptadorBD-RecuH</i>, se ha creado la instancia <i>especSens</i>, de clase <i>EspecSensor</i>, e inicializado completamente con los valores de <i>especNewSens</i>, también de <i>EspecSensor</i>, junto con los obtenidos de la base de datos. El contenido de <i>especSens</i> ha quedado almacenado de manera persistente. - La instancia <i>especSens</i>, de <i>EspecSensor</i>, se ha asociado al catálogo <i>cSens</i>, de <i>CatDeSensores</i>. - Se ha agregado la instancia <i>especSens</i>, de clase <i>EspecSensor</i>, al HashMap <i>especsSens</i> (multiobjeto), componente del catálogo <i>CatDeSensores</i>, <i>cSens</i>. - La instancia del catálogo <i>cSens</i>, <i>especSens</i>, de <i>EspecSensor</i>, se ha asociado a <i>altasMng</i>, de <i>GestorAltas</i>. - Se ha creado la instancia <i>sensorNID</i>, de <i>SensorV</i>, inicializada con el valor del atributo <i>sensorID</i>, de clase <i>SensorV</i>, de la instancia de <i>EspecSensor</i>, <i>especSens</i>. La nueva instancia <i>sensorID</i> ha quedado asociada a <i>altasMng</i>, de <i>GestorAltas</i>.

Sección 4. *Diseño de las Colaboraciones*

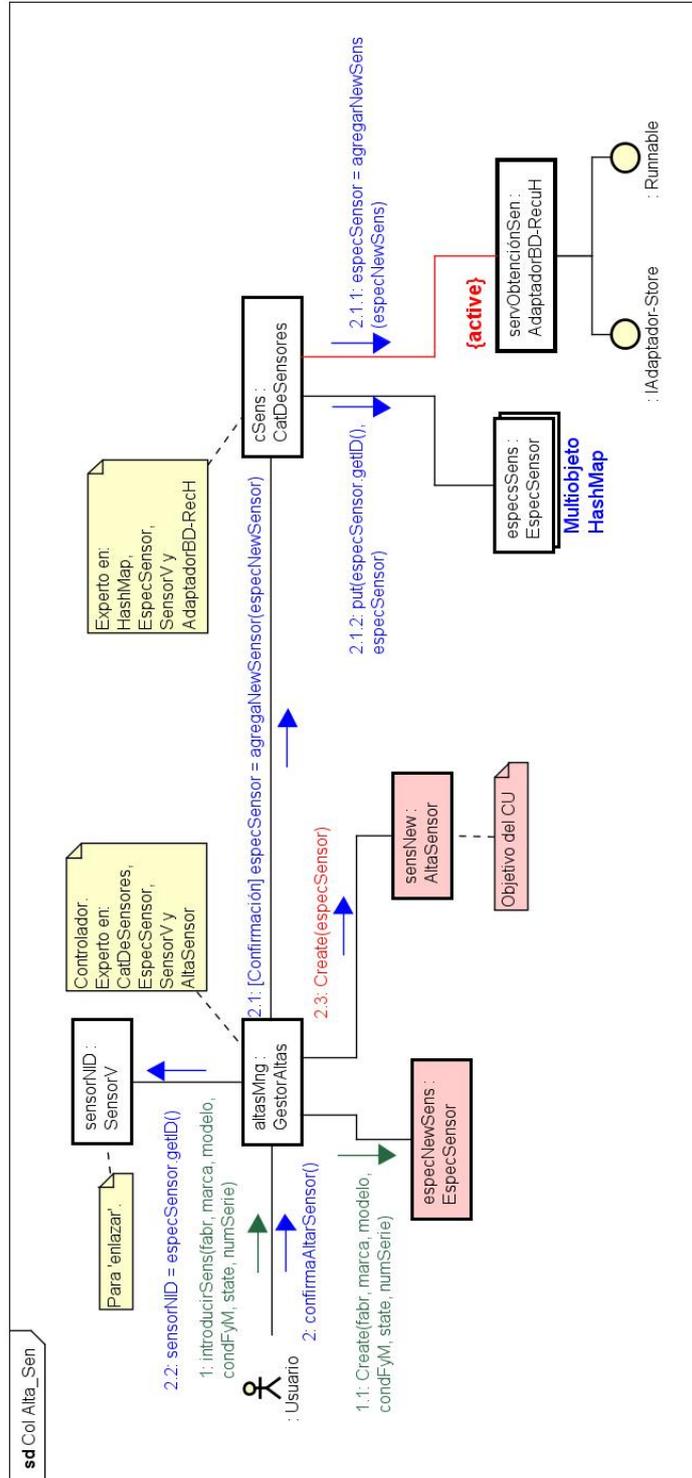
5. (1 punto) A partir del contrato de la operación 'AltaPresas' y de la correspondiente secuencia representada en el DS, que ha indicado en la pregunta 4, complete el diagrama de colaboración en UML. Consigne cada mensaje con los patrones GRASP (Experto, Creador, etc.) o cualquier otro que lo justifique. Si añade responsabilidades no explicitadas en el contrato (porque crea que es importante señalarlas), explíquelas brevemente.

Solución propuesta:

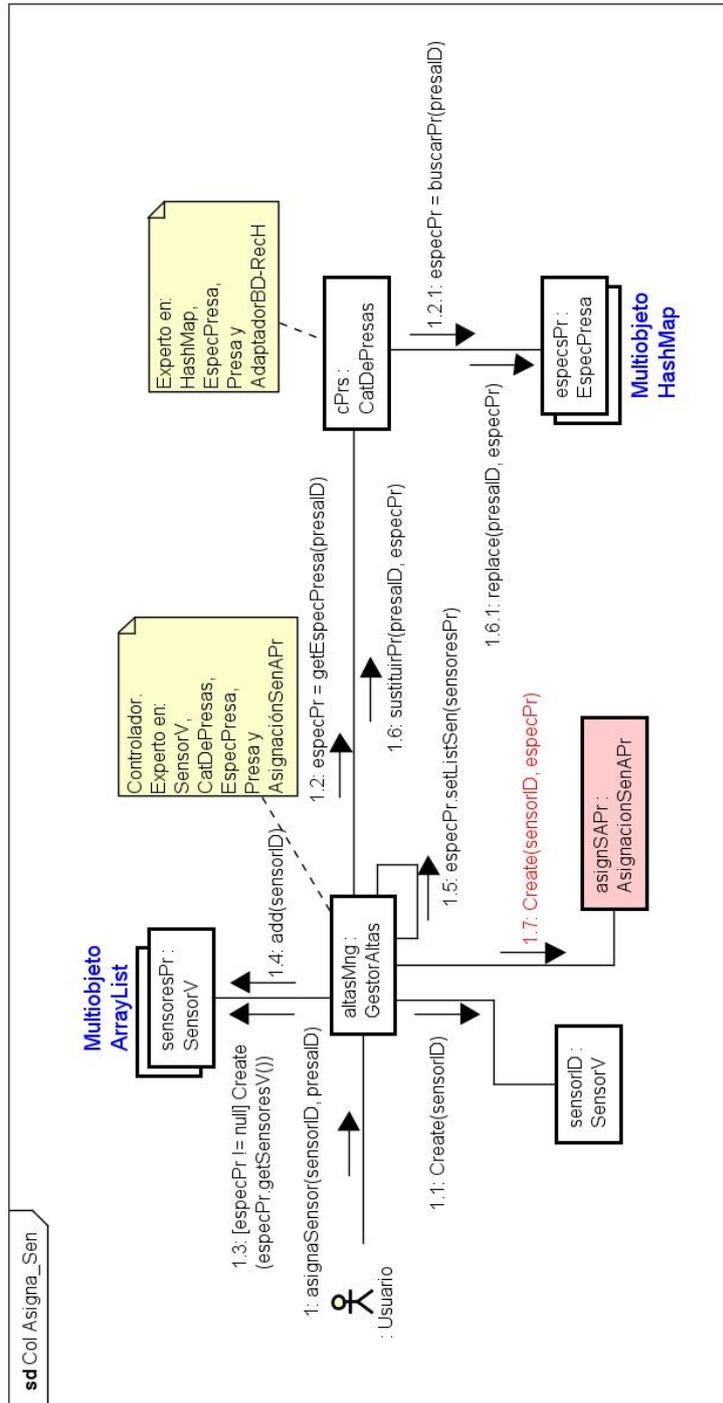


6. (1 punto) A partir del contrato de la operación 'AltaSensor' que haya indicado en la pregunta 4, complete el diagrama de colaboración en UML. Consigne cada mensaje con los patrones GRASP (Experto, Creador, etc.) o cualquier otro que lo justifique. Si añada responsabilidades no explicitadas en el contrato (porque crea que es importante señalarlas), explíquelas brevemente.

Solución propuesta:



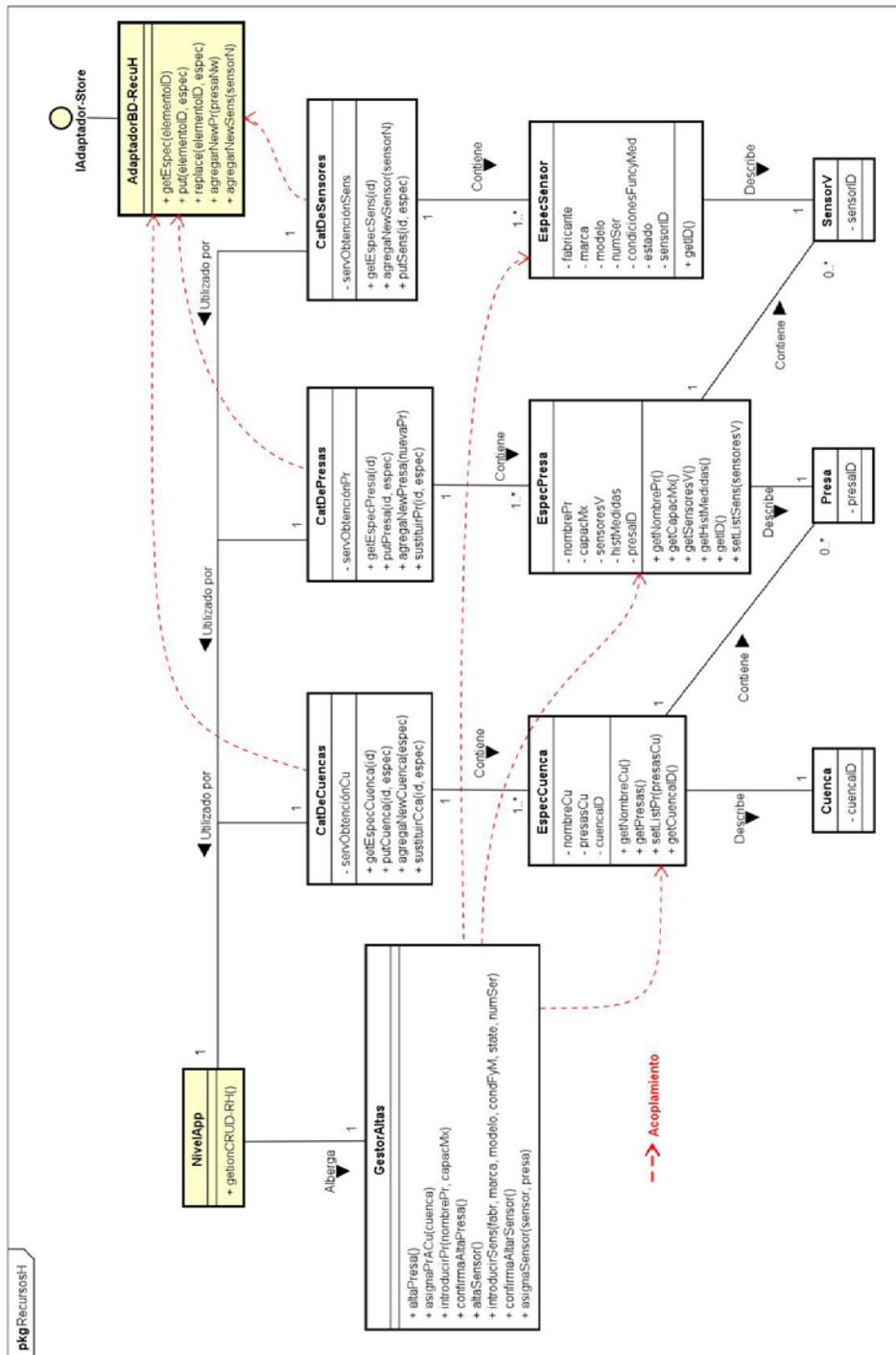
Ejemplo de diagrama de colaboración para la asignación de un sensor a una presa:



Sección 5. Evaluación del **Diagrama de Clases** de diseño

7. (1 punto) Elabore un diagrama de clases para el caso de uso que se está tratando <<AgregarNuevaPres<> (DCD), centrado en la clase que va a implementar la responsabilidad más característica del caso de uso, la que mejor defina la naturaleza de lo que se hace en él (Alta). Represente los nombres de todos los atributos, asociaciones (con la navegabilidad) y métodos de esa clase (excepto 'setters' y 'getters' irrelevantes) y de las que estén directamente involucradas con ella en el caso de uso.

Solución propuesta:



Notas aclaratorias:

- En el DCD sólo se representan las clases software que intervienen significativamente en el funcionamiento del caso de uso. Dicho funcionamiento consiste en crear 2 elementos pertenecientes al modelo de datos, por lo que se ha decidido que esas tareas se realicen en el controlador del caso de uso y en los objetos que manejan las colecciones en el modelo de datos. Por este motivo, las clases **AltaPresa**, **AltaSensor** y **AsignacionSenAPr** no tienen relevancia funcional, al limitarse a recoger los datos que aporta el actor, y no aparecen en este diagrama. De igual forma, tampoco se representa la interacción entre estos objetos software y los actores externos (entre ellos, el actor principal del caso de uso).
- Además de gran parte de los objetos software del modelo de dominio, en este diagrama sí se incluyen todas las clases puramente software que posibilitan el funcionamiento diseñado. Precisamente por la premisa de restringir el funcionamiento del caso de uso al dominio del negocio, y a mantenerlo desacoplado de cualquier manipulación de los elementos externos, se hace imprescindible utilizar un mecanismo, como el de los adaptadores (**AdaptadorBD-RecuH**), para implementarlo.
- Otra de las diferencias con otros diagramas (como el modelo de dominio) es que aquí se incluyen las asociaciones obtenidas en la elaboración del diseño dinámico del funcionamiento (pregunta 4, 5 y 6). Esas asociaciones se han dibujado con una línea punteada en rojo en el diagrama, y significan un acoplamiento adicional a las que representan las relaciones de inclusión o de composición del diagrama del modelo de dominio.

Sección 6. Evaluación de la **Transformación del Diseño en Código**

8. (0'5 puntos) A partir de los anteriores diagramas de clases y colaboraciones, elabore y defina la clase que haya establecido, en el desarrollo anterior, como responsable de controlar la correcta secuencia de acciones en el caso de uso <<AgregarNuevaPresa>>. Incluya las definiciones de todas las variables que la componen (miembros), pero escriba solamente la definición completa del cuerpo para el método (o métodos) principal o más significativo: <<se omite el método>>. Ignore los pequeños detalles de sintaxis -el objetivo es evaluar la capacidad fundamental para transformar el diseño en código-. Utilice la sintaxis de Java.

ATENCIÓN: lo que hay entre signos, <<se omite el método>>, es un ejemplo, usted debe sustituirlo por el nombre que haya asignado al método principal que haya elegido.