

La elaboración de este ejercicio está dirigida a través de una secuencia de **9 preguntas**, agrupadas en **6 secciones**, en las que se solicita realizar determinadas operaciones y tareas de diseño. **La puntuación** es sobre un **total de 11 puntos** (10 más 1 punto de la sección BP). **No todas** las cuestiones **puntúan igual**. Puede utilizar la extensión que necesite para el documento, pero conteste a las preguntas de cada sección en páginas diferentes. Por favor, lea TODO el ejercicio, hasta el final.

**Se permite el uso INDIVIDUAL de cualquier material.**

**¡ATENCIÓN!:** este examen consiste en el diseño de un caso de uso concreto que se define en la pregunta número 2. Lo que se presenta inmediatamente es el planteamiento del escenario de dicho caso de uso, por lo que se recomienda leerlo rápida y *transversalmente* y acudir a la definición del caso de uso, en dicha pregunta 2, para contestar el resto de las preguntas del examen.

### **Planteamiento del escenario de estudio.**

El escenario en el que se situará el funcionamiento del caso de uso (pregunta 2), consiste en un *paquete* para la **gestión sanitaria de una explotación ganadera**. (SanGranja).

Dicho módulo (o *paquete*) forma parte de una aplicación dedicada a la gestión integral ganadera (i-FarM) de la que, para situar este escenario, puede encontrar información ampliada en este [otro documento](#).

El negocio general del usuario de estas aplicaciones es el cuidado y la cría de animales para el consumo humano de sus productos.

Esta descripción del escenario, en el que se ubica el caso de uso que se va a implementar en el ejercicio, se refiere al módulo SanGranja. Lógicamente, interactúa con algunos de los servicios que provee i-FarM (considerado externo a SanGranja) pero, también, con otros servicios aportados por sistemas de apoyo externos (como el que llamaremos Sistema de Información, el sistema de gestión del almacén común con toda la información –Datos– del sistema integral y que, por tanto, también daría soporte a i-FarM).

De esta forma, el propósito fundamental del módulo SanGranja es ocuparse de:

- La gestión Sanitaria de los animales y de las instalaciones. Es decir, de la monitorización, diagnóstico, tratamiento y seguimiento de la salud de cada animal.

El ciclo vital de cada individuo suele transcurrir en unas instalaciones o recintos (Corrales) en los que se supervisan sus condiciones ambientales, la alimentación especializada o los tratamientos sanitarios que recibe. Para realizar esta funcionalidad, se va a suponer que cada animal lleva implantado un dispositivo biométrico que recoge la información necesaria. Estos datos se mantienen en el Sistema de Información global de la explotación, en la parte que le corresponde tanto a i-FarM como a SanGranja,.

De la misma manera que ocurre con la aplicación general i-FarM, **el objetivo fundamental de la construcción del módulo SanGranja es que**, además de funcionar colaborativa e independientemente de i-FarM, **sea fácilmente escalable y adaptable a la dimensión de la explotación, a la estructura organizativa con que se realiza, a la naturaleza de su ganadería, a las enfermedades que les afectan y a sus tratamientos.**

Con estas prerrogativas, el funcionamiento de la Gestión Sanitaria del negocio (SanGranja) incluye:

- Un comportamiento automatizado, online y *desatendido* (aunque puede monitorizarse bajo demanda del usuario), que recoge los valores biométricos del ganado, los procesa a través de un Sistema Experto de Salud auxiliar (externo) y registra los resultados en el Sistema de Información global (también externo). Además de esta monitorización, se encarga de gestionar un sistema de alarmas de la salud del ganado.
- El mantenimiento de los expedientes sanitarios de los individuos de la Cabaña. Es decir, el mantenimiento de la información sanitaria sobre el estado de los animales, que se ubica en el Sistema de Información global externo.
- El mantenimiento de la Caracterización de Enfermedades (síntomas, vectores de transmisión, tratamientos, etc.) que se gestionan en el módulo y cuya información también reside en el Sistema de Información global externo.
- Gestión Sanitaria de la Estabulación. Es decir, el mantenimiento de la información sobre la ubicación de los animales, en un Corral u otro, que, por motivos sanitarios, varía respecto a la planificación normal de la explotación (aislamiento, etc.). Como se ha indicado, el ciclo vital de los animales transcurre, principalmente, en recintos específicos de las instalaciones de la granja (Corrales, establos, chiqueros, jaulas, etc.). La gestión de la producción (implementado en algún módulo de i-FarM) se encarga de planificar la estancia de cada animal en un Corral determinado y de programar sus Traslados de uno a otro. El registro de este seguimiento también forma parte de la información de cada animal, que se mantiene (por el módulo que corresponda) en el Sistema de Información global externo.
- El Análisis de los datos sobre la situación y la evolución sanitarias del ganado. En este objetivo se incluye la realización de simulaciones sobre la evolución de una enfermedad, una de las opciones que constituirá el caso de uso que se desarrolla en este ejercicio.
- El mantenimiento de los parámetros de configuración y funcionamiento del módulo. Obviamente, para que esto sea posible es imprescindible que todos los componentes implementados sean funcionalmente independientes, adaptables y comprensibles.

Detalles y simplificaciones admitidas:

- Lo más importante: Para delimitar sustancialmente el ámbito del código que hay que construir, este estudio se circunscribe a la implementación de la funcionalidad estricta del escenario descrito (SanGranja), y no a cómo se realiza la interacción ni con los Actores principales (la Capa de Presentación, IU) ni con los servicios que proveen los sistemas de apoyo externos (Capa de Servicios y de Acceso a Datos Externos). Con esta limitación, además de *centrar* la atención sobre el objetivo del desarrollo, se obliga a que su implementación se independice de cómo se concreten esos elementos externos (los de la IU o los de esos sistemas de apoyo externos, incluido el almacén externo en el que se gestionan todos los datos que SanGranja pueda necesitar: el Sistema de Información global externo).

Esto es aplicable en todos los niveles de granularidad y, por consiguiente, tanto en la evaluación de los casos de uso primarios del módulo SanGranja, como en todos los pasos de la implementación del caso de uso definido en la pregunta 2.

- De esta forma, las únicas tareas en la elaboración de la solución del ejercicio, a partir de la pregunta 2, están destinadas a:
  - Comprender los requisitos funcionales del caso de uso especificado en esa pregunta. Es decir, a definir esa funcionalidad, qué datos maneja y cómo se deben utilizar.
  - Especificar ese funcionamiento a través de su implementación con código (diseño), pero de forma que ese código sea funcionalmente independiente, adaptable y comprensible.

Es decir, no hay que preocuparse de cómo se presentan los resultados ni de cómo se guardan los datos o cómo se obtienen, sino de qué datos se necesitan para el funcionamiento del caso de uso y cómo deben organizarse para que el código los utilice de manera desacoplada, flexible y comprensible. Los datos necesarios para el caso de uso están en el Sistema de Información global externo y los mecanismos para su obtención no forman parte de la funcionalidad que se estudia en este ejercicio, pero sí en qué estructuras se organizan para ser utilizados de la manera descrita.

- De la misma manera, se asume que tanto i -FarM como SanGranja tienen todos los dispositivos e instalaciones necesarios, así como tienen implementados todos los mecanismos que se precisan para su manejo, de forma que el correcto funcionamiento del caso de uso sea posible por sí mismo.
- En cada pregunta se ampliarán los detalles que se precisen para elaborar la respuesta que se pide en ella.

## Enunciado de las preguntas.

Sección 1. Evaluación de los **Casos de Uso**. (Fase de Inicio). Se trata de ubicar y caracterizar, globalmente, el sistema software que va a desarrollarse: cuáles son sus funciones principales (casos de uso primarios), qué Actor o Actores las dirigen y con qué sistemas o actores de apoyo externos necesitan interactuar para cumplir con su objetivo.

1. (0'5 puntos) Represente, en un diagrama UML de casos de uso, los casos de uso primarios (Elementary Business Process) más importantes, sus actores principales, los de apoyo y las interacciones correspondientes para el módulo SanGranja.

**Se recuerda:** de aquí en adelante, todas las preguntas se refieren al caso de uso especificado en la siguiente pregunta. El objetivo es que realice el diseño para que también admita las otras funcionalidades y opciones definidas para este módulo SanGranja.

2. (1 punto) Con la siguiente descripción del **caso de uso** <<*SimularPropagaciónEnfermedad\_X*>>, **escríbalo en un formato completo** (se recomienda la variante '*en dos columnas*') **y un estilo esencial** (excluyendo los detalles técnicos de nivel bajo). Incluya tanto el flujo en el escenario principal de éxito como 2 extensiones o flujos alternativos que pudieran ser frecuentes:

El escenario supuesto que justifica este caso de uso es que el Sistema de Seguimiento del Estado Sanitario de los animales ha detectado una patología, contagiosa y mortal, en un determinado número de individuos de uno de los Corrales. Aunque la enfermedad está caracterizada en el sistema (Vector de Contagio, etc.), no se dispone ni de tratamiento farmacológico ni de vacuna para paliar los efectos de su proliferación. Por este motivo se hace necesaria otra estrategia que actúe sobre la movilidad de los animales entre Corrales (o su aislamiento, según la capacidad de las instalaciones de la granja). Para ello, dentro de la funcionalidad incluida en el Análisis de los datos, se requiere una opción que permita realizar la simulación de cómo prolifera el contagio entre los animales para una topología concreta de su distribución en los Corrales, para una planificación prefijada de su traslado entre ellos y para las reglas de transmisión del contagio.

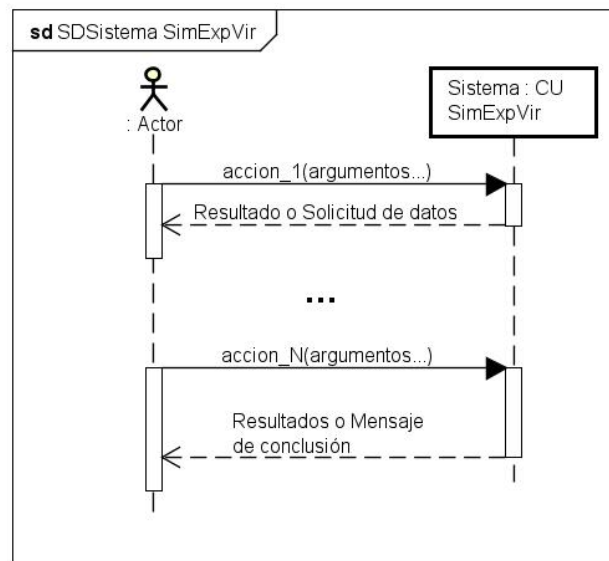
En definitiva, **lo que hace** el caso de uso es calcular el número previsto de animales infectados, en cada Corral que haya seleccionado el Usuario y en cada día sucesivo por el que transcurre la simulación. El **objetivo o resultado** del caso de uso es una relación bidimensional del número de contagios: en cada Corral seleccionado para su observación y en cada día de la simulación.

Se va a considerar que al seleccionar la opción de este tipo específico de simulación (con la caracterización del Vector de Contagio y con las reglas para el cálculo de su expansión) ya se ha determinado para qué enfermedad del catálogo se va a realizar, por lo que el caso de uso se inicia cuando el Usuario establece en qué Corrales desea hacer la observación (**un subconjunto de los que utiliza la explotación**) y durante cuántos días. Finaliza cuando el sistema presenta los resultados mencionados (el número de animales contagiados en cada Corral seleccionado y, todo ello, en cada día de la simulación).

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.

### Simplificaciones y sugerencias:

- Recuerde que en esta actividad se describe la secuencia de la interacción entre el actor principal (con un objetivo de uso concreto: obtener esos resultados) y la respuesta del sistema, el cual se considera una '*caja negra*'. Es decir, el actor no tiene ningún conocimiento de los elementos de software que hacen funcionar al sistema, ni siquiera de cuáles son internos o externos a la aplicación (base de datos, etc.).
- Para el desarrollo posterior (el diseño dinámico detallado) y para *aclearar* la comprensión respecto a la secuencia de operaciones planteada en la respuesta a esta pregunta, se recomienda elaborar un Diagrama de Secuencia del Sistema en el que sólo intervienen el Actor y el Sistema que, precisamente, se representa como una '*caja negra*', como un único objeto:



Nota: Este diagrama ni es lo que se pide en el enunciado de ninguna de las preguntas, ni puntúa.

Sección 2. Evaluación del **Modelado Conceptual**. (Fase de Elaboración. Comprensión de los requisitos funcionales y Diseño Lógico del funcionamiento mediante el Modelo de Dominio).

3. (3 puntos) En relación con el caso de uso anterior, <<SimularPropagaciónEnfermedad\_X>>, 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.

La descripción del comportamiento 'de uso' que se requiere para el caso de uso es la siguiente:

La caracterización de la expansión del contagio de esta enfermedad (Vector de Contagio) viene definida por los parámetros:

- E: promedio diario de encuentros, contactos o situaciones de riesgo proclives a que un individuo infectado pueda transmitir la enfermedad a uno sano.
- pr: probabilidad de que en una situación 'E', de riesgo de contagio, éste se produzca. Representa la eficacia en la transmisión.

En este caso, si tenemos una población P de animales, en un Corral concreto, de la que un número de individuos N están infectados, la proporción de sanos será:  $\left(1 - \frac{N}{P}\right)$ . Así, si ese número de infectados,  $N_d$ , corresponde a un día 'd', el incremento de contagios en el siguiente día (d+1) será:  $\Delta N_{d+1} = pr \times E \times \left(1 - \frac{N_d}{P}\right) \times N_d$ .

El cálculo anterior se corresponde con una situación en la que la población P de un Corral 'i' se mantiene invariable. Sin embargo, la explotación de la granja establece una programación de traslados de los individuos de un Corral 'j' a otro 'i'. Por consiguiente, el cálculo debe añadir los contagios importados, producidos por los visitantes infectados que han sido trasladados desde otro Corral, a los contagios locales (los de la fórmula anterior).

Vamos a llamar  $traslados_{origen\ j,\ destino\ i}$  al número de reses que, en un día determinado (d+1), se ha programado trasladar desde el Corral de origen (j) al Corral i. De ellos, el número de infectados dependerá de cómo se ha extendido la enfermedad en el día previo (d) y en el Corral de origen (j). Es decir:  $infectadosExt_{d+1,destino\ i,de\ origen\ j} = trasladados_{d+1,origen\ j,con\ destino\ i} \times \frac{N_{d,origen\ j}}{P_{origen\ j}}$ .

Para simplificar notablemente el cálculo (y los índices) se va a asumir que la información disponible es un número entero correspondiente al promedio, en un número extenso de días (mayor que el de la simulación), de los animales que se trasladan de un sitio a otro.

En ese caso, haciendo el cálculo para todos los Corrales j que tienen programado trasladar animales al Corral i:

$$infectadosExt_{d+1,i} = \sum_{j=1}^{j=n\ recintos\ totales, j \neq i} trasladados_{origen\ j,con\ destino\ i} \times \frac{N_{d,origen\ j}}{P_{origen\ j}}$$

Con esto, el cálculo del número total de animales contagiados, en un Corral  $i$  cualquiera y en un día  $d+1$ , se realiza de esta forma:

$$N_{d+1,i} = N_{d,i} + pr \times E \times \left(1 - \frac{N_{d,i}}{P_i}\right) \times \left(N_{d,i} + \sum_{j=1}^{j=n, j \neq i} \text{traslados}_{\text{origen } j, \text{con destino } i} \times \frac{N_{d,j}}{P_j}\right)$$

*Ecuación 1. Cálculo del número total de infectados en un día y en un corral determinado.*

Realizando estos cálculos para cada Corral que se quiera observar, se desea obtener:

- El número total de contagiados cada día de la simulación.
- El correspondiente porcentaje de contagiados respecto a su población residente P.

Como referencia comparativa, para cada día, también se quieren esos mismos valores anteriores, pero referidos al total de animales de los Corrales observados.

Por ejemplo, si se selecciona hacer la simulación en 3 de los 4 Corrales que tiene una explotación, el aspecto visual de los resultados deseados podría ser el correspondiente a la tabla sombreada en azul:

	Corral 1		Corral 2		Corral 4		Total	
Población	300		8		75		383	
Traslados / día V	7		0		1		8	
Población residente	296		9		76		381	
Contagiados inicial	2		0		0		2	
Visitantes del Corral 1			1		1		2	
Visitantes del Corral 2	0				0		0	
Visitantes del Corral 3	2		0		1		3	
Visitantes del Corral 4	1		0				1	
	Contagiados	%	Contagiados	%	Contagiados	%	Contagiados	%
Día 0	2	0,67568%	0	0,00000%	0	0,00000%	2	0,524934%
Día 1	9	3,04054%	0	0,00000%	0	0,00000%	9	2,362205%
Día 2	43	14,52703%	0	0,00000%	0	0,00000%	43	11,286089%
Día 3	190	64,18919%	0	0,00000%	0	0,00000%	190	49,868766%
Día 4	296	100,00000%	2	22,22222%	2	2,63158%	300	78,740157%
Día 5	296	100,00000%	9	100,00000%	12	15,78947%	317	83,202100%
Día 6	296	100,00000%	9	100,00000%	57	75,00000%	362	95,013123%
Día 7	296	100,00000%	9	100,00000%	76	100,00000%	381	100,000000%
Día 8	296	100,00000%	9	100,00000%	76	100,00000%	381	100,000000%
Día 9	296	100,00000%	9	100,00000%	76	100,00000%	381	100,000000%
Día 10	296	100,00000%	9	100,00000%	76	100,00000%	381	100,000000%



En conclusión:

El funcionamiento consiste en realizar unos cálculos (ecuaciones o reglas de transmisión de la enfermedad) cuyos parámetros provienen de:

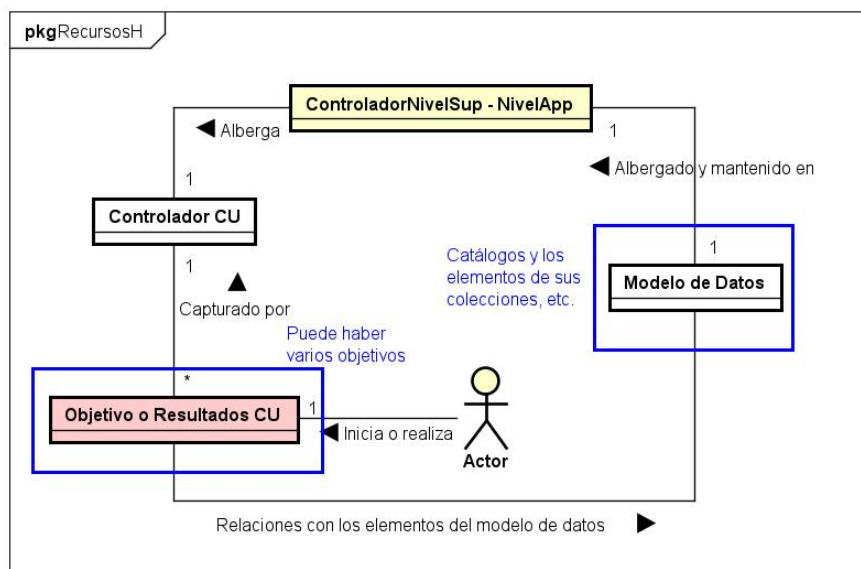
- La caracterización del contagio de la enfermedad.
- La información de cada Corral. Sin embargo, al observar la Ecuación 1 se concluye que parte de esos parámetros se corresponden con la información estática de ese Corral (la que se mantiene en el Sistema de Información global externo); otra parte, con el mismo tipo de información pero de los otros Corrales; y, por último, otros datos se refieren a los resultados de la simulación, en un día anterior (es decir, los que **no están contenidos** en la información de cada Corral que mantiene el sistema), correspondientes a todos los Corrales de la explotación.

De lo anterior también se concluye que, aunque el resultado se limite a un subconjunto de los Corrales de la explotación, la vinculación de los datos que impone la Ecuación 1 obliga a hacer los cálculos con todos ellos (al menos, con los implicados en la programación de traslados).

El resultado de este funcionamiento (es decir, el objetivo deseado en este caso de uso) es un listado del número de contagiados calculado, que se debería poder visualizar como una tabla, un gráfico, etc. En concreto, esos resultados se obtienen de una sucesión de cálculos, para cada día de la simulación, en los que cada uno, a su vez, contiene el número total de contagiados, en cada Corral seleccionado, más el del total de la población correspondiente a esos mismos Corrales.

Simplificaciones y sugerencias:

- En este diagrama, la descripción del funcionamiento lógico se elabora representando al actor que maneja el caso de uso y los objetos estrictamente conceptuales (no los de tipo software) a los que se les va a asignar la responsabilidad de las operaciones *lógicas* que constituyen dicho funcionamiento. Como guía para '*aclarar ideas*', se recomienda utilizar esta plantilla:



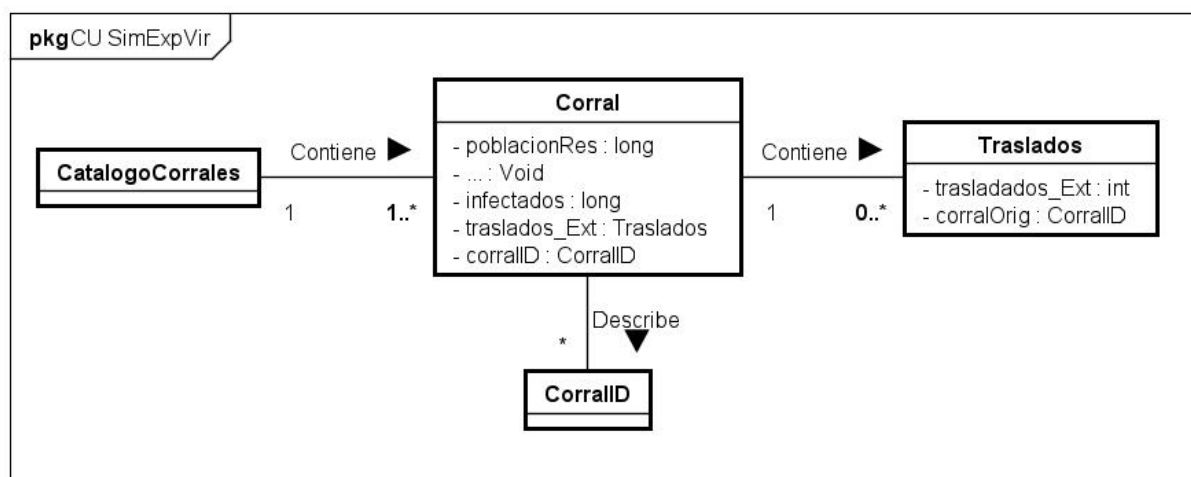
- Para simplificar el funcionamiento, los cálculos de la Ecuación 1 y tanto el manejo de los datos contenidos en las estructuras del 'Modelo de Datos' (los datos que se mantienen, desde el 'ControladorNivelSup', en el Sistema de Información global externo), como su representación, se sugieren algunas simplificaciones que se refieren al contenido de cada elemento Corral de la colección correspondiente a las instalaciones de la explotación:
  - En la simulación, los cálculos en cada Corral comienzan con un valor inicial de animales contagiados en '*un día*'. Dicho valor puede corresponder al último conocido (*real*, no *calculado*), que recopila la parte del Sistema de Monitorización y Seguimiento Sanitario y que proporciona al Sistema de Información global externo, del cual se obtiene al construir ese elemento Corral.
  - La programación de animales trasladados de un Corral a otro depende de la situación específica en que se encuentre cada uno de sus correspondientes *ciclos vitales*. Es decir, depende del día que se considere, lo que imposibilita el uso sencillo de esa información en *cualquier día* de la simulación. La simplificación consiste en considerar que, de alguna manera, el ControladorNivelSup (el que maneja el flujo de trabajo de todo el módulo SanGranja y el único que tiene acceso al Sistema de Información global externo) se *las apaña* para consultar la programación de esos traslados y, para cada Corral, añade la información correspondiente al cálculo del promedio, en un período amplio de días, del número de animales que se trasladan desde cualquier otro Corral. De esta manera, estos datos se independizan del día en el que se realizan los cálculos de la simulación. Estas operaciones se hacen fuera del caso de uso (con anterioridad a él), pero sus consecuencias quedan reflejadas en la estructura de cada elemento Corral, en la que se agrega una colección de elementos con el número de animales trasladados a ese Corral y el identificador del Corral de procedencia.
  - Al haber traslados desde otros Corrales, la Población residente en cada uno se ve modificada respecto a la *nominal*. Dado que se ha *forzado* a que los animales trasladados no dependan del día para el que se realizan los cálculos de la simulación, también se puede calcular, *a priori*, el valor de la ocupación *constante* de cada Corral como un balance de movimientos:

$$\begin{aligned}
 poblacionRes_i &= Población_i - \sum \text{traslados a otros Corrales} \\
 &+ \sum \text{visitantes de otros Corrales}
 \end{aligned}$$

Como se ha indicado, este cálculo también se puede realizar fuera del caso de uso y, para mejorar los resultados de la simulación, se recomienda que se utilicen estos valores en la Ecuación 1, en lugar de los correspondientes  $P_i$ .

Todas estas sugerencias están destinadas a conseguir que las estructuras de datos de los elementos que componen el 'Modelo de Datos' **se mantengan desacopladas**. Aunque en el Modelo de Dominio se utilicen estos *objetos conceptuales*, la lógica del funcionamiento del caso de uso está definida aquí, precisamente, por el contenido y la estructura de esos *objetos conceptuales*. Por tanto, para que esté desacoplado el funcionamiento del código (en el diseño dinámico detallado), es primordial que los *objetos conceptuales* del Modelo de Dominio (su estructura *conceptual*) también lo estén.

En definitiva, con las explicaciones anteriores, la propuesta que se hace para los elementos Corral (componente, entre otros, del mencionado Modelo de Datos) podría tener estas características (**¡¡Ojo!!** No está completa y sólo se incluyen los elementos –atributos– descritos más arriba):



Sección 3. Evaluación de la **Asignación de Responsabilidades y Diseño de la Interacción**.  
(Fase de Elaboración. Diseño Dinámico Detallado del caso de uso).

El Diseño Dinámico Detallado consiste en especificar cómo se realiza cada colaboración, cada interacción que se produce en el desarrollo del '*flujo básico de éxito*' del caso de uso y entre cada elemento de código que interviene en dicho funcionamiento.

Por tanto, hay que representar cada mensaje que el Actor envía al Controlador del caso de uso, y los que se intercambian entre éste y cualquier otra instancia que intervenga, así como los mensajes que transitan entre ellas, con los correspondientes argumentos de cada mensaje, los objetos que devuelvan, los condicionantes para que se produzcan ('*guardas*'), iteraciones, etc. Es idéntico a definir el código. En cualquier caso, cada objeto representado en los diagramas es una instancia, por lo que debe estar definida por su nombre y la clase a la que pertenece (su tipo).

En las respuestas a las 2 preguntas siguientes es donde se elabora la implementación de este funcionamiento del código, de este diseño dinámico detallado. Se pueden hacer en el orden que se prefiera, porque una respuesta se apoya en la otra y se complementa (por lo que deben ser absolutamente coherentes), pero siguiendo las pautas que indican sus enunciados.

4. (3 puntos) Circunscrito al caso de uso que nos ocupa, <<*SimularPropagaciónEnfermedad\_X*>>, construya un **Diagrama de Interacción** en UML. Represente el actor, sus eventos y el paso de mensajes entre cada instancia de las clases software que componen el sistema para este caso de uso.

**¡ATENCIÓN!:** se pide un Diagrama de Interacción porque se deja al estudiante la elección de responder con un **Diagrama de Secuencia Detallado** (DS) o con los **Diagramas de Colaboración**, correspondientes a cada evento que el Actor envíe al Controlador y las sucesivas acciones que se producen tanto en él como en las otras instancias del código. Son representaciones excluyentes (o utiliza el DS o utiliza los Diagramas de Colaboración), totalmente equivalentes, y **sólo se va a calificar una de ellas** (no lo que se pueda deducir de la mezcla de las evidencias encontradas entre una y otra).

En ambos casos, conviene agrupar la secuencia de acciones que se producen en el '*flujo básico de éxito*' del caso de uso (por ejemplo, mediante el diagrama de secuencia del sistema –DSS— que se propuso elaborar al final de la pregunta 2) en torno a las *operaciones principales* de su funcionamiento: los hitos importantes que, de alguna manera, describen y caracterizan ese grupo de acciones.

Estas operaciones principales son las que se utilizan en la representación con los Diagramas de Colaboración: en cada diagrama se representa todo el funcionamiento que corresponde a una operación principal.

En el funcionamiento del caso de uso, vamos a denominar '*ParamSim*' a la operación que contiene los eventos del Actor y las acciones *internas* correspondientes tanto a la inicialización del caso de uso (creación e inicialización de las instancias que se requieran en ese momento) como a la asignación de los valores requeridos para realizar los cálculos (que serían la selección de los Corrales y la delimitación del número de días para los cálculos).

De igual forma, '*RealizarSim(días)*' es la operación que contiene las acciones para realizar los cálculos subsiguientes hasta generar el resultado. Por motivos de simplicidad, se ha *traspasado* la asignación de la extensión de la simulación, última perteneciente a la operación '*ParamSim*', para que sirva de '*disparo*' en esta otra.

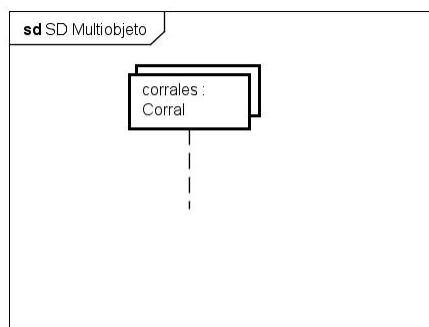
Por consiguiente, si se elige representar el diseño dinámico detallado mediante los Diagramas de Colaboración, hay que hacer **uno para la operación '*ParamSim*' y otro para la operación '*RealizarSim(días)*'**. Consigne cada mensaje con los patrones GRASP (Experto, Creador, etc.) o cualquier otro que lo justifique. Recuerde que, además de especificar el nombre de todas las instancias, debe numerar el orden en que se producen los mensajes.

Sugerencias:

- Inicie el diagrama de la operación '*ParamSim*' con el 1<sup>er</sup> mensaje entre el Actor y la instancia del objeto que actúe como Controlador. A partir de aquí, sitúe los mensajes de las acciones que se desencadenan en otras instancias y numere tanto el orden como la jerarquía en que se producen. **¡¡Ojo!! Sólo es posible intercambiar mensajes desde un objeto hacia un componente** (un miembro) **suyo**: la línea que los une representa esa relación, la accesibilidad y su visibilidad; lo cual debe ser coherente en todos los diagramas. Si hay algún '*bucle*', tenga especial cuidado al ordenar el instante en el que se producen los mensajes que lo componen. Si se utiliza una instancia auxiliar (frecuentemente un multiobjeto vacío), recuerde situar su creación fuera de la iteración. Así mismo, en cualquier operación, utilice la creación de instancias, justo, cuando sea necesario utilizarlas y no antes de disponer de los valores necesarios para su inicialización.

Si esta operación incluye otros eventos (mensajes) desde el Actor, realice el mismo procedimiento, en el mismo orden en que se producen, hasta completarla.

- Tanto en este diagrama, como en el DS, los multiobjetos, colecciones, listas, etc., se representan mediante un doble recuadro '*descentrado*':



- En el otro diagrama, especifique el funcionamiento para la operación de simulación '*RealizarSim(días)*', con las mismas recomendaciones de la otra operación.

La elección de un tipo de diagrama es excluyente respecto a la otra representación. Si se elige representar el diseño dinámico detallado mediante el Diagrama de Secuencia detallado (DS), hay que hacer uno que recoja todas las operaciones del '*flujo básico de éxito*' del caso de uso, uno en el que se represente el paso de mensajes entre el actor y el Controlador, así como entre éste y las distintas instancias implicadas en el funcionamiento, pero **NO** entre el Actor y el Sistema como si fuera una '*caja negra*' (no lo que se denomina un Diagrama de Secuencia del Sistema, DSS).

Si este diagrama le resulta muy extenso para representarlo con claridad en las dimensiones del papel, puede separarlo en la parte correspondiente a la operación '*ParamSim*' y en la correspondiente a '*RealizarSim(días)*'. Pero recuerde que, además de especificar el nombre de todas las instancias, debe situar cada una, con su línea de tiempo, en la posición temporal que le corresponde respecto a las demás (mensajes de creación de instancias), de la misma forma que el resto de los mensajes (aquí no se numeran al situarlos correctamente en la correspondiente línea temporal). Esto es aún más importante si separa el diagrama DS en 2: las instancias creadas en la 1ª operación, '*ParamSim*', aparecen desde el inicio de las líneas de tiempo de la operación '*RealizarSim(días)*', si se utilizan en ella.

Sugerencias:

- Inicie el diagrama con las líneas de tiempo de los objetos que ha representado en el Modelo de Dominio: el Actor, el que actúe como Controlador, así como los objetos del Modelo de Datos, los que haya representado en el Modelo de Dominio (pero ahora son instancias de clases software) y que necesite para especificar el funcionamiento de cada operación. Estos objetos siempre *existen desde el principio* del caso de uso.
- Al empezar, 1º especifique el funcionamiento para la operación de inicialización '*ParamSim*'. Si hay '*bucles*', tenga especial cuidado al situar sus límites y, si utiliza una instancia auxiliar (frecuentemente un multiobjeto vacío), sitúe su creación fuera de ellos. Así mismo, en cualquier operación, utilice la creación de instancias justo cuando sea necesario utilizarlas y no antes de disponer de los valores que sean necesarios para su inicialización.
- Especifique el funcionamiento para la operación de simulación '*RealizarSim(días)*', con las mismas recomendaciones de la otra operación.

**¡ATENCIÓN!:** Tanto si esta pregunta se responde con 2 Diagramas de Colaboración, como si se hace dividiendo en partes el Diagrama de Secuencia, no es posible valorarlas asignando una puntuación máxima a cada una (por ejemplo: 1,5 más 1,5). Esto es debido a que el funcionamiento que se especifica en cada diagrama, o en cada parte, se refiere a la implementación de un único caso de uso, indivisible.

5. (1 punto) A partir del Diagrama de Interacción presentado en la pregunta 4, escriba y desarrolle el **contrato de la operación** '*ParamSim*' que corresponde a la inicialización del caso de uso (la creación e inicialización de las instancias que se requieran en ese momento) y a la asignación de los valores requeridos para realizar los cálculos de la simulación.

La escritura del contrato de una operación consiste en la transcripción '*literal*' de la especificación de su funcionamiento. Por este motivo **es imprescindible utilizar una sintaxis específica** para esta representación. También es ineludible que los nombres de las instancias, las clases a las que pertenecen, los de los componentes a los que se asigna algún valor o los de aquellas entre las que se establece alguna asociación, **sean absolutamente coherentes con lo representado en el Diagrama de Interacción** (cualquiera de ellos).

Sección 4. Evaluación del **Diagrama de Clases** de diseño. (Fase de Elaboración. **Diseño Estático Detallado** del caso de uso).

Se trata de representar la estructura tanto de las clases que intervienen en el funcionamiento como de todas las relaciones que lo permiten (composición, asociación, dependencia, realización, especialización, etc.)

6. (1 punto) Elabore un diagrama de clases para el caso de uso que se está tratando <<*SimularPropagaciónEnfermedad\_X*>> (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 (ControlSim, Calculador, o como lo haya llamado en los diagramas precedentes). Represente los nombres de todos los atributos, asociaciones (con su navegabilidad) y los métodos (excepto 'setters' y 'getters' irrelevantes), tanto de esa clase como de las que estén directamente involucradas con ella en el funcionamiento del caso de uso.

En este diagrama, al menos, deben aparecer las clases software que provienen de los mismos *objetos conceptuales* utilizados en el Modelo de Dominio, con sus atributos, tipos y relaciones representados allí, pero complementados con los correspondientes métodos (con sus tipos y sus argumentos) que se han deducido en los diagramas del diseño dinámico. También se debe complementar con las relaciones de asociación, dependencia, herencia, etc., así como con los objetos, puramente software (clases), que se han obtenido, o elaborado específicamente, en dichos diagramas, para hacer posible el funcionamiento del código.

## Sección 5. Evaluación de la **Transformación del Diseño en Código**.

No se trata de valorar su capacidad para codificar en un lenguaje, sino de validar la identidad entre la especificación del funcionamiento que ha diseñado y el del código que le corresponde.

7. (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 <<SimularPropagaciónEnfermedad\_X>>. Incluya las definiciones de todas las variables que la componen (miembros), pero escriba solamente la definición completa del cuerpo para el método principal o los métodos más significativos: <<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 o los nombres que haya asignado a los métodos elegidos.



Sección 6. Preguntas opcionales **BP**. Motivación.

8. (0'5 puntos) Indique qué principios GRASP ha utilizado en el ejercicio y qué responsabilidades ha asignado guiándose por ellos.
9. (0'5 puntos) Indique qué patrones GoF ha utilizado en el ejercicio y qué mejoras ha obtenido, con su uso, en la elaboración de este desarrollo o en el comportamiento final del código que ha diseñado.