



CURSO 2014 - 2015

ACTIVIDAD EVALUABLE Y CALIFICABLE

1. Portada con:

Asignatura: 71013035 – Diseño de Software

Año de la práctica:

Centro Asociado al que pertenece:

Tutor que le da asistencia:

Datos personales: Nombre y apellidos:

DNI o número de expediente:

Contacto (teléfono o correo electrónico):

Localidad de residencia:

Datos de coste:

Horas de dedicación al estudio de los contenidos:

Nº de actividades no evaluables realizadas y horas de dedicación:

Horas de dedicación para realizar esta actividad:

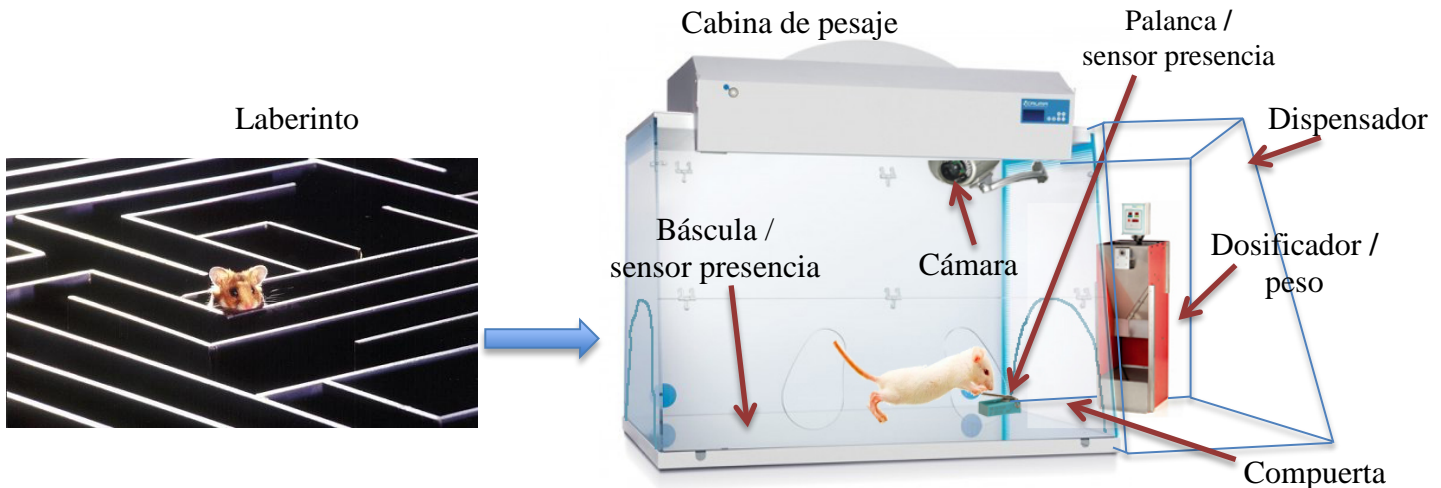
Juan del Rosal, 16
28040, Madrid

Tel: 91 398 89 10
Fax: 91 398 89 09

swdesign@issi.uned.es

2. El enunciado y planteamiento del caso de estudio.

El dominio del problema es un **sistema de información que da soporte, a los investigadores, para el seguimiento y estudio del comportamiento de una población de ratones sometidos a experimentos en el laboratorio (CompLab).**



El escenario del *Experimento* es un entorno habitable, en el que está la *Población de Ratones* estudiados, y un acceso laberíntico a dos *Dispensadores* de alimento: uno líquido (agua y, posiblemente, otros productos de composición conocida) y otro de comida (posiblemente, también, con otros productos sólidos de composición conocida). Cada *Dispensador* está situado en una salida distinta del laberinto; con el acceso, cerrado, que se abre mediante un dispositivo mecánico (pulsador o palanca) y precedido por una *Cabina* de pesaje que incluye una cámara (la cual se activa mediante algún sensor de presencia). Es decir, para acceder al contenido de un *Dispensador*, el individuo debe salir por la puerta correspondiente del laberinto, pasar por la *Cabina* de pesado (es detectado, pesado y fotografiado, en esta secuencia) y accionar la apertura del recipiente de alimento. Una vez abierto, el *Dispensador* provee una única *Dosis* y, cuando se cierra (*Raton* otra vez en la báscula de pesaje, detectado por el sensor de presencia de la *Cabina* y pesado por 2ª vez), el resto que no ha tomado el *Raton* se pesa, se registra (la diferencia) y se desecha.

Cada individuo del experimento tiene un chip (puede ser RFID o cualquier otra tecnología inductiva) que lo identifica en la puerta del *Dispensador* y en la *Cabina* de pesaje. Para ejecutar el *Experimento*, la programación de los *Dispensadores* es tal que no permite su apertura, a cada individuo, con un intervalo menor a 3 horas (para el líquido) ni de 5 horas (para el sólido). Un *Raton* comienza el *Experimento* cuando abre por primera vez cualquiera de

los dispensadores, momento en el que se da de alta como *participante*. Si el *Raton* no acciona la apertura de ningún *Dispensador* en 48 horas, el sistema le da de baja, automáticamente, como *participante* en el *Experimento*.

El sistema se comporta de manera que:

- El *Experimento* se realice según está programado (identificación de los *Ratones*, alta y baja de individuos, apertura y cierre de los *Dispensadores*, etc.)
- Se recojan y registren adecuadamente todos los datos necesarios, con los sensores instalados.
- Provea información a los investigadores en cualquier instante, tanto de la marcha del *Experimento* como de los datos de cualquier individuo (identificador, datos estáticos, *historia*, ...). '*historia*' es una tabla que recoge los datos históricos de los *Tratamientos* a los que se le ha sometido. Un *Tratamiento* contiene una *listaDeTomas*; cada *Toma* de la lista contiene el conjunto de los datos registrados, entre los que se encuentra *dosisTomada*, y *DosisTomada* consiste en la especificación de la dosis (*especDosis*) y la *cantidad* tomada.

El sistema ComporLab debe permitir diseñar e implantar distintos tipos de experimentos. En cada caso, el experimento consiste en realizar una acción (*tratamiento*) sobre los ratones con el objetivo de obtener unos resultados medibles. Es de gran importancia que el sistema registre los datos de cada aplicación del *tratamiento* (*Toma*) en un almacén permanente durante el transcurso del experimento y, una vez concluido, se añadan al historial de cada ratón. En este caso, el *Experimento* consiste en someter a un *Tratamiento* (suministrando, en cada *Toma*, productos en *Dosis* de composición fija y conocida: *DosisLiq* y *DosisSol*) a una *Poblacion participante* de *Ratones*, durante un período de tiempo –*duracion*– y registrar los datos para que los *Investigadores* puedan obtener conclusiones.

Los detalles y simplificaciones admitidas son:

- Inicialmente, los *Ratones* y sus propiedades estáticas están registrados en una base de datos (*Poblacion*).
- Se consideran experimentos simples, es decir, durante el intervalo de estudio –*duracion*– no se cambia a ningún individuo de la población ni se modifican las composiciones de las *Dosis* administradas.
- El escenario del laboratorio contiene sensores que advierten del mal funcionamiento de los dispositivos. Además, está diseñado para evitar medidas falsas o erróneas. Por ejemplo, la *Cabina* de

pesaje sólo admite a un individuo y está separada físicamente del *Dispensador* (el *Raton*, o está en un sitio o está en otro)

- El registro permanente de los datos de una *Toma* sólo debe realizarse cuando la acción ha concluido completa y correctamente, es decir, después de administrarse una dosis, que se haya cerrado la puerta, se haya pesado por 2ª vez (en esa sesión) y haya sido detectado por 2ª vez en el sensor de presencia de la cabina (en este orden)

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

RECOMENDACIONES: Durante la lectura del caso de estudio, preste especial atención a las palabras ‘marcadas’ (con otro tipo de letra, etc.) Después de tratar de entender el planteamiento anterior, léase todos los enunciados de las cuestiones siguientes, hasta el final. Esté atento a algunos indicios que le pueden ayudar para el enfoque de sus respuestas. Es posible que se le pida que describa o relacione algún elemento; lo cual podría hacerle sospechar que debería contar con él.

PREGUNTAS GUÍA: Estas preguntas no forman parte del ejercicio. Su objetivo es ayudar a que dirija sus conclusiones adecuadamente.

¿Qué hace, exactamente, este software?

¿De dónde provienen los estímulos? Es decir ¿quiénes son los actores y cuál debe ser su comportamiento (estímulos), frente al que debe reaccionar el software?

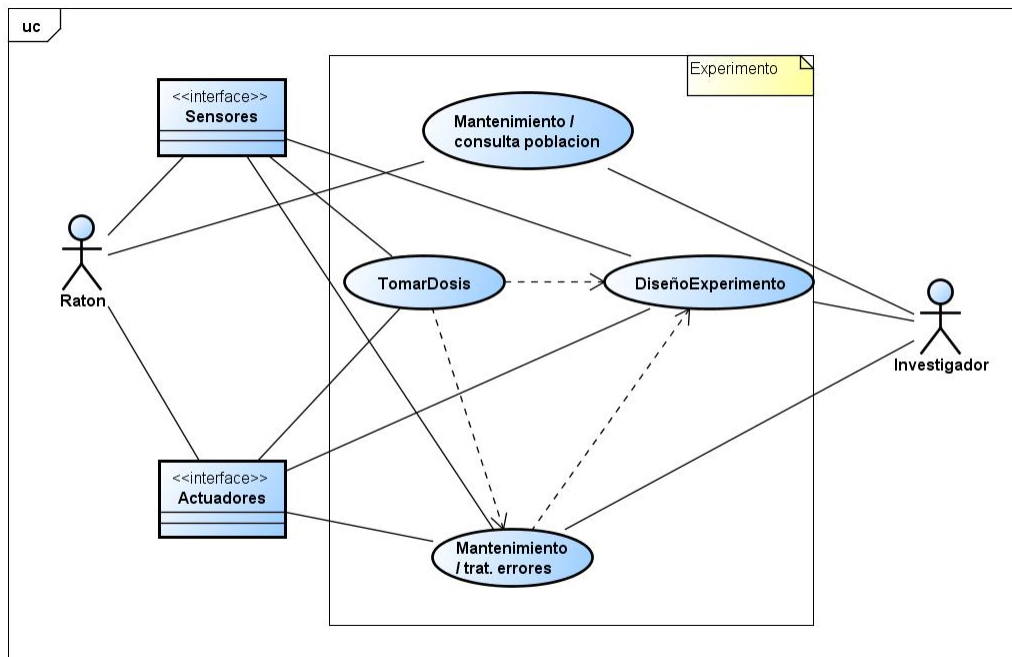
Una vez ‘aislado’ el software y comprendido el objetivo primordial de su funcionamiento ¿qué secuencia de operaciones debe realizar, como reacción a los estímulos, para que el funcionamiento sea el deseado (Caso de Uso)?

En el comportamiento del software, la situación es parecida: un objeto reacciona ante un estímulo que proviene de otro objeto o del exterior (un actor). Esa es, precisamente, la tarea de asignar responsabilidades. Por tanto, la pregunta es: ¿qué objeto atiende un estímulo y cómo reacciona ante él? Nunca hay que perder de vista que, en definitiva, **estamos hablando de código** y que, esos estímulos, se traducen en llamadas a métodos. Por consiguiente, es inaceptable una invocación a un método inexistente o aquellas llamadas en las que se solicita una reacción que es imposible realizar

porque no se dispone de la información necesaria (falta de parámetros, etc.) o ésta es inaccesible.

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

1. (0'5 puntos) En relación a este *Experimento*, identifique al menos 4 casos de uso primarios y sus actores correspondientes. Represente los resultados en un diagrama de casos de uso de UML.



powered by Astah

2. (1 punto) Escriba el caso de uso <<TomarDosis>> en un estilo y formato completo y esencial. Incluya tanto el escenario principal de éxito (flujo básico correspondiente a una *Toma*, una sesión en la que un *Raton* –ya dado de alta como participante en el *experimento*, es decir, con un *tratamiento* asignado— entre en la *Cabina*, acceda al *Dispensador*, se administre una *Dosis* y salga al laberinto otra vez) como 2 extensiones o flujos alternativos que pudieran ser frecuentes. No escriba un encabezamiento demasiado elaborado del caso de uso (es decir, omita *propósito*, *resumen*...); en su lugar, afronte directamente el transcurso típico de los acontecimientos.

Caso de uso: **Tomar Dosis**

Estilo informal, extendido y esencial.

Evolución típica de los acontecimientos

Acciones del actor	Respuesta del sistema
1. El caso de uso comienza cuando el ratón se acerca, por primera vez en esa toma, a la entrada de la cabina de pesaje.	2. El sistema identifica al ratón y determina si es el inicio de una toma.
3. El ratón, con su peso, acciona la báscula de la cabina de pesaje.	4. El sistema recoge la medida del peso, acciona la cámara y registra la fotografía.
5. El ratón acciona la palanca de acceso al dispensador.	6. El sistema identifica al ratón y comprueba que es el individuo de la toma activa. Comprueba que está autorizado para recibir una dosis, consulta su composición, acciona la provisión en el dosificador, la pesa y abre la compuerta de entrada.
7. El ratón toma la dosis y sale del dispensador.	8. El sistema detecta la salida del ratón y cierra la compuerta, pesa la dosis restante y limpia el comedero del dosificador. Registra la fecha, la hora y la dosis administrada.
9. El ratón, con su peso, acciona la báscula de la cabina de pesaje, tras tomar la dosis.	10. El sistema recoge la medida del peso, tras haberse administrado la dosis.
11. El ratón se acerca a la entrada de la cabina de pesaje para salir.	12. El sistema identifica al ratón y determina si es el final de una toma. En ese caso, se registran permanentemente los datos de la toma en el historial del tratamiento.

Alternativas

- Paso 5 bis: Los sensores no detectan el sentido del movimiento. Podría ocurrir que, tras pesarse, el ratón volviera a salir. En ese caso, se desechan las medidas y termina el caso de uso.
- Paso 6 bis: Si, al accionar la palanca, el ratón no estuviera autorizado, tarde o temprano volvería a salir. Se prosigue como en el paso 5 bis.
- Pasos 3 y 9. El ratón puede accionar la báscula múltiples veces. El sistema recoge la última medida de peso y foto.
- Paso 9 bis. Con este planteamiento puede darse el caso de que entre otro ratón, en la cabina de pesaje, mientras el primero esté en el dispensador. La única forma de evitar que se falseen las mediciones es mediante otra compuerta (normalmente abierta), situada en la entrada de la cabina de pesaje. De este modo, tras el Paso 3, el sistema cierra esa compuerta. A partir de aquí, el sistema debe accionar su apertura en cualquier momento en el que se identifique al ratón de la toma activa en el sensor de la entrada de la cabina de pesaje. De igual forma, debe cerrarla en todas las situaciones del Paso 3.

La escritura del caso de uso tiene, como objetivo principal, comprender el funcionamiento del negocio. Es la condición fundamental para poder construir una aplicación que reproduzca ese funcionamiento. En este caso de uso, el software **debe registrar un conjunto de parámetros observables en un ratón**.

En principio, las reacciones del ratón son imprevisibles. No se puede suponer una intencionalidad ni un comportamiento coherente con ella: puede que entre en la cabina y salga 5 veces, antes de pulsar la palanca; o puede que no la pulse nunca o puede que no entre nunca... Por ello, para poder recoger la medidas que interesen, es necesario restringir los movimientos y, en alguna medida, condicionar el comportamiento del ratón. Parte del diseño del experimento consiste en definir las barreras físicas que limitan sus movimientos y situar los sensores para obtener medidas fiables en el instante más adecuado.

El comportamiento impredecible del ratón requiere adecuarlo a la naturaleza asíncrona de los sensores; por lo que, el comportamiento de la aplicación, debe estar controlada y dirigida por **eventos**. Un evento determinado tiene una significación diferente dependiendo de los eventos anteriores y de la situación en la que estaba antes el ratón. La lógica del comportamiento del negocio se convierte en la de una *máquina de estados finitos*.

Para fijar ideas, conviene hacer un cuadro con los distintos sensores del experimento, los eventos que producen y las lecturas que aportan:

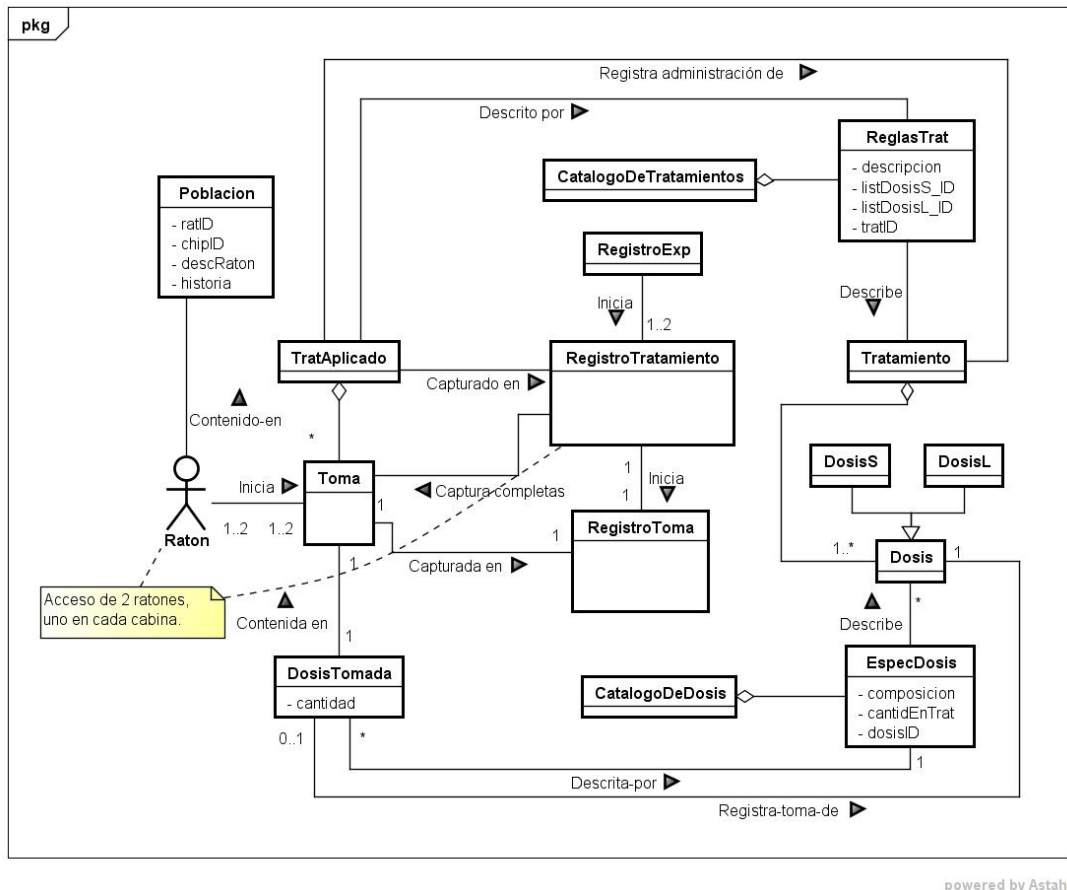
Nombre	Descripción	Evento	Parámetro
sensID1	Presencia en la entrada de la cabina de pesaje	ev1_ID	Raton_ID
bascula1	Peso del ratón en la cabina de pesaje	ev2_P	Peso
palanca	Dispositivo para abrir la puerta del dispensador	ev3_ON	Dosif_ID
sensID2	Presencia en la puerta del dispensador	ev4_ID	Raton_ID
bascula2	Peso de la dosis no tomada, en el plato del dosificador	ev5_Pdr	Peso

De igual forma, con los accionadores:

Nombre	Descripción	Comando	Devuelve
camara	Recoge la imagen del ratón.	disparo	Foto
puerta	Acceso al dispensador	abrir cerrar	
dosificador	Provee la dosis	dosis	
comedero	Recoge la dosis suministrada y la sobrante.	limpia	

Sección 2. Evaluación del *Modelado Conceptual*

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

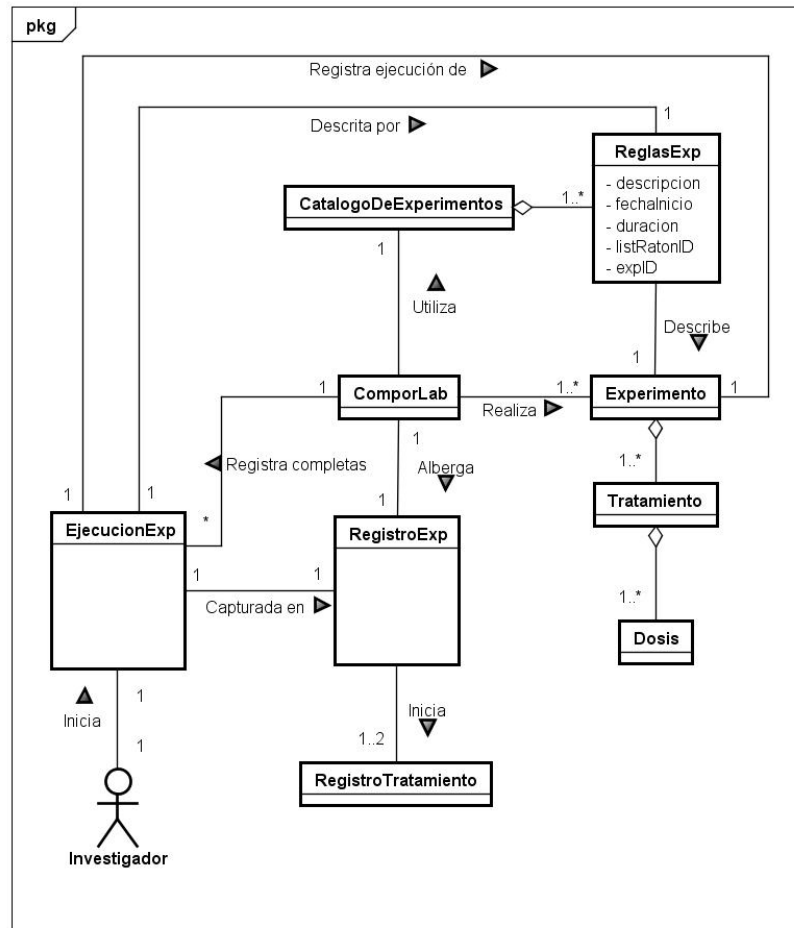


Es extremadamente importante comenzar a identificar ya qué información tiene cada elemento (objeto) porque, más adelante, al aplicar el principio de Experto, cada objeto sólo debe manejar la información contenida en él, la que conoce. A su vez, esto favorecerá enormemente el acoplamiento bajo.

A la hora de reflejar el comportamiento del caso de uso, el modelo, ya se ha tomado una gran cantidad de decisiones que afectarán a la construcción (diseño). Precisamente, cuando se afronta el diseño, habrá que refinar el modelo de dominio y, posiblemente, modificar algunas de esas decisiones.

Como desde el punto de vista del comportamiento es irrelevante si el estímulo proviene de un sensor o del ratón, se ha optado por ignorar la existencia de los sensores y actuadores. Sí se tendrán en cuenta en el diseño.

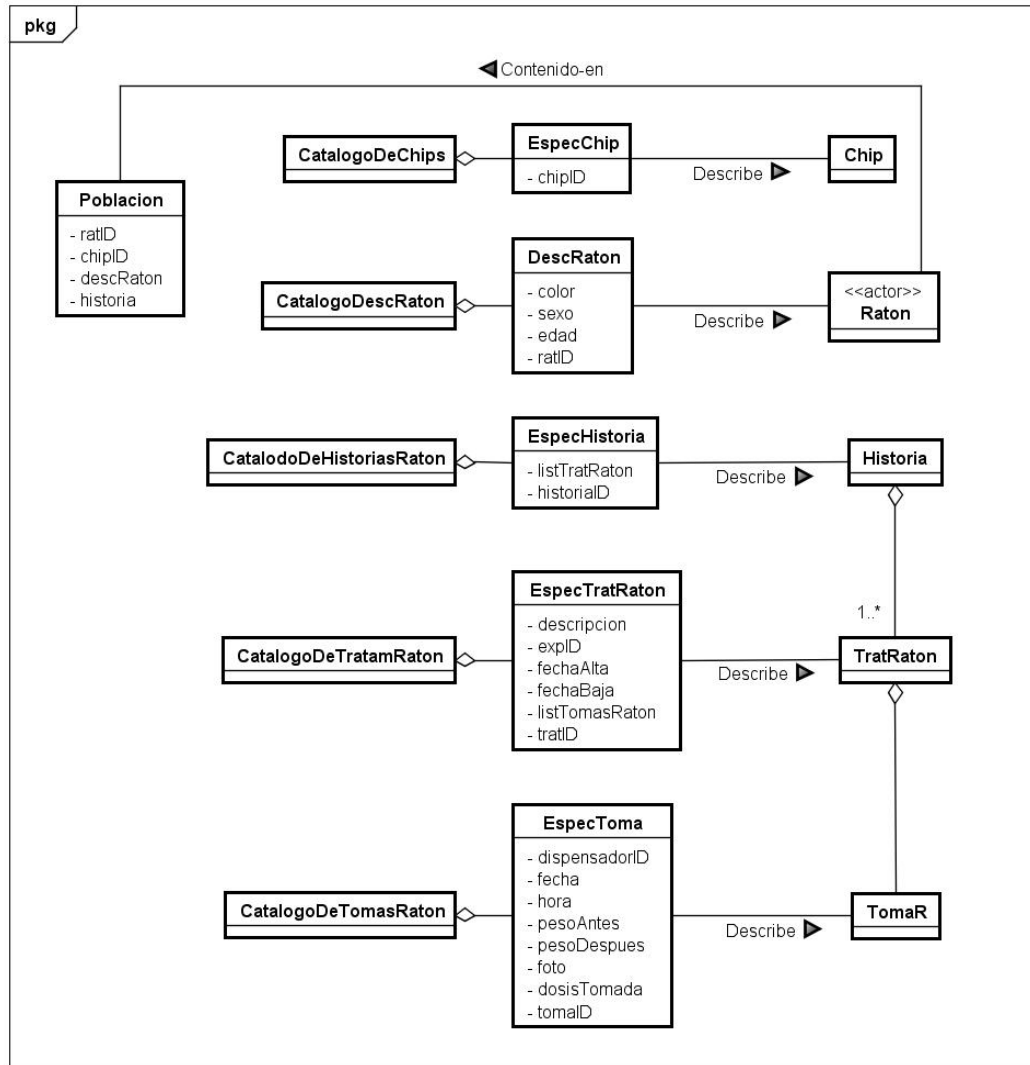
Básicamente, el esquema de PdV se repite, anidado, una y otra vez. Para observar esto, y ubicar modelo de dominio del caso de uso dentro del sistema CompOrLab, sería conveniente representar el modelo de dominio del sistema, en el nivel superior al del caso de uso TomarDosis.



powered by Astah

El paralelismo con PdV es obvio: ComporLab (Tienda) registra experimentos (Venta) descritos en un catálogo. Sólo puede ejecutarse un experimento en cada laboratorio (entorno) a la vez y lo inicia el investigador. A partir de ahí, cada registro 'dispara' al siguiente con la llegada de algún evento que signifique un cambio en la 'atención' o la operación que se va a realizar. Así, una vez que se ha definido qué experimento se está realizando, el RegistroExp crea un RegistroTratamiento, que se quedará a la escucha de que se acerque algún ratón para realizar una toma correspondiente al tratamiento activo que le esté asignado. En realidad crea dos registros, uno por cada tipo dosificador o cada salida del laberinto. Esto no significa más que se abren dos sesiones (registros), cada una pendiente de un juego de sensores, actuadores, etc. A partir de aquí se considerará un único RegistroTratamiento.

La decisión de qué tratamiento-dosis se administra al ratón proviene de su historial. Se vuelve a observar el esquema 'información etiquetada – catálogo' de PdV del subsistema Población:



powered by Astah

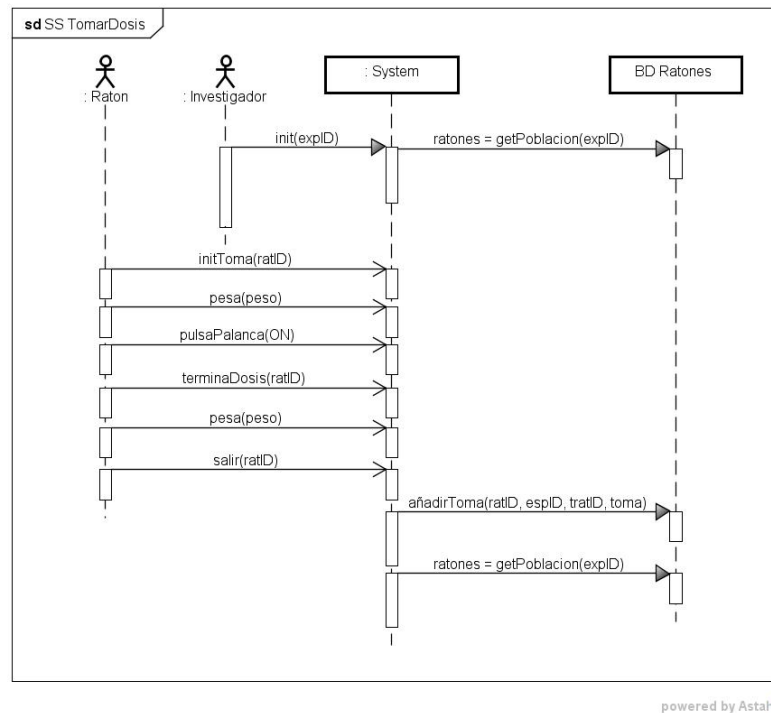
Sección 3. Evaluación de los **Eventos del Sistema** **Caso de Uso**

4. (0'5 puntos) Circunscrito al caso de uso anterior <<TomarDosis>>, construya un Diagrama de Secuencia del Sistema (DSS) **caso de uso** en UML. Represente los actores y los eventos del sistema. A partir de este DSS, especifique los contratos de dos operaciones principales: <<se omite la operación A>> y <<se omite la operación B>>.

ATENCIÓN: lo que hay entre corchetes <<se omite...>> es un ejemplo, usted lo debe sustituir por su propia operación.

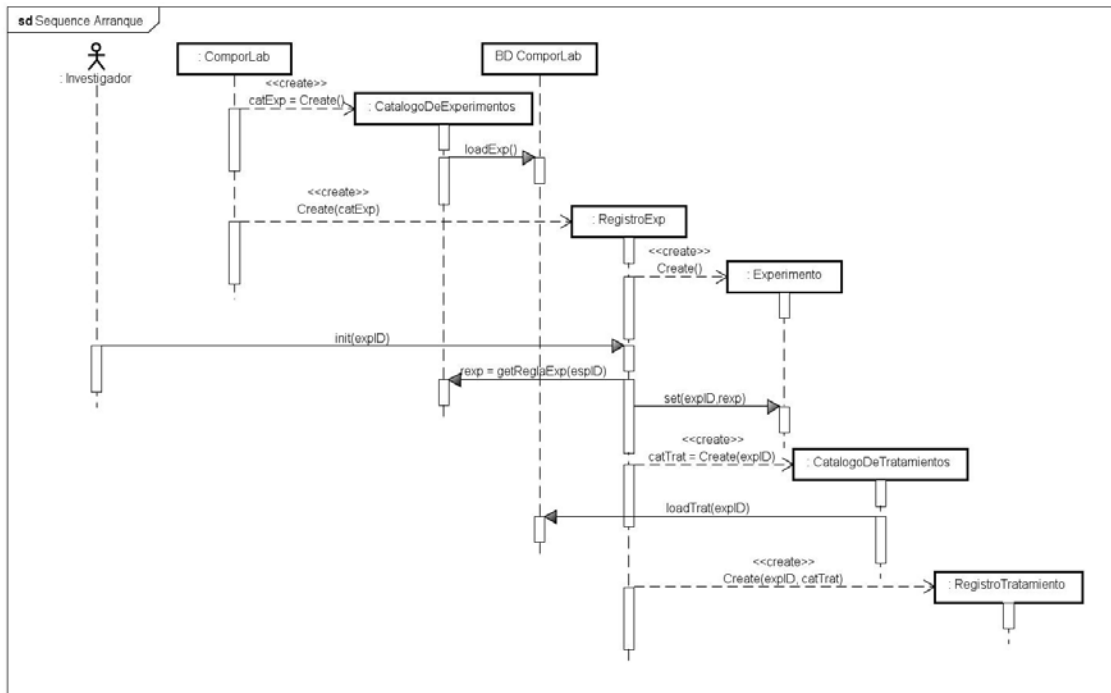
A pesar del enunciado (**en el examen persiste la misma errata**) la cuestión se refiere al diseño, a la migración de los objetos del MD hacia las clases software y a la secuencia de mensajes y acciones entre estas clases.

Obviando los sensores y actuadores, el DS **del Sistema** en relación al caso de uso TomarDosis podría ser el siguiente:



powered by Astah

Antes de desarrollar el diagrama de secuencia del caso de uso resulta conveniente, por claridad, plantear el de sus antecedentes: el arranque.



powered by Astah

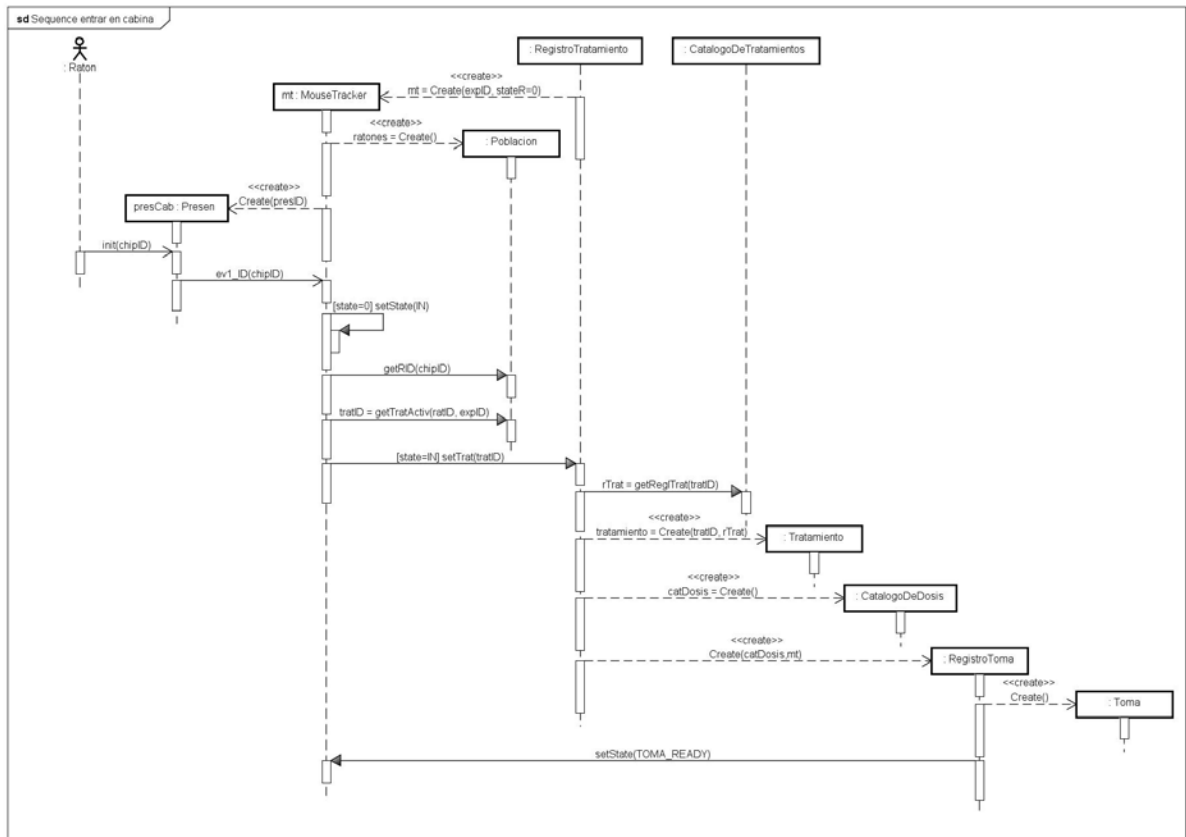
Nota: se ha abreviado el ciclo de carga de los catálogos devolviendo la instancia, directamente, en su creación.

Esta es la situación anterior al inicio de una toma; es decir, a que llegue un ratón y sea detectado en la puerta de la cabina. Por eso, y para controlar la situación del ratón (cambio de estado), se va a registrar el listener de los sensores en un controlador 'factory': MouseTracker. MouseTracker requiere manejar la información del ratón, por lo que tiene que acceder al subsistema Población (gestor experto en él). MouseTracker actúa de *monitor*, de *observer* de los sensores y los cambios de estado del ratón y de *register*, notificando a los objetos implicados.

Población va a manejar varios catálogos: CatalogoDeChips, CatalogoDescripRat, CatalogoDeHistorias, CatalogohistTratR y CatalogoTomasR. Con ellos, la funcionalidad de Población va a permitir obtener RatID y el TratID, activo para ese ratón, a partir del chipID. También permite consultar la autorización para una toma o añadir una toma terminada a la historia.

Volviendo al diagrama de secuencia del sistema, se han representado los estímulos que genera el ratón: initToma(chipID), pesa(peso), pulsaPalanca(ON), terminaDosis(ratID), pesa(peso) y salir(ratID). Se podrían renombrar y agrupar en 4 operaciones 'de peso': 'Entrar en cabina', 'Primera pesada y foto', 'Consumir dosis' y 'Salir'. A su vez, 'Consumir dosis' se podría descomponer en 'Abrir dosificador', 'Preparar dosis' y 'Tomar'.

El diagrama de secuencia de la operación 'Entrar en cabina' podría ser:

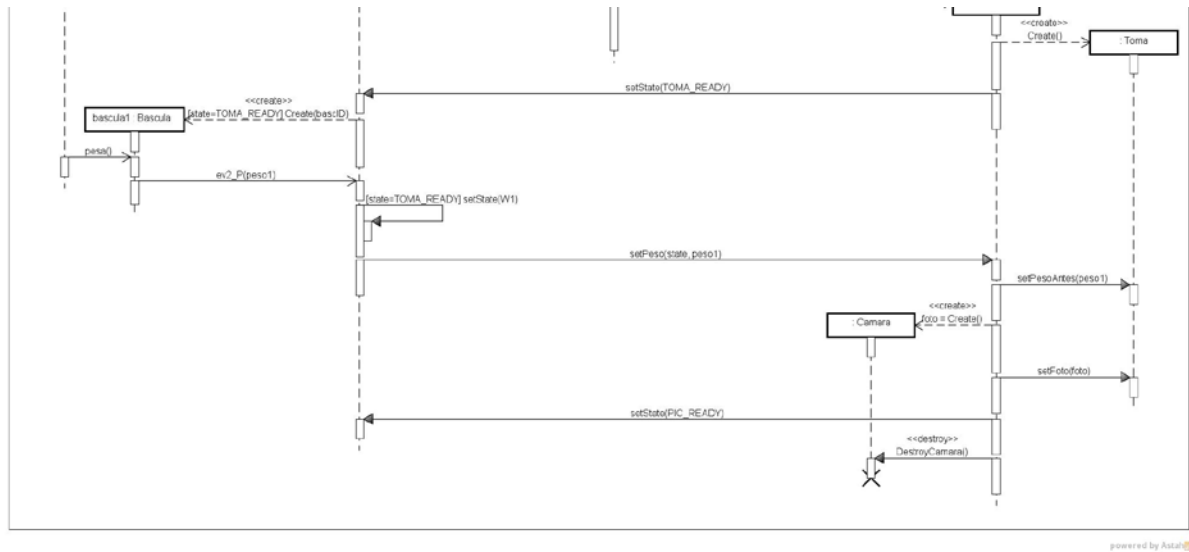


powered by Astah

Contrato CO1: **entrarEnCabina**

Operación:	init(chipID)
Referencias cruzadas:	Caso de Uso: Tomar Dosis.
Precondiciones:	<ul style="list-style-type: none"> Hay un experimento en curso, exp. Hay un RegistroTratamiento, rt, que maneja un CatalogoDeTratamientos, ct. Hay un MouseTracker, mt, que maneja Poblacion, ratones, y registra los eventos del sensor presCab. El estado del ratón, state, es 0.
Postcondiciones:	<ul style="list-style-type: none"> Se asoció presCab a un tipo de dosificador tipoDosis. Se creó una instancia de Tratamiento tratamiento. chipID se asoció a un ratón, a una DescRaton, a una Historia y a tratamiento, activo para el experimento, en base a la coincidencia del identificador en los catálogos de Poblacion, ratones, y el de tratID en ct. Se creó una instancia de CatalogoDeDosis catDosis. Se creó una instancia de RegistroToma regToma. Se creó una instancia de Toma toma. state pasó a valer IN, tras ser detectado; y TOMA_READY tras la creación de toma. Se creó una instancia del sensor Bascula bascula1, que notifica sus eventos a mt.

La siguiente operación, 'Primera pesada y foto', es más simple aún:

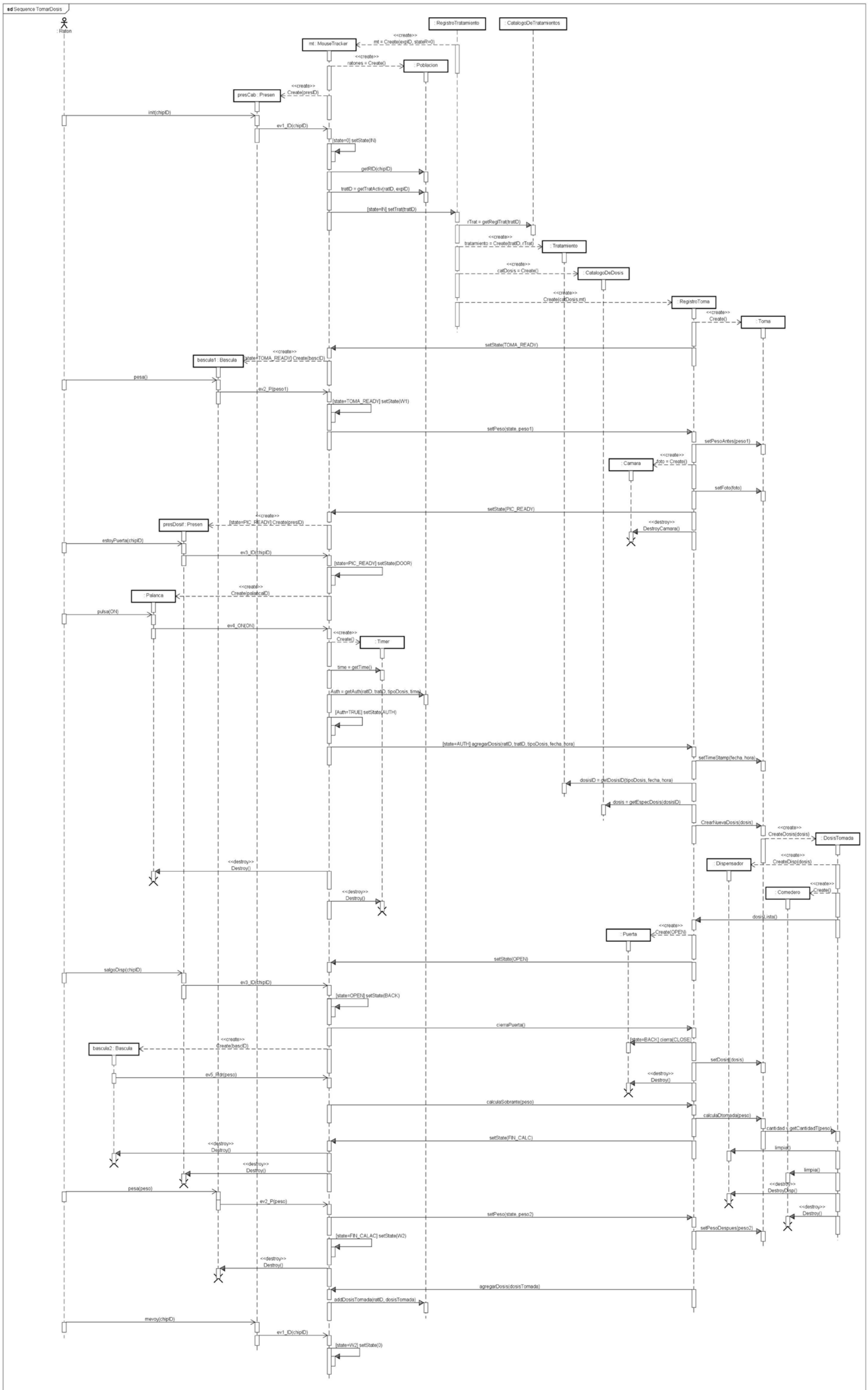


Contrato CO2: [peso1yFoto](#)

Operación:	pesa()
Referencias cruzadas:	Caso de Uso: Tomar Dosis.
Precondiciones:	<ul style="list-style-type: none"> – Hay un experimento en curso, exp. – Hay un tratamiento en curso, tratamiento. – Hay una instancia toma preparada para modificar sus atributos. – Hay una instancia del sensor Bascula, bascula1, preparada para notificar el evento ev2_P a mt y la lectura de la magnitud peso. – El estado del ratón, state, es TOMA_READY.
Postcondiciones:	<ul style="list-style-type: none"> – Se creó una instancia del actuador Camara. – El atributo pesoAntes, de toma, pasó a valer la magnitud peso, leída en el sensor bascula1. – El atributo foto, de toma, pasó a valer la imagen obtenida de la cámara. – Se creó una instancia de CatalogoDeDosis catDosis. – Se creó una instancia de RegistroToma regToma. – Se creó una instancia de Toma toma. – state pasó a valer W1 y PIC_READY tras haber modificado los respectivos atributos de toma. – Se eliminó la instancia de Camara.

Las otras dos operaciones se desarrollarán en la siguiente sección.

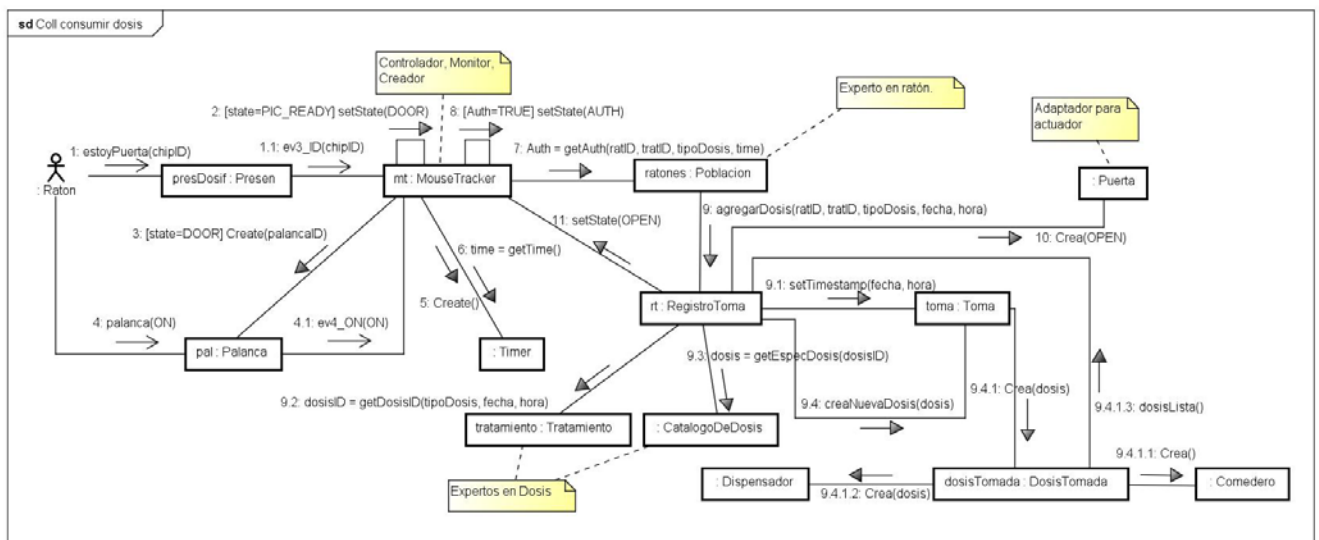
A continuación, se incluye el diagrama, completo y detallado, propuesto para todo el caso de uso. Para su legibilidad, se presenta en un formato A3.



Sección 4. Evaluación de la **Asignación de Responsabilidades** y **Diseño de Colaboraciones**

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

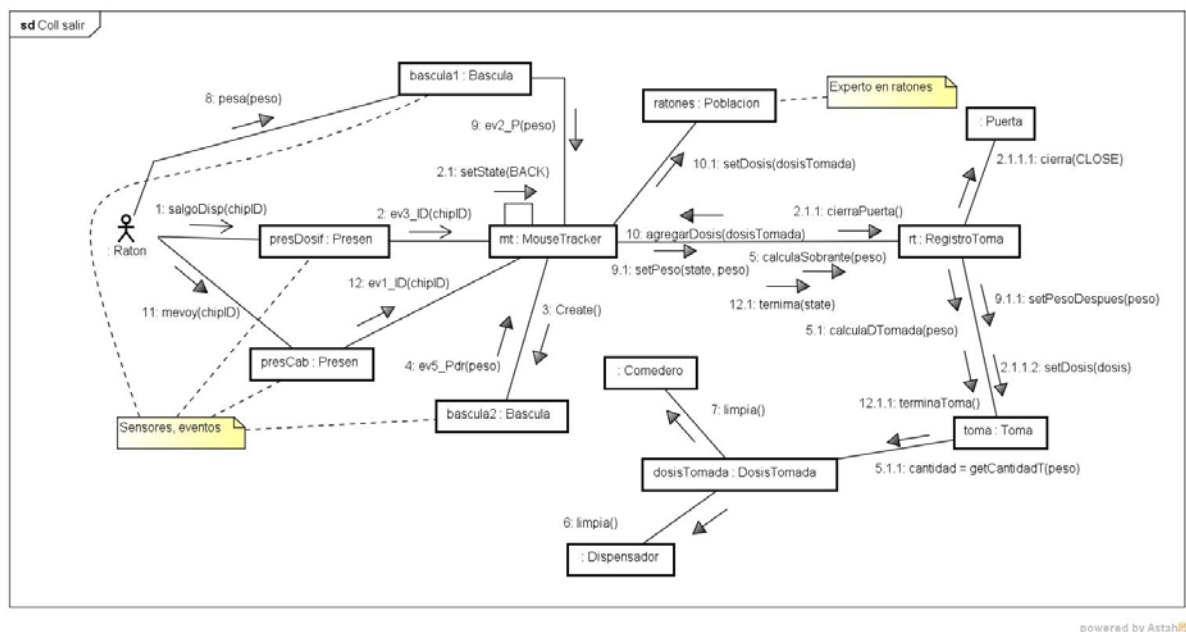
ATENCIÓN: lo que hay entre corchetes <<se omite...>> es un ejemplo, usted lo debe sustituir por su propia operación.



Operación consumirDosis.

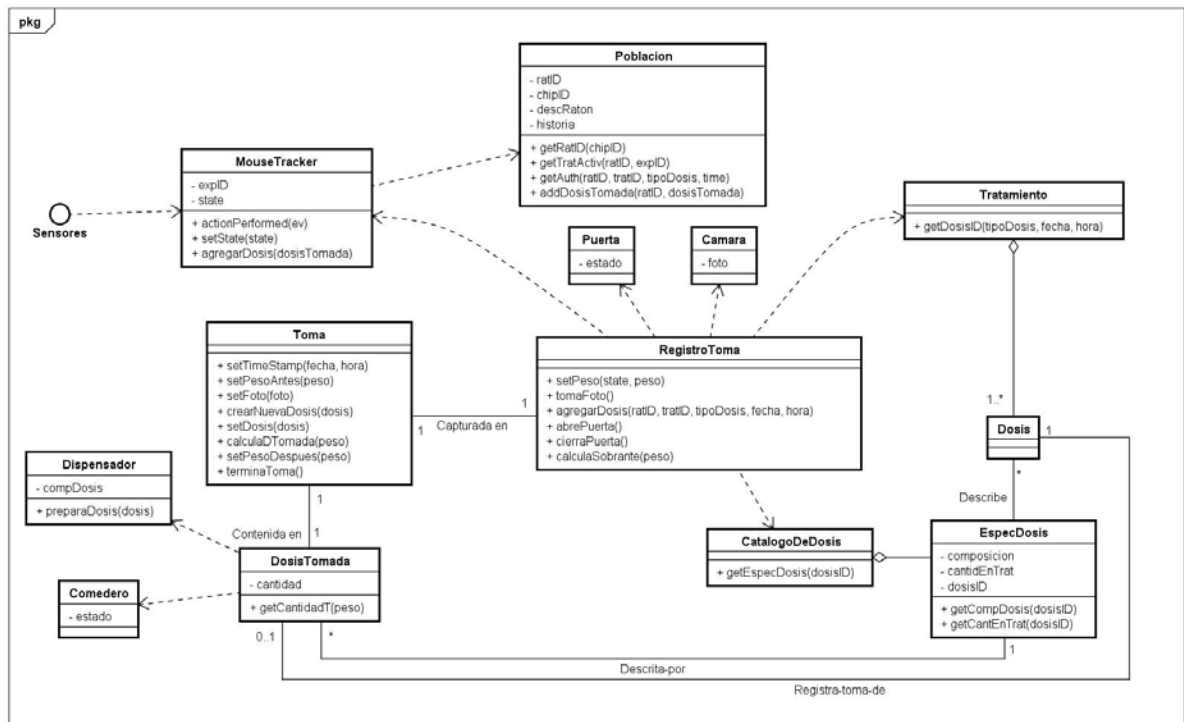
6. (2'5 puntos) A partir del contrato de la operación <<se omite la operación B>> que haya indicado en el punto 4, complete el diagrama de colaboración en UML. Consigne cada mensaje con los patrones GRASP (Experto, Creador, etc.) o cualquier otro que lo justifique. Si añade responsabilidades no explicitadas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

ATENCIÓN: lo que hay entre corchetes <<se omite...>> es un ejemplo, usted lo debe sustituir por su propia operación.



Operación CO4 Salir.

7. (0'5 puntos) Elabore un diagrama de clases parcial para la clase *RegistroToma* y las clases más importantes que intervengan en el caso de uso desarrollado. Represente los nombres de todos sus atributos, asociaciones (con la navegabilidad) y métodos.



powered by Astah

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 *RegistroToma*. Incluya las definiciones de todas las variables que la componen (miembros), pero escriba solamente la definición completa del cuerpo para el método (función miembro): *<<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 corchetes <<se omite...>> es un ejemplo, usted lo debe sustituir por **su** propio método.