71013035 - Diseño del Software 2021

Centro Asociado de la UNED en Bizkaia
Tutor: Aziz Mulud

Martín Romera Sobrado
Bilbao

Horas de estudio de los contenidos hasta la fecha: **103 horas**
Horas de dedicación para realizar esta actividad: **20(PEC) + 22(PUF) horas**
Número de actividades no evaluables realizadas: **22 actividades**
21 de enero de 2021

1. Cuestiones

1.1. Fase de Inicio: Evaluación de los Casos de Uso

1.1.1. Casos de uso primarios / Ejercicio 1

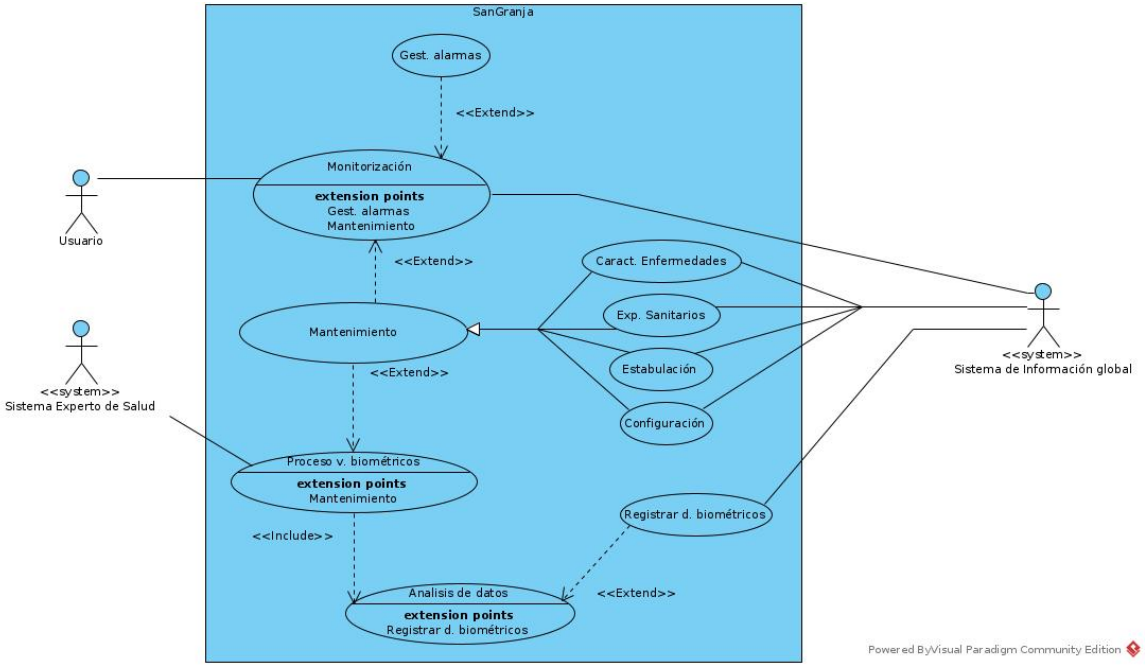
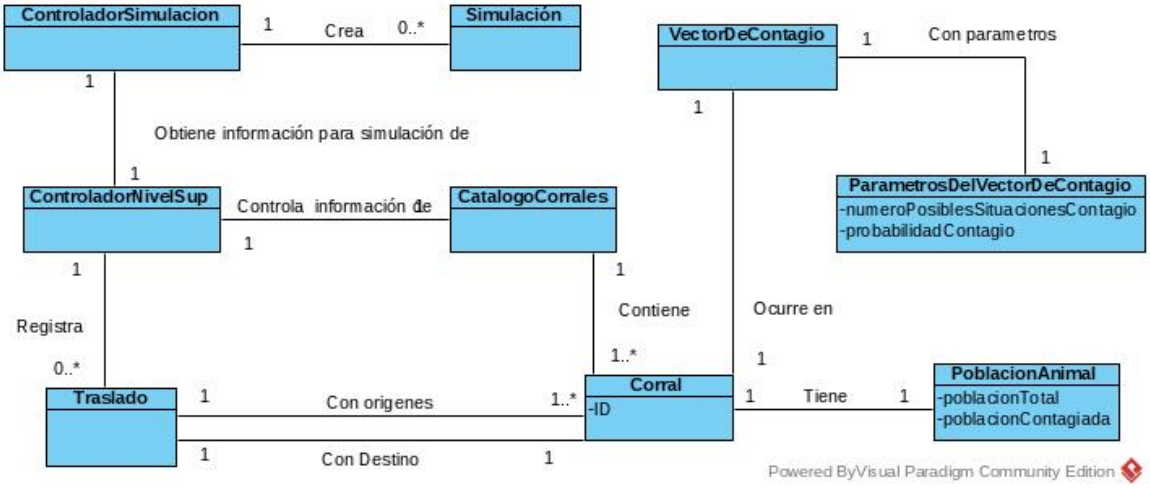


Figura 1: Diagrama de Casos de Uso primarios.

1.1.2. Caso de uso «SimularPropagaciónEnfermedad_X» / Ejercicio 2

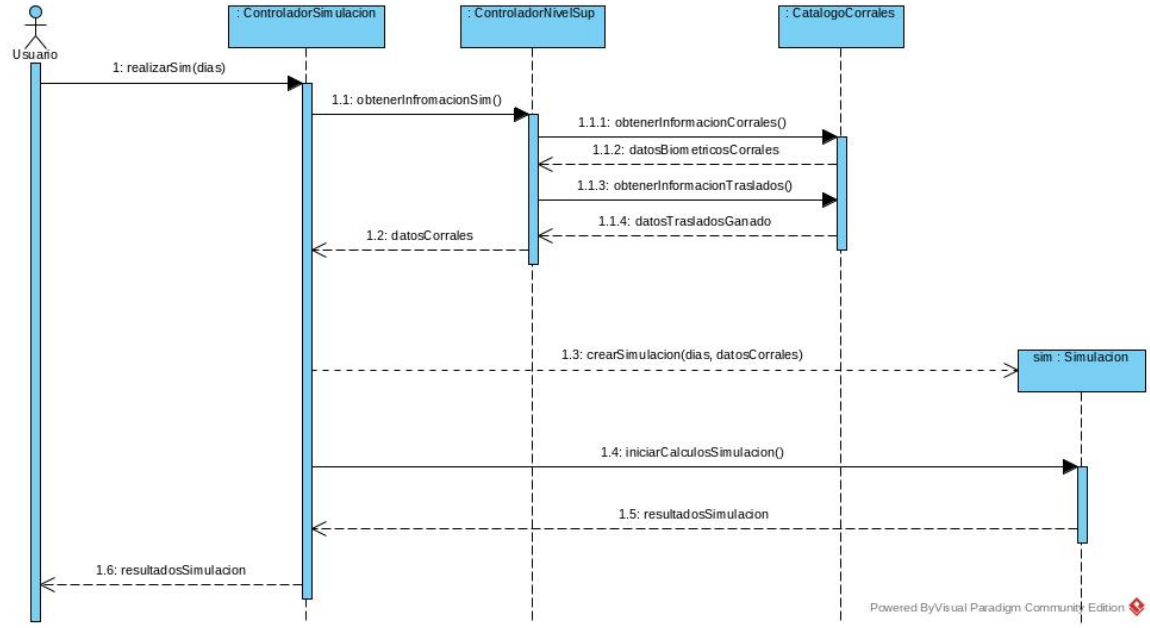
Actor principal:	*Usuario*
Escenario príncipal de éxito	
Acción del actor	*Responsabilidad del Sistema*
1. El *Usuario* establece en qué *Corrales* desea hacer la observación.	2. El *Sistema* toma muestras de los datos biométricos de cada *Corral* seleccionad 3. El *Sistema* realiza los cálculos para estimar la previsión de la propagación de la enfermedad en los corrales para un día. *El sistema repite el paso 3 para todos los días que haya establecido el usuario* 4. El *Sistema* presenta los resultados obtenidos de la iteración del paso 3 en forma de una relación bidimensional.
Escenario alternativo 1: El usuario interrumpe el proceso de cálculos	
Pasos 1, 2 y 3 se mantienen de la misma manera	
4. El *Usuario* decide interrumpir el proceso de cálculo. 6a El *Usuario* acepta la petición 6b El *Usuario* no acepta la petición	5. El *Sistema* realiza una petición de confirmación al *Usuario* 7a El *Sistema* desecha los cálculos y vuelve al estado en el que se encontraba previo a la simulación. 7b *Sistema* reanuda los cálculos y continua en el paso 3 del escenario principal.
Flujo alternativo 2: Alguno de los corrales no tiene animales enfermos	
Pasos 1 y 2 se mantienen de la misma manera	
	3. El *Sistema* detecta que hay un corral sin animales infectados 4. El *Sistema* elimina ese corral para los cálculos de la simulación 5. El *Sistema* reanuda los cálculos y continua en el paso 3 del escenario principal.

1.2. Fase de Elaboración: Modelado Conceptual / Ejercicio 3

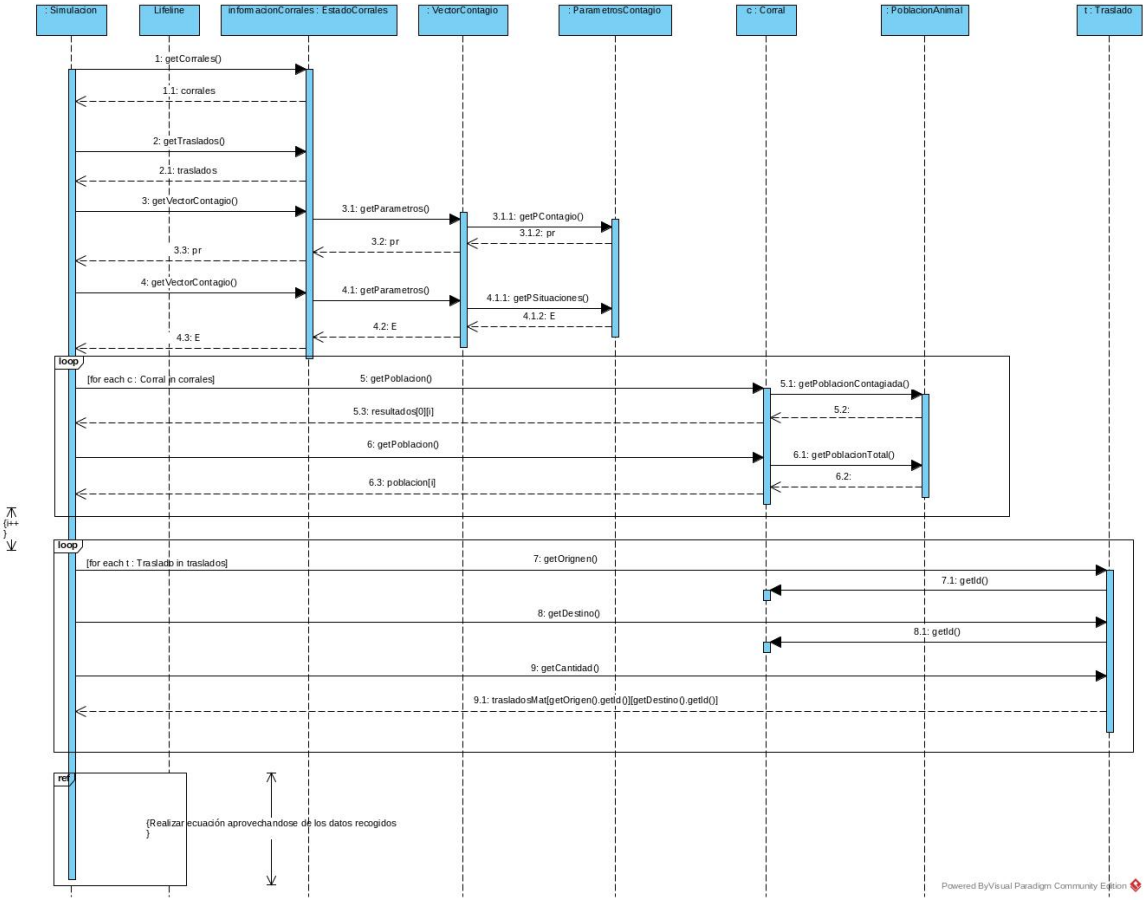


1.3. Fase de Elaboración: Diseño Dinámico Detallado del caso de uso

1.3.1. Diagrama de interacción / Ejercicio 4



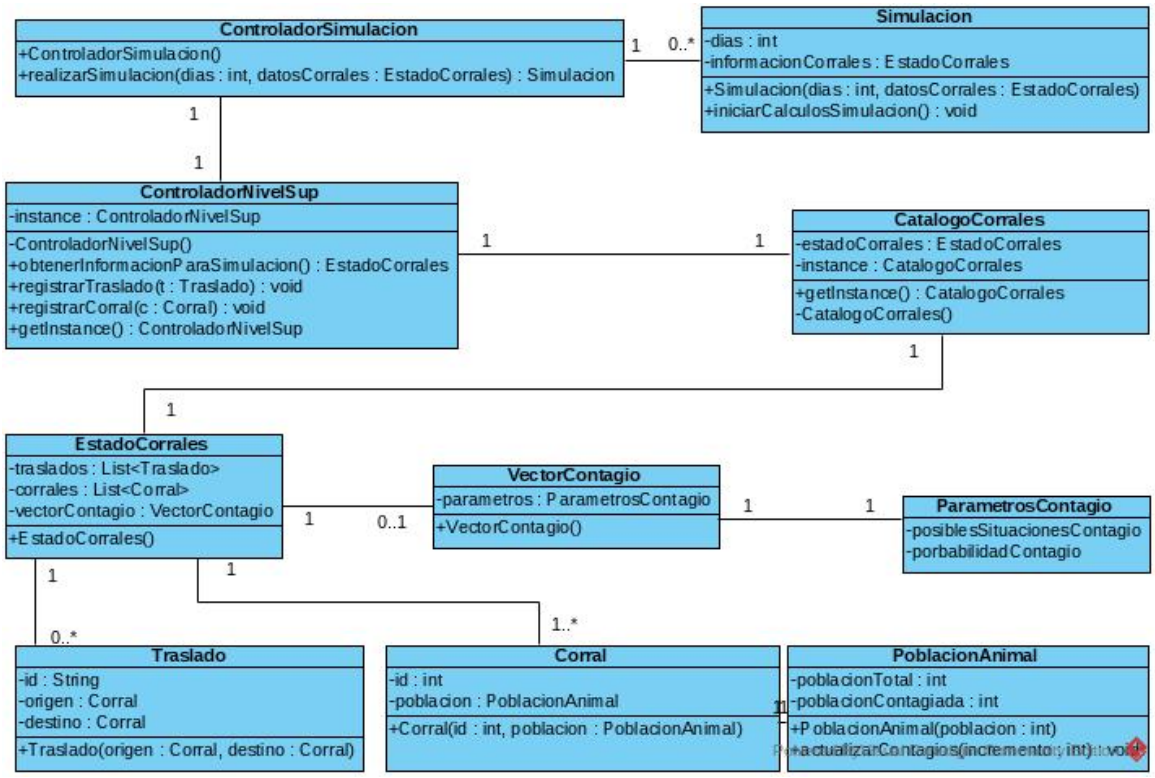
A continuación se añade un diagrama de secuencia que explica como funciona en profundidad la operación `iniciarCalculosSimulacion()`.


: Simulacion
Lifeline
infomacionCorrales : EstadoCorrales
: VectorContagio
: ParametrosContagio
c : Corral
: PoblacionAnimal
t : Traslado
1: getCorrales()
1.1: corrales
2: getTraslados()
2.1: traslados
3: getVectorContagio()
3.1: getParametros()
3.1.1: getPContagio()
3.1.2: pr
3.2: pr
3.3: pr
4: getVectorContagio()
4.1: getParametros()
4.1.1: getPSituaciones()
4.1.2: E
4.2: E
4.3: E
loop
[for each c : Corral in corrales]
5: getPoblacion()
5.1: getPoblacionContagiada()
5.2:
5.3: resultados[0][i]
6: getPoblacion()
6.1: getPoblacionTotal()
6.2:
6.3: poblacion[i]
{i++
}
loop
[for each t : Traslado in traslados]
7: getOrignen()
7.1: getId()
8: getDestino()
8.1: getId()
9: getCantidad()
9.1: trasladosMat[getOrigen().getId()][getDestino().getId()]
ref
{Realizar ecuación aprovechandose de los datos recogidos
}
Powered By Visual Paradigm Community Edition

1.3.2. Contrato de operación: ParamSim / Ejercicio 5

Nombre	`ParamSim`
Responsabilidades	Realizar una simulación sobre la propagación de una enfermedad que haya brotado en los corrales druante unos día determinados
Referencias Curzadas	*Caso de Uso*: «SimularPropagaciónEnfermedad_X»
Notas	Para la evolución de la enfermedad de un día a otro se utilizará la siguiente formula: $$\Delta N_{d+1} = pr \times E \times \left(1 - \frac{N_d}{P} \times N_d\right)$$ Siendo E el promedio de encuentros un día, pr la probabilidad de contagio, P la población animal en un Corral, y N_d el número de animales infectados el día d. En cuanto para calcular la importación de infectados del exterior de un corral utilizaremos la siguiente formula: $$IE_{d+1,i} = \sum_{j=1}^{j=n,j\neq i} T_{j,i} \times \frac{N_{d,j}}{P_j}$$ Siendo $IE_{d+1,i}$ los infectados importados del exterio en el corral i el día d+1, T_j, i son los traslados del corral j a i N_d, j es el número de contagiados en el corral j el día d y P_j es la población animal del corral j. En conclusión la formula del algoritmo de calculo de la simulación quedaría de la siguiente forma $$N_{d+1,i} = N_{d,i} + pr \times E \times \left(1 - \frac{N_{d,i}}{P_i}\times\right) \times \left(N_{d,i} + \sum_{j=1}^{j=n,j\neq i} T_{j,i} \times \frac{N_{d,j}}{P_j}\right)$$ Los traslados deberán estar preprogramados.
Excepciones	Si el proceso se ve interrumpido, no se mostrará ninguna información sobre la simulación que estaba en curso.
Salida	Muestra al usuario en forma de relación bidimensional la evolución de la enfermedad a lo largo de los días.
Precondiciones	Las instancias del `ControladorSimulacion`, `ControladorNivelSup` y `CatalogoCorrales` deberán estar iniciadas en el sistema, además de que `CatalogoCorrales` deberá constar de al menos un `Corral` registrado, y todos los traslados que vayan a ocurrir en el lapso de tiempo de la simulación deberán haber sido registrados para obtener unos resultados útiles. Además el hardware desde el que el usuario inicia la simulación deberá tener acceso a la instancia `ControladorSimulacion`.
Postcondiciones	El usuario podrá examinar los resultados de la simulación desde un hardware especifico para ello (como un terminal) que tenga acceso a la instancia `ControladorSimulacion`.

1.4. Fase de Elaboración: Diseño Estático Detallado del caso de uso / Ejercicio 6



1.5. Transformación del Diseño en Código / Ejercicio 7

La solución a este apartado puede encontrarse en forma de código fuente dentro del directorio `src/` en el paquete de java `es.uned.dspuf`.

1.6. Motivación / Ejercicios 8 y 9

Los patrones de responsabilidades de GRASP que se han utilizado, la más notable es la de *Experto en Información* cuyo patrón se ha utilizado para centralizar la información correspondiente a cada elemento. También ha facilitado una descentralización de la información (algo que a primera vista puede parecer contradictorio) ya que ciertas fracciones de información, como la `PoblacionAnimal` de los `Corrales` o los `ParametrosContagio` de `VectorContagio` se encuentran en clases separadas, apoyandose en el planteamiento que se hizo en el diagrama de clases conceptuales del ejercicio 3. También se hace uso del patrón de *Controlador* en la clase `ControladorNivelSup` quien es quien se encarga de funcionar como sistema principal. Aunque el nombre pueda engañar y que en parte también cumpla la función de *Controlador* de segundo nivel, ya que controla la instancia de `Simulacion`, la clase `ControladorSimulacion` sigue el patrón de *Creador* inicializando la instancia de `Simulacion`. También el esquema de clases siguen un patrón de *Bajo Acoplamiento* asignando unas pocas responsabilidades a clases muy diversas, para que futuros cambios no generen un impacto tan grande en el sistema. También sigue un patrón de *Alta Cohesión* ya que muchos de los metodos

de las clases son sencillos, reutilizables y comprensibles. El único caso que no cumpliria eso serían las clases cercanas a la `Simulacion` que cumplen una tarea más especifica y compleja.

Los patrones más notable de los patrones GoF utilizados es *Singleton*. Tanto `ControladorNivelSup` como `CatalogoCorrales` utilizan este patrón ya que son clases únicas que realmente son independientes entre ellas, facilitando así la interfaz de acceso entre subsistemas. `ControladorSimulacion` podría utilizar este patrón también, pero para este prototipo no erá realmente necesario. Otro patrón destacable es el de *Factoría* utilizado en el `ControladorSimulacion` encargandose de proporcionar los datos necesarios a `Simulacion` aunque despues delegando a este la lógica de gestión de esos datos de forma que no cumple .ª raja tabla.el patrón. Finalmente el patrón de *Fachada* utilizado para la clase `EstadoCorrales` que funciona como punto de acceso dentro del algoritmo de de la simulación para acceder a los datos del corral, lo que simplifica mucho el acceso a ellos.