



UNIVERSIDAD DEL BÍO-BÍO

FACULTAD DE CIENCIAS EMPRESARIALES

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN Y TECNOLOGÍAS DE LA
INFORMACIÓN.

**PROYECTO DE TÍTULO “SISTEMA DE OPTIMIZACIÓN DE RACIONES PARA
VACUNOS WAGYU”**

ALUMNOS:

Alexander Fell

Claudia Fuentes

PROFESOR GUÍA:

Sr. Pablo Sáez

CARRERA:

Ingeniería Civil Informática

**Informe de Proyecto de Título presentado en conformidad a los requisitos para
obtener el título de Ingeniero Civil en Informática.**

Diciembre de 2010

Tabla de Contenido

Resumen	5
Capítulo I. Introducción.....	6
1.1 Introducción.....	7
Capítulo II. Presentación de la Organización y Situación Actual	9
2.1 Reseña de la Empresa	10
2.2 Descripción de la situación actual	10
2.2.1 Proceso de cálculo de raciones	11
Capítulo III. Especificación del proyecto	16
3.1 Descripción del problema	17
3.2 Descripción general del proyecto	18
3.2.1 Objetivo General.....	18
3.2.2 Objetivos Específicos	18
3.2.3 Metodología de desarrollo a utilizar.....	19
3.3 Estudio de Factibilidad Económica	22
3.3.1 Cálculo de Costos	22
3.3.2 Estimación de Beneficios	22
3.3.3 Valor actual Neto (VAN)	24
Capítulo IV. Especificación de Requisitos	26
4.1 Actores del Sistema	27
4.2 Requisitos Funcionales	27
4.3 Requisitos No Funcionales	29
4.4 Método de Optimización	30
4.4.1 Método Simplex	31
4.4.2 Modelo matemático	31
Capítulo V. Análisis	35
5.1 Diagrama de casos de uso	36
5.2 Especificación de casos de uso	38
5.2.1 Gestión General de Etapas.....	39
5.2.2 Gestión de Compuestos y sus aportes	45
5.2.3 Gestión de Precio y Stock.....	50
5.2.4 Gestión Específica de Raciones y Restricciones	52
5.2.5 Optimización y Reportes	58
5.2.6 Control de Acceso y Gestión de nombre de usuario y contraseña	60
5.3 Modelo Conceptual.....	62
5.3.1 Descripción del modelo conceptual.....	62
5.4 Diagrama de Secuencia.....	65
Capítulo VI. Diseño	66
6.1 Arquitectura del sistema	67
6.2 Diagrama de Paquetes de la Arquitectura.....	70

6.3	Diagrama de clases	71
6.4	Modelo Entidad Relación	73
6.3.1	Descripción de Tablas	73
6.5	Interfaz de Usuario.....	80
6.6	Diagramas de Secuencia (nivel de diseño)	80
Capítulo VII. Plan de Pruebas		81
7.1	Estrategia de prueba a utilizar.....	82
7.1.1	Los elementos a probar.....	82
7.1.2	Las características.....	82
7.2	Casos de prueba	82
Capítulo VIII. Conclusión		84
8.1	Conclusión	85
8.2	Trabajos Futuros	87
Bibliografía.....		88
Glosario		89
Anexo A “Diagramas de secuencia”		95
Anexo B “Prototipo Interfaz de Usuario”.....		100
Anexo C “Diagrama de secuencia a nivel de diseño”		108
Anexo D “Casos de prueba”.....		115
Anexo E “Ejecución de los casos de prueba (versión Alfa)”		137
Anexo F “Ejecución de los casos de prueba (versión Beta)”		164
Anexo G “Raciones de Prueba y Resultados de Optimización		189

Resumen

Agrícola Mollendo S.A. es una empresa dedicada a la crianza, engorda y venta de carne de vacunos wagyu¹. La empresa se encuentra ubicada en la zona de Los Ángeles y su objetivo principal es lograr un producto de calidad, que cumpla con los requisitos para ser exportado. Hoy en día la cantidad de exportaciones no es la que la empresa desea, por lo que ha decidido tomar medidas para lograr cumplir con los estándares de calidad requeridos. Una de las medidas es lograr raciones alimenticias para los animales nutricionalmente balanceadas, al mínimo costo.

El objetivo de este proyecto es ayudar en la generación de raciones óptimas, es decir, raciones que cumplan con los requisitos nutricionales mínimos al menor costo posible. Para lograr lo anterior, se debe especificar como entrada una serie de compuestos alimenticios, cada uno de éstos tiene distintos aportes nutricionales, los cuales se deben especificar en el programa. Se debe ingresar también información sobre las etapas de engorda y las restricciones, tanto nutricionales mínimas, como de stock y de compuestos. El software debe ser capaz de generar una ración óptima para cada etapa de engorda.

Una vez implementado el software, se logró obtener una ración óptima, la cual generó excelentes resultados. En la sección análisis de factibilidad de este informe, se puede apreciar el gran ahorro económico que brinda el software a la organización, en cuanto al precio de las raciones que genera el programa, versus las generadas por el ingeniero agrónomo. El ahorro no está sólo relacionado con los costos de las raciones, sino también en relación al tiempo, debido a que el ingeniero agrónomo se demoraba aproximadamente cuatro horas en realizar un plan de alimentación y el programa lo realiza en pocos segundos.

¹ **Wagyu**: El Wagyu es una raza bovina originaria de Japón, de la ciudad de Kōbe. Su carne es muy apreciada por su textura y sabor. También se conoce como Japanese Black. Por Wagyu (和牛, wagyū) se refiere a varios tipos de crías de ganado, con una fuerte tendencia genética al marmoleo de la carne y a la producción de un alto porcentaje de grasas insaturadas del tipo oleaginoso. El nombre proviene de los kanji 和 (wa, que significa "armonía" pero históricamente ha estado muy ligado a Japón) y 牛 (ushi o gyū, vaca/vacuno/ternera). El significado es, simplemente, "vaca japonesa".

Capítulo I. Introducción

1.1 Introducción

Las tecnologías de información (TI) en estos días son un gran apoyo para las grandes empresas, instituciones y para la mayoría de la población, debido a que permiten realizar las actividades en un tiempo menor, con mayor eficiencia y precisión. Es por lo anterior que las organizaciones se ven obligadas a incorporar cada vez más tecnología en sus procesos productivos, puesto que sin ellas pierden competitividad, ubicándose en una posición de desventaja respecto al resto del mercado.

La empresa Agrícola Mollendo se ha percatado de la importancia de la incorporación de TI en sus procesos productivos. Dentro de los procesos a los que desea incorporar TI, se encuentra el de generación de raciones alimenticias para sus vacunos. Hoy en día, este proceso se lleva a cabo de una forma manual y poco exacta, obteniendo resultados no óptimos, que producen pérdidas a la empresa, debido a que la calidad de la carne no es la esperada.

Agrícola Mollendo realiza exportaciones de su producto al mercado internacional, especialmente a Japón, siendo este último un mercado muy exigente en cuanto a la calidad. Para seleccionar las carnes a exportar, la empresa debe cerciorarse de que éstas cumplan con los requisitos mínimos de calidad impuestos por los estándares internacionales. Para evaluar la calidad de la carne, existen procesos de análisis, donde ésta se clasifica en doce diferentes niveles, siendo el nivel doce el de mejor calidad. El mercado más rentable para Agrícola Mollendo es el japonés, sin embargo, éste exige una calidad mínima de seis. La cantidad de animales que Agrícola Mollendo produce con este nivel de calidad no es el deseado por la misma, en consecuencia, los animales que no alcanzan el nivel de calidad para ser exportados deben venderse al mercado nacional, el cual es mucho menos rentable para la empresa. Por lo anteriormente mencionado, Agrícola Mollendo desea aumentar el porcentaje de vacunos que alcanzan el nivel seis y superior.

Para que un wagyu alcance el nivel de exportación, uno de los factores influyentes es la alimentación. Ésta debe cumplir con los requisitos mínimos nutricionales para que el animal se desarrolle de la forma deseada. Así, para lograr aumentar el volumen de exportación a Japón, parte del plan que ha decidido llevar a cabo Agrícola Mollendo, es el

desarrollo de un sistema que permita realizar el proceso de raciones alimenticias para sus vacunos wagyu, de una manera automática y flexible, que entregue raciones óptimas, las cuales deberían ayudar a aumentar la calidad de las carnes.

En el presente informe, se dará a conocer la organización y la situación actual de la misma, luego se presentará la especificación del proyecto. Después, se mostrará la especificación de requisitos, continuando con un análisis y diseño. Se dará también a conocer el plan de pruebas aplicado al programa y finalmente se presentarán algunas conclusiones y trabajos futuros.

Capítulo II. Presentación de la Organización y Situación Actual

2.1 Reseña de la Empresa

Agrícola Mollendo S.A es una empresa chilena fundada en el año 2002. Se encuentra ubicada en la zona de Los Angeles, VIII Región de Chile. Desde sus orígenes, ha dirigido todos sus esfuerzos hacia la producción y distribución de carne de Wagyu.

La empresa es propiedad de la familia Gras, la cual también participa en los negocios inmobiliarios, de ingeniería y construcción, acuicultura y en la industria del vino.

El cuidado del medio ambiente está fuertemente presente en la empresa. El feedlot² de Mollendo cuenta con aprobación de sus Prácticas Medio Ambientales (COREMA, cert. 023 del 30 de marzo de 2009) y está certificado por el Servicio Agrícola Ganadero de Chile (SAG).

Cabe destacar, que todos sus campos están bajo certificación PABCO A (Plantel de Animal bajo Control Oficial).

2.2 Descripción de la situación actual

La empresa se encarga básicamente de la engorda de una especie bovina conocida como Wagyu, para luego venderlo al mercado nacional e internacional. El proceso hoy en día, a grandes rasgos, es el siguiente:

Se compran los terneros de aproximadamente 200 kilogramos, los cuales se engordan hasta alcanzar aproximadamente 300 kilogramos. A esta etapa se le conoce como recría.

Una vez alcanzados los 300 kilogramos, los terneros pasan a una segunda etapa, con una ración diferente a la de la etapa anterior. En esta fase se sigue engordando a los wagyu hasta alcanzar los 450 kilogramos.

² Es un área donde los animales para carne permanecen confinados largos períodos, que generalmente superan los 150 días, en corrales de dimensiones determinadas donde reciben la alimentación y tratamientos correspondientes, según su destino final.

Cuando llegan a los 450 kg pasan a la tercera etapa. Acá se continúa alimentando a los waguay como en las etapas anteriores, pero con diferentes raciones. Se mantienen en dicha etapa, hasta que pesen aproximadamente 600 kilos.

Finalmente, cuando los Wagyu pesan más de 600 kilogramos, éstos están listos para ser faenados.

En la Figura 1 se muestra el proceso de alimentación, destacando el flujo entre las etapas de engorda y la diferenciación del tipo de raciones para cada una.

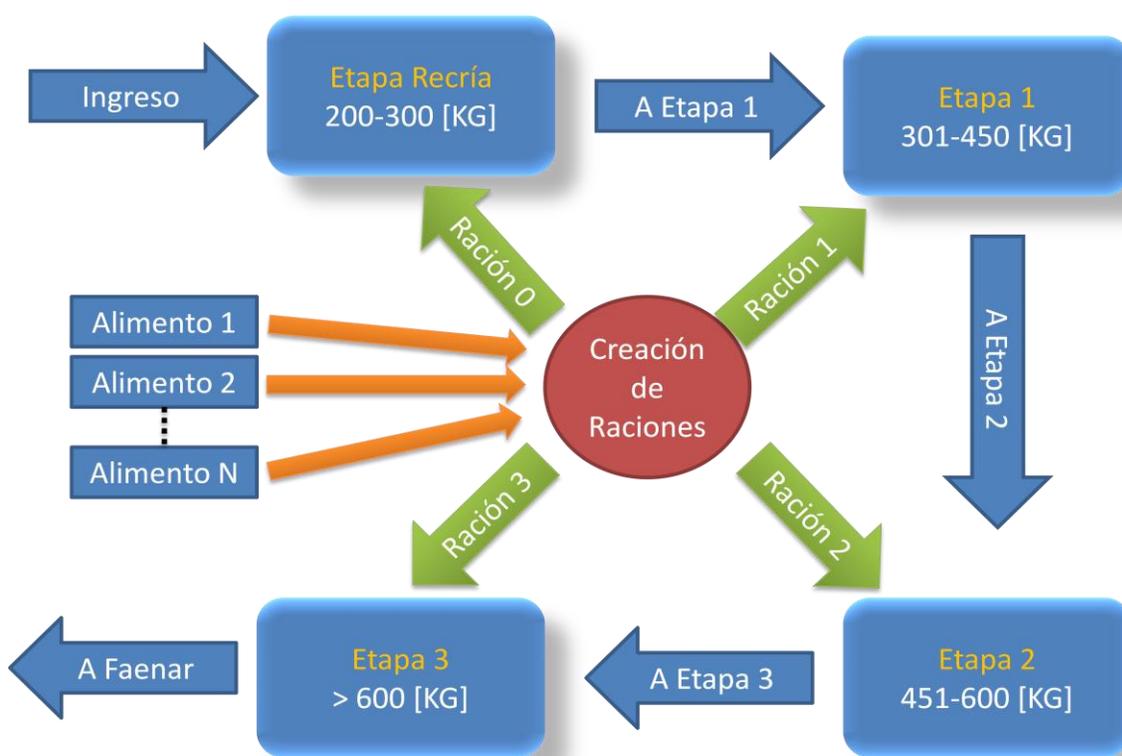


Figura 1. Proceso de alimentación de vacunos waguay.

2.2.1 Proceso de cálculo de raciones

Actualmente el proceso de creación de raciones se hace en una planilla Excel de manera manual, es decir ajustando valores según se estime conveniente, intentando aproximar la combinación de alimentos que entregue la ración de menor costo para cada una de las etapas, cumpliendo con las restricciones.

El cálculo de raciones para alimentación de vacunos Wagyu, involucra una gran cantidad de variables a considerar. Agrícola Mollendo consta con personal capacitado, el cual se mantiene informado sobre los alimentos disponibles en el mercado y también gestiona los alimentos producidos por la organización. Cada alimento tentativo para participar en la alimentación de los vacunos, debe ser previamente analizado por un laboratorio. El análisis entregado proporciona información respecto al contenido nutricional del alimento, es decir los distintos nutrientes y su porcentaje de participación en el alimento o compuesto³. En la Tabla 1 podemos apreciar un ejemplo donde se muestran los distintos compuestos que conforman las raciones alimenticias y cada uno de ellos con sus respectivos aportes nutricionales.

Composición	MS	PT	TDN	MCAL	FDN	CA	P	\$
ensilaje de maíz	0,33	0,07	0,70	2,67	0,41	0,25	0,15	28,00
coseta orafti	0,16	0,08	0,68	2,47	0,46	2,72	0,16	11,00
vinaza	0,59	0,77	0,78	2,83	0,17	0,00	0,00	81,00
bagazo cebada	0,33	0,00	0,00	0,00	0,00	0,00	0,00	0,00
maíz grano húmedo	0,73	0,08	0,88	3,19	0,09	0,06	0,34	108,00
mezcla soprodi	0,89	0,13	0,79	3,15	0,19	0,14	0,75	127,00
paja de trigo	0,90	0,04	0,45	1,62	0,85	0,18	0,05	32,00
urea	0,99	2,87	0,00	0,00	0,00	0,00	0,00	200,00
ensilaje de pradera	0,35	0,09	0,60	2,27	0,50	0,67	0,16	14,00

Tabla 1. Compuestos y sus aportes nutricionales.

Nótese que en la Tabla 1 existe una cantidad específica de aportes nutricionales (MS, PT, TDN, MCAL, FDN, CA, P). A continuación se describirá cada aporte nutricional y su respectiva sigla.

MS (*Materia Seca*): Es el residuo de la extracción de agua de cierta materia, mediante un calentamiento de la misma.

PT (*Proteínas totales*): Es la cantidad de proteínas totales que tiene determinado compuesto. La proteína es un elemento imprescindible para el desarrollo del animal, ya que

³ Desde aquí en adelante se utilizará el término compuesto para referirse a todo lo que conforma una ración alimenticia, tanto alimentos como otros compuestos químicos. Se estima que este término abarca a todo lo que se puede incluir en una ración alimenticia.

prácticamente todos los procesos biológicos dependen de la presencia o la actividad de este tipo de molécula. [1]

TDN (*Total nutrientes digestibles*): Es una variable, entregada por un método matemático para el cálculo aproximado de la energía liberada por un ingrediente dado. Este método además de valorar energéticamente a un alimento partiendo de ensayos de digestibilidad, puede valorar la energía existente en % o en Kg.

MCAL (*Mega calorías*): Se define la caloría como la cantidad de energía calorífica necesaria para elevar un grado centígrado la temperatura de un gramo de agua pura, desde 14,5 °C a 15,5 °C, a una presión estándar de una atmósfera. La kilocaloría (símbolo kcal) es igual a 1.000 cal. A partir de lo anterior la mega caloría equivale a mil kilocalorías. [1]

FDN (*Fibra detergente neutro*): Es la porción de la muestra de alimento que es insoluble en un detergente neutro (método de los detergentes de Van Soest). La fibra detergente neutro está básicamente compuesta por celulosa, hemicelulosa, lignina y sílice, y se le denomina pared celular. La misma se correlaciona inversamente con el consumo voluntario de MS.

CA (*Calcio*): Además de su función en la construcción y mantenimiento de huesos y dientes, el calcio también tiene otras funciones metabólicas. El calcio influye en varios procesos biológicos como por ejemplo: en la contracción de los músculos, liberación de neurotransmisores, regulación de algunas enzimas, trasmisión nerviosa, regulación de los latidos cardiacos, mantiene el tono muscular y controla la irritabilidad nerviosa.[1]

Potasio: El Potasio ocupa el tercer lugar en la composición mineral de los animales; unos 2/3 del mismo están localizados en la piel y en los músculos. El Potasio está ubicado principalmente en el interior de las células. Éste mantiene el balance de agua corporal; ayuda en la activación de varios sistemas enzimáticos, como el de transferencia y utilización de la energía, la síntesis de proteínas y el metabolismo de los carbohidratos. [1]

Como podemos apreciar en la Tabla 1, por ejemplo, la vinaza está compuesta de 59% de materia seca, de la cual, el 77% corresponde a proteínas totales, el 78% a total

nutrientes digestibles, etc. La información presentada en la Tabla 1 es la que entrega el laboratorio al ingeniero agrónomo a cargo, después del análisis a los compuestos.

Esta información es utilizada para la formulación de cada etapa de engorda como se puede apreciar en la Tabla 2.

Alimento	cantidad T.C.O	Mat.Seca	PT	TDN	MCA L	FDN	CA	P	\$
ensilaje de maíz	6,00	1,97	0,14	1,38	5,27	0,80	0,49	0,30	168,00
coseta orafti	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
vinaza	0,15	0,09	0,07	0,07	0,25	0,02	0,00	0,00	12,15
maíz grano húmedo	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
mezcla soprodi	1,50	1,34	0,17	1,05	4,21	0,25	0,19	1,00	190,50
paja de trigo	0,50	0,45	0,02	0,20	0,73	0,38	0,08	0,02	16,00
sal molido	0,10	0,09	0,00	0,00	0,00	0,00	1,44	0,72	31,50
bicarbonato	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
sal molido E	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
urea	0,08	0,08	0,23	0,00	0,00	0,00	0,00	0,00	16,00
ensilaje de pradera	10,00	3,47	0,33	2,08	7,88	1,74	2,33	0,56	140,00
aportes totales	18,33	7,49	0,95	4,79	18,34	3,19	4,53	2,60	574,15
requerimientos		6,74	0,90	5,08	18,38		3,80	1,80	
balance		0,75	0,05	-0,29	-0,04		0,73	0,80	

Tabla 2. Ejemplo de formulación de ración para la etapa de recría.

Como se puede apreciar en la Tabla 2, en la primera columna se encuentran los compuestos seleccionados para participar en esta etapa. Los valores que el ingeniero agrónomo a cargo varía manualmente para obtener una ración son los que pertenecen a la

segunda columna rotulada como cantidad T.C.O⁴. A partir de éstos y los datos de la Tabla 1, se generan los valores de los aportes nutricionales, que pueden apreciarse en la Tabla 2 desde la tercera columna en adelante. En la fila denominada aportes totales se suman los valores de los aportes correspondientes a cada nutriente, para evaluar si se cumplen los requisitos mínimos con las cantidades especificadas en la segunda columna. La fila “requerimientos” indica, tal como su nombre lo dice, los requisitos nutricionales mínimos que se deben cumplir para esta etapa. La fila balance corresponde a la resta de “Aportes totales” y “requerimientos”, obteniendo el margen en relación al cumplimiento de los requisitos, por ejemplo en la columna 3 de “Aportes totales” el aporte de materia seca es 7,49 y en la misma columna en la fila de “requerimientos” el requisito mínimo de materia seca es 6,74, por lo tanto se obtiene un “balance” positivo, el cual indica que se cumple con el requerimiento mínimo excediéndose 0,75 kilogramos.

Para la fila del compuesto vinaza, el aporte de la materia seca se calcula de la siguiente forma: como se puede apreciar en la Tabla 1 el 59% de la vinaza corresponde a materia seca, para obtener las proteínas totales correspondientes a este compuesto, se multiplica la cantidad T.C.O por el 0.59 correspondiente a la materia seca y luego el resultado (0,0885) debe ser multiplicado por el porcentaje de proteína, también mostrada en la Tabla 1, es decir 0,77, obteniéndose así el valor de 0,07 kilogramos.

Este proceso se realiza para cada una de las etapas de engorda, obteniéndose actualmente un resultado aceptable, pero no óptimo.

⁴ T.C.O (tal como ofrecido): se refiere al alimento como “materia verde” es decir con agua incorporada, a diferencia de la materia seca, que es el compuesto sin agua.

Capítulo III. Especificación del proyecto

En este capítulo se darán a conocer los aspectos más generales del proyecto que se llevará a cabo, se comenzará con la descripción del problema que se desea resolver, luego se indicarán los objetivos generales y específicos del desarrollo del proyecto. A continuación se realizará una descripción más detallada sobre el proyecto, se indicará la metodología con la que se desarrolló, el enfoque a utilizar y se describe el tipo de lenguaje usado para modelar.

3.1 Descripción del problema

Mollendo S.A., empresa a la cual está destinado el sistema, produce un animal de alta calidad en términos de marmoleo⁵ para satisfacer a mercados exigentes. En general, un buen nivel de marmoleo es una condición que se obtiene contando con un animal de genética wagyu y la alimentación adecuada para que logre la máxima infiltración de grasa posible, mientras más alto es el nivel de marmoleo (por sobre 6 en la tabla japonesa), mejor es su precio en el mercado. Para esto se requiere de un tiempo de espera mayor que el del sistema de engorda tradicional y además de una gran inversión en alimentación. Para lograr un alto nivel de marmoleo, lo transcendental es la alimentación brindada a los wagyu.

La alimentación debe ser equilibrada nutricionalmente, considerando que, un animal engordado con cierto alimento, por más que la cantidad sea suficiente, éste puede no lograr obtener el marmoleo necesario para entrar en el mercado. Así, se destaca la importancia de establecer y cumplir con los requisitos nutricionales necesarios, para lograr obtener un animal con un marmoleo adecuado.

Además, se debe tener siempre en cuenta el aspecto económico, por lo que los gastos de engorda no deben ser extremadamente altos, pues restarían rentabilidad a la empresa. Ésta es la principal dificultad que tiene Agrícola Mollendo hoy en día, debido a que el cálculo de las raciones alimenticias se realiza de forma manual, lo cual no entrega una ración óptima en cuanto a los costos y combinación de alimentos para cumplir con los requerimientos alimenticios.

⁵ El **marmoleo** es la cantidad de grasa entreverada dentro de la carne y se observa principalmente en el área del ojo de costilla, en un corte hecho entre las costillas duodécima y decimotercera. El marmoleo es el principal factor a tomar en cuenta por el consumidor para determinar la calidad de la carne. Mientras mayor sea el nivel de marmoleo, de mejor calidad será la carne, puesto que ésta tendrá mejor sabor y será más jugosa.

Es por esto que se desea desarrollar un sistema que permita entregar raciones nutricionalmente balanceadas (cumpliendo con los requerimientos nutricionales) al mínimo costo.

3.2 Descripción general del proyecto

A continuación se detallarán los objetivos del proyecto a realizar, se especificará el objetivo general, para luego desglosarlo en objetivos específicos. Orientado al desarrollo del proyecto, se dará a conocer la metodología de desarrollo a utilizar, junto con una descripción detallada de la misma.

3.2.1 Objetivo General

Implementación de un sistema de optimización de raciones para la alimentación de vacunos wagyu. El software permitirá optimizar el costo de las raciones para la alimentación de vacunos wagyu, colaborando en el proceso de alimentación de Agrícola Mollendo y proporcionando una disminución de los costos de alimentación de los wagyu dentro de lo que permite la dieta alimenticia de dichos animales.

3.2.2 Objetivos Específicos

- Planteamiento de un modelo matemático adecuado para resolver el problema de optimización.
- Creación de una base de datos que permita almacenar la información relacionada con la dieta de los animales, incluyendo las distintas raciones calculadas, compuestos, nutrientes, entre otros.
- Creación de una interfaz que permita al usuario:
 - Ingresar nuevos alimentos, con sus aportes nutricionales y precios correspondientes.
 - Seleccionar qué alimentos desea utilizar para realizar las raciones, por cada etapa de engorda.
 - Ingresar restricciones de cantidad de alimentos máximo o mínimo por etapa de crecimiento.
 - Ingresar los requisitos mínimos de nutrientes, necesarios por etapa.

- Obtener raciones previamente realizadas. Para esto el usuario, debió haber guardado las raciones previas, utilizando el programa.
- Obtener reportes de raciones calculadas.
- Se desarrollará un módulo de optimización, el cual tendrá como entrada el modelo matemático (restricciones nutricionales y otras) y como salida entregará la ración óptima (en cuanto a los alimentos y su precio) para engordar a un animal, esto por las etapas de engorda existentes.
- Se contemplará dentro del programa un módulo de generación de reportes, con los resultados de la optimización. Específicamente se detallará la cantidad de cada alimento a utilizar, en la ración de cada etapa. El programa entregará estos reportes, en formato EXCEL.

3.2.3 Metodología de desarrollo a utilizar

Se utilizará el modelo de desarrollo iterativo incremental, con un enfoque orientado a objetos, el cual permite realizar la documentación a medida que se desarrolla el proyecto. Se utilizarán estándares UML para las etapas de análisis y diseño.

3.2.3.1 Modelo iterativo incremental.

El modelo incremental combina elementos del modelo lineal secuencial (aplicados repetidamente) con la filosofía interactiva de construcción de prototipos. El modelo incremental aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un «incremento» del software [2]. Por cada incremento se realiza una etapa de análisis, diseño, código (implementación) y prueba. El resultado de cada incremento es una versión del software (véase Figura 2). Se realizan tantos incrementos como se estime necesario, para obtener el producto final.

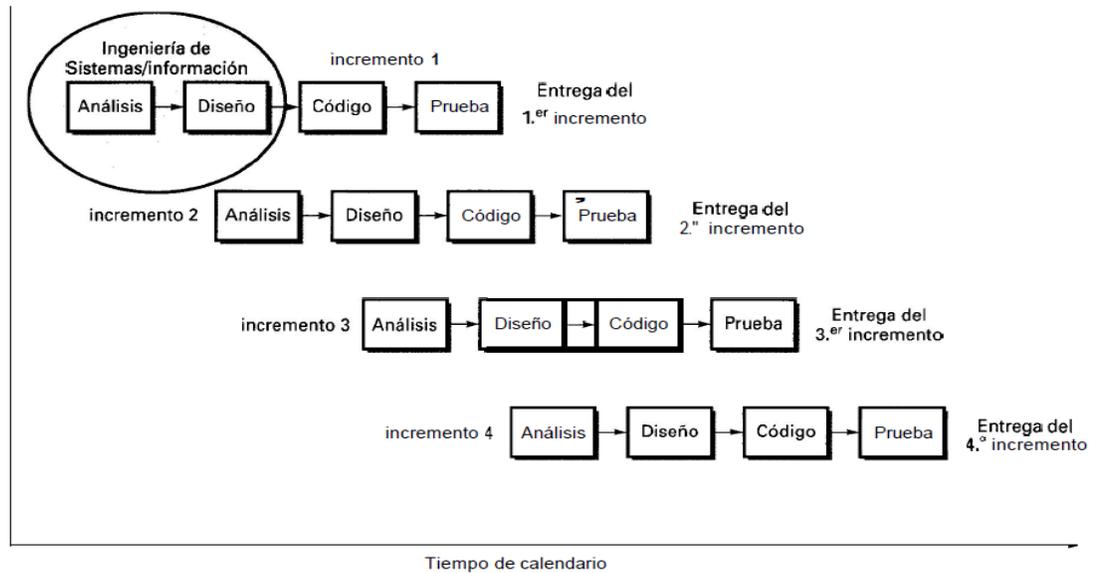


Figura 2. Modelo iterativo Incremental [2].

Este proyecto se planea desarrollar en dos incrementos:

- Primer Incremento:
 - Obtención de requisitos: se realizará en base a reuniones con el cliente.
 - Análisis: Se analizarán los requisitos y se llevará a cabo la documentación relacionada con esta etapa.
 - Prototipo: Se creará un prototipo de interfaz gráfica, para que sea aprobada por el usuario. Si se requieren cambios, se establecerán y documentarán apropiadamente.
 - Diseño: Se realizará un diseño del sistema inicial basado en UML.
 - Implementación: Se enfocará en:
 - Desarrollar un módulo capaz de resolver el problema de optimización.
 - Se generará la base de datos y las respectivas conexiones a ésta.
 - Se implementará la lógica básica, para el funcionamiento del programa.
 - Se implementará una interfaz gráfica básica.

- Pruebas: Se realizarán pruebas, para verificar el funcionamiento del programa. Se registrarán los resultados y se llevarán a cabo las correcciones pertinentes.
- Segundo Incremento:
 - Análisis: Se analizará la documentación y los cambios realizados en los requisitos, hasta esta etapa.
 - Diseño: Se especificará detalladamente lo que se ha realizado en la etapa anterior.
 - Implementación:
 - Se enfocará en cumplir los requerimientos de interfaz gráfica, especificados por el usuario.
 - Se desarrollará el módulo de generación de reportes.
 - Se corregirán errores encontrados en la etapa anterior.
 - Pruebas: Se realizarán pruebas exhaustivas, del funcionamiento completo del programa y se corregirán los defectos encontrados.
- Documentación: Durante todo el desarrollo del proyecto, se llevará a cabo la documentación pertinente a cada etapa.

3.3 Estudio de Factibilidad Económica

En esta sección se mostrará la relación costo beneficio que implica la creación e implementación del software. Los gastos a considerar en este análisis de factibilidad son: costos de desarrollo, costo de puesta en marcha y costos del software. Luego se compararán con los beneficios que generará la utilización del software.

3.3.1 Cálculo de Costos

En la Tabla 3 se detalla el costo de desarrollo y puesta en marcha del proyecto. Esta información será utilizada para evaluar la factibilidad del proyecto.

Categoría	Unidades	*Precio Unitario (UF)	Cantidad (días)	Valor Total (pesos)
Programador	2	21.007,32	10	210.073
Mantenimiento anual	1	21.447,95	5	107.235
Base de datos Oracle 10g	1	-	-	84.636 ⁶
Total				401.944

Tabla 3. Costos relacionados con el desarrollo e implementación del software.

3.3.2 Estimación de Beneficios

Para la estimación de beneficios se consideraron las horas que el ingeniero agrónomo dedica a generar raciones utilizando el método actual. A continuación se detalla el cálculo:

El ingeniero agrónomo tarda aproximadamente 4 horas en generar un plan de alimentación para cuatro etapas. Esta actividad debe realizarse una vez al mes como mínimo. El salario aproximado del ingeniero agrónomo es de \$ 3.000.000. A partir de lo anterior se obtiene:

- Salario diario: $\$3.000.000 / 30 \text{ días} = \100.000 diarios.

⁶ Licencia de Oracle estándar calculada con el precio del dólar del día 20-12-2010

- Salario por hora (suponiendo que trabaja 8 horas al día): \$10.000/8 horas = \$12.500 por hora.
- Valor del cálculo de raciones mensual: \$12.500 * 4 horas = \$50.000 mensuales
- Valor de cálculo de raciones anual: \$600.000

Otro beneficio considerado es el ahorro en el costo de las raciones al utilizar el software. A continuación se detalla en la Tabla 4 la comparación entre los precios (pesos chilenos) de las raciones generadas por el programa y las raciones generadas por el ingeniero agrónomo a cargo de la alimentación. Nótese la columna rotulada como: “cumple con requisitos mínimos”, la cual indica si la ración cumple con los requisitos mínimos nutricionales para la etapa especificada. Lo anterior se agrega debido a que, se pueden generar raciones más baratas, que no cumplan con los requisitos mínimos nutricionales.

Nombre de las etapas	Valores generados por el ingeniero agrónomo	Valores generados por el software	Ahorro/perdida	Cumple con requisitos mínimos ⁷	
				IA	SW
Recría	574,15	742,79	-168,64	No	Sí
Etapa 1	803,1	761,13	41,97	No	Sí
Etapa 2	1088,92	648,65	440,27	No	Sí
Etapa 3	1084,85	614,33	470,52	No	Sí
Total	-	-	784,12	-	-

Tabla 4. Comparación planes de alimentación ingeniero agrónomo vs. Software⁸

Con la información obtenida en la Tabla 4, se puede realizar una estimación del ahorro o pérdida generado al utilizar el programa. En la Tabla 5 se utilizará la información anteriormente obtenida, para calcular, según la cantidad de animales por etapa, el ahorro o pérdida anual.

Nombre de las etapas	Ahorro/pérdida	Cantidad de animales	Ahorro/pérdida anual
Recría	-168,64	300	-18.466.080
Etapa 1	41,97	700	10.723.335
Etapa 2	440,27	1600	257.117.680
Etapa 3	470,52	800	137.391.840
Total		3400	386.766.775

Tabla 5. Estimación de ahorros o pérdidas anuales por el uso del programa.

⁷ IA: Ingeniero agrónomo, SW: Software.

⁸ Los valores se encuentran en pesos chilenos.

A continuación, en la Tabla 6 se presentará el flujo de caja con una proyección de cinco años.

Flujos de Caja						
	Año 0	Año 1	Año 2	Año 3	Año 4	Año 5
Costos Programadores	- 210.073	-	-	-	-	-
Costo de la Base de Datos	- 84.636	-	-	-	-	-
Costos de Mantención	-	- 107.235	- 107.235	- 107.235	- 107.235	- 107.235
Ahorro/ Pérdida Estimada por calculo de ración	-	386.766.775	386.766.775	386.766.775	386.766.775	386.766.775
Ahorro hrs. trabajo ingeniero agrónomo	-	600.000	600.000	600.000	600.000	600.000
Flujos de Caja Netos	- 294.709	387.259.540	387.259.540	387.259.540	387.259.540	387.259.540
VAN(5%)	\$	1.676.336.435				

Tabla 6. Proyección del Proyecto a 5 años.

Se debe notar que éste análisis de factibilidad no incluye, debido a la alta complejidad que esto implica, el beneficio otorgado por la mejora esperada en la calidad de la carne, lo que debería aumentar aún más los beneficios al utilizar el programa.

3.3.3 Valor actual Neto (VAN)

El valor actual neto es un indicador que trae a valor presente los flujos de caja generados por un proyecto. Este indicador es el más utilizado para el evaluar la factibilidad económica de un proyecto. Al ser este un proyecto informático, el valor actual neto se evalúa para un período de vida útil de 5 años.

El cálculo del VAN se realiza con la siguiente fórmula:

$$\sum_{i=1}^n \frac{FC_i}{(1-k)^i} - I_0$$

Donde:

- n es el total de años de vida útil del proyecto, en este caso 5.
- i representa el año correspondiente.
- FC_i es el flujo de caja neto del i -ésimo año.
- k es la tasa de interés para los proyectos informáticos, que corresponde a un 5%
- I_0 es la inversión inicial.

En base a los flujos de caja presentados en la Tabla 6 se calcula el indicador VAN utilizando Excel, el cual entrega como resultado: \$ 1.676.336.435. Como el resultado de este indicador es positivo, el proyecto es económicamente factible.

Capítulo IV. Especificación de Requisitos

Los requisitos del sistema especifican qué es lo que el sistema debe hacer y sus propiedades esenciales y deseables [2]. Para el presente proyecto, los requisitos fueron especificados por el ingeniero agrónomo a cargo de la alimentación en Agrícola Mollendo, a través de varias entrevistas y muestras de avances parciales. Los requisitos se fueron refinando y completando con cada entrevista, como también, con la muestra de avances parciales de ciertas funcionalidades implementadas.

Al finalizar la etapa de captura y análisis de requisitos se obtuvieron tanto los requisitos funcionales como los no funcionales especificados a continuación.

4.1 Actores del Sistema

Ingeniero agrónomo a cargo de la alimentación: Es el encargado de todo el proceso de engorda de los vacunos. Es quien decide los compuestos que se comprarán y en qué cantidades. Por lo anterior, es el encargado de calcular las raciones alimenticias para los vacunos, por lo tanto será el único actor que se relacionará con el sistema.

4.2 Requisitos Funcionales

Al utilizar una metodología iterativa incremental, los requisitos se implementaron en diferentes incrementos, a continuación se detallarán los requisitos y en qué incremento dentro de la metodología utilizada se desarrollaron.

Primer incremento

El sistema deberá permitir al ingeniero agrónomo:

R1. Gestionar las etapas de engorda.

R.1.1. Agregar nuevas etapas de engorda, las cuales son utilizadas para calcular la optimización.

R.1.2 Eliminar etapas de engorda.

- R2.** Asignar el porcentaje de consumo de alimento, respecto al peso vivo⁹ del animal, por cada etapa de engorda.
- R3.** Asignar un rango de peso que defina el peso de ingreso y el peso de salida de cada etapa de engorda.
- R4.** Modificar la cantidad de animales que estarán presente en cada etapa de engorda. Esta información debe ser utilizada para realizar la optimización.
- R5.** Gestionar compuestos, tales como: ensilaje de maíz, coqueta, vinaza, bagazo de cebada, maíz grano húmedo, mezcla soprodi, paja de trigo, sal molido, urea, entre otros.
 - R.5.1 Eliminar un compuesto.
 - R.5.2 Agregar un compuesto, con sus respectivos aportes nutricionales.
 - R.5.3 Modificar todos los aspectos de los compuestos.
 - R.5.3.1. Modificar algunos o todos los aportes nutricionales de determinado compuesto.
 - R.5.3.2. Modificar el stock disponible de cada compuesto.
 - R.5.3.3. Modificar el precio de determinado compuesto.
- R6.** Gestionar restricciones de cantidad de compuesto para cada etapa, es decir agregar, modificar o eliminar restricciones de cantidad.
 - R.6.1 Modificar las restricciones de cantidad de compuesto por etapa, éstas pueden ser máximas o mínimas.
 - R.6.2 Eliminar restricciones de cantidad de compuesto por etapa.
 - R.6.3 Agregar restricciones de cantidad de compuesto por etapa, tanto mínimas como máximas. Las nuevas restricciones deben ser consideradas a la hora de realizar la optimización.
- R7.** Seleccionar los compuestos que participarán en cada etapa.
- R8.** Modificar requisitos mínimos nutricionales respecto a todos los aportes por cada etapa de engorda.

⁹ Peso vivo se refiere simplemente al peso del animal (cuando éste está vivo). En los rumiantes, la cantidad de alimento que éstos pueden ingerir se encuentra en función de su peso vivo, denominándose así: porcentaje de consumo de peso vivo.

R9. El sistema debe obtener las raciones por cada etapa que minimice los costos, a partir de una selección previa, de parte del usuario, de los compuestos que utilizará para cada etapa de engorda con sus respectivas restricciones.

R.9.1 Todas las variables involucradas en la creación de raciones (restricciones, compuestos, nutrientes, precio, stock, etapas, porcentaje consumo peso vivo, cantidad de animales por etapa y compuestos seleccionados por etapas) y los resultados de la optimización deberán ser almacenados en la base de datos.

Segundo incremento

El sistema deberá permitir al ingeniero agrónomo:

R10. El sistema deberá entregar reportes en formato Excel de las optimizaciones realizadas, tanto las históricas como las actuales. El formato de los reportes es similar al presentado en la Tabla 2.

R11. Cuando se ingrese al sistema, éste deberá solicitar un nombre de usuario y clave de acceso, los datos deberán ser verificados y de ser correctos permitir el acceso, de lo contrario permitirá ingresar la clave 3 veces. Después de 3 intentos erróneos (intentos por sesión), el programa se bloqueará 5 minutos. La siguiente vez, si después de los 3 intentos no ingresa un nombre de usuario y clave correcta, se bloqueará por 10 minutos. De esta forma el programa se bloquea 5 minutos por intento erróneo para cada sesión.

R12. El sistema deberá mostrar una pantalla resumen que contenga, por cada etapa: los compuestos que se utilizarán con su respectivo precio, stock y restricciones como también los requisitos mínimos nutricionales.

4.3 Requisitos No Funcionales

- El sistema debe tener una interfaz amistosa que permita al usuario utilizar el sistema sin mayor dificultad. Para desarrollar la interfaz, se creará un prototipo, el cual será

evaluado por el usuario y a partir de sus sugerencias y una metodología iterativa, se logrará crear una interfaz amigable.

- El tiempo de respuesta para la realización del cálculo de optimización no debe superar los 5 minutos.
- El computador que alojará el sistema debe cumplir con los siguientes requisitos mínimos:
 - Tener instalado JRE 1.6¹⁰ o superior.
 - Memoria RAM de 512MB (recomendado para una correcta ejecución)
 - 1GB de memoria virtual
 - Espacio en el disco duro de 2.26 GB (espacio requerido por el motor de base de datos Oracle mas la aplicación en sí).
 - Adaptador de video de 256 colores
 - Procesador con 200 MHz
 - Excel versión 2003 o superior, para visualizar los reportes.

4.4 Método de Optimización

Para la optimización de raciones se utilizará el método simplex debido a que el problema presentado cumple con las condiciones para ser resuelto por dicho método, esto es, las restricciones y función objetivo deben ser ecuaciones lineales (variables con exponente menor o igual que uno). Otra condición, es que, lo que se desea optimizar es sólo un factor, en este caso el precio de las raciones. Un elemento determinante en la elección de este método es que es capaz de entregar una solución, asegurando que la misma, es con certeza la óptima, si ésta existe. Además al utilizar el método simplex se obtienen resultados en un tiempo aceptable, considerando un problema de alta complejidad (gran cantidad de restricciones y variables).

A continuación se mostrará en mayor detalle el método simplex, para luego definir el modelo matemático generado para ser resuelto por dicho método.

¹⁰ JRE (Java Runtime Enviroment) es un conjunto de utilidades que permite ejecutar un programa Java.

4.4.1 Método Simplex

Es un proceso algebraico que permite resolver problemas de programación lineal.[3] Los problemas de programación lineal se relacionan con la optimización, a través de la maximización o minimización de una función, la cual, está sujeta a restricciones, que pueden ser de igualdad o desigualdad.

Se ha comprobado que el método simplex es muy eficiente para resolver problemas de optimización, debido a que se aplica en forma rutinaria, dándose a entender por rutinaria una única forma de resolverlo que es repetitiva y se basa principalmente en operaciones matemáticas como multiplicación y división en matrices, lo que permite que los computadores de hoy en día puedan resolver el método de una manera muy rápida. Los conceptos fundamentales del método simplex surgen de la geometría, donde ésta muestra como el método simplex opera, buscando el punto óptimo entre todas las soluciones factibles en los vértices.

En el caso de esta tesis el método simplex fue utilizado para realizar una minimización de costos en el cálculo de raciones alimenticias para vacunos wayu, y estará sujeto a todas las restricciones alimenticias y de stock que se encuentran inmersas en el problema.

4.4.2 Modelo matemático

Para dar solución al requisito número nueve (véase en la sección 4.2 “requisitos funcionales”) se debe, en primera instancia, plantear el problema a resolver. Como éste es un problema de optimización de raciones, se debe crear el modelo matemático, lo que se conoce como planteamiento del problema, para luego poder aplicar el método simplex y lograr obtener una solución. Es por lo anterior que a continuación presentaremos el modelo matemático que será resuelto por el modulo de optimización a desarrollar.

Función objetivo

$$Z = \sum_{i=1...C} \sum_{j=1}^n x_{ij} * \$P_i$$

Donde:

x_{ij} es la cantidad de kilogramos de un compuesto i que participa en una ración j . Pudiendo tomar valor cero, si el compuesto no fue seleccionado para participar en esa ración.

$\$P_i$ es el precio del compuesto i .

n es el número total de etapas ingresadas para el plan de alimentación.

N es la cantidad de nutrientes.

C es la cantidad total de compuestos.

Nota: para efectos del modelo “etapa” y “ración”, son la misma cosa. Ambas van de $j=1 \dots n$.

Sujeto a:

Nutrientes por ración

El número de nutrientes en una ración es fijo, debido a que al realizar raciones alimenticias para vacunos, los únicos nutrientes relevantes son los que se mencionaron anteriormente (materia seca, proteínas totales, fibra detergente neutro, etc.). La expresión (1) representa las restricciones nutricionales mínimas a las cuales se encuentra sujeta la función objetivo:

$$(1) \quad \sum_{i=1}^C a_{ki} * x_{ij} \geq r_{kj} \quad \forall j= 1 \dots n, \forall k= 1 \dots N$$

Donde:

a_{ki} es la cantidad del k -ésimo nutriente que aporta el compuesto i .

r_{kj} representa la restricción nutricional mínima del k -ésimo nutriente para la etapa j .

A partir de la expresión (1), se ejemplificará, en la Figura 3, las restricciones nutricionales mínimas a las que está sujeta la función objetivo.

$$\begin{array}{cccccc}
 \sum_{\forall i \in C_i} a_{1i} * x_{i1} & + & 0 & + \dots + & 0 & \geq r_{11} \\
 \sum_{\forall i \in C_i} a_{2i} * x_{i1} & + & 0 & + \dots + & 0 & \geq r_{21} \\
 \sum_{\forall i \in C_i} a_{3i} * x_{i1} & + & 0 & + \dots + & 0 & \geq r_{31} \\
 \vdots & & \vdots & & \vdots & \vdots \\
 \sum_{\forall i \in C_i} a_{ki} * x_{i1} & + & 0 & + \dots + & 0 & \geq r_{k1} \\
 0 & + & \sum_{\forall i \in C_i} a_{1i} * x_{i2} & + \dots + & 0 & \geq r_{12} \\
 0 & + & \sum_{\forall i \in C_i} a_{2i} * x_{i2} & + \dots + & 0 & \geq r_{22} \\
 0 & + & \sum_{\forall i \in C_i} a_{3i} * x_{i2} & + \dots + & 0 & \geq r_{32} \\
 \vdots & & \vdots & & \vdots & \vdots \\
 0 & + & \sum_{\forall i \in C_i} a_{ki} * x_{i2} & + \dots + & 0 & \geq r_{k2} \\
 0 & + & 0 & + \dots + & \sum_{\forall i \in C_i} a_{1i} * x_{ij} & \geq r_{1j} \\
 0 & + & 0 & + \dots + & \sum_{\forall i \in C_i} a_{2i} * x_{ij} & \geq r_{2j} \\
 0 & + & 0 & + \dots + & \sum_{\forall i \in C_i} a_{3i} * x_{ij} & \geq r_{3j} \\
 \vdots & & \vdots & & \vdots & \vdots \\
 0 & + & 0 & + \dots + & \sum_{\forall i \in C_i} a_{ki} * x_{ij} & \geq r_{kj}
 \end{array}$$

Figura 3. Matriz de restricciones.

Restricciones de stock para algunos compuestos.

$$\sum_{i=1}^n x_{ij} \leq R_i$$

Donde:

R_i es la cantidad en stock del compuesto i .

n es el total de compuestos con restricción de stock.

Restricciones de cantidad de compuestos por etapa.

$$\text{MIN}_{ij} \leq x_{ij} \leq \text{MAX}_{ij}$$

Donde:

MIN_{ij} es el límite mínimo para el compuesto i en la ración j.

MAX_{ij} es el límite máximo para el compuesto i en la ración j.

Capítulo V. Análisis

En este capítulo se presentará el diagrama de casos de uso que se llevó a cabo para el proyecto, a partir de los requisitos obtenidos en el capítulo anterior. Además se presentará la especificación de todos los casos de uso mostrados en el diagrama. Para finalizar este capítulo se mostrarán el modelo conceptual y los diagramas de secuencia desarrollados para este proyecto.

5.1 Diagrama de casos de uso

En la Figura 4 se presenta el diagrama de casos de uso para el sistema de optimización de raciones para vacunos wagyu. En este diagrama se agruparon los casos de uso según las distintas funcionalidades identificadas por el cliente. Cada grupo de casos de uso se encuentra dentro de un rectángulo azul con el correspondiente título que define la funcionalidad del conjunto.

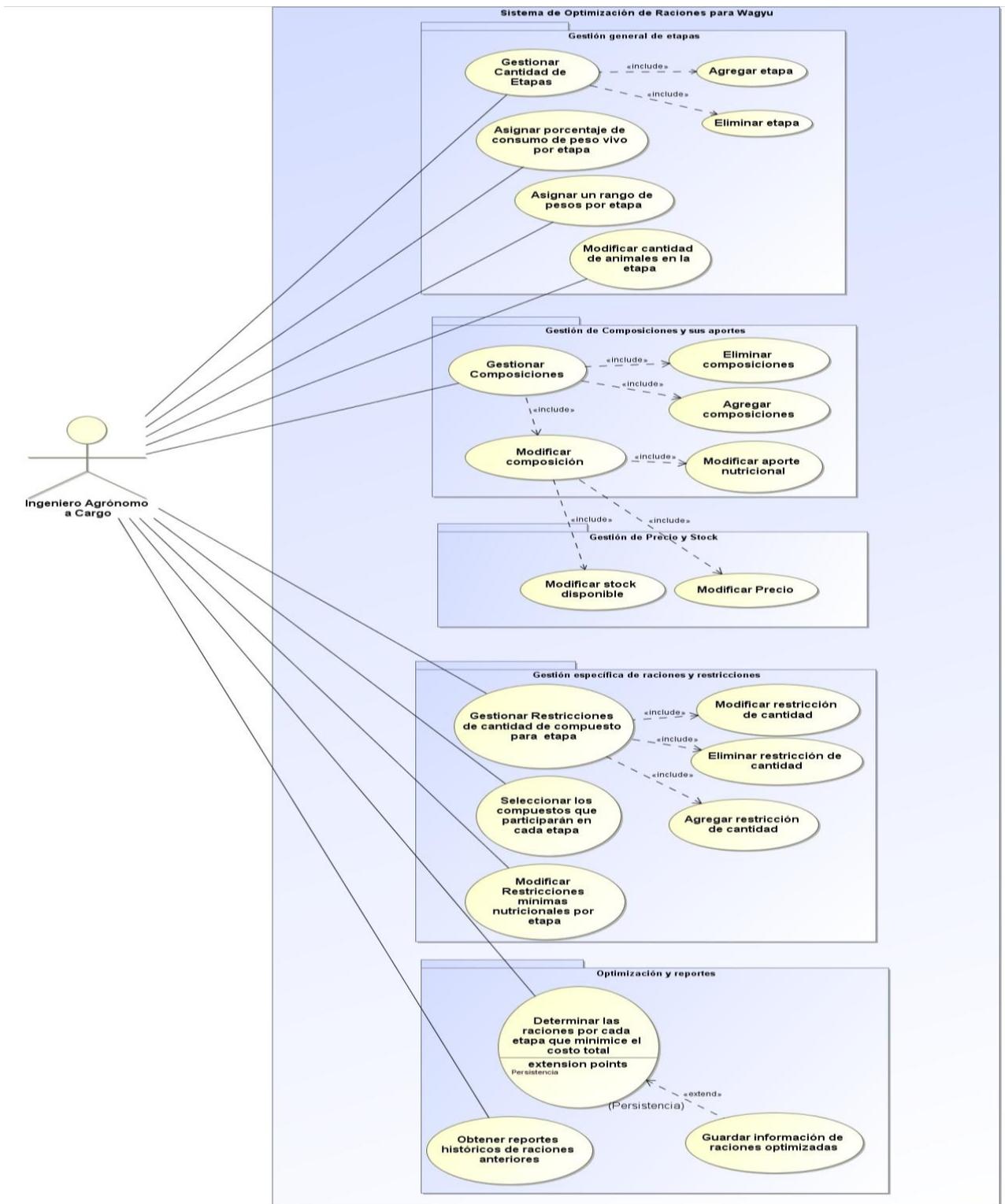


Figura 4. Diagrama de casos de uso.

5.2 Especificación de casos de uso

Las especificaciones de casos de uso o también llamados caso expandido de uso es una explicación detallada de los casos de uso de alto nivel. En éstos se muestra la interacción básica entre el actor y el sistema, permitiendo obtener un entendimiento más profundo y acabado de los procesos y requerimientos. Una de las características principales de las especificaciones de casos de uso, es que se desarrollan con un lenguaje coloquial que permite la fácil comprensión entre el analista y el cliente, ayudando en el completo entendimiento de los requisitos.[4]

A continuación se presentarán las especificaciones de casos de uso, clasificadas en las siguientes secciones:

- Gestión general de etapas
- Gestión de compuestos y sus aportes
- Gestión de precio y stock
- Gestión específica de raciones y restricciones
- Optimización y reportes
- Control de acceso y gestión de nombre de usuario y contraseña

5.2.1 Gestión General de Etapas

Número caso de uso	:	1.
Nombre caso de Uso	:	Gestionar Cantidad de etapas.
Breve descripción	:	El ingeniero a cargo agrega o elimina etapas según desee.
Actor	:	Ingeniero encargado de la alimentación.
Pre-condiciones	:	
Post-condiciones	:	Se agrega(n) o elimina(n) la(s) etapa(s) que el Ingeniero a cargo estima.
Flujo principal		
Ingeniero encargado de la alimentación		Sistema
1. Este caso de uso comienza cuando el Ingeniero a cargo desea gestionar la cantidad de etapas que se utilizarán para generar las raciones.		2. Muestra en pantalla las etapas con su respectivo nombre, rango de pesos, porcentaje de consumo de alimento, respecto al peso vivo del animal que se utilizará para el cálculo de la ración y la cantidad de animales en la etapa.
3. Selecciona la etapa que desea gestionar.		4. Muestra las opciones: a) agregar una etapa después de la seleccionada. b) eliminar la etapa seleccionada.
5. Selecciona una de las dos opciones: a) Agregar una etapa, ver caso de uso <i>agregar etapa</i> . b) Eliminar etapa, ver caso de uso <i>eliminar etapa</i> .		6. Muestra los cambios realizados.
Repetir 2-3-4-5-6 hasta que no se deseen agregar y/o eliminar más etapas.		
Flujo Alternativo		
2.a Error al cargar etapas desde la base de datos. 1. El sistema notifica el error y sugiere iniciar la base de datos.		

Número caso de uso	:	1.1
Nombre caso de Uso	:	Agregar Etapa
Breve descripción	:	El ingeniero a cargo, agrega una nueva etapa para que ésta sea considerada en la formulación de las raciones.
Actor	:	Ingeniero encargado de la alimentación.
Pre-condiciones	:	El ingeniero a cargo, ha seleccionado donde desea agregar la etapa.
Post-condiciones	:	Se agrega una nueva etapa con todos los datos necesarios para calcular la ración.
Flujo principal		
Ingeniero encargado de la alimentación		Sistema
1. Este caso de uso comienza cuando el Ingeniero a cargo desea agregar una nueva etapa para la formulación de las raciones.		2. El sistema muestra en la pantalla los campos vacíos para: <ul style="list-style-type: none"> a) Nombre de la etapa b) Porcentaje de consumo de peso vivo para la etapa. c) Rango de peso para la etapa. d) Cantidad de animales en la etapa.
3. Llena los campos vacios para generar la nueva etapa.		4. Agrega la etapa con toda la información correspondiente y la muestra en pantalla.
Flujo Alternativo		
4.a Valores inválidos para nueva etapa <ul style="list-style-type: none"> 1. El sistema notifica que alguno o todos los campos para crear una nueva etapa tienen valores no permitidos. Los valores no permitidos son: <ul style="list-style-type: none"> a) Nombre de la etapa repetido o en blanco. b) El porcentaje menor que cero o mayor que 100. c) Peso de ingreso mayor que el peso de salida, pesos con valores negativos o valores que no sean números. d) Cantidad de animales negativa o valores que no sean números. 2. Muestra un mensaje de error, vuelve a 2, pero con los campos erróneos en rojo para que sean ingresados nuevamente. 		

Número caso de uso	:	1.2
Nombre caso de Uso	:	Eliminar Etapa
Breve descripción	:	El ingeniero a cargo, elimina una etapa para que ésta no sea considerada en la formulación de las raciones.
Actor	:	Ingeniero encargado de la alimentación.
Pre-condiciones	:	Debe existir al menos una etapa ingresada. Se debe haber seleccionado la etapa que desea eliminar.
Post-condiciones	:	Se elimina correctamente la etapa seleccionada por el Ingeniero a cargo.
Flujo principal		
Ingeniero encargado de la alimentación		Sistema
1. Este caso de uso comienza cuando el Ingeniero a cargo desea eliminar una etapa.		3. El sistema muestra un mensaje que le pregunta al Ingeniero a cargo si está seguro que desea eliminar la etapa.
2. Presiona eliminar.		
4. Presiona "SI"		5. Elimina la etapa y muestra los cambios realizados.
Flujo Alternativo		
3.a Ingeniero no desea eliminar la etapa seleccionada.		
1. El ingeniero presiona "NO". Fin caso de uso.		

Número caso de uso	:	2.
Nombre caso de Uso	:	Asignar porcentaje de consumo de peso vivo por etapa.
Breve descripción	:	El Ingeniero a cargo de la alimentación, desea cambiar el porcentaje de peso vivo que se utilizará para determinar la cantidad máxima, en kilos, de la ración.
Actor	:	Ingeniero encargado de la alimentación.
Pre-condiciones	:	Debe existir por lo menos una etapa.
Post-condiciones	:	Se asigna un nuevo porcentaje de peso vivo que se considerará para el cálculo de la ración.
Flujo principal		
Ingeniero encargado de la alimentación		Sistema
<ol style="list-style-type: none"> Este caso de uso comienza cuando el Ingeniero a cargo desea modificar el porcentaje de consumo de peso vivo, para determinada etapa. Selecciona la etapa a la cual desea modificar el porcentaje de consumo de peso vivo. 		<ol style="list-style-type: none"> El sistema permite al Ingeniero a cargo, modificar el porcentaje de consumo de peso vivo para la etapa seleccionada.
<ol style="list-style-type: none"> Modifica el porcentaje asociado a la etapa. 		<ol style="list-style-type: none"> Verifica el valor ingresado y lo muestra en la interfaz.
Repetir pasos 2-3-4-5 hasta que no se deseen modificar más porcentajes de consumo de peso vivo.		
Flujo Alternativo		
<ol style="list-style-type: none"> 5. a Valor asignado al porcentaje es inválido. <ol style="list-style-type: none"> El sistema notifica que el porcentaje nuevo ingresado no se encuentra dentro del rango aceptado (mayor que 0% y menor que 100%). Volver a 3. 		

Número caso de uso	:	3.
Nombre caso de Uso	:	Asignar un rango de pesos por etapa.
Breve descripción	:	El ingeniero a cargo, puede reasignar nuevos pesos de ingreso y salida a la(s) etapa(s) que desee.
Actor	:	Ingeniero encargado de la alimentación.
Pre-condiciones	:	Debe existir por lo menos una etapa.
Post-condiciones	:	Los nuevos pesos de entrada y salida para la(s) etapa(s) seleccionada(s) se asignan correctamente.
Flujo principal		
Ingeniero encargado de la alimentación		Sistema
<ol style="list-style-type: none"> Este caso de uso comienza, cuando el Ingeniero a cargo desea modificar los pesos de entrada y/o salida para una determinada etapa. Selecciona la etapa para la cual desea modificar los pesos de entrada y/o salida. 		<ol style="list-style-type: none"> Permite que los campos donde aparecen los pesos de entrada y salida sean editables.
<ol style="list-style-type: none"> Ingresar nuevos valores para los campos correspondientes a los pesos de entrada y/o salida. 		<ol style="list-style-type: none"> Verifica y muestra los nuevos pesos de entrada y salida para la etapa.
Repetir los pasos 2-3-4-5 hasta que no se deseen modificar más pesos de entrada y/o salida.		
Flujo Alternativo		
<ol style="list-style-type: none"> El peso de entrada y/o salida no son válidos. <ol style="list-style-type: none"> Los valores para los pesos de entrada no son válidos, ya sea porque uno o ambos son negativos o porque el peso de ingreso es superior al peso de salida, por lo que el sistema mostrará un error. Volver a 4. 		

Número caso de uso	:	4.
Nombre caso de Uso	:	Modificar cantidad de animales por etapa.
Breve descripción	:	El usuario solicita al programa modificar la cantidad de animales para una determinada etapa.
Actores principales	:	Ingeniero encargado de la alimentación.
Actores secundarios	:	
Pre-condiciones	:	Debe existir por lo menos una etapa.
Post-condiciones	:	Se modifica la cantidad de animales en determinada etapa.
Flujo principal		
Ingeniero encargado de la alimentación		Sistema
1. Este caso de uso comienza cuando el ingeniero a cargo de la alimentación desea modificar la cantidad de animales por etapa.		2. Muestra al ingeniero a cargo las etapas existentes.
3. Selecciona una etapa para modificar la cantidad de animales en ésta. 4. Modifica el valor asociado a la cantidad de animales.		5. El sistema registra y muestra correctamente los cambios realizados.
Se realizan los pasos 2-3-4-5 hasta que no se deseen realizar más modificaciones.		
Flujo Alternativo		
5. a Cantidad de animales inválida		
1. El sistema muestra un error en la nueva cantidad de animales ingresada, debido a que se introdujo un valor negativo u otro dato no numérico.		
2. Volver al paso 2.		

5.2.2 Gestión de Compuestos y sus aportes

Número caso de uso	:	5.
Nombre caso de Uso	:	Gestionar compuesto.
Breve descripción	:	Modificar, eliminar o agregar compuestos dejando estos cambios guardados de forma persistente.
Actor	:	Ingeniero encargado de la alimentación.
Pre-condiciones	:	Para la modificación y eliminación de compuestos, la composición debe estar registrada.
Post-condiciones	:	La gestión realizada a las compuestos queda guardada de forma persistente.
Flujo principal		
Ingeniero encargado de la alimentación		Sistema
1. Este caso de uso comienza cuando el Ingeniero a cargo desea editar, eliminar o agregar un compuesto.		
2. Solicita al sistema los compuestos existentes.		3. Muestra todas los compuestos de manera estructurada y ordenadas, similar a como se hace en una planilla Excel.
4. Realiza una de las siguientes acciones: a) Modificar composición, ver caso de <i>uso modificar compuesto</i> . b) Eliminar composición, ver caso de <i>uso eliminar compuesto</i> . c) Agregar composición, ver caso de <i>uso agregar nueva compuesto</i> .		
Repetir paso 4 cuantas veces sea necesario.		

Número caso de uso	:	5.1
Nombre caso de Uso	:	Eliminar compuesto.
Breve descripción	:	Eliminar un compuesto del sistema.
Actor	:	Ingeniero encargado de la alimentación.
Pre-condiciones	:	Deben existir compuestos previamente registrados en el sistema.
Post-condiciones	:	Se eliminan, además del compuesto, todos los aportes nutricionales del mismo.
Flujo principal		
Ingeniero encargado de la alimentación		Sistema
1. Este caso de uso comienza cuando el Ingeniero a cargo selecciona eliminar un compuesto.		2. Muestra una lista con los compuestos previamente ingresadas.
3. Selecciona el compuesto a eliminar. 4. Solicita al sistema la eliminación de la composición.		5. Muestra un mensaje de confirmación.
6. Confirma la eliminación.		7. Elimina el compuesto.
Repetir los pasos 2-3-4-5-6 mientras desee eliminar más compuestos		
Flujo Alternativo		
6. a Cancelación de eliminación.		
1. No desea eliminar el compuesto seleccionado ir a 2.		

Número caso de uso	:	5.2
Nombre caso de Uso	:	Agregar nueva compuesto.
Breve descripción	:	Registrar un nuevo compuesto, para ello se ingresan los aportes nutricionales de ésta según se especifican en el informe de laboratorio.
Actor	:	Ingeniero encargado de la alimentación.
Pre-condiciones	:	
Post-condiciones	:	Se registran en el sistema todos los aportes nutricionales del compuesto (con valores válidos), según el laboratorio.
Flujo principal		
Ingeniero encargado de la alimentación		Sistema
1. Este caso de uso comienza cuando el Ingeniero a cargo selecciona agregar compuesto.		2. Muestra una nueva fila con los campos necesarios para agregar un nuevo compuesto.
3. Ingresa el nombre del compuesto y los aportes nutricionales correspondientes.		4. Verifica que el nombre no esté repetido y que los valores de los aportes nutricionales para éste, se encuentren dentro del dominio predefinido.
Repetir pasos 1-2-3-4 mientras se desee seguir ingresando nuevos compuestos.		
Fin caso de uso		
Flujo Alternativo		
<p>4.a Valor de aporte nutricional incorrecto.</p> <p>1. el sistema notifica al Ingeniero que el valor del aporte nutricional no se encuentra en el rango predefinido. El rango predefinido son números reales positivos.</p> <p>2. volver al paso 2.</p> <p>4. b Nombre de compuesto repetido.</p> <p>1. el sistema notifica al Ingeniero que un compuesto con ese nombre ya ha sido ingresado.</p> <p>2. volver al paso 2</p>		

Número caso de uso	:	5.3
Nombre caso de Uso	:	Modificar compuesto.
Breve descripción	:	Editar una, algunas o todas las características de un compuesto.
Actor	:	Ingeniero encargado de la alimentación.
Pre-condiciones	:	Deben existir compuestos previamente registradas en el sistema.
Post-condiciones	:	Los cambios deben quedar guardados de manera persistente si el Ingeniero a cargo así lo desea.
Flujo principal		
Ingeniero encargado de la alimentación		Sistema
<p>1. Este caso de uso comienza cuando el Ingeniero a cargo desea editar un compuesto.</p>		<p>2. El sistema ofrece las siguientes opciones:</p> <ul style="list-style-type: none"> a) Modificar aporte nutricional. b) Modificar precio. c) Modificar stock disponible.
<p>3. Selecciona una de las siguientes opciones:</p> <ul style="list-style-type: none"> a) Modificar el valor de uno o más nutrientes que conforman un compuesto, ver caso de uso <i>Modificar aporte nutricional</i>. b) Modificar el precio de algún compuesto, ver caso de uso <i>Modificar Precio</i>. c) Modificar stock disponible del compuesto, ver caso de uso <i>Modificar Stock Disponible</i>. <p>Repetir paso 2 cuantas veces sea necesario.</p>		

Número caso de uso	:	5.3.1
Nombre caso de Uso	:	Modificar aporte nutricional.
Breve descripción	:	Editar una, algunas o todos los valores que componen el aporte nutricional de un compuesto.
Actor	:	Ingeniero encargado de la alimentación.
Pre-condiciones	:	Deben existir compuestos previamente registradas en el sistema.
Post-condiciones	:	Los cambios realizados tienen valores correctos.
Flujo principal		
Ingeniero encargado de la alimentación		Sistema
1. Este caso de uso comienza cuando el Ingeniero a cargo desea modificar el valor de un campo, del aporte nutricional de un compuesto.		2. El sistema muestra los compuestos previamente ingresados y sus correspondientes aportes nutricionales.
3. Selecciona el valor del compuesto correspondiente al aporte nutricional que desea modificar.		5. El sistema muestra el cambio realizado en el campo correspondiente.
4. Modifica el valor.		
Flujo Alternativo		
5a. Error en el valor ingresado.		
1. El sistema notifica al Ingeniero que el valor que ingresó es inválido. Los valores inválidos, son aquellos números negativos o datos no numéricos. Volver a 4.		

5.2.3 Gestión de Precio y Stock

Número caso de uso	:	5.3.2
Nombre caso de Uso	:	Modificar precio.
Breve descripción	:	Editar el precio correspondiente a un compuesto.
Actor	:	Ingeniero encargado de la alimentación.
Pre-condiciones	:	Deben existir compuestos previamente registradas en el sistema.
Post-condiciones	:	Los cambios realizados tienen valores coherentes.
Flujo principal		
Ingeniero encargado de la alimentación		Sistema
1. Este caso de uso comienza cuando el Ingeniero a cargo desea modificar el valor del precio correspondiente a un compuesto.		2. El sistema muestra los compuestos previamente ingresados y sus correspondientes precios.
3. Selecciona el compuesto al cual desea modificar su precio.		5. El sistema muestra el cambio realizado en el campo correspondiente.
4. Modifica el valor.		
Flujo Alternativo		
5a. Error en el valor ingresado.		
2. El sistema notifica al Ingeniero que el valor que ingresó es inválido. Los valores inválidos son números negativos o datos no numéricos. Volver a 4.		

Número caso de uso	:	5.3.3
Nombre caso de Uso	:	Modificar Stock.
Breve descripción	:	Editar el stock correspondiente a un compuesto.
Actor	:	Ingeniero encargado de la alimentación.
Pre-condiciones	:	Deben existir compuestos previamente registrados en el sistema.
Post-condiciones	:	Los cambios realizados tienen valores coherentes.
Flujo principal		
Ingeniero encargado de la alimentación		Sistema
1. Este caso de uso comienza cuando el Ingeniero a cargo desea modificar la cantidad disponible (stock) correspondiente a un compuesto.		2. El sistema muestra los compuestos previamente ingresados y su correspondiente stock.
3. Selecciona el compuesto al cual desea modificar su stock.		5. El sistema muestra el cambio realizado en el campo correspondiente.
4. Modifica el valor.		
Flujo Alternativo		
5a. Error en el valor ingresado.		
3. El sistema notifica al Ingeniero que el valor que ingresó es inválido. Los valores inválidos son números negativos o datos no numéricos. Volver a 4.		

5.2.4 Gestión Específica de Raciones y Restricciones

Número caso de uso	:	6.
Nombre caso de Uso	:	Gestionar Restricciones de cantidad de compuesto para etapa.
Breve descripción	:	El usuario solicita al programa modificar, agregar o eliminar alguna restricción de cantidad de un determinado compuesto para la etapa seleccionada.
Actores principales	:	Ingeniero encargado de la alimentación.
Actores secundarios	:	
Pre-condiciones	:	Haber seleccionado la etapa a la cual se desea gestionar sus restricciones y también los compuestos tentativos a participar en la ración.
Post-condiciones	:	El compuesto se encuentra restringido, en cuanto a cantidad, para la etapa seleccionada.
Flujo principal		
Ingeniero encargado de la alimentación		Sistema
1. Este caso de uso comienza cuando el ingeniero a cargo de alimentación, desea imponer una restricción para un compuesto en determinada etapa.		2. Muestra al ingeniero a cargo de la alimentación, los compuestos que ha seleccionado para que formen parte de la ración para la etapa.
3. El Ingeniero a cargo de la alimentación puede realizar una de las siguientes acciones: a) Modificar una restricción de cantidad, ver caso de uso <i>Modificar restricción de cantidad</i> . b) Eliminar una restricción de cantidad, ver caso de uso <i>Eliminar restricción de cantidad</i> . c) Agregar una restricción de cantidad, ver caso de uso <i>Agregar restricción de cantidad</i> .		4. El sistema registra y muestra correctamente los cambios realizados.
Repetir pasos 1-2-3-4 mientras se desee seguir gestionando restricciones.		

Número caso de uso	:	6.1
Nombre caso de Uso	:	Modificar restricciones de cantidad.
Breve descripción	:	El usuario solicita al programa, modificar una restricción impuesta previamente a un determinado compuesto.
Actores principales	:	Ingeniero encargado de la alimentación.
Actores secundarios	:	
Pre-condiciones	:	Debe haber, al menos, una restricción impuesta en el compuesto para que ésta pueda ser modificada.
Post-condiciones	:	Se cambia la restricción para un determinado compuesto según lo desea el Ingeniero a cargo.
Flujo principal		
Ingeniero encargado de la alimentación		Sistema
1. Este caso de uso comienza cuando el ingeniero a cargo de la alimentación, desea cambiar una restricción impuesta a un compuesto, en determinada etapa.		2. Muestra al ingeniero a cargo una lista con los compuestos y sus respectivas restricciones.
3. Selecciona una restricción que desee cambiar.		5. El sistema registra y muestra correctamente los cambios realizados.
4. Cambiar el valor y/o tipo de restricción (\geq ó \leq) asignado al compuesto.		
Se realizan los pasos 2-3-4-5 hasta que no se deseen efectuar más modificaciones.		
Flujo Alternativo		
5. a Restricción inválida		
1. El sistema muestra un error en la nueva restricción ya ingresada, debido a una incoherencia con otras restricciones o por un valor inválido. Los valores inválidos son números negativos o datos no numéricos.		
2. Volver al paso 2.		

Número caso de uso	:	6.2
Nombre caso de Uso	:	Eliminar restricciones de cantidad.
Breve descripción	:	El usuario solicita al programa eliminar una restricción impuesta previamente, a un determinado compuesto.
Actores principales	:	Ingeniero encargado de la alimentación.
Actores secundarios	:	
Pre-condiciones	:	Debe haber, al menos, una restricción impuesta en el compuesto para que ésta pueda ser eliminada.
Post-condiciones	:	Se eliminará la restricción para un determinado compuesto, según lo desea el Ingeniero a cargo.
Flujo principal		
Ingeniero encargado de la alimentación		Sistema
1. Este caso de uso comienza, cuando el ingeniero a cargo de la alimentación, desea eliminar una restricción impuesta a un compuesto en determinada etapa.		2. Muestra al ingeniero a cargo una lista con los compuestos y sus respectivas restricciones.
3. Selecciona la restricción que desee eliminar.		4. El sistema registra y muestra correctamente los cambios realizados
Se realizan los pasos 2-3-4-5-6 hasta que no se deseen realizar más modificaciones.		

Número caso de uso	:	6.3
Nombre caso de Uso	:	Agregar restricciones de cantidad.
Breve descripción	:	El usuario solicita al programa agregar una nueva restricción a imponer a un determinado compuesto.
Actores principales	:	Ingeniero encargado de la alimentación.
Actores secundarios	:	
Pre-condiciones	:	Debe haber, seleccionado la etapa a la cual desea agregarle una restricción de cantidad de alimento. Debe haber, al menos, un compuesto seleccionado, para que forme parte de la ración para la etapa seleccionada.
Post-condiciones	:	Se agrega la restricción para un determinado compuesto en la etapa seleccionada.
Flujo principal		
Ingeniero encargado de la alimentación		Sistema
1. Este caso de uso comienza cuando el ingeniero a cargo de la alimentación, desea agregar una restricción a un compuesto en determinada etapa.		2. Muestra al ingeniero a cargo una lista con los compuestos seleccionados para la etapa en la que se encuentra trabajando.
3. Selecciona el compuesto al cual desea agregar la restricción.		4. Muestra una interfaz a través de la cual el Ingeniero podrá agregar un valor y el tipo de restricción (\geq ó \leq) para el compuesto.
5. Ingresar un valor y un tipo de restricción para el compuesto.		6. Verifica la restricción y la muestra en la interfaz.
Flujo Alternativo		
5. a Restricción inválida		
1. El sistema muestra un error en la nueva restricción ya ingresada, debido a una incoherencia con otras restricciones o por un valor inválido. Los valores inválidos son números negativos o datos no numéricos.		
1. Volver al paso 4.		

Número caso de uso	:	7.
Nombre caso de Uso	:	Seleccionar los compuestos que participan en cada etapa.
Breve descripción	:	El usuario selecciona los compuestos tentativos a participar en cada etapa de engorda.
Actores principales	:	Ingeniero encargado de la alimentación.
Actores secundarios	:	
Pre-condiciones	:	Deben existir compuestos ingresados en el sistema. La etapa para la cual se desea seleccionar compuestos, debe ser previamente escogida.
Post-condiciones	:	Se seleccionan los compuestos que participarán en cada etapa de engorda, para la futura realización de la optimización.
Flujo principal		
Ingeniero encargado de la alimentación		Sistema
1. Este caso de uso comienza cuando el ingeniero a cargo de la alimentación, desea seleccionar los compuestos que participaran en determinada etapa, para la generación de raciones.		2. Muestra al ingeniero a cargo una lista con todos los compuestos disponibles a utilizar.
3. Selecciona los compuestos que desee utilizar para determinada etapa.		4. El sistema registra y muestra correctamente los cambios realizados.
Se realizan los pasos 2-3-4 hasta que no se deseen seleccionar más compuestos.		

Número caso de uso	:	8.
Nombre caso de Uso	:	Modificar Restricciones Mínimas Nutricionales por etapa.
Breve descripción	:	Editar una restricción de un aporte nutricional para una determinada etapa.
Actor	:	Ingeniero encargado de la alimentación.
Pre-condiciones	:	Deben existir restricciones previamente registradas, para la etapa, en el sistema.
Post-condiciones	:	El sistema registra el valor cambiado para la(s) restricción(es) elegida(s).
Flujo principal		
Ingeniero encargado de la alimentación		Sistema
<ol style="list-style-type: none"> Este caso de uso comienza cuando el Ingeniero a cargo de la alimentación desea gestionar las restricciones asociadas a los aportes nutricionales, para la creación de las raciones. El ingeniero a cargo de la alimentación selecciona la etapa, para la cual desea gestionar las restricciones de los aportes nutricionales 		<ol style="list-style-type: none"> Muestra una interfaz, con los espacios que corresponden a las restricciones nutricionales.
<ol style="list-style-type: none"> Selecciona el campo correspondiente al nutriente al cual, desea modificar su restricción. 		<ol style="list-style-type: none"> Permite que el campo sea modificable.
<ol style="list-style-type: none"> Ingresar el nuevo valor. 		<ol style="list-style-type: none"> Verifica que el valor esté dentro del dominio de las restricciones.
Repetir los pasos 2-3-4-5 hasta que no se deseen editar más campos.		
Flujo Alternativo		
<ol style="list-style-type: none"> valor incorrecto <ol style="list-style-type: none"> el sistema notifica que el valor está incorrecto. Los valores incorrectos son números negativos o datos no numéricos. Volver al paso 4. 		

5.2.5 Optimización y Reportes

Número caso de uso	:	9.
Nombre caso de Uso	:	Determinar las raciones por cada etapa que minimice el costo total.
Breve descripción	:	El usuario solicita al programa la obtención de las raciones que minimicen los costos para cada etapa a partir de una selección de compuestos.
Actores principales	:	Ingeniero encargado de la alimentación.
Actores secundarios	:	
Pre-condiciones	:	Debe estar registrada la información sobre los compuestos y sus correspondientes aportes nutricionales. Deben haber sido ingresadas las restricciones mínimas nutricionales de cada etapa. Se deben haber seleccionado los compuestos a participar en cada etapa de engorda.
Post-condiciones	:	El cálculo de la optimización fue realizado exitosamente o el programa indica que no es posible encontrar una solución para esa combinación de compuestos.

Flujo principal

Ingeniero encargado de la alimentación	Sistema
1. Este caso de uso comienza cuando el ingeniero a cargo de la alimentación desea determinar las raciones por cada etapa que minimicen el costo total.	2. Muestra un menú al ingeniero a cargo de la alimentación, con la opción optimizar ración.
3. Escoge la opción de optimizar ración.	4. Realiza la optimización e indica al ingeniero a cargo de la alimentación, cual es la cantidad de cada compuesto que debe colocar en la ración.
	5. Despliega la opción de guardar la optimización realizada.
6. Selecciona guardar el cálculo realizado	7. Guarda la optimización realizada.
	8. Muestra botón que permite obtener el reporte en EXCEL.
9. Presiona el botón y selecciona el plan de alimentación del cual desea obtener el reporte.	10. Genera y muestra el reporte.

Flujo Alternativo

4. a El sistema no puede realizar el cálculo de la minimización de los costos, porque el problema no tiene solución. 1. El sistema muestra un mensaje al usuario.
4. b El sistema no puede realizar el cálculo de la minimización de los costos, porque el ingeniero no seleccionó ningún compuesto. 1. Volver al paso 4
6. a El ingeniero a cargo no desea guardar la optimización realizada. 1. fin de caso de uso
9. a El ingeniero no desea un reporte. 1. El ingeniero no presiona el botón "Reportes en Excel".

Número caso de uso	:	10.
Nombre caso de Uso	:	Obtener reporte histórico de planes de alimentación anteriores.
Breve descripción	:	Generar un reporte en formato Excel de acuerdo a las raciones óptimas calculadas anteriormente.
Actores principales	:	Ingeniero a cargo de la Alimentación.
Actores secundarios	:	
Pre-condiciones	:	Debe haber al menos una optimización guardada.
Post-condiciones	:	Se genera un archivo Excel con el plan de alimentación seleccionado.
Flujo principal		
Ingeniero encargado de la alimentación		Sistema
1. Este caso de uso comienza cuando el ingeniero a cargo de alimentación desea obtener un reporte de una optimización realizada anteriormente.		2. Muestra todos los planes de alimentación realizados anteriormente, con su nombre y fecha.
3. Selecciona un plan de alimentación.		
		4. Obtiene la información referente al cálculo realizado en el caso de uso <i>Determinar las raciones por cada etapa que minimice el costo total:</i> - Identificador - Fecha -Restricciones de cantidad ingresadas Además, obtiene los valores calculados: - Cantidad de cada alimento a utilizar para la raciones en cada etapa de engorda.
		5. Crea el archivo Excel con los valores y la información obtenida.
		6. Muestra la ruta donde se guardó el archivo generado.
Flujo Alternativo		
2. a No hay cálculos de raciones guardados.		
1. El sistema muestra un mensaje indicando que no existen optimizaciones previas.		

5.2.6 Control de Acceso y Gestión de nombre de usuario y contraseña

Número caso de uso	:	11.
Nombre caso de Uso	:	Control de acceso al iniciar el programa.
Breve descripción	:	El programa verifica la autenticidad del usuario a través de la utilización de un nombre de usuario y contraseña.
Actores principales	:	Ingeniero a cargo de la Alimentación.
Actores secundarios	:	
Pre-condiciones	:	Se debe haber registrado un nombre de usuario y contraseña previamente en la base de datos.
Post-condiciones	:	Se permite el uso del programa si el nombre de usuario y contraseña son correctos, se cierra el programa en caso contrario.
Flujo principal		
Ingeniero encargado de la alimentación		Sistema
1. Este caso de uso comienza cuando el ingeniero a cargo de alimentación desea ingresar al programa de Optimización de Raciones para vacunos Wagyu.		2. El programa carga la información necesaria de la base de datos y muestra una pantalla con los campos nombre de usuario y contraseña.
3. El usuario ingresa el nombre de usuario y contraseña en los campos señalados.		4. El programa verifica que el nombre de usuario y contraseña sean correcto.
		5. El nombre de usuario y contraseña es efectivamente correcto.
		6. El programa se inicia normalmente.
Flujo Alternativo		
2. a Se ha intentado ingresar previamente al programa sin éxito y no ha pasado el tiempo de espera correspondiente a: 5 minutos por sesión errada.		
1. Se muestra un mensaje, indicando que el programa se cerrará, debido a que no ha pasado el tiempo necesario. El programa muestra una aproximación de cuántos minutos se debe esperar, para ingresar al programa de forma normal.		
5. a El nombre de usuario y contraseña no son correctos.		
1. El programa muestra un mensaje, indicando que el nombre de usuario y contraseña son incorrectos. Se vuelve a 2 únicamente, si el usuario no ha intentado ingresar un nombre de usuario y contraseña válidos, 3 veces en esta sesión.		
5. b El nombre de usuario y contraseña no son correctos y ya se ha intentado ingresar sin éxito 3 veces en esta sesión.		
1. El programa muestra un mensaje, donde indica que se cerrará por 5 minutos multiplicado por la cantidad de sesiones erradas (5 minutos si es la primera vez, 10 la segunda 15 la tercera, etc.).		

Número caso de uso	:	12.
Nombre caso de Uso	:	Cambiar el nombre de usuario y/o contraseña.
Breve descripción	:	El programa permite cambiar el nombre de usuario y contraseña.
Actores principales	:	Ingeniero a cargo de la Alimentación.
Actores secundarios	:	
Pre-condiciones	:	El ingeniero a cargo debe haber ingresado (autenticado) correctamente al programa.
Post-condiciones	:	Se permite el cambio de nombre de usuario y contraseña, datos que serán utilizados para ingresar al programa en la próxima sesión.
Flujo principal		
Ingeniero encargado de la alimentación		Sistema
1. Este caso de uso comienza cuando el ingeniero a cargo de alimentación desea cambiar el nombre de usuario y/o contraseña.		2. El programa muestra una pantalla que permite la gestión del nombre de usuario y contraseña.
3. El usuario ingresa un nuevo nombre de usuario		
4. El usuario ingresa una nueva contraseña en los campos señalados, la que debe repetir 2 veces.		5. El programa verifica que el nombre de usuario no esté en blanco, en caso de que quiera cambiar el nombre de usuario.
		6. El programa verifica que la nueva contraseña no esté en blanco y además verifica que los 2 campos donde se ingresa la contraseña tengan la misma contraseña.
		7. El programa permite el cambio de usuario y/o contraseña.
Flujo Alternativo		
5. a Se ha ingresado un nombre de usuario en blanco. 1. Se muestra un mensaje, indicando que el nombre de usuario está en blanco, por lo que el programa no cambió el nombre de usuario.		
6. a Se ha ingresado una contraseña vacía. 1. El programa muestra un mensaje, indicando que la contraseña no puede ser vacía, por lo que la nueva contraseña no se asignará.		
6. b Los campos para ingresar la contraseña no tienen el mismo contenido (no tienen la misma contraseña). 2. El programa indica que las contraseñas puestas en ambos campos no coinciden, por lo que no se puede asignar una nueva contraseña.		

5.3 Modelo Conceptual

Un modelo conceptual se enfoca principalmente en describir el dominio de un problema real, por lo que no es un modelo para el diseño del software, por lo anterior no es conveniente incluir elementos como bases de datos, interfaces gráficas, entre otros conceptos relacionados directamente al diseño del software en sí [4].

En la Figura 5 se puede apreciar, el modelo conceptual que representa el problema de optimización de raciones para vacunos wagyu. A continuación se brindará una explicación detallada del modelo.

5.3.1 Descripción del modelo conceptual

El concepto “Etapa” representa a las etapas de engorda en las que deben ser clasificados los animales, de acuerdo a su peso de ingreso. La etapa tiene un nombre, peso de ingreso, peso salida, porcentaje de consumo de peso vivo y cantidad de animales.

Una “Ración” tiene un único atributo llamado “id ración”, un ración es generada para una etapa de engorda, específica, en un plan de alimentación. Es por lo anterior que existe la relación “generada para” entre Ración y Etapa, donde una ración es generada sólo para una etapa y a una etapa se le pueden haber generado varias raciones, en distintos planes de alimentación.

Una “Restricción de Stock” posee un único atributo “cantidad/disponibilidad”, este atributo representa a la cantidad disponible, de determinado compuesto, que participará en un plan de alimentación. Un plan de alimentación puede estar limitado por cero o muchas restricciones de stock, una restricción de stock, limita sólo a un plan de alimentación. Una restricción de stock impone un límite en la cantidad de cierto compuesto que participará en un plan de alimentación.

Un “Compuesto” posee cero o muchos nutrientes y un nutriente forma parte de uno o muchos compuestos. Esta relación, como puede apreciarse, es de 0...* a 0...*, cuando

esto ocurre, se debe crear una tabla a partir de esta relación, sin embargo debido a que se está presentando un modelo conceptual y éste no es tan cercano a la implementación, la tabla será obviada, sin embargo, se incluirá en el diagrama de clases que se mostrará en el capítulo de diseño.

La relación que existe entre un compuesto y una ración, es que un compuesto participa en cero o muchas raciones y en una ración participan cero o muchos compuestos, como puede apreciarse sucede lo mismo que con la relación compuesto nutriente, del párrafo anterior.

Una “RestriccionCompuesto” limita a un sólo compuesto y un compuesto puede estar limitado por cero o muchas restricciones de compuestos, en distintas raciones. La diferencia que existe entre este concepto y la “Restricción de Stock”, es que una restricción de compuesto es específica para una ración, es decir, una restricción de compuesto se puede aplicar a una ración y una ración puede ser limitada por cero o muchas restricciones de compuesto.

Una “RestriccionNutricionalMinima” es una limitación de cantidad de un nutriente, esto quiere decir que una restricción nutricional mínima es específica para un solo nutriente, un nutriente puede tener una o muchas restricciones nutricionales mínimas, para distintas raciones. Además una ración está sujeta a una o muchas restricciones nutricionales mínimas y una restricción nutricional mínima es para una única ración.

Un plan de alimentación es creado al realizar una optimización. Un plan de alimentación está compuesto por una o muchas raciones, estas raciones son generadas para una etapa específica; además un plan de alimentación está limitado por restricciones de stock asignadas a determinados compuestos que participan en el mismo plan.

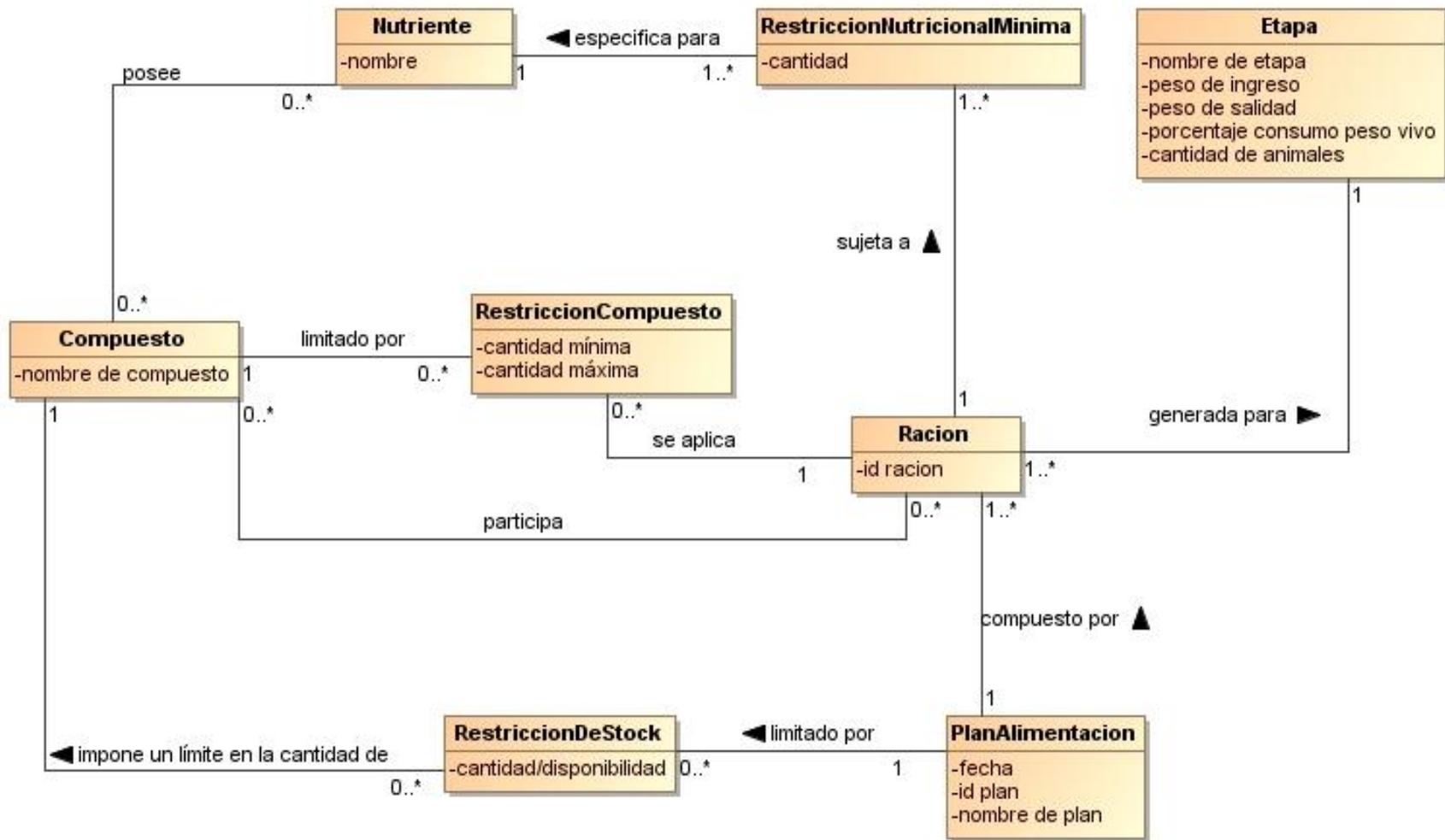


Figura 5. Modelo conceptual para optimización de raciones para vacunos wagyu.

5.4 Diagrama de Secuencia

El diagrama de secuencia es una representación que muestra, en determinado escenario de un caso de uso, los eventos generados por actores externos, su orden y los eventos internos del sistema[4]. Los diagramas de secuencia se utilizan principalmente para detallar de forma gráfica y temporal la comunicación que existe entre los objetos de una aplicación, para determinado caso de uso y también entre el actor y el sistema, dependiendo del nivel al que esté orientado el diagrama.

Los diagramas de secuencia desarrollados para este proyecto se encuentran en el Anexo A, el cual, está disponible en la copia digital de este informe. Se han seleccionado los diagramas más significativos, que representan al resto, en la forma y estilo de realizar las interacciones.

Capítulo VI. Diseño

En este capítulo se presentará un modelo de la arquitectura del sistema, donde se mostrará un detalle de cada parte que lo compone. También se dará a conocer una descripción de qué es un diagrama de clases, un modelo entidad relación e interfaz de usuario, para luego presentar los modelos respectivos desarrollados para el proyecto.

6.1 Arquitectura del sistema

En la Figura 6 se presenta a grandes rasgos la arquitectura del sistema.

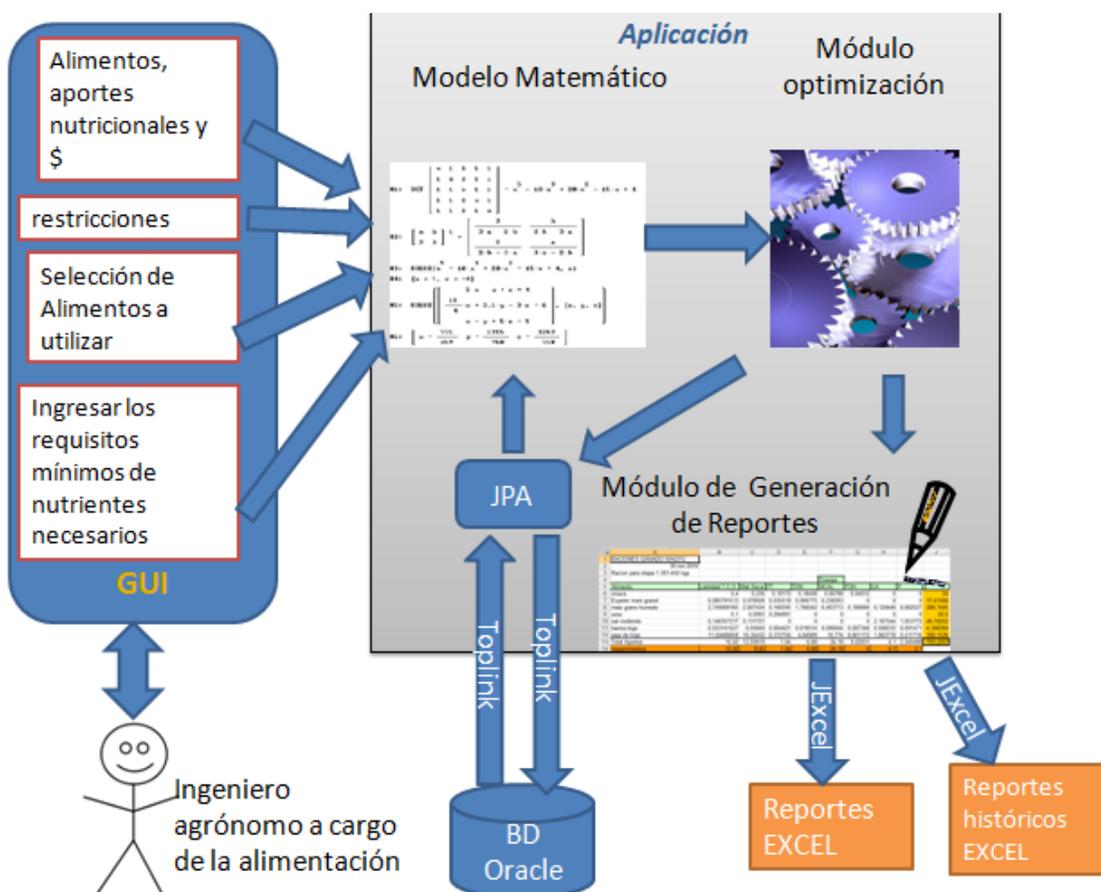


Figura 6. Arquitectura básica del sistema.

A continuación se presentará una breve descripción de cada módulo que compone la aplicación.

GUI: representa la interfaz gráfica explicada más adelante en “Interfaz de Usuario”. La interfaz gráfica es el fragmento del software, con el cual, interactúa el usuario, es decir es la parte “visual” del programa. Aquí el Ingeniero agrónomo a cargo de la alimentación ingresa información como restricciones, aportes nutricionales, precio y stock disponible de los compuestos; selecciona los compuestos a utilizar en cada etapa, etc.

Modelo Matemático: toda la información capturada en la GUI, en conjunto con la información que se tiene en la base de datos, se transforma para crear el modelo matemático mostrado en la sección 4.4 “modelo matemático” del capítulo IV, donde se genera una función objetivo y se limita con restricciones que se deben cumplir.

Módulo de Optimización: El módulo de optimización recibe el modelo matemático y lo resuelve utilizando el método simplex. Si las variables en la función objetivo adquieren valores que se ajustan a las restricciones, entonces el problema tiene solución, por lo que se despliega en pantalla y se ofrecerá la opción de guardar la optimización, en conjunto, con guardar el plan de alimentación recién generado. Si el problema no tiene solución, se despliega un mensaje que indica esta situación, por lo que el usuario deberá hacer algún cambio, que generalmente implica ser menos restrictivo con las cantidades y requisitos mínimos, para que el problema tenga solución.

JPA: Son las siglas en inglés para Java Persistence API¹¹, es la API de persistencia desarrollada para Java EE e incluida en el estándar EJB3. El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos, como así pasaba con EJB2, y permitir usar objetos regulares (conocidos como POJOs). Una de las implementaciones específicas de JPA es *toplink* que permite trabajar con el gestor de bases de dato Oracle.

Módulo de Generación de Reportes: El módulo de generación de reportes se encarga de crear un archivo Excel con toda la información del plan de alimentación, dándole un nombre al archivo y guardándolo donde el usuario especifique. El módulo de generación de reportes obtiene reportes, tanto para la ración actualmente calculada, como para raciones

¹¹ interfaz de programación de aplicaciones o API (del inglés Application Programming Interface) es el conjunto de métodos que ofrece cierta librería para ser utilizado por otro software como una capa de abstracción.

históricas, calculadas anteriormente. Para generar los reportes en Excel, se utilizó la API de código abierto denominada *JExcel*, la cual permite leer, escribir y modificar archivos Excel de manera dinámica.

Reportes Excel: Todos los reportes son generados por el módulo de generación de reportes. Cada reporte Excel contiene pestañas, en cada pestaña se encuentra la ración correspondiente a una etapa, también se incluyen los aportes de los compuestos participantes, y el precio final de la ración. Se muestran además en el reporte, la suma de los aportes por nutriente de la ración generada, para poder compararla con los requisitos mínimos que se deben cumplir, también, indicados en la pestaña correspondiente.

Base de datos Oracle: Oracle es un gestor de base de datos. La versión utilizada es Oracle 10g, la cual, tiene una versión de uso libre y gratuita, que fue la utilizada para interactuar con el programa. Esta versión, al ser gratuita, tiene limitaciones en cuanto a capacidad de almacenamiento, proceso y funcionalidad. Sin embargo este gestor de base de datos fue escogido para interactuar con el sistema debido a su robustez, confiabilidad, gratuidad y libre utilización.

6.2 Diagrama de Paquetes de la Arquitectura

Los paquetes permiten describir la arquitectura de un sistema mediante la notación UML. En la Figura 7 se muestran tres paquetes, correspondientes a la arquitectura de tres capas implementada en el desarrollo de este proyecto.

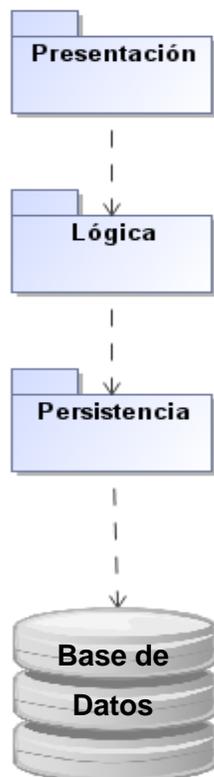


Figura 7. Diagrama de paquetes de la aplicación.

En la capa de presentación se encuentran todas aquellas clases que tienen que ver con el aspecto visual de la aplicación y por ende las que interactúan directamente con el ingeniero agrónomo. En la capa lógica se encuentran todas las clases relacionadas con la lógica del negocio, aquí se implementan todos los procesos que se llevan a cabo para gestionar las etapas, raciones, restricciones, es decir, el proceso de generación de un plan de alimentación. En la capa de persistencia, se encuentran todos los elementos relacionados con el almacenamiento de información en la base de datos, aquí se encuentran todas las clases que controlan la forma en que los objetos son guardados de manera persistente.

6.3 Diagrama de clases

El diagrama de Clases “describe gráficamente las especificaciones de las clases de software y de las interfaces (Java, por ejemplo) en una aplicación”[4]. Los diagramas de clases, generalmente contienen información acerca de las clases, asociaciones, atributos, interfaces, métodos, información sobre el tipo de atributos, navegabilidad y dependencias.

El diagrama de clases presentado en la Figura 8 es parte de la capa lógica del programa. Se incluye la especificación de este diagrama, ya que se considera el más importante debido a que representa mejor el proceso de negocio y además es la parte central del programa implementado.

Se debe considerar que el diagrama que se muestra en la Figura 8 fue simplificado, por términos de espacio, por lo que se obviaron algunos métodos y clases. El programa, en realidad, fue desarrollado con más clases, las cuales incluyen las utilizadas para la interfaz gráfica (del paquete de presentación) y la conexión con la base de datos (del paquete de persistencia).



Figura 8. Diagrama de Clases para optimización de raciones para vacunos wagy.

6.4 Modelo Entidad Relación

El modelo entidad relación o también conocido como MER, es un modelo de datos conceptual de alto nivel, es muy utilizado para el modelado de base de datos, ya que permite mostrar los datos como entidades y especificar sus atributos y relaciones [5]. Este modelo permite al diseñador crear una visión conceptual de la implementación de la persistencia de su sistema, lo cual facilita la creación de la base de datos en un futuro y la comunicación con el usuario a la hora de explicarle en qué consistirá la base de datos.

En la Figura 9 se muestra el modelo entidad relación para el sistema de optimización de raciones para wagyu.

6.3.1 Descripción de Tablas

Etapa: Esta entidad representa las etapas de engorda existentes en el desarrollo de un plan de alimentación para vacunos wagyu. En la Tabla 7 se muestran los atributos de esta entidad y su descripción.

<i>Nombre</i>	<i>Descripción</i>
PESO_INGRESO	Indica el peso en el que un animal es apto para entrar a una etapa.
PESO_SALIDA	Indica el peso límite de un animal para estar en la etapa, al superarlo debe ingresar a otra etapa.
IDETAPA	Es el identificador de la etapa.
NOMBRE_ETAPA	Es el nombre que se le asignará a la etapa.
PORCENTAJE_PESO_VIVO	Es el porcentaje de alimentación máximo que puede comer un animal dependiendo de su peso.
CANTIDAD_ANIMALES	Es el número de animales que se encuentran en la etapa de engorda.
VIGENTE	Es un indicador del estado en el que se encuentra la etapa, es decir, si la etapa participa en el plan de alimentación actual, su estado vigente será uno, de lo contrario cero.

Tabla 7. Atributos de la entidad Etapa y su descripción.

Ración: Esta entidad representa a las distintas raciones que participarán en diferentes planes de alimentación. En la Tabla 8 se muestran los atributos de esta entidad y su descripción.

<i>Nombre</i>	<i>Descripción</i>
ID_RACIÓN	Es el identificador de la ración.
ID_ETAPA_CF	Es una clave foránea que muestra el identificador de la etapa para la cual fue generada una ración.
ID_PLAN_ALI_CF	Es una clave foránea que muestra el identificador del plan de alimentación en el que participará la ración.

Tabla 8. Atributos de la entidad Ración y su descripción.

Plan Alimentación: Esta entidad representa al plan de alimentación que se llevará a cabo al realizar la optimización. En la Tabla 9 se muestran los atributos de esta entidad y su descripción.

<i>Nombre</i>	<i>Descripción</i>
FECHA	Indica la fecha en que fue creado el plan de alimentación.
ID_PLAN_ALI	Es el identificador del plan de alimentación.
NOMBRE_PLAN	Es el nombre que el usuario le da al plan de alimentación que calculó.

Tabla 9. Atributos de la entidad Plan de alimentación y su descripción.

Restricción nutricional mínima: Esta entidad representa a las restricciones nutricionales mínimas que debe cumplir una ración. En la Tabla 10 se muestran los atributos de esta entidad y su descripción.

<i>Nombre</i>	<i>Descripción</i>
CANTIDAD_REST_MIN	Indica la cantidad mínima de cierto nutriente que debe participar en una ración.
ID_REST_NUT_MIN	Es el identificador de la restricción nutricional mínima.
ID_RACION_CF	Es una clave foránea que muestra el identificador de la ración para la cual, fue creada la restricción nutricional mínima.
NOMBRE_NUTRIENTE_CF	Es una clave foránea que muestra el identificador del nutriente, para el cual es

	especificada la restricción nutricional mínima.
--	---

Tabla 10. Atributos de la entidad Restricción nutricional mínima y su descripción.

Restricción Compuesto: Es una entidad que representa a una restricción creada para un compuesto y asociada a una ración. En la Tabla 11 se muestran los atributos de esta entidad y su descripción.

<i>Nombre</i>	<i>Descripción</i>
ID_RESTRICCION_COMPUESTO	Es el identificador de la restricción de un compuesto.
CANTIDAD_MIN	Es la cantidad mínima de un compuesto que se le debe dar a un animal que pertenece a una ración.
CANTIDAD_MAX	Es la cantidad máxima de un compuesto que se le puede dar a un animal que pertenece a una ración.
ID_RACION_RES_COMP_CF	Es una clave foránea que muestra el identificador de la ración a la cual se aplica la restricción de compuesto.
ID_COMPUESTO_CF	Es una clave foránea que muestra el identificador del compuesto que es limitado por la restricción.

Tabla 11. Atributos de la entidad Restricción Compuesto y su descripción.

Participa: Es una entidad que representa a la participación de un compuesto en una ración. En la Tabla 12 se muestran los atributos de esta entidad y su descripción.

<i>Nombre</i>	<i>Descripción</i>
PRECIO	Es el precio del compuesto que participa en la ración.
CANTIDAD	Es la cantidad del compuesto en kilogramos que se le debe dar a los animales que pertenecen a una ración, esta cantidad es obtenida luego de realizada la optimización.
ID_RACION_CF	Es una clave foránea que muestra el identificador de la ración a la cual pertenece la participación.
ID_COMPUESTO_CF	Es una clave foránea que muestra el

	identificador del compuesto que participará en la ración.
ID_PARTICIPA	Es el identificador de la entidad participa.

Tabla 12. Atributos de la entidad Plan de alimentación y su descripción.

Restricción de Stock: Es una entidad que representa a la restricción de stock para un compuesto que se utiliza dentro del un plan de alimentación. En la Tabla 13 se muestran los atributos de esta entidad y su descripción.

<i>Nombre</i>	<i>Descripción</i>
CANTIDAD_REST_STOCK	Es la cantidad máxima de disponibilidad en kilogramos de un compuesto para un plan de alimentación.
ID_COMPUESTO_CF	Es una clave foránea que muestra el identificador del compuesto que es limitado por la restricción de stock para un plan de alimentación.
ID_PLAN_ALI_CF	Es una clave foránea que muestra el identificador del plan de alimentación, que es limitado por la restricción de stock.
ID_RESTRICCION_STOCK.	Es el identificador de la entidad restricción de stock

Tabla 13. Atributos de la entidad Restricción de Stock y su descripción.

Nutriente: Es una entidad que representa a los nutrientes que tienen los compuestos. En la Tabla 14 se muestran los atributos de esta entidad y su descripción.

<i>Nombre</i>	<i>Descripción</i>
NOMBRE_NUTRIENTE	Es el nombre del nutriente.

Tabla 14. Atributos de la entidad Nutriente y su descripción.

Cantidad: Es una entidad que representa a la cantidad específica de nutrientes que tiene un compuesto. En la Tabla 15 se muestran los atributos de esta entidad y su descripción.

<i>Nombre</i>	<i>Descripción</i>
CANTIDAD_NUTRIENTE	Es la cantidad de cierto nutriente que tiene un compuesto.

NOMBRE_NUTRIENTE_CF	Es una clave foránea que muestra el identificador del nutriente.
ID_COMPUESTO_CF	Es una clave foránea que muestra el identificador del compuesto que posee una cantidad de cierto nutriente.
ID_CANTIDAD	Es el identificador de la entidad Cantidad.

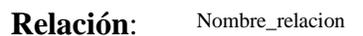
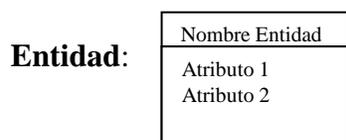
Tabla 15. Atributos de la entidad Cantidad y su descripción.

Compuesto: Es una entidad que representa a los compuestos que participan en la optimización. En la Tabla 16 se muestran los atributos de esta entidad y su descripción.

<i>Nombre</i>	<i>Descripción</i>
NOMBRE_COMPUESTO	Es el nombre del compuesto.
ID_COMPUESTO	Es el identificador de la entidad compuesto.
VIGENTE	Es un indicador del estado en el que se encuentra el compuesto, es decir si el compuesto participa en el plan de alimentación actual su estado vigente será uno, de lo contrario cero.

Tabla 16. Atributos de la entidad Compuesto y su descripción.

En la Figura 9 mostrada más adelante, se utiliza la notación denominada “crow’s foot”. A continuación se explicará la simbología utilizada para la generación de este modelo.



Modalidad y Cardinalidad:

≥ 0 — Cero o muchos

≥ 1 — Uno o muchos

$= 1$ — Uno y sólo uno.

≤ 1 — Cero o uno.

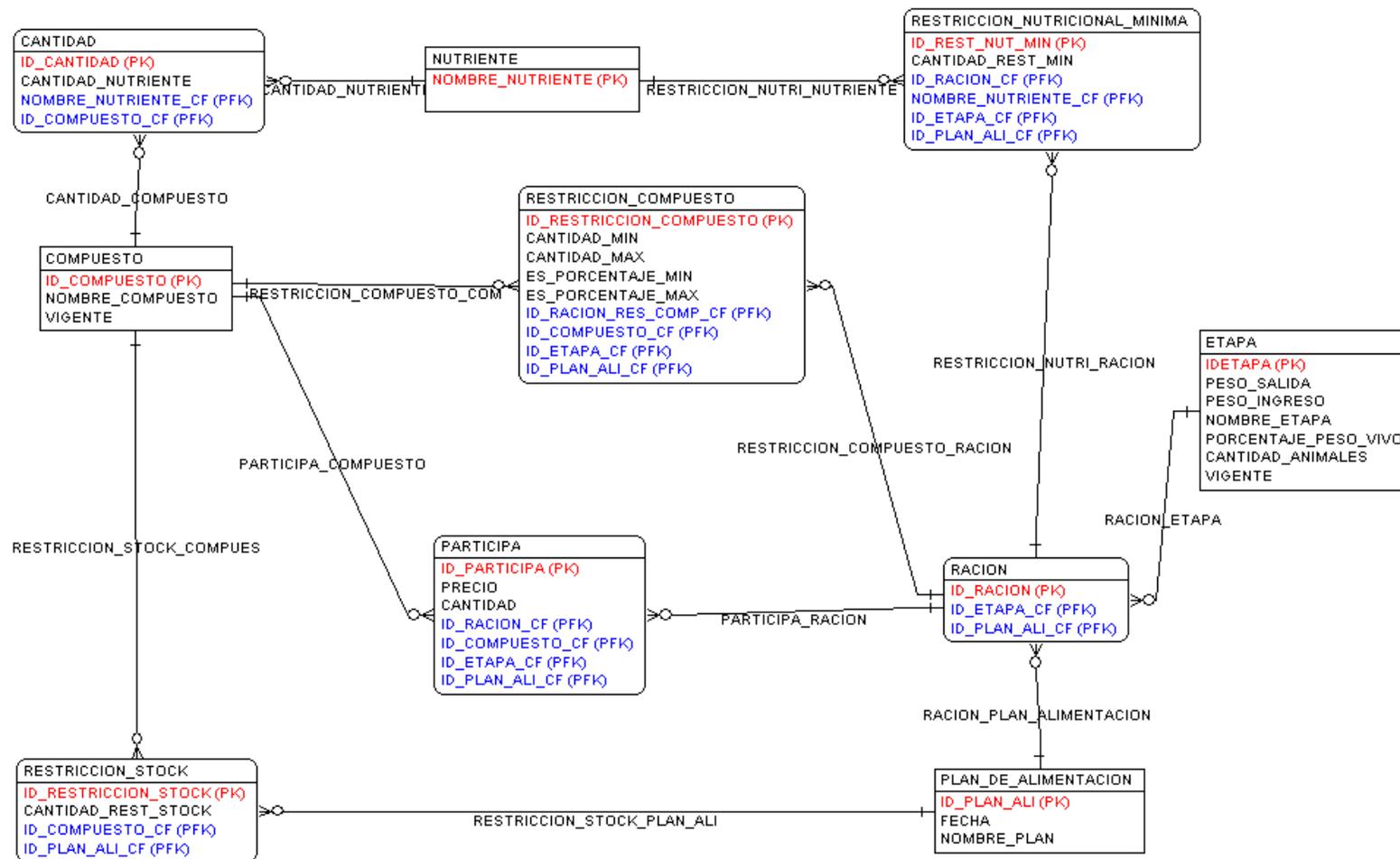


Figura 9. Modelo Entidad Relación para optimización de raciones para vacunos waxy.

6.5 Interfaz de Usuario

Una interfaz de usuario (GUI) presenta un mecanismo amigable que permite al usuario interactuar con el sistema, ya que esta última le proporciona una apariencia visual única al programa [6]. La interfaz de usuario debe siempre enfocarse en las necesidades del cliente, tratando de realizar una GUI de muy fácil entendimiento y manejo. Mientras más consistentes e intuitivas sean las interfaces, más fácil será para el usuario familiarizarse con el sistema, pudiendo incluso disminuir el tiempo de aprendizaje que requiere el usuario para utilizar el programa.

Los prototipos de interfaz de usuario, desarrollados para este proyecto se encuentran en el Anexo B, el cual, está disponible en la copia digital de este informe.

6.6 Diagramas de Secuencia (nivel de diseño)

Debido a que ya existe un mayor entendimiento del problema y se tiene una noción sobre cómo será implementado el programa, se pueden realizar diagramas de secuencia a un nivel más específico. Estos diagramas facilitan la implementación del software, debido a que son más detallados, de esta forma ayudan a visualizar la manera en la que los objetos deberán interactuar.

En el Anexo C, el cual, está disponible en la copia digital de este informe, se presentarán los diagramas de secuencia, a nivel de diseño, que se utilizaron para guiar el proceso de implementación (programación). Se desarrollaron únicamente los diagramas que se consideran más relevantes y que representan o contienen funcionalidades de otros casos de uso, evitando así la redundancia.

Capítulo VII. Plan de Pruebas

En este capítulo se presenta una descripción de las estrategias de pruebas que se utilizarán en el desarrollo de este proyecto, donde se incluirán los elementos y las características a probar, además se dará a conocer qué son los casos de prueba.

7.1 Estrategia de prueba a utilizar

Se ha decidido realizar la estrategia de prueba basadas en requerimientos, estas se llevan a cabo mediante la creación de un conjunto de pruebas para cada caso de uso, es decir para cada requerimiento.

Estas pruebas son de validación, por lo que intentan demostrar que el sistema ha implementado los requerimientos de forma adecuada. Las pruebas a desarrollar tienen que ser lo más completas posible, para probar cada requerimiento exhaustivamente. Estas pruebas deben permitir, comprobar el perfecto cumplimiento de los requisitos, como también, se debe verificar que el programa no permita realizar acciones inválidas.[2]

7.1.1 Los elementos a probar

- Validación de datos según el formato requerido
- Validación de funcionalidades

7.1.2 Las características

Validación de datos según el formato requerido: Verificar la validez de los datos ingresados, por ejemplo si los tipos de datos ingresados son igual a los solicitados.

Validación de funcionalidades: Se verificará el cumplimiento de lo detallado en los casos de uso para cada funcionalidad del sistema.

7.2 Casos de prueba

Los casos de prueba son un detalle de las variables o situaciones de ejecución que serán evaluadas para comprobar el correcto funcionamiento del programa, en ellos se especifica que es lo que se debe evaluar y cuáles son los resultados esperados. Debido a que los casos de prueba se utilizan para corroborar el buen funcionamiento del programa, se

debe realizar un caso de prueba por cada requisito, es decir por cada caso de uso, para así probar que todas las funcionalidades que el programa debe realizar se llevan a cabo perfectamente.

Los casos de prueba realizados se encuentran detallados en el Anexo D y la ejecución de los mismos en el Anexo E y F. Estos anexos se encuentran disponibles en la copia digital de este informe.

Capítulo VIII. Conclusión

8.1 Conclusión

El desarrollo de este software nos ha permitido valorar el importante rol que juegan las tecnologías de información hoy en día y cómo el correcto uso de éstas, no sólo facilitan la comunicación social, la conectividad nacional e internacional tan importante en tiempos de la globalización; en diferentes ámbitos del quehacer humano, generando también beneficios económicos, al permitir optimizar recursos materiales y de mano de obra.

El software desarrollado, brinda como principal beneficio una disminución de los costos de las raciones alimenticias, ajustándose, al mismo tiempo, a los requisitos mínimos necesarios. El ahorro anual aproximado que Agrícola Mollendo obtendrá al utilizar el software es de \$386.766.775 pesos chilenos, como puede apreciarse, disminuye en grandes cantidades los costos de las raciones, pero éste no es el único beneficio, sino también, al utilizar el software y obtener raciones óptimas, mejora la calidad de la carne, lo que contribuye al aumento de las exportaciones.

Existen varios otros beneficios que se pueden obtener con el uso del software. Uno de ellos es la gestión centralizada de los compuestos, stock y su precio. El software también puede asistir a la hora de realizar un análisis histórico de raciones y sus precios, al poseer éste la capacidad de generar reportes de raciones anteriores. El programa también da espacio para realizar simulaciones de raciones permitiendo al ingeniero agrónomo a cargo de la alimentación, realizar cambios en la formulación de las raciones de manera rápida y flexible, ya sea, agregando o eliminando compuestos, modificando aportes nutricionales, agregando o eliminando restricciones a los compuestos, modificando los requisitos mínimos nutricionales de las etapas, variando los precios de los compuestos, agregando, quitando o modificando restricciones de stock, entre varias otras funcionalidades ofrecidas.

Además de entregar raciones óptimas, el programa, basado en la cantidad de stock disponible de los compuestos, es capaz de calcular la duración de la ración e indicar qué compuesto es el que se acabará primero. Esta última funcionalidad le ofrece al ingeniero agrónomo a cargo información sobre qué compuesto debe comprar próximamente.

Al ser este un programa basado en un modelo matemático que es resuelto utilizando, el ya maduro método simplex, se tiene la certeza que el resultado obtenido es

siempre el óptimo, y en caso de que éste no existiera, el programa se lo da a conocer al usuario. Debido a lo anterior, el programa le permite al encargado de la generación de raciones alimenticias, obtener un plan de alimentación óptimo en pocos minutos, liberándolo de la laboriosa tarea que se llevaba a cabo anteriormente a través de un proceso de prueba y error.

Agrícola Mollendo posee una gran cantidad de hectáreas para sembrar sus propios cultivos, de esta forma puede producir diversos alimentos que pudieran formar parte de las raciones alimenticias. En base a lo anterior, otra funcionalidad posible del software desarrollado, es que permite a Agrícola Mollendo simular la generación de raciones, con el objetivo de evaluar cuáles podrán ser los cultivos más rentables a la hora de sembrar, ayudando aún más a abaratar los costos en la generación de futuras raciones.

Una vez que el software comience a ser utilizado en la organización, esperamos que sea un verdadero aporte para la misma, permitiéndole conseguir el objetivo que se han propuesto. Además de lograr, a través de la utilización del software, tomar decisiones inteligentes, más confiables y en menor tiempo, permitiendo a Agrícola Mollendo adaptarse más fácilmente a los cambios del entorno y más específicamente, liberar al ingeniero agrónomo de la tediosa tarea de generar raciones, otorgándole así más tiempo para realizar otras labores.

8.2 Trabajos Futuros

Con el fin de otorgarle mayor funcionalidad al programa y permitir un control más estricto del proceso de engorda, algunas de las mejoras y que se podrían realizar en el software son:

- Incorporar aún mayor flexibilidad al programa, permitiendo variar la cantidad de nutrientes, otorgándole así, la capacidad de generar raciones optimas aplicables a cualquier proceso de engorda, es decir no sólo para vacunos wagyu, sino para cualquier tipo de animal.
- Incorporar un modelo de engorda y crecimiento, que permita en función del tiempo, la etapa en la que el animal se encuentra clasificado y su peso, estimar cuánto tiempo se demorará el animal en alcanzar el peso necesario para pasar a la siguiente etapa de engorda.
- Incorporar un módulo de seguimiento individual para cada wagyu, que permita evaluar su evolución en relación a su peso y la etapa en la que se encuentra. Lo anterior permitirá tomar decisiones en relación a la generación de raciones.
- Implementar un módulo de optimización mejorado, el cuál utilizará el método simplex para los problemas donde existe solución óptima y cuando la solución no exista, se utilizará un método alternativo, tal como algoritmos genéticos, simulated annealing, etc., para obtener una solución lo más cercano a la óptima. El método complementario será escogido a partir de un estudio de todos los posibles métodos y se seleccionará la mejor opción.

Bibliografía

1. Hill, J., D. Kolb, and C. Hill, *Química para el nuevo milenio*. 1999: Prentice-Hall.
2. Sommerville, I., *Ingeniería del software*. 2005: Pearson Educación.
3. Hillier, F., et al., *Introducción a la Investigación de Operaciones*. 1982: McGraw-Hill México.
4. Larman, C., *UML y Patrones*. Una introducción al análisis y diseño orientado a objetos y al proceso unificado”, Editorial: Prentice Hall, 2001.
5. Elmasri, R., S. Navathe, and R. García, *Sistemas de Bases de Datos: conceptos fundamentales*. 1997: Addison-Wesley.
6. Deitel, H. and P. Deitel, *Cómo programar en Java*. 2004: Pearson Education.
7. Winblad, A., *Software orientado a objetos*. 1993: Ediciones Diaz De Santos SA.
8. Backes, M., B. Pfitzmann, and M. Waidner, *Security in business process engineering*. Business Process Management, 2003: p. 1019-1019.
9. *Netbeans IDE*. [en línea]. <<http://netbeans.org/features/ide/index.html>> [consultado: 09 diciembre 2010].
10. Toplink. *Toplink*. [en línea]. <<http://www.oracle.com/technetwork/middleware/ias/toplink-wp-1-130636.pdf>> [consultado: 9 diciembre 2010].
11. Toplink. *Toplink*. [en línea]. <<http://www.oracle.com/technetwork/middleware/toplink/overview/index.html>> [consultado: 09 diciembre 2010].
12. Larman, C., *UML y Patrones*. Segunda ed, ed. Pearson. 2003.

Glosario

Orientación a objetos

Modelo de desarrollo de software muy utilizado hoy en día, mediante el cual se da solución a determinados problemas a través de operaciones computacionales que se desarrollan en el contexto de objetos. En la orientación a objetos, los objetos y las clases son los pilares fundamentales mientras que los métodos, mensajes y la herencia producen los mecanismos primarios. [7]

Lenguajes de programación como C++, Java, Eiffel, Python, entre otros, se basan en la orientación a objetos; tratando de visualizar soluciones en base a abstracciones de datos conocidas como clases, donde un objeto es una instancia de ellas. Los métodos, mensajes y herencia son los que permiten la interacción entre objetos y facilitan el desarrollo de sistemas modulares y de más fácil modificación.

UML

Se utiliza para especificar, visualizar y documentar modelos de sistemas de software, incluyendo su estructura, artefactos y diseño, de manera que cumpla con todos los requisitos. (UML puede utilizarse para el modelado del negocio y también para modelar sistemas no software) [8]. Este lenguaje de modelado es uno de los más utilizados y conocidos hoy en día, ya que permite dar soporte al desarrollo de software, sin especificar una metodología a utilizar. UML cuenta con varios diagramas que permiten representar diferentes aspectos del desarrollo de software.

Netbeans

Es un entorno de desarrollo integrado, que se conoce como IDE, que permite desarrollar software y consta con todas las herramientas necesarias para llevar a cabo aplicaciones de escritorio, empresariales, web y aplicaciones móviles con la plataforma Java, como también con C/C++, PHP, JavaScript, Groovy, and Ruby. Las principales características de Netbeans son: que incluye un editor multi-lenguajes, un depurador de errores, perfiles de integración y archivos de control de versiones.[9]

Toplink

TopLink permite a una aplicación almacenar y recuperar, de forma transparente, objetos Java, usando una base de datos relacional u otro sistema de almacenamiento [10]. Específicamente Toplink es un paquete de mapeo objeto-relacional (ORM) desarrollado para programadores Java. Éste provee un marco de trabajo para almacenar objetos Java en una base de datos relacional o para convertir objetos Java a documentos XML.

En el programa desarrollado se utilizó la versión de código abierto de Toplink, llamada Toplink Essentials, la cual es también la implementación de referencia del EJB¹² 3.0 Java Persistence API (JPA)¹³ y la edición comunitaria de código abierto del producto Toplink de Oracle. Toplink Essentials es una versión limitada de Toplink, por lo que no posee todas las características de Toplink como sincronización entre aplicaciones agrupadas, algunas políticas de validación de caché, etc.[11]

Básicamente Toplink Essentials (como también Toplink) permite comunicar una aplicación programada en Java con una base de datos, tal como se muestra en la Figura 9.

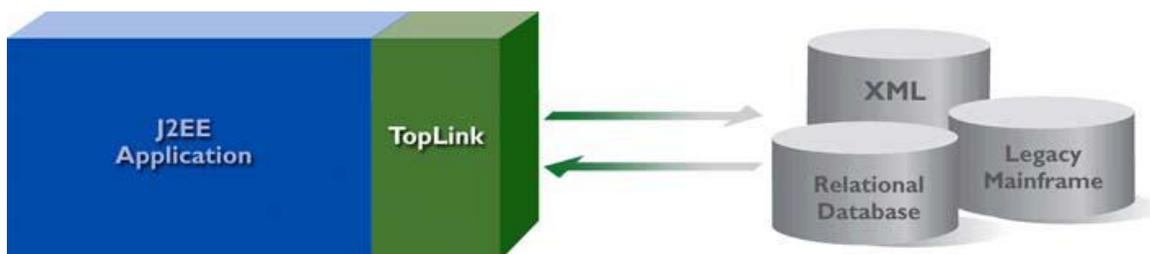


Figura 10. Rol de Toplink en una aplicación.[10]

Oracle 10g

Es un sistema gestor de base de datos relacional (RDBMS), que puede ser utilizado libremente. Oracle 10 g puede ser conectado a aplicaciones desarrolladas con distintos lenguajes de programación, tales como, PHP, Java, .NET, XML, y aplicaciones de código

¹² EJB, por sus siglas en inglés de Enterprise JavaBeans, es una arquitectura de componente del lado del servidor para la construcción de aplicaciones empresariales modulares.

¹³ Véase sección “Arquitectura del sistema”.

abierto. Este gestor de base de datos es muy utilizado hoy en día debido a que se caracteriza por una fácil escalabilidad y migración. [10]

Diagrama de caso de uso

Los diagramas de casos de uso permiten identificar gráficamente las funcionalidades del sistema a desarrollar, además sirven para conocer a los actores que interactuarán con el programa, como también la relación entre estos y los casos de uso. Existen líneas que muestran la comunicación entre los actores y los casos de uso. Los diagramas de casos de uso permiten conocer de forma general el contexto del sistema, permitiendo así, al analista de sistemas comunicarse con el cliente de una forma más clara y rápida.

Modelo Conceptual

El modelo conceptual “es una representación de conceptos en un dominio del problema”[12]. La realización de un modelo conceptual se considera un paso esencial de un análisis orientado a objetos y éste es el primer diagrama que se lleva a cabo con esa orientación. En UML, para generar un modelo conceptual, se crean diagramas de estructura estática. Los diagramas de estructura estática representan conceptos que generalmente se traducirán a objetos, también muestran asociaciones entre conceptos, las cuales se convertirán en relaciones. Finalmente para cada diagrama de estructura se identifican propiedades, las cuales, se traducirán en atributos [4].

Elementos de los diagramas de secuencia

Objeto: Es una instancia de una clase que participará en el diagrama de secuencia interactuando con distintos objetos.

Actores: Son los que interactúan con el sistema, pueden ser parte del diagrama de secuencia, puesto que estos se comunican con el sistema, ya que realizan eventos que producen cierta reacción en el programa.

Línea de vida del objeto: Representa la vida del objeto, dentro de la funcionalidad que se está modelando, esta línea muestra las acciones y reacciones del objeto al comunicarse con otros.

Envío de mensajes: Estos se representan por una flecha horizontal que une las líneas de vida de dos objetos, uno que es el emisor y otro el receptor del mensaje, este mensaje representa la comunicación entre estas dos entidades.

Creación y destrucción de un objeto: La creación de un objeto se representa mediante el envío de un mensaje puntual que da inicio a la línea de vida de un objeto y la destrucción de un objeto se lleva a cabo cuando éste ya ha cumplido su función y esto se denota con una cruz en la línea de vida.

Clase

Una clase “es una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, métodos, relaciones y semántica” [4]. Una clase contiene:

- Nombre: Ejemplo “SolicitudRación”.
- Atributo: “es un valor lógico de un dato de un objeto” [4]. Ejemplo “fechaDeInicio”. Los atributos pueden ser privados(-), públicos(+) y protegidos(#).
- Métodos: en UML es una implementación o algoritmo específico de la operación de una clase. Informalmente es un procedimiento de software que puede ejecutarse en respuesta a un mensaje[4].

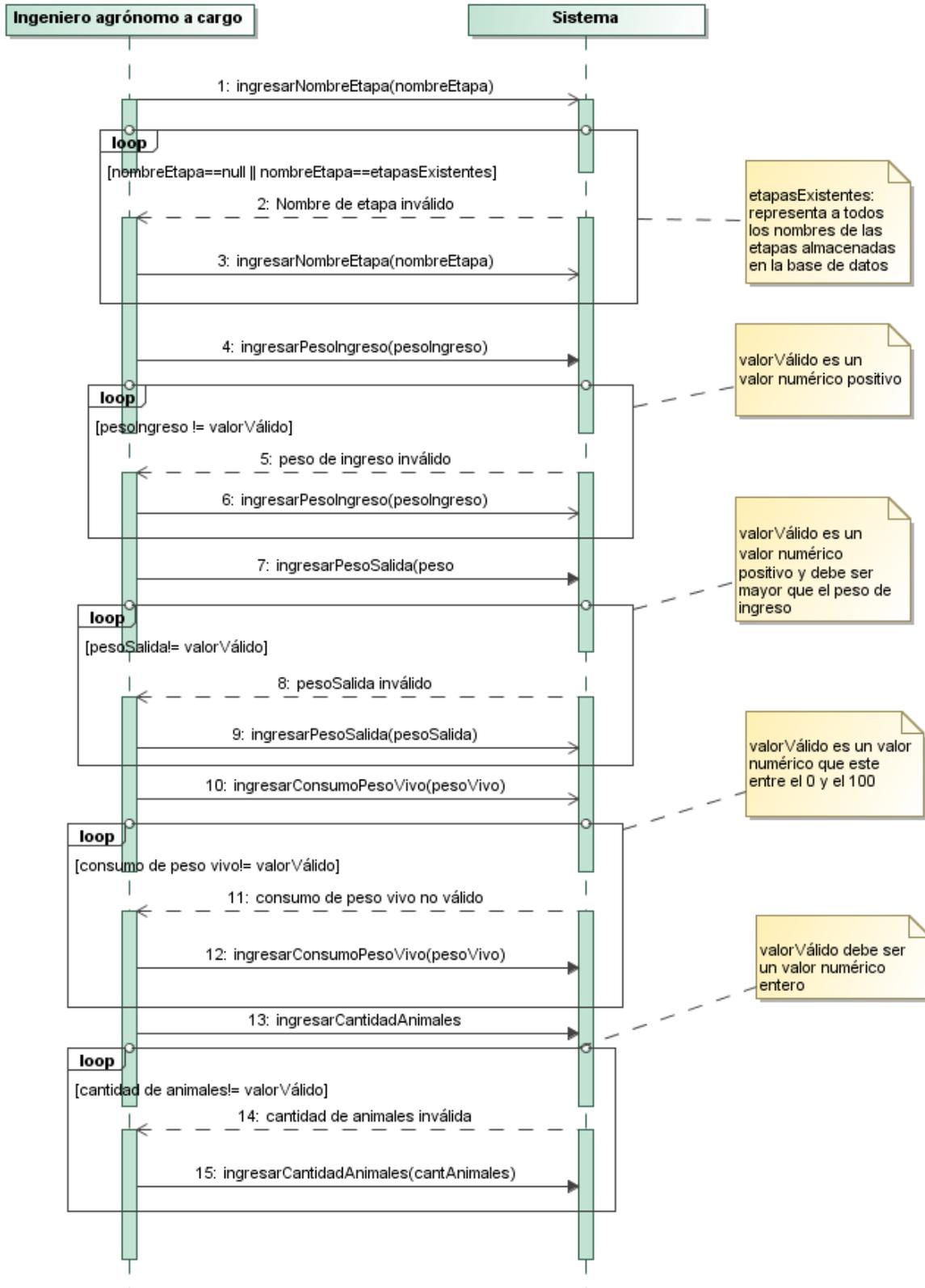
Asociaciones

“Una asociación es una relación entre tipos de objetos (o más concretamente, instancias de estos tipos) que indica alguna conexión significativa e interesante”[4]. Las asociaciones son relaciones semánticas entre clases, éstas pueden tener nombres, navegabilidad y multiplicidad, la última es muy importante, puesto que muestra cuantas instancias de una clase se pueden relacionar con una instancia de otra clase, las multiplicidades pueden ser 0..1; 0..*; 1; 1..*; *, donde * significa muchos. [4]

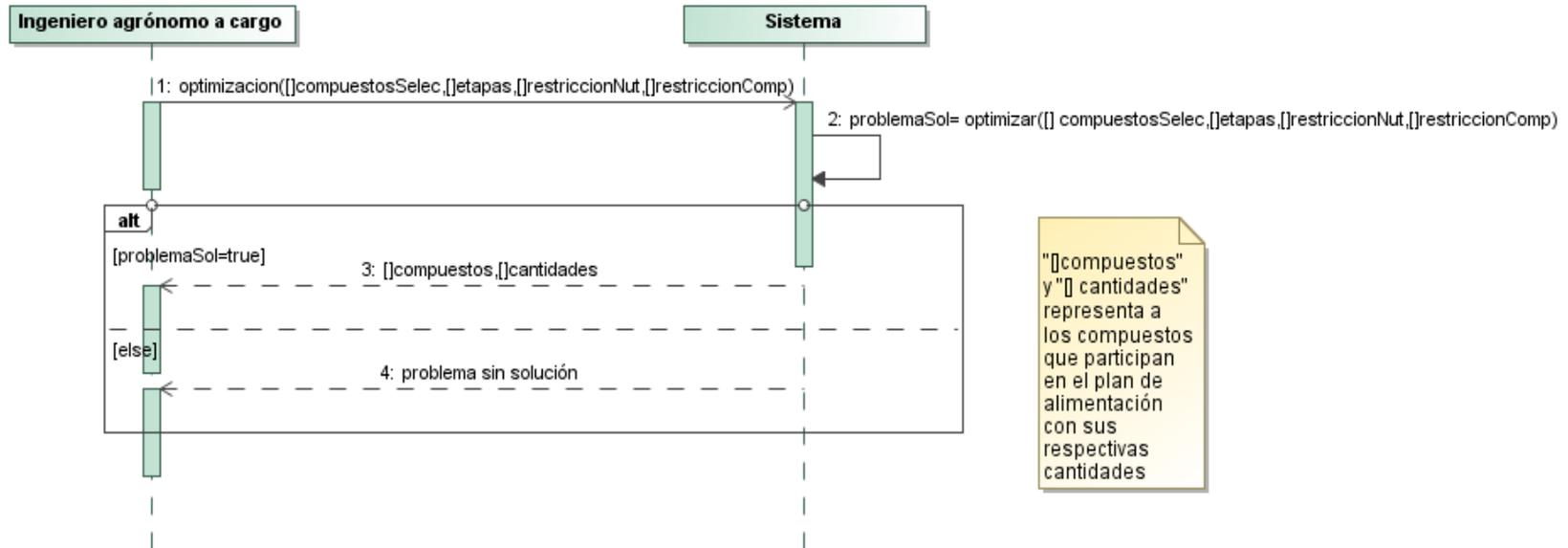
Las asociaciones en UML se representan con una línea entre clases, la cual tiene un nombre que es el de la asociación, en los extremos del segmento se debe colocar la multiplicidad que corresponda y existe la posibilidad de dar una dirección de lectura a la asociación, el símbolo utilizado es ► . En cambio la navegabilidad se representa con una flecha que indica la forma de navegar entre clases que va desde el objeto origen hasta el destino.

Anexo A “Diagramas de secuencia”

Agregar Etapa

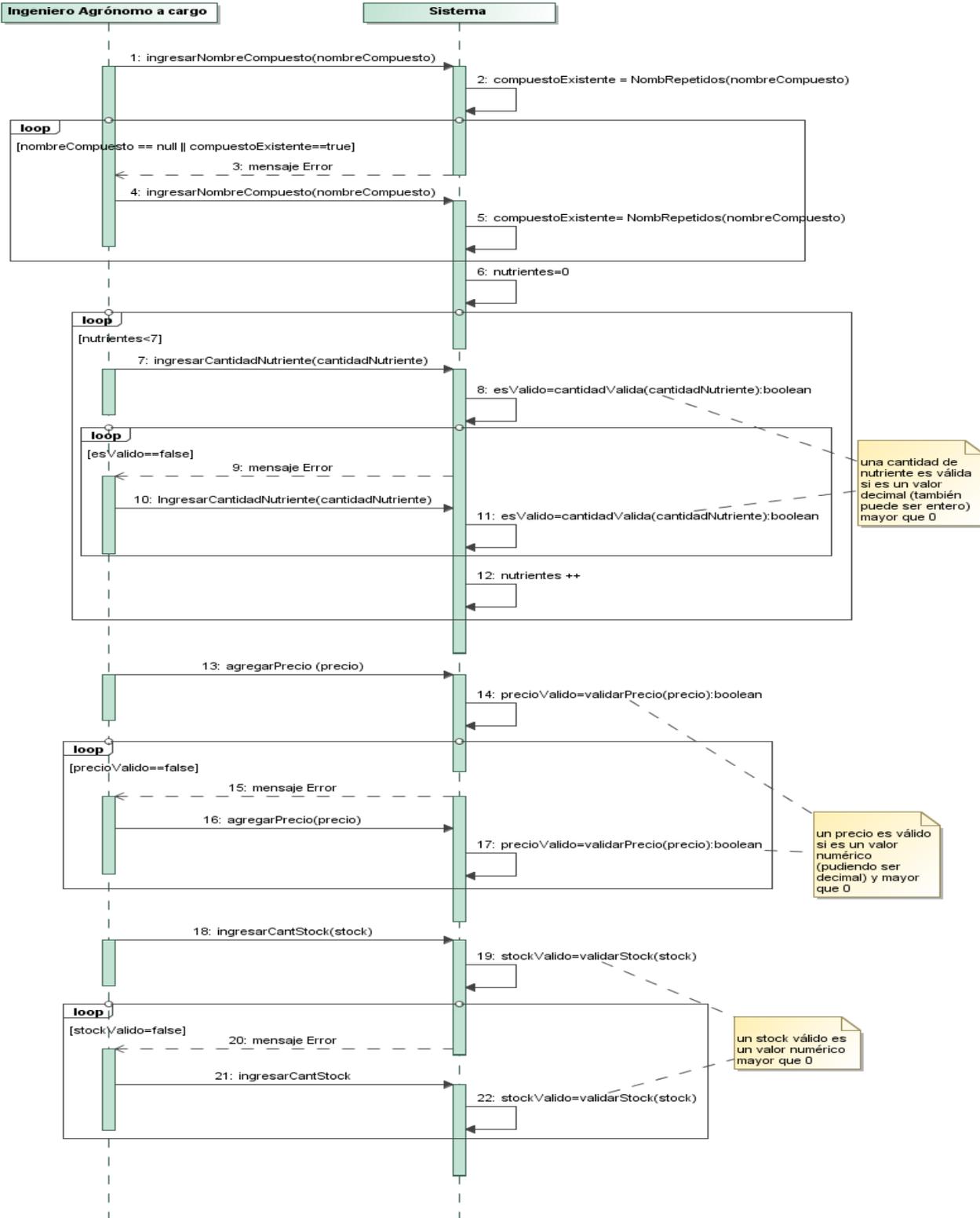


Optimización de raciones

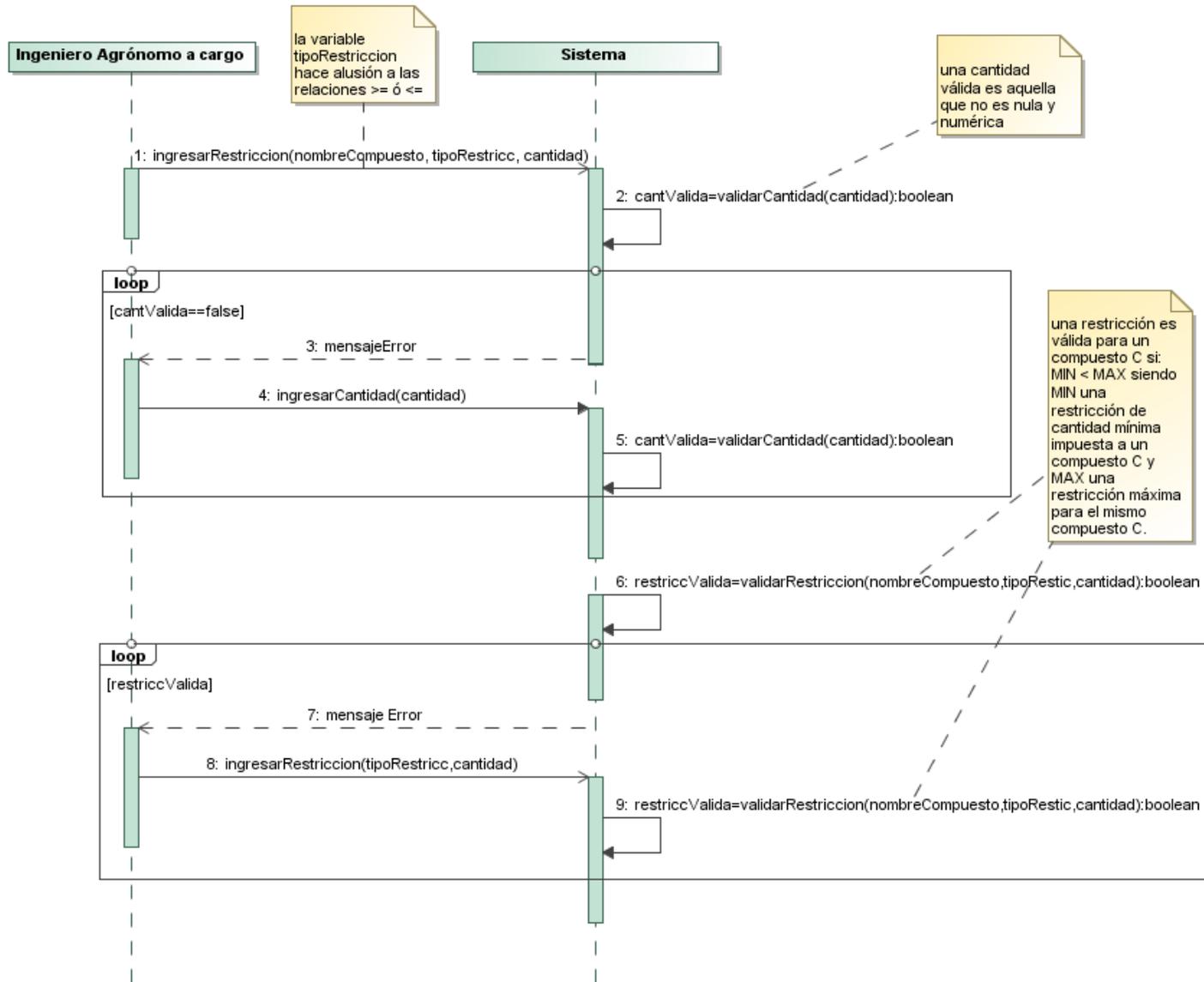


"[]compuestos" y "[]cantidades" representa a los compuestos que participan en el plan de alimentación con sus respectivas cantidades

Agregar compuesto



Agregar restricción de cantidad



Anexo B “Prototipo Interfaz de Usuario”

Gestión general de etapas

Optimización de Raciones para Wagyu

[Gestion General de Etapas](#)
[Gestión de Composiciones y sus Aportes](#)
[Gestión de Precio y Stock](#)
[Gestión Específica de Raciones y Restricciones](#)
[Optimización y Reportes](#)
[Reportes Históricos](#)

Gestión General de Etapas

Selección	Etapas	Peso Ingreso	Peso Salida	% Consumo Peso Vivo	Promedio consumo materia verde	Cantidad Animales
<input type="checkbox"/>	recria	200	300	4	10.0	300
<input type="checkbox"/>	Etapa1	301	400	4	14.02	200

[Agregar Etapa](#)
[Eliminar Etapa](#)
[Ordenar Etapas](#)

Gestión de compuestos y sus aportes

Optimización de Raciones para Wagyu

Gestion General de Etapas | **Gestión de Composiciones y sus Aportes** | Gestión de Precio y Stock | Gestión Específica de Raciones y Restricciones | Optimización y Reportes | Reportes Históricos

Gestión de composiciones y sus aportes

Compuesto	MS	PT	TDN	MCAL	FDN	CA	P
Paja de trigo	0.9	0.04	0.45	1.62	0.85	0.18	0.05
ensilaje de pradera	0.35	0.09	0.6	2.27	0.5	0.67	0.16
vinaza	0.59	0.77	0.78	2.83	0.17	0	0
mezcla soprodi	0.89	0.13	0.79	3.15	0.19	0.14	0.75
sal molido	0.9	0	0	0	0	16	8
E.maiz	0.33	0.07	0.7	2.67	0.41	0.25	0.15
coseta oraffi	0.16	0.08	0.68	2.47	0.46	2.72	0.16
prueba	0	0	0	0	0	0	0

Agregar Composición

Eliminar Composición

Gestión de precio y stock

Optimización de Raciones para Wagyu

[Gestion General de Etapas](#)
[Gestión de Composiciones y sus Aportes](#)
[Gestión de Precio y Stock](#)
[Gestión Específica de Raciones y Restricciones](#)
[Optimización y Reportes](#)
[Reportes Históricos](#)

Compuesto	Cantidad [Kg]	Precio
Paja de trigo		1
ensilaje de pradera		1
vinaza	1000	1
mezcla soprodi		1
sal mollendo	2000	1
E.maiz		123
coseta oraffi		1
prueba		123

Gestión específica de raciones y restricciones

Optimización de Raciones para Wagyu

Gestion General de Etapas | Gestión de Composiciones y sus Aportes | Gestión de Precio y Stock | **Gestión Específica de Raciones y Restricciones** | Optimización y Reportes | Reportes Históricos

recria
Etapa1

Compuestos a utilizar

Paja de trigo
 ensilaje de pradera
 vinaza
 mezcla soprodu
 sal molido
 E.maiz
 cosecha orafti
 prueba

Requisitos Nutricionales Mínimos

MS	PT	TDN	MCAL	FDN	CA	P
<input type="text" value="1.0"/>						

Restricciones

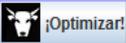
Compuesto	Restricción Mínima [KG]	Restricción Máxima [KG]
vinaza		0.4

Optimización y reportes

Optimización de Raciones para Wagyu

Gestion General de Etapas | Gestión de Composiciones y sus Aportes | Gestión de Precio y Stock | Gestión Específica de Raciones y Restricciones | **Optimización y Reportes** | Reportes Históricos

Optimización y Reportes

Compuestos y Restricciones a Utilizar 

recria

Compuestos seleccionados para la etapa

Nombre Compuesto	Precio	Stock	Restricción mínima	Restricción máxima
mezcla soprodu	1			
sal molido	1	2000		
E.maiz	123			
vinaza	1	1000		0.4

Requerimientos nutricionales mínimos

MS	PT	TDN	MCAL	FDN	CA	P
1.0	1.0	1.0	1.0	1.0	1.0	1.0

Etapa1

Compuestos seleccionados para la etapa

Nombre Compuesto	Precio	Stock	Restricción mínima	Restricción máxima
vinaza	1	1000		0.4
mezcla soprodu	1			
sal molido	1	2000		

Requerimientos nutricionales mínimos

MS	PT	TDN	MCAL	FDN	CA	P
1.0	1.0	1.0	1.0	1.0	1.0	1.0

Resultados de la optimización

recria

Compuesto	Cantidad
mezcla soprodi	6.911207299600333
ensilaje de pradera	0.5921687440076688
vinaza	0.4

Costo de la ración:

Etapa1

Compuesto	Cantidad
mezcla soprodi	8.643042350907521

Costo de la ración:

El compuesto limitante es: vinaza

Duración de la ración 8.333 días

Obtener reportes históricos

Optimización de Raciones para Wagyu

Gestión Específica de Raciones y Restricciones Optimización y Reportes **Reportes Históricos**

Gestión General de Etapas Gestión de Composiciones y sus Aportes Gestión de Precio y Stock

Reportes Históricos

Buscar ración por:

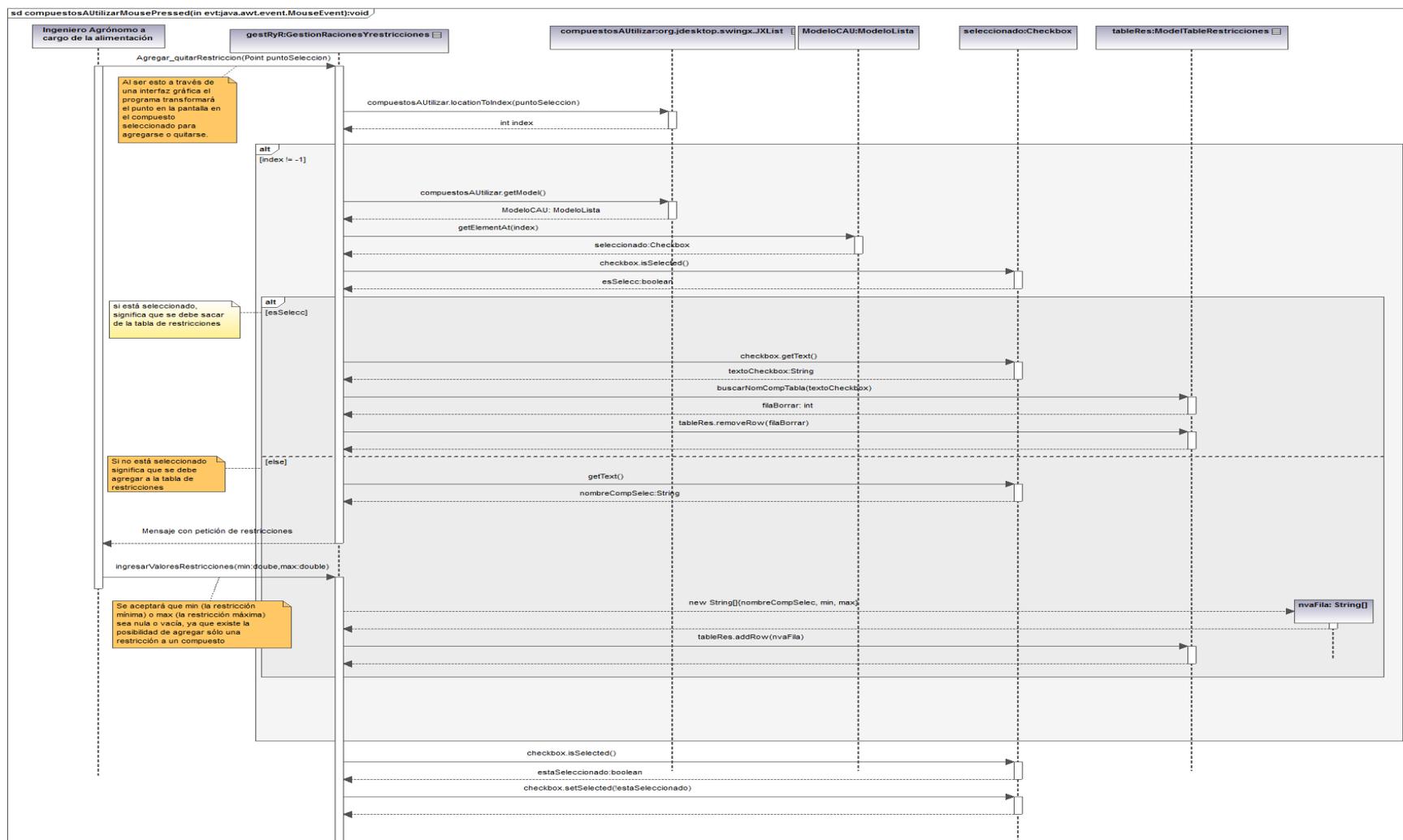
Nombre Racion Diciembre ▼

Fecha

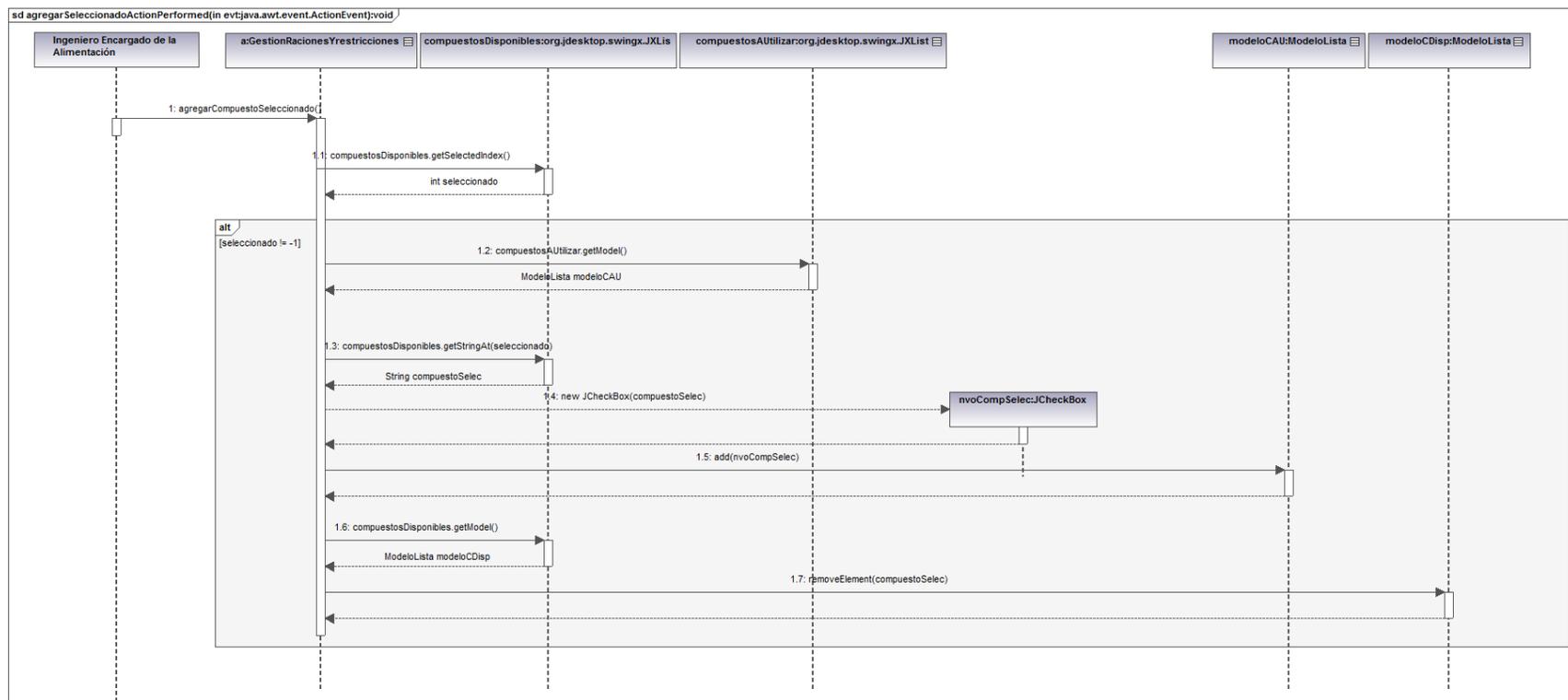
Generar Reporte Excel

Anexo C “Diagrama de secuencia a nivel de diseño”

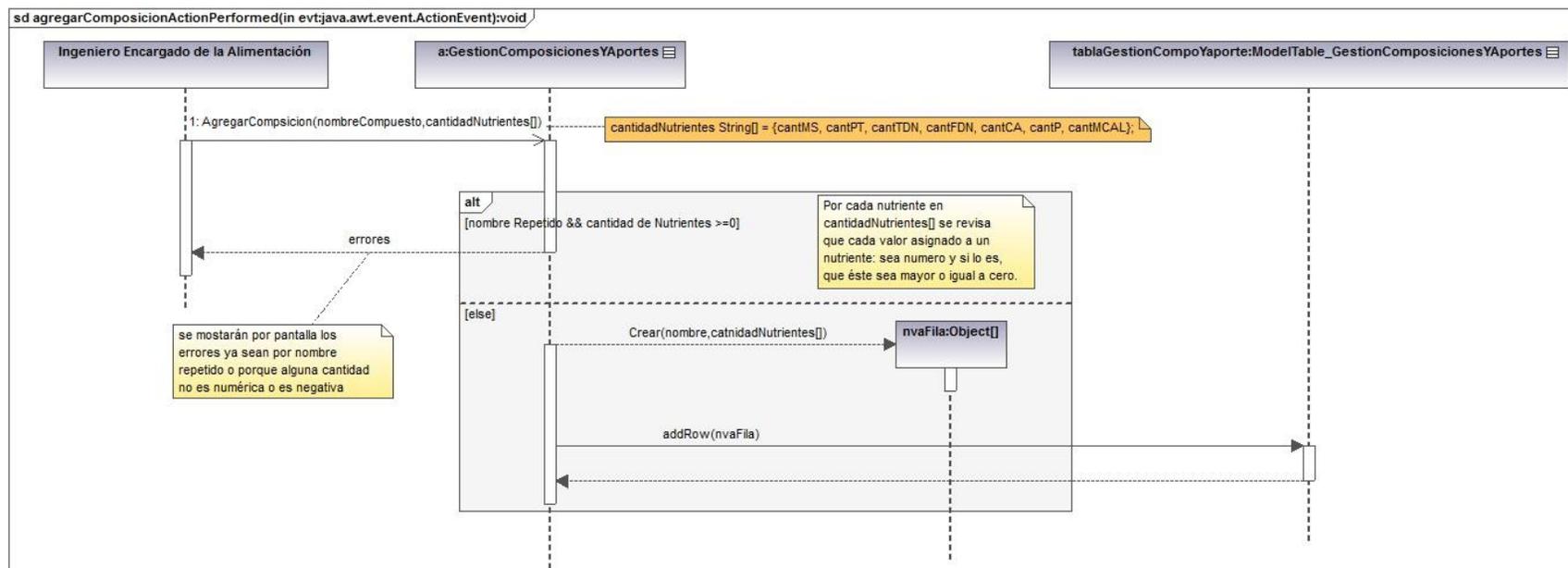
Agregar y quitar restricciones



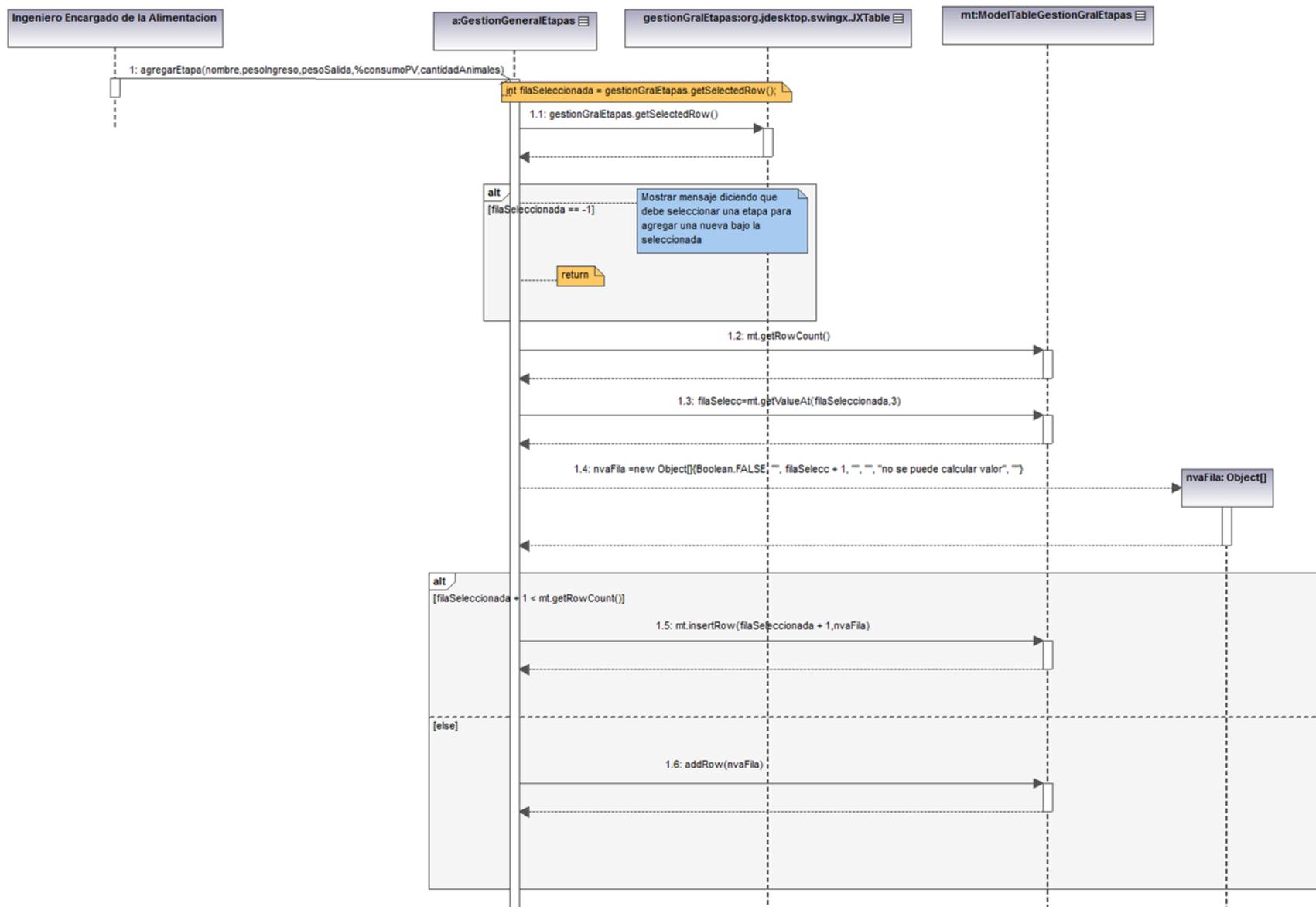
Agregar compuesto seleccionado



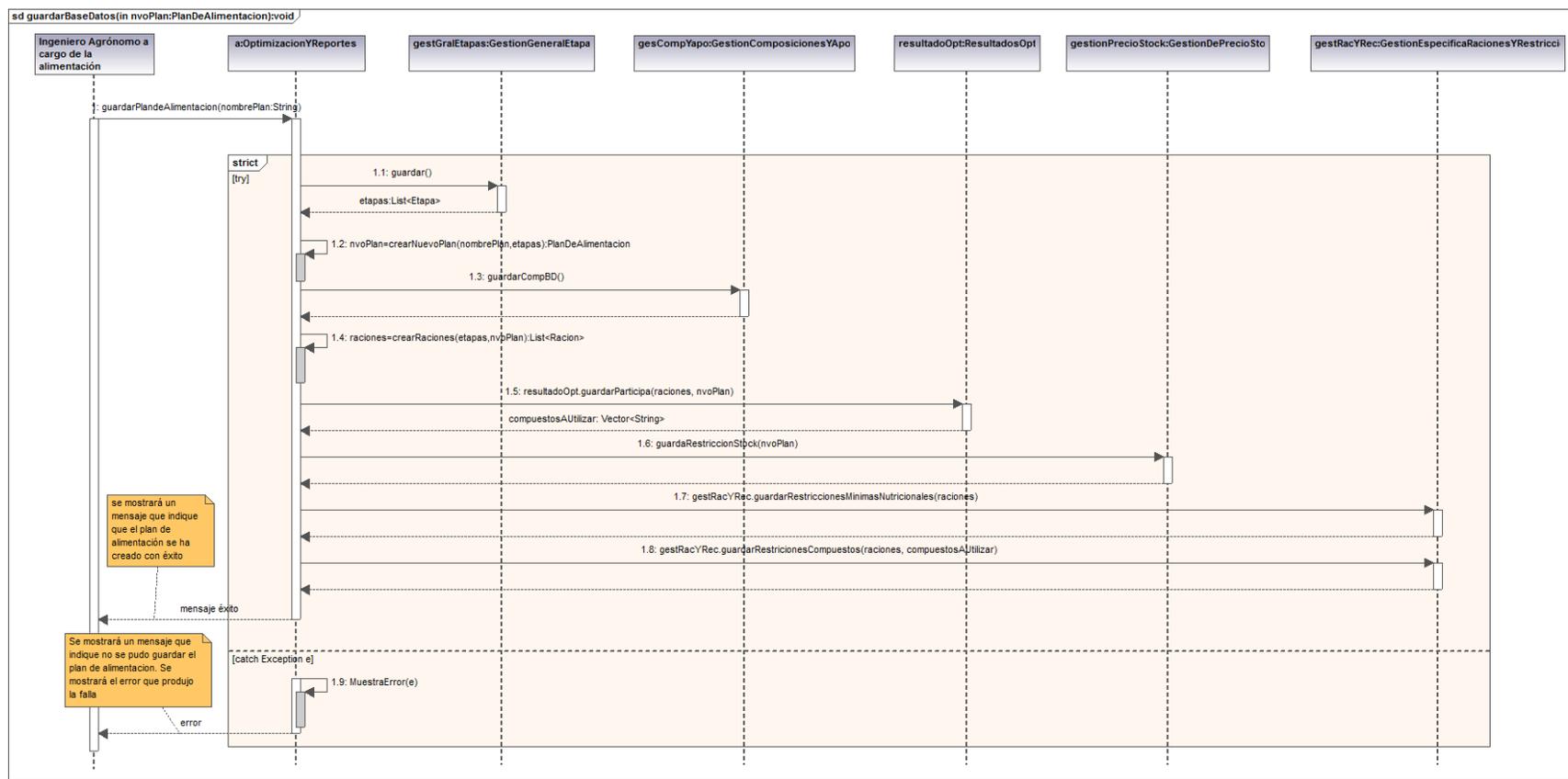
Agregar nueva composición



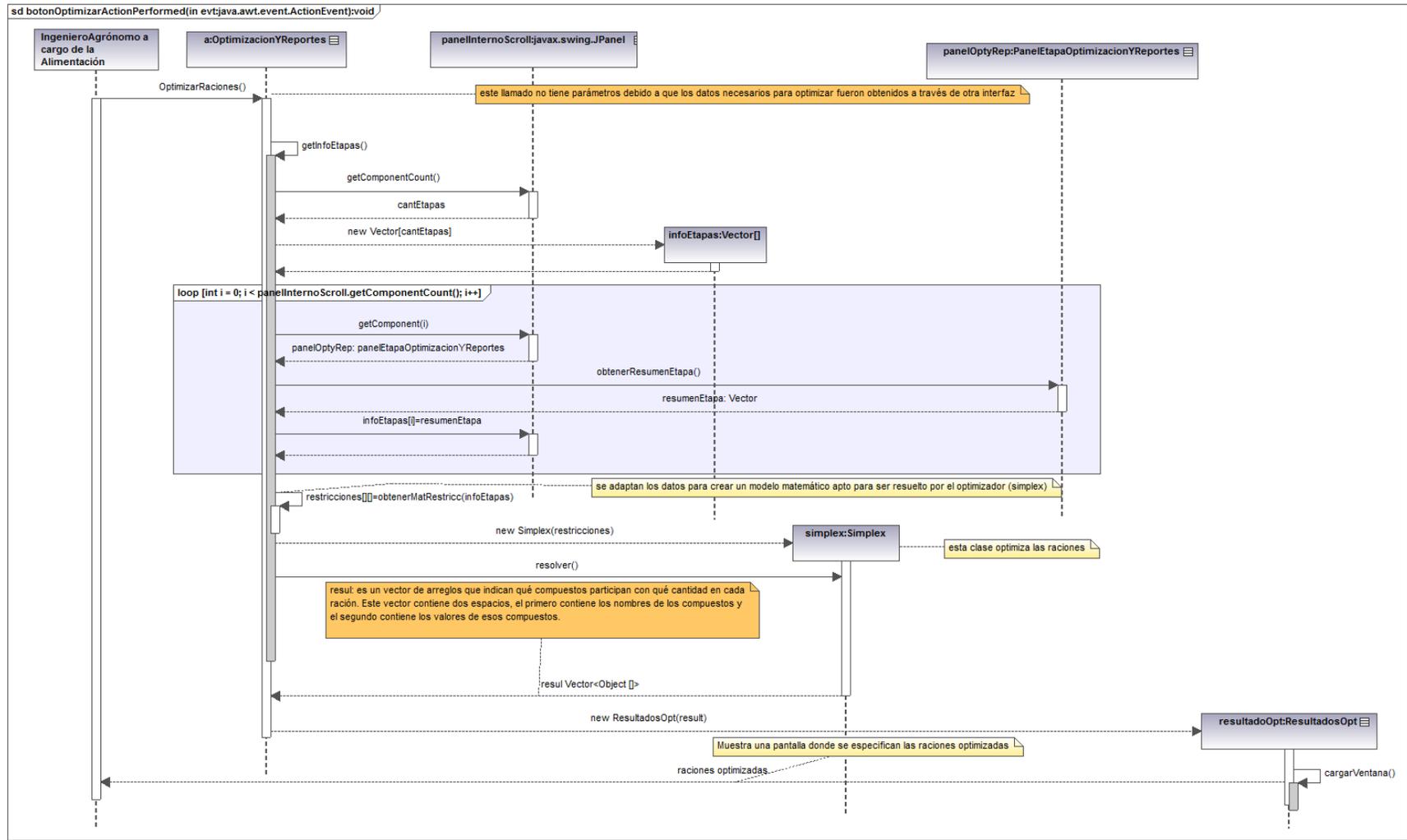
Agregar nueva etapa



Guardar en base de datos



Optimizar Raciones



Anexo D “Casos de prueba”

Caso de Prueba “Agregar etapa”
Código Caso de Uso: 1.1
Descripción de la Prueba: Verificar que el funcionamiento de “Agregar etapa” cumpla con todo lo descrito por el caso de uso.
Condiciones de Ejecución: El sistema donde se realizan las pruebas deberá contar con los recursos necesarios para la óptima ejecución de la prueba, es decir, acceso a la base de datos y el programa con todas sus librerías.
<p>Entrada / Pasos de Ejecución:</p> <ul style="list-style-type: none"> ✓ Al seleccionar la etapa anterior, bajo la cual se desea agregar una nueva etapa, debe añadirse una nueva fila a la tabla, que permite realizar la operación indicada, donde los campos en blanco deben estar marcados en rojo. ✓ Verificar que al agregar un nombre a la etapa, éste no esté repetido, si lo está, el programa debe marcar la celda en rojo. ✓ Verificar que al agregar el nuevo peso de ingreso y salida, porcentaje consumo de peso vivo y cantidad de animales, estos sean valores numéricos positivo, de no serlos las celdas con error deberán ser marcadas en rojo. ✓ Verificar que el peso de ingreso sea menor que el peso de salida, de no serlo, el programa debe marcar la celda con error en rojo. ✓ Verificar que el porcentaje de consumo de peso vivo sea mayor que cero y menor o igual a 100, de no serlo, el programa debe marcar la celda con error en rojo. ✓ Verificar que al agregar la etapa, ésta quede almacenada correctamente en la base de datos, luego de realizar la optimización y guardar el plan de alimentación. ✓ Verificar la existencia de la nueva etapa en la pestaña “Gestión específica de raciones y restricciones”. <p>Nota: si existen celdas con error, el programa no debe permitir el cambio de pestaña.</p>
Evaluación de la Prueba: Se esperará que el 100% de las actividades no presenten errores.

Sin embargo, las actividades que presenten errores deberán corregirse para que el sistema funcione adecuadamente.

Caso de Prueba “Eliminar etapa”
Código Caso de Uso: 1.2
<p>Descripción de la Prueba: Verificar que el funcionamiento de “Eliminar etapa” cumpla con todo lo descrito por el caso de uso.</p>
<p>Condiciones de Ejecución: El sistema donde se realizan las pruebas deberá contar con los recursos necesarios para la óptima ejecución de la prueba, es decir, acceso a la base de datos y el programa con todas sus librerías.</p> <p>Para eliminar una etapa se debe haber agregado una, o deben existir etapas en la base de datos.</p>
<p>Entrada / Pasos de Ejecución:</p> <ul style="list-style-type: none"> ✓ Al seleccionar la etapa y presionar el botón eliminar se muestra la pantalla que permite la confirmación de la eliminación. ✓ Verificar si al realizar una eliminación de una etapa, ésta es eliminada de la interfaz. ✓ Verificar si al realizar una eliminación de una etapa, ésta es marcada como no vigente en la base de datos, si es que ahí se encuentra. ✓ Verificar si al eliminar una etapa con errores, ésta es eliminada de la interfaz y permite cambiarse de pestaña. ✓ Verificar que en la pestaña “Gestión específica de raciones y restricciones” no exista la etapa eliminada. ✓ Verificar que la etapa eliminada no participe en la optimización.
<p>Evaluación de la Prueba: Se esperará que el 100% de las actividades no presenten errores. Sin embargo, las actividades que presenten errores deberán corregirse para que el sistema funcione adecuadamente.</p>

Caso de Prueba “Eliminar composición”
Código Caso de Uso: 5.1
Descripción de la Prueba: Verificar que el funcionamiento de “Eliminar composición” cumpla con todo lo descrito por el caso de uso.
<p>Condiciones de Ejecución: El sistema donde se realizan las pruebas deberá contar con los recursos necesarios para la óptima ejecución de la prueba, es decir, acceso a la base de datos y el programa con todas sus librerías.</p> <p>Para llevar a cabo este caso de prueba deben existir compuestos en la base de datos o debe haber sido agregado al menos un compuesto.</p>
<p>Entrada / Pasos de Ejecución:</p> <ul style="list-style-type: none"> ✓ Al seleccionar el compuesto y presionar el botón eliminar se muestra la pantalla que permite la confirmación de la eliminación. ✓ Verificar si al realizar una eliminación de una composición, ésta es eliminada de la interfaz. ✓ Verificar si al realizar una eliminación de una composición, ésta es marcada como no vigente en la base de datos, si es que ahí se encuentra. ✓ Verificar que en las pestañas “Gestión de precio y stock” y “Gestión específica de raciones y restricciones” no se encuentre disponible la composición eliminada.
Evaluación de la Prueba: Se esperará que el 100% de las actividades no presenten errores. Sin embargo, las actividades que presenten errores deberán corregirse para que el sistema funcione adecuadamente.

Caso de Prueba “Agregar Nueva Composición”
Código Caso de Uso: 5.2
Descripción de la Prueba: Verificar que el funcionamiento de “Agregar nueva composición” cumpla con todo lo descrito por el caso de uso.
Condiciones de Ejecución: El sistema donde se realizan las pruebas deberá contar con los recursos necesarios para la óptima ejecución de la prueba, es decir, acceso a la base de datos y el programa con todas sus librerías.
<p>Entrada / Pasos de Ejecución:</p> <ul style="list-style-type: none"> ✓ El botón agregar nueva composición, al apretarse, debe generar una nueva fila al final de la tabla de compuestos. La nueva fila debe tener todos los campos marcados con rojo (al no tener valores los campos, se consideran erróneos). ✓ Al agregar el nombre del compuesto, el programa debe verificar que el nuevo nombre ingresado no esté repetido. Si el nombre del nuevo compuesto ingresado está repetido, el programa debe marcar el campo del nombre con rojo. ✓ Al ingresar los valores para los aportes nutricionales, el programa debe verificar que los valores sean numéricos y que sean positivos, en caso contrario debe marcar las celdas con error. ✓ Si el compuesto está correctamente ingresado, una vez que se realice la optimización, si se presiona el botón guardar, el nuevo compuesto ingresado debe guardarse en la base de datos. ✓ Se debe verificar la existencia del nuevo compuesto en la pestaña “Gestión de Precio y Stock”. ✓ El nuevo compuesto ingresado debe participar al generarse el modelo para la optimización, con los aportes nutricionales ingresados en esta sección, únicamente, si éste fue seleccionado, para ser considerado en la generación de una ración para una o más etapas. ✓ Verificar que en la pestaña “Gestión específica de raciones y restricciones” se

encuentre disponible la composición agregada.

Nota: si existen celdas con error, el programa no debe permitir el cambio de pestaña.

Evaluación de la Prueba: Se esperará que el 100% de las actividades no presenten errores. Sin embargo, las actividades que presenten errores deberán corregirse para que el sistema funcione adecuadamente.

Caso de Prueba “Modificar aporte nutricional”
Código Caso de Uso: 5.3.1
<p>Descripción de la Prueba: Verificar que el funcionamiento de “Modificar aporte nutricional” cumpla con todo lo descrito por el caso de uso.</p>
<p>Condiciones de Ejecución: El sistema donde se realizan las pruebas deberá contar con los recursos necesarios para la óptima ejecución de la prueba, es decir, acceso a la base de datos y el programa con todas sus librerías.</p> <p>Para realizar este caso de prueba, deben existir compuestos en la base de datos.</p>
<p>Entrada / Pasos de Ejecución:</p> <ul style="list-style-type: none"> ✓ Al hacer doble clic para modificar un aporte nutricional el programa deberá verificar que el valor sea numérico, positivo y de no serlo, deberá marcar en rojo la celda con error. ✓ Verificar si al realizar una modificación del aporte nutricional, ésta queda registrada en la base de datos, luego de realizar la optimización y guardar el plan de alimentación. <p>Nota: si existen celdas con error, el programa no debe permitir el cambio de pestaña.</p>
<p>Evaluación de la Prueba: Se esperará que el 100% de las actividades no presenten errores. Sin embargo, las actividades que presenten errores deberán corregirse para que el sistema funcione adecuadamente.</p>

Caso de Prueba “Modificar precio”
Código Caso de Uso: 5.3.2
Descripción de la Prueba: Verificar que el funcionamiento de “Modificar precio” cumpla con todo lo descrito por el caso de uso.
<p>Condiciones de Ejecución: El sistema donde se realizan las pruebas deberá contar con los recursos necesarios para la óptima ejecución de la prueba, es decir, acceso a la base de datos y el programa con todas sus librerías.</p> <p>Para modificar el precio deben existir compuestos en la base de datos.</p>
<p>Entrada / Pasos de Ejecución:</p> <ul style="list-style-type: none"> ✓ En la pestaña de Gestión de Precio y Stock, el programa debe permitir hacer clic sobre la celda correspondiente al campo “stock” del compuesto deseado, para luego dejar escribir una cantidad de stock. ✓ Al hacer doble clic para modificar un precio el programa deberá verificar que el valor sea numérico, positivo y de no serlo, deberá marcar en rojo la celda con error. ✓ Verificar que el precio modificado sea el que se utiliza en la optimización. ✓ Verificar si al realizar una modificación del precio, ésta queda registrada en la base de datos, luego de realizar la optimización y guardar el plan de alimentación. <p>Nota: si existen celdas con error, el programa no debe permitir el cambio de pestaña.</p>
<p>Evaluación de la Prueba: Se esperará que el 100% de las actividades no presenten errores. Sin embargo, las actividades que presenten errores deberán corregirse para que el sistema funcione adecuadamente.</p>

Caso de Prueba “Modificar Stock”
Código Caso de Uso:5.3.3
<p>Descripción de la Prueba: Verificar que el funcionamiento de “modificar Stock” cumpla con todo lo descrito por el caso de uso.</p>
<p>Condiciones de Ejecución: El sistema donde se realizan las pruebas deberá contar con los recursos necesarios para la óptima ejecución de la prueba, es decir, acceso a la base de datos y el programa con todas sus librerías.</p> <p>Para modificar la cantidad de stock de una composición, se debe haber ingresado el compuesto al que se le desea gestionar la cantidad de stock o se debe haber cargado de la base de datos.</p>
<p>Entrada / Pasos de Ejecución:</p> <ul style="list-style-type: none"> ✓ En la pestaña de Gestión de Precio y Stock, el programa debe permitir hacer clic sobre la celda correspondiente al campo “stock” del compuesto deseado, para luego dejar escribir una cantidad de stock. ✓ El valor correspondiente a la cantidad de stock puede ser un valor no entero, pero debe ser mayor que 0. El programa debe marcar como errónea la cantidad de stock si ésta no es un valor numérico o si es menor que 0. ✓ El programa debe considerar como correcto el valor de String vacío (“”), ya que, esto simboliza que el compuesto no tiene restricción de stock. ✓ Se debe verificar que la cantidad de stock se tome en cuenta a la hora de realizar el cálculo de la duración de la ración, si y solo si, el compuesto con una restricción de stock fue seleccionado para participar en alguna de las raciones generadas por el programa. ✓ La cantidad de stock se debe guardar en la base de datos si el compuesto fue seleccionado para participar en la ración.

Nota: si existen celdas con error, el programa no debe permitir el cambio de pestaña.

Evaluación de la Prueba: Se esperará que el 100% de las actividades no presenten errores. Sin embargo, las actividades que presenten errores deberán corregirse para que el sistema funcione adecuadamente.

Caso de Prueba “Modificar restricciones de cantidad de compuesto”
Código Caso de Uso: 6.1
<p>Descripción de la Prueba:</p> <p>Se verificará que el programa permita modificar restricciones de cantidad a los compuestos seleccionados para formar parte de la formulación de raciones.</p>
<p>Condiciones de Ejecución: El sistema donde se realizan las pruebas deberá contar con los recursos necesarios para la óptima ejecución de la prueba, es decir, acceso a la base de datos y el programa con todas sus librerías.</p> <p>Para modificar una restricción se debe haber agregado una restricción, por lo menos, un compuesto.</p>
<p>Entrada / Pasos de Ejecución:</p> <ul style="list-style-type: none"> ✓ El programa debe permitir modificar una restricción de cantidad para un determinado compuesto. Para lo anterior al hacer clic sobre una celda en la tabla rotulada como “restricciones”, el programa debe permitir editar el campo, en el cual, se hace clic (a menos que éste sea el nombre) para ingresar un nuevo valor. ✓ El programa debe verificar que el nuevo valor editado se encuentre dentro del dominio de las restricciones y que sea un valor numérico mayor que 0 ó que sea un valor en blanco (String vacío “”), en caso contrario debe marcar la celda como errónea. ✓ Se verificará que los datos agregados se vean reflejados tanto en la interfaz, como en la base de datos. Para el último caso, se guardarán las restricciones cuando se presione el botón “Guardar”.
<p>Evaluación de la Prueba: Se esperará que el 100% de las actividades no presenten errores. Sin embargo, las actividades que presenten errores deberán corregirse para que el sistema funcione adecuadamente.</p>

Caso de Prueba “Eliminar restricciones de cantidad de compuesto”
Código Caso de Uso: 6.2
<p>Descripción de la Prueba:</p> <p>Se verificará que el programa permita eliminar restricciones de cantidad asignadas a los compuestos.</p>
<p>Condiciones de Ejecución: El sistema donde se realizan las pruebas deberá contar con los recursos necesarios para la óptima ejecución de la prueba, es decir, acceso a la base de datos y el programa con todas sus librerías.</p> <p>Para eliminar una restricción se debe haber ingresado por lo menos una restricción a un compuesto.</p>
<p>Entrada / Pasos de Ejecución:</p> <ul style="list-style-type: none"> ✓ El programa deberá permitir eliminar una restricción asignada a un compuesto. ✓ Para lo anterior se deberá hacer clic sobre el nombre del compuesto que tiene asignada una restricción de cantidad. Una vez hecho lo anterior, el cuadrado a la izquierda del nombre del compuesto se debe mostrar vacío y el nombre del compuesto en la tabla de restricciones se debe borrar, junto con las restricciones asociadas a éste. ✓ Para lo anterior, en caso de haber errores en las celdas y al quitar las restricciones sobre un compuesto, se deben también eliminar las celdas con errores. ✓ Una vez realizada la optimización, el programa no deberá considerar la restricción que fue eliminada. <p>Nota: si existen celdas con error, el programa no debe permitir el cambio de pestaña.</p>
<p>Evaluación de la Prueba: Se esperará que el 100% de las actividades no presenten errores. Sin embargo, las actividades que presenten errores deberán corregirse para que el sistema funcione adecuadamente.</p>

Caso de Prueba “Agregar restricciones de cantidad de compuesto”
Código Caso de Uso: 6.3
Descripción de la Prueba: Verificar que el funcionamiento de “Agregar restricciones de cantidad de compuesto” cumpla con todo lo descrito por el caso de uso.
<p>Condiciones de Ejecución: El sistema donde se realizan las pruebas deberá contar con los recursos necesarios para la óptima ejecución de la prueba, es decir, acceso a la base de datos y el programa con todas sus librerías.</p> <p>Para agregar restricciones de cantidad de compuesto por etapa, deben existir compuestos y etapas previamente ingresados o cargados de la base de datos.</p>
<p>Entrada / Pasos de Ejecución:</p> <ul style="list-style-type: none"> ✓ Al seleccionar la pestaña “Gestión específica de raciones y restricciones” verificar que se muestren los espacios dentro de la pantalla para permitir realizar la operación indicada, esto es: una tabla en la parte inferior de la pantalla que contenga 3 columnas rotuladas como: “Nombre”, “Cantidad mínima [kg]”, “Cantidad máxima [kg]” ✓ Se debe verificar que al hacer clic sobre un compuesto que se encuentre bajo el recuadro con título “Compuestos a Utilizar” debe aparecer el nombre del compuesto en la tabla, en la parte inferior de la pantalla bajo el título de “restricciones” y debe quedar marcado el cuadrado que se encuentra a la izquierda del nombre del compuesto. ✓ Se probará que el programa verifique los valores del dominio de las restricciones, es decir, que el valor de la restricción debe ser mayor o igual a cero y que las restricciones no sean excluyentes entre sí. ✓ Verificar que las restricciones sean consideradas al realizar la optimización. ✓ Una restricción vacía (“”) indica que no existe una restricción para la columna donde se encuentra el valor vacío. Por ejemplo si se selecciona un compuesto y no

hay nada en la columna de “restricción mínima” eso significa que el compuesto no tiene restricción mínima. Por lo tanto se debe verificar que una restricción vacía no se marque con error.

Nota:

- una restricción con ambas columnas vacías (restricción mínima y máxima), por convención se toma como si no tuviera restricciones y no son consideradas como restricciones a la hora de realizar la optimización.
- si existen celdas con error, el programa no debe permitir el cambio de pestaña.

Evaluación de la Prueba: Se esperará que el 100% de las actividades no presenten errores. Sin embargo, las actividades que presenten errores deberán corregirse para que el sistema funcione adecuadamente.

Caso de Prueba “Seleccionar los compuestos que participan en cada etapa”
Código Caso de Uso: 7
<p>Descripción de la Prueba: Verificar que el funcionamiento de “Seleccionar los compuestos que participan en cada etapa” cumpla con todo lo descrito por el caso de uso.</p>
<p>Condiciones de Ejecución:</p> <p>El sistema donde se realizan las pruebas deberá contar con los recursos necesarios para la óptima ejecución de la prueba, es decir, acceso a la base de datos y el programa con todas sus librerías.</p> <p>Para seleccionar los compuestos que participan en cada etapa, deben existir compuestos y etapas en la base de datos o deben haber sido ingresados utilizando la interfaz.</p>
<p>Entrada / Pasos de Ejecución:</p> <ul style="list-style-type: none"> ✓ Al dirigirse a la pestaña “Gestión específica de raciones y restricciones”, verificar que se muestren en “<i>compuestos disponibles</i>”, los compuestos agregados en la pestaña “Gestión de compuestos y aportes”. ✓ Verificar que los compuestos eliminados en la pestaña “Gestión de compuestos y aportes” no aparezcan en “<i>compuestos disponibles</i>”. ✓ Verificar que la pestaña permita seleccionar y deseleccionar correctamente los compuestos disponibles para ser utilizados en la ración. ✓ Verificar que si no se selecciona ningún compuesto, el programa debe marcar en rojo el recuadro “<i>compuestos a utilizar</i>”.
<p>Evaluación de la Prueba: Se esperará que el 100% de las actividades no presenten errores. Sin embargo, las actividades que presenten errores deberán corregirse para que el sistema funcione adecuadamente.</p>

Caso de Prueba “Modificar Restricciones Mínimas Nutricionales por etapa”
Código Caso de Uso: 8
<p>Descripción de la Prueba: Verificar que el funcionamiento de “Modificar Restricciones Mínimas Nutricionales por etapa” cumpla con todo lo descrito por el caso de uso.</p>
<p>Condiciones de Ejecución:</p> <p>El sistema donde se realizan las pruebas deberá contar con los recursos necesarios para la óptima ejecución de la prueba, es decir, acceso a la base de datos y el programa con todas sus librerías.</p> <p>Para modificar las restricciones mínimas nutricionales que limitan cada etapa, deben existir restricciones para cada etapa en la base de datos.</p>
<p>Entrada / Pasos de Ejecución:</p> <ul style="list-style-type: none"> ✓ Al hacer doble clic para modificar una restricción mínima nutricional, el programa deberá verificar que el valor sea numérico, positivo y de no serlo, deberá marcar en rojo la celda con error. ✓ Verificar que las restricciones modificadas sean las que se utilizan en la optimización. ✓ Verificar, si al realizar una modificación de las restricciones, éstas quedan registradas en la base de datos, luego de realizar la optimización y guardar el plan de alimentación. <p>Nota: si existen celdas con error, el programa no debe permitir el cambio de pestaña.</p>
<p>Evaluación de la Prueba: Se esperará que el 100% de las actividades no presenten errores. Sin embargo, las actividades que presenten errores deberán corregirse para que el sistema funcione adecuadamente.</p>

Caso de Prueba “Determinar las raciones por cada etapa que minimice el costo total.”
Código Caso de Prueba: 9
<p>Descripción de la Prueba:</p> <p>Verificar el correcto funcionamiento del caso de uso “Determinar las raciones por cada etapa que minimice el costo total.”</p>
<p>Condiciones de Ejecución:</p> <p>El sistema donde se realizan las pruebas deberá contar con los recursos necesarios para la óptima ejecución de la prueba, es decir, acceso a la base de datos y el programa con todas sus librerías.</p>
<p>Entrada / Pasos de Ejecución:</p> <ul style="list-style-type: none"> ✓ Verificar que el sistema conste de todos los datos necesarios para realizar la optimización, es decir nutrientes, compuestos y restricciones. ✓ Verificar que los datos, para realizar la optimización, sean entregados al módulo de optimización correctamente, es decir en el formato deseado y con su correcta validación. ✓ Verificar que la realización de la optimización se haya llevado a cabo correctamente, es decir que entregue una salida adecuada. Si el problema tiene solución debe retornar la solución correcta, de lo contrario debe retornar un mensaje que indique que el problema no puede solucionarse. ✓ Verificar que el almacenamiento de los datos, relacionados con el resultado de la optimización sea correcto. <p>Nota: Cuando se menciona “adecuado” y “correcto” se refiere a que concuerde con lo definido en los requisitos.</p>
<p>Evaluación de la Prueba:</p> <p>Se esperará que el 100% de las actividades no presenten errores. Sin embargo, las actividades que presenten errores deberán corregirse para que el sistema funcione adecuadamente.</p>

Caso de Prueba “Obtener reporte histórico de raciones anteriores”
Código Caso de Uso: 10
<p>Descripción de la Prueba:</p> <p>Se verificará que el programa permita obtener reportes con los resultados de la optimización seleccionada.</p>
<p>Condiciones de Ejecución: El sistema donde se realizan las pruebas deberá contar con los recursos necesarios para la óptima ejecución de la prueba, es decir, acceso a la base de datos y el programa con todas sus librerías.</p> <p>Para obtener reportes correctamente primero se debe haber guardado el plan de alimentación.</p>
<p>Entrada / Pasos de Ejecución:</p> <ul style="list-style-type: none"> ✓ Se debe verificar que todos los planes de alimentación generados anteriormente estén disponibles por nombre y por fecha. ✓ Se debe verificar que en el reporte generado en Excel, estén todos los compuestos seleccionados para participar en las raciones para las etapas. ✓ Para cada compuesto seleccionado en cada ración, se debe verificar que el aporte que éste brinda a la ración sea correcto por cada nutriente (materia seca, proteínas, etc.) y que el precio también esté correcto. ✓ Se debe verificar que la ración generada muestre correctamente los aportes reales de la misma y que éstos sean mayores que los requisitos. ✓ Se debe verificar que la fecha de creación de la ración es correcta. ✓ Se debe verificar que en el archivo Excel, se genere la misma cantidad de pestañas que la cantidad de etapas y además que cada pestaña tenga el nombre de la etapa. ✓ Se debe verificar que los requerimientos mínimos de la etapa sean los correspondientes al nombre de la etapa que se muestra, tanto en la pestaña, como en la misma planilla. ✓ Se debe verificar que los pesos de entrada y salida mostrados en la planilla Excel

sean los correspondientes a la etapa.

- ✓ Se deben comprobar tanto las sumas de los aportes como el precio total de la ración.

Evaluación de la Prueba: Se esperará que el 100% de las actividades no presenten errores. Sin embargo, las actividades que presenten errores deberán corregirse para que el sistema funcione adecuadamente.

Caso de Prueba “Control de acceso al iniciar el programa”
Código Caso de Uso: 11
<p>Descripción de la Prueba:</p> <p>Se verificará que el programa autentique correctamente al usuario bajos las políticas de autenticación definidas en el caso de uso.</p>
<p>Condiciones de Ejecución: El sistema donde se realizan las pruebas deberá contar con los recursos necesarios para la óptima ejecución de la prueba, es decir, acceso a la base de datos y el programa con todas sus librerías.</p> <p>Para poder ingresar, se debe haber creado previamente un usuario y contraseña en la base de datos.</p>
<p>Entrada / Pasos de Ejecución:</p> <ul style="list-style-type: none"> ✓ Se debe verificar que el programa permita el ingreso al programa con el nombre de usuario y contraseña correctos. ✓ Se debe verificar que el programa permita intentar ingresar un nombre de usuario y contraseña errada 3 veces, para luego bloquearse. ✓ Se debe comprobar que al bloquearse el programa, éste no debe permitir intentar ingresar a él durante 5 minutos por sesiones anteriores fallidas (5 minutos por la primera sesión, 10 minutos por la segunda sesión, etc.). ✓ Se debe comprobar que el programa notifique correctamente que en la sesión se han terminado los 3 intentos y se bloqueará por el tiempo correcto. ✓ Se debe comprobar que cuando se ingresa un nombre de usuario y contraseña errados el programa lo indique en la pantalla.
<p>Evaluación de la Prueba: Se esperará que el 100% de las actividades no presenten errores. Sin embargo, las actividades que presenten errores deberán corregirse para que el sistema funcione adecuadamente.</p>

Caso de Prueba “Cambiar el nombre de usuario y/o contraseña”
Código Caso de Uso: 12
<p>Descripción de la Prueba:</p> <p>Se verificará que el programa permita cambiar de nombre de usuario y contraseña correctamente.</p>
<p>Condiciones de Ejecución: El sistema donde se realizan las pruebas deberá contar con los recursos necesarios para la óptima ejecución de la prueba, es decir, acceso a la base de datos y el programa con todas sus librerías.</p> <p>Para poder cambiar el nombre de usuario y contraseña se debe haber ingresado (autenticado) al programa correctamente.</p>
<p>Entrada / Pasos de Ejecución:</p> <ul style="list-style-type: none"> ✓ Se debe verificar que el programa permita cambiar el nombre de usuario siempre y cuando el nuevo nombre de usuario no esté en blanco (vacío). ✓ Se debe verificar que se permita cambiar de contraseña siempre y cuando la contraseña no esté en blanco y que los 2 campos utilizados para ingresar la nueva contraseña coincidan en el contenido (tengan la misma contraseña) ✓ Se debe comprobar que se muestren los mensajes adecuados en caso de realizar las operaciones exitosas o cuando no permita un cambio ya sea de usuario o contraseña.
<p>Evaluación de la Prueba: Se esperará que el 100% de las actividades no presenten errores. Sin embargo, las actividades que presenten errores deberán corregirse para que el sistema funcione adecuadamente.</p>

Anexo E “Ejecución de los casos de prueba (versión Alfa)”

Control de Pruebas N° 1			
Caso de prueba revisado		Agregar etapa	
Versión del sistema		Alfa	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se selecciona una etapa anterior, bajo la cual se desea ingresar una nueva etapa.	Se crea una nueva fila bajo la etapa seleccionada por el usuario, con los campos: etapas, peso de salida, consumo de peso vivo y cantidad de animales en color rojo.	Sí
2	Se agrega una nueva etapa con un nombre ya existente.	Marca en rojo la celda donde se ingresó el nombre de la etapa.	Sí
3	Se agrega una nueva etapa con un nombre nuevo.	La celda se muestra de un color normal, es decir no presenta errores.	Sí
4	Se agrega un peso de ingreso - 500 y salida "234dadf".	Marca en rojo las celdas donde se ingreso un número negativo y letras.	Sí
5	Se agrega un peso de ingreso de 300 y un peso de salida de 200.	Marca en rojo la celda peso de salida e indica mediante un tooltip "el peso de ingreso no puede ser superior al peso de salida".	Sí
6	Se agrega un peso de ingreso 250 y un peso de salida de 500.	La celda se muestra de un color normal, es decir no presenta errores.	Sí
7	Se agrega una cantidad de animales -700.	Marca en rojo la celda donde se ingreso un número negativo.	Sí
8	Se agrega una cantidad de	La celda se muestra de un color	Sí

	animales 700.	normal, es decir no presenta errores.	
9	Se ingresa un porcentaje de consumo de peso vivo -4.	Marca en rojo la celda porcentaje consumo peso vivo e indica mediante un tooltip "el porcentaje debe estar entre 0 y 100".	Sí
10	Se ingresa un porcentaje de consumo de peso vivo igual a 200.	Marca en rojo la celda porcentaje consumo peso vivo e indica mediante un tooltip "el porcentaje debe estar entre 0 y 100".	Sí
11	Se ingresa un porcentaje de consumo de peso vivo igual a 4	La celda se muestra de un color normal, es decir no presenta errores.	Sí
12	Se agrega una nueva etapa "etapa4"	La etapa queda correctamente guardada en la base de datos	Sí
13	Se presiona la pestaña "Gestión específica de raciones y restricciones" y se comprueba que en la sección etapas, esté la etapa ingresada correctamente en la pestaña "Gestión general de etapas".	En la pestaña "Gestión específica de raciones y restricciones" se encuentra la etapa ingresada en la pestaña "Gestión general de etapas".	Sí
14	Se intenta salir de la pestaña cuando existen celdas marcadas en rojo, es decir con errores.	Entrega un mensaje de error "Existen errores, por favor ingrese datos válidos en las celdas marcadas con rojo".	Sí
Correcciones: No se debe llevar a cabo ninguna corrección.			

Control de Pruebas N° 2			
Caso de prueba revisado		Eliminar etapa	
Versión del sistema		Alfa	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se selecciona una etapa y se presiona el botón eliminar	Se muestra un mensaje de confirmación “Esta seguro que desea eliminar las siguientes etapas” y a continuación muestra el nombre de la etapa que se desea eliminar.	Sí
2	Se presiona sí, después del mensaje presentado en 1.	La etapa ya no se muestra en la interfaz.	Sí
3	Se presiona si después del mensaje presentado en 1.	La etapa es marcada como no vigente en la base de datos.	Sí
4	Se selecciona una etapa que tiene errores para ser eliminada.	Se muestra un mensaje de confirmación “Esta seguro que desea eliminar las siguientes etapas” y a continuación muestra el nombre de la etapa que se desea eliminar.	Sí
5	Se presiona si después del mensaje presentado en 4.	La etapa seleccionada es eliminada de la interfaz y es posible cambiarse de pestaña.	Sí
6	Se presiona la pestaña “Gestión específica de raciones y restricciones” y se comprueba que en la sección etapas, no esté la etapa eliminada en la pestaña “Gestión general de etapas”.	En la pestaña “Gestión específica de raciones y restricciones” no se encuentra la etapa ingresada en la pestaña “Gestión general de etapas”.	Sí

7	Se realiza una optimización luego de eliminar una etapa.	La etapa eliminada no está en la optimización.	Sí
<i>Correcciones: No se debe llevar a cabo ninguna corrección.</i>			

Control de Pruebas N° 3			
Caso de prueba revisado		Eliminar composición	
Versión del sistema		Alfa	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se selecciona una compuesto y se presiona el botón eliminar.	Se muestra un mensaje de confirmación “Esta seguro que desea eliminar” y a continuación muestra el nombre del compuesto seleccionado.	Sí
2	Se presiona si después del mensaje presentado en 1.	El compuesto ya no se muestra en la interfaz.	Sí
3	Se presiona si después del mensaje presentado en 1.	El compuesto es marcado como no vigente en la base de datos.	Sí
4	Se presiona la pestaña “Gestión de precio y stock” y se comprueba que en la sección compuestos, no esté el compuesto eliminado.	El compuesto eliminado no se encuentra en la pantalla “Gestión de precio y stock”, sin embargo se produce un error y no se puede salir de esta pestaña.	No
5	Se presiona la pestaña “Gestión específica de raciones y restricciones” y se comprueba que en la sección compuestos, no esté el compuesto eliminado.	El compuesto eliminado no se encuentra en la pantalla “Gestión específica de raciones y restricciones”.	Sí
Correcciones:			
N°	Correcciones		
4	Corregir la función “modificarVentana” de la clase Gestión de precio y stock, para que al cargar la pestaña “Gestión de precio y stock”, luego de eliminar un compuesto, ésta permita salir de ella.		

Control de Pruebas N° 4			
Caso de prueba revisado		Agregar nueva composición.	
Versión del sistema		Alfa	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se selecciona un compuesto y se presiona el botón agregar composición.	Se crea una nueva fila, con los campos en color rojo, debido a que no existen valores.	Sí
2	Se agrega una nueva composición con un nombre ya existente.	Marca en rojo la celda donde se ingresó el nombre de la composición.	Sí
3	Se agrega una nueva composición con un nombre nuevo.	La celda se muestra de un color normal, es decir no presenta errores.	Sí
4	Se ingresan los siguientes valores para los aportes nutricionales: 1, -6, 100, 0.8, 2, -7.8, 0 y “aporte”.	Las celdas donde se ingresaron los números 1, 100, 0.8, 2 y 0 se muestra de un color normal, es decir no presentar errores. Sin embargo, en las celdas donde se ingresaron los datos -6,-7.8 y “aporte” se muestran en color rojo, puesto que son números negativos o valores no numericos.	Sí
5	Se presiona la pestaña “Gestión de precio y stock” y se comprueba que en la sección compuestos, esté el compuesto agregado.	El compuesto agregado se encuentra en la pantalla “Gestión de precio y stock”.	Sí
6	Se presiona la pestaña “Gestión	El compuesto agregado se	Sí

	especifica de raciones y restricciones” y se comprueba que en la sección compuestos, esté el compuesto agregado.	encuentra en la pantalla “Gestión específica de raciones y restricciones”.	
7	Se verifica luego de realizada la optimización y de haber apretado el botón guardar, que el compuesto agregado esté en la base de datos.	El compuesto agregado si se encuentra almacenado en la base de datos en un estado vigente.	Sí
8	Se verifica si la nueva composición es tomada como entrada en el módulo de optimización, con sus respectivos aportes, siempre y cuando haya sido seleccionado en la pestaña “Gestión específica de raciones y restricciones”.	El compuesto sí es recibido como parámetro de entrada al módulo de optimización, cuando la composición es seleccionada en la pestaña “Gestión específica de raciones y restricciones”.	Sí
9	Se ingresan -1 en todos los aportes nutricionales para un nuevo compuesto y se intenta cambiar de pestaña.	El programa no permite cambiar de pestaña, muestra el mensaje: “Existen errores, por favor ingrese datos válidos en las celdas marcadas con rojo”.	Sí
Correcciones: No se debe llevar a cabo ninguna corrección.			

Control de Pruebas N° 5			
Caso de prueba revisado		Modificar aporte nutricional	
Versión del sistema		Alfa	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se modifican distintos aportes nutricionales, los valores ingresados son: 1, -6, 100, 0.8, 2, -7.8, 0 y un string “prueba”.	Las celdas donde se ingresaron los números 1, 100, 0.8, 2 y 0 se muestra de un color normal, es decir no presentar errores. Sin embargo, en las celdas donde se ingresaron los valores -6,-7.8 y “prueba” se muestran en color rojo, puesto que son números negativos y letras.	Sí
2	Se verifica luego de realizada la optimización y de haber apretado el botón guardar que los aportes modificados estén en la base de datos.	Los aportes nutricionales modificados quedan registrados en la base de datos.	Sí
3	Se intenta salir de la pestaña cuando existen celdas marcadas en rojo, es decir con errores.	Entrega un mensaje de error “Existen errores, por favor ingrese datos válidos en las celdas marcadas con rojo” y es imposible cambiarse de pestaña.	Sí
Correcciones: No se debe llevar a cabo ninguna corrección.			

Control de Pruebas N° 6			
Caso de prueba revisado		Modificar precio	
Versión del sistema		Alfa	
Cumple con compilación			
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se modifican distintos precios, los valores ingresados son: 30, -6, 100, 0.8, 2, -7.8, 0 y un string "prueba".	Las celdas donde se ingresaron los números 30, 100, 0.8, 2 y 0 se muestra de un color normal, es decir no presentan errores. Sin embargo, en las celdas donde se ingresaron los valores -6, -7.8 y "prueba" se muestran en color rojo, puesto que son números negativos y letras. Existe un error puesto que acepta el valor 0 y este debería ser marcado en rojo.	No
2	Se verifica luego de realizada la optimización y de haber apretado el botón guardar, que los precios modificados estén en la base de datos.	El precio modificado queda registrado en la base de datos.	Sí
3	Se intenta salir de la pestaña cuando existen celdas marcadas en rojo, es decir con errores.	Entrega un mensaje de error "Existen errores, por favor ingrese datos válidos en las celdas marcadas con rojo" y es imposible cambiarse de pestaña.	Sí
4	Se verifica que el nuevo precio modificado sea utilizado en la optimización	El precio es utilizado en la optimización.	Sí

Correcciones: No se debe llevar a cabo ninguna corrección.

N°	Correcciones
1	Se debe validar que el valor de un precio debe ser mayor que 0

Control de Pruebas N° 7			
Caso de prueba revisado		Modificar stock	
Versión del sistema		Alfa	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se modifican distintos stock, los valores ingresados son: 30000, -6, 100, 0.8, 2, -7.8, 0, un string "prueba" y "".	Las celdas donde se ingresaron los valores 30000, 100, 0.8, 2, 0 y "" se muestran de un color normal, es decir no presentan errores. Sin embargo, en las celdas donde se ingresaron los valores -6,-7.8 y "prueba" se muestran en color rojo, puesto que son números negativos y letras.	Sí
2	Se verifica que el stock sea considerado a la hora de realizar el cálculo de la duración de la ración, siempre y cuando, el compuesto, al cual se le aplicó la restricción de stock, participe en el cálculo de la ración.	El stock es considerado para el cálculo de la duración de la ración.	Sí
3	Se intenta salir de la pestaña cuando existen celdas marcadas en rojo, es decir con errores.	Entrega un mensaje de error "Existen errores, por favor ingrese datos válidos en las celdas marcadas con rojo" y es imposible cambiarse de pestaña.	Sí
4	Se verifica, luego de realizada la optimización y de haber apretado el botón guardar que el stock	El stock es almacenado en la base de datos.	Sí

	modificado esté almacenado en la base de datos.		
<i>Correcciones: No se debe llevar a cabo ninguna corrección.</i>			

Control de Pruebas N° 8			
Caso de prueba revisado		Modificar restricciones de cantidad de compuesto.	
Versión del sistema		Alfa	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Para el compuesto vinaza se le asignó la restricción mínima -6 y máxima -7.8	Las celdas fueron marcadas con error.	Sí
2	Para el compuesto e. maíz se le asignó la restricción “prueba” y “”	La celda con el string “prueba” fue marcada en rojo, sin embargo la celda con el string “” fue marcada como correcta.	Sí
3	Para el compuesto Maíz grano húmedo se le asignó la restricción mínima 5 y máxima 4	La celda con la restricción máxima 4 es marcada en rojo y muestra un tooltip con el mensaje “La restricción mínima no puede ser mayor que la máxima”	Sí
4	Se verifica que al modificar una restricción de valor 6 por una de valor 3 esto se vea reflejado en la interfaz.	Al modificar el valor de una restricción se ve reflejado en la interfaz.	Sí
5	Se intenta salir de la pestaña cuando existen celdas marcadas en rojo, es decir con errores.	Entrega un mensaje de error “Existen errores, por favor ingrese datos válidos en las celdas marcadas con rojo” y es imposible cambiarse de pestaña.	Sí
6	Se verifica luego de realizada la optimización y de haber apretado	Las restricciones son almacenadas en la base de datos.	Sí

	el botón guardar que la restricción modificada esté almacenada en la base de datos.		
7	Se agrega para vinaza la restricción mínima 6 y máximo 9.	El programa muestra las restricciones y no marca ninguna con error.	Sí
8	Se agrega para e. maíz una restricción mínima de 4 y se deja en blanco la restricción máxima.	El programa muestra las restricciones y no marca ninguna con error.	Sí
9	Se agrega para Coseta Orafti la restricción máxima de 5.3 y se deja en blanco la restricción mínima.	El programa muestra las restricciones y no marca ninguna con error.	Sí
<i>Correcciones: No se debe llevar a cabo ninguna corrección.</i>			

Control de Pruebas N° 9			
Caso de prueba revisado		Eliminar restricciones de cantidad de compuesto.	
Versión del sistema		Alfa	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se ingresan los valores para la restricción mínima 5 y máxima 4 y se eliminan estas restricciones con error	El programa si permite eliminar la restricción	Sí
2	Se selecciona un compuesto y se le asignan restricciones mínima 5 y máxima 4 y luego se elimina haciendo clic en el checkbox del compuesto.	El programa permite la eliminación de la fila de restricciones, sin embargo al intentar cambiarse de pestaña lanza un mensaje de error "Existen errores, por favor ingrese datos válidos en las celdas marcadas en rojo"	No
3	Se agrega una restricción al compuesto E.maiz mínima de 4 kg y luego se elimina.	El programa no considera la restricción al optimizar	Sí
Correcciones			
N°	Correcciones		
2	Si una fila tiene un error y se elimina, se deben también eliminar los errores, para que estos no sean considerados a la hora de hacer el cambio de pestaña.		

Control de Pruebas N° 10			
Caso de prueba revisado		Agregar restricciones de cantidad de compuesto.	
Versión del sistema		Alfa	
Cumple con compilación			
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se hace clic en el pestaña “Gestión específica de raciones y restricciones” para verificar la existencia del espacio para agregar una restricción	El programa muestra el espacio (tabla) para ingresar la restricción.	Sí
2	Se selecciona un compuesto, haciendo clic en su checkbox correspondiente.	El programa agrega el nombre el compuesto en la tabla inferior para ingresarle las respectivas restricciones. También marca el checkbox dentro de la casilla de “compuestos a utilizar”	Sí
3	Se le hace clic al compuesto vinaza y se le asigna la restricción mínima -6 y máxima -7.8	Las celdas fueron marcadas con error.	Sí
4	Para el compuesto e. maíz se le asignó la restricción “prueba” y “”	La celda con el string “prueba” fue marcada en rojo, sin embargo la celda con el string “” fue marcada como correcta.	Sí
5	Para el compuesto Maíz grano húmedo se le asignó la restricción mínima 5 y máxima 4	La celda con la restricción máxima 4 es marcada en rojo y muestra un tooltip con el mensaje “La restricción mínima no puede ser mayor que la	Sí

		máxima”	
6	Se verificará que las restricciones sean consideradas al realizar una optimización. A al compuesto coseta orafti se le impone una restricción mínima de 5 kg	El programa la considera y aparece el valor señalado como parte de la ración	Sí
7	Se selecciona un compuesto para agregarle una restricción y éstas son dejadas en blanco	El programa no lo marca como error pero tampoco las considera en la optimización.	Sí
8	Se asigna una restricción de -5 al compuesto vinaza y luego se hace clic en la pestaña de “optimización y reportes”	El programa indica que existen errores y deben arreglarse antes de poder cambiar de pestaña.	Sí
<i>Correcciones: No existen correcciones</i>			

Control de Pruebas N° 11			
Caso de prueba revisado		Seleccionar los compuestos que participan en cada etapa	
Versión del sistema		Alfa	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se hace clic en la pestaña “Gestión específica de raciones y restricciones”.	El programa NO muestra en “ <i>compuestos disponibles</i> ” todos los compuestos agregados y cargados en la pestaña “Gestión de compuestos y aportes”	No
2	Se eliminan en la pestaña “Gestión específica de raciones y restricciones” los compuestos: Mezcla Soprodu y Coseta Orafti	El programa no muestra en “ <i>compuestos disponibles</i> ” los compuestos que fueron eliminados en “Gestión de compuestos y aportes”.	Sí
3	Se selecciona Sal Mollendo y Ensilaje de pradera. Luego se deselecta Ensilaje de pradera. Después se seleccionan todos los compuestos. Finalmente se deselectan todos.	El programa muestra, al seleccionar uno o muchos compuestos, los nombres de los compuestos en “ <i>compuestos a utilizar</i> ” y los quita de “ <i>compuestos disponibles</i> ” al deselectarlos.	Sí
4	No se selecciona ningún compuesto en la etapa 1.	El programa marca con un recuadro rojo la sección de “ <i>compuestos a utilizar</i> ” y no deja cambiarse de pestaña.	Sí
Correcciones: No existen correcciones			
N°	Correcciones		
1	Se debe recargar correctamente la pestaña y agregar todos los compuestos		

	disponibles. Además se deben eliminar aquellos compuestos que fueron eliminados en la pestaña “Gestión de compuestos y aportes”
--	---

Control de Pruebas N° 12			
Caso de prueba revisado		Modificar restricciones mínimas nutricionales por etapa.	
Versión del sistema		Alfa	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se ingresan los siguientes requisitos nutricionales mínimos: MS: -8 PT: "asdlfkj" TDN:"" (vacío) MCAL:1.0 FDN: ""(vacío) CA: 0.8 P: 11000	El programa marca en rojo las celdas de: MS,PT,TDN,FDN. El resto de las celdas son consideradas sin error y no permite cambiar de pestaña.	Sí
2	Se le asignará como requisito mínimo 0.5 a todos los campos de "requisitos nutricionales mínimos" para la etapa de recría (única etapa en esta prueba), luego se procederá a optimizar y a guardar la ración creada.	El programa permite optimizar la ración y guardarla. La suma de los aportes nutricionales de la ración, para cada nutriente es efectivamente mayor o igual que el señalado como restricción mínima nutricional. Al revisar la base de datos los requisitos mínimos ingresados se encuentran correctamente guardados.	Sí
Correcciones: No existen correcciones			

Control de Pruebas N° 13			
Caso de prueba revisado		Determinar las razones por cada etapa que minimice el costo total.	
Versión del sistema		Alfa	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se ingresaron los datos mostrados en el Anexo G	El programa retornó lo que se muestra en el Anexo G. Los resultados obtenidos fueron comparados con un programa específico de optimización llamado LINGO ¹⁴ y concuerdan con lo que éste último entrega. Los datos se almacenan correctamente en la base de datos.	Sí
2	Se ingresaron datos insuficientes (se seleccionó un compuesto por etapa únicamente) para la realización de una optimización.	El programa indica, al intentar realizar la optimización que el problema ingresado no tiene solución.	Sí
Correcciones: No existen correcciones			

¹⁴ LINGO es una herramienta diseñada para modelar y resolver ecuaciones lineales, no lineales (convexas, no convexas/globales), cuadráticas, restringidas cuadráticamente, estocásticas, etc. En este caso sólo se utilizó una versión de prueba para resolver problemas de programación lineal.

Control de Pruebas N° 14			
Caso de prueba revisado		Obtener reportes históricos de raciones anteriores.	
Versión del sistema		Alfa	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se crean 4 planes de alimentación diferentes, con los nombres: Plan 1, Plan 2, Plan 3, Plan4. Después se hace clic en la pestaña reportes históricos y se verifica la existencia de los 4 planes agregados por nombre y por fecha.	El programa muestra los 4 planes de alimentación ingresados previamente por nombre y por fecha.	Sí
2	Con los planes anteriores, se le pide al programa crear los reportes en Excel.	El programa crea los reportes y los guarda en el lugar que el usuario indica. Todos los reportes tienen los compuestos seleccionados para participar en las raciones en las distintas etapas. Las cantidades de los compuestos concuerdan con los de la optimización. Los aportes nutricionales concuerdan con los ingresados para la generación de la ración. Los precios mostrados son correctos y la suma de éstos entrega el precio total de la ración correctamente.	Sí

		<p>La ración generada cumple con los requisitos nutricionales mínimos necesarios.</p> <p>La fecha de creación concuerda con la fecha actual.</p> <p>La cantidad de pestañas del archivo Excel concuerda con la cantidad de etapas y con sus respectivos nombres.</p> <p>Los requerimientos mínimos nutricionales de las etapas ingresadas corresponden al nombre de la etapa que se muestra en la planilla.</p> <p>Los pesos de entrada y salida corresponden a los que se ingresaron en el programa.</p> <p>La suma de los aportes nutricionales por cada requisito nutricional es correcta.</p>	
<p><i>Correcciones: No existen correcciones</i></p>			

Control de Pruebas N° 15			
Caso de prueba revisado		Control de acceso al iniciar el programa	
Versión del sistema		Alfa	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se ingresa con el nombre de usuario “mollendo” y contraseña “wagyu” (ambos correctos)	El programa deja ingresar a la ventana principal y se inicia correctamente.	Sí
2	Se ingresa el nombre de usuario incorrecto y la contraseña incorrecta una vez u luego se ingresan el nombre de usuario y contraseña correctamente.	El programa muestra en pantalla un mensaje que dice: “nombre de usuario y contraseña incorrectos” la primera vez. La segunda vez si deja ingresar a la ventana principal.	Sí
3	Se ingresan incorrectamente el nombre de usuario y contraseña 3 veces (en la primera sesión)	El programa indica que se cerrará por 5 minutos	Sí
4	Tras haber intentado ingresar 3 veces en una sesión y cerrarse el programa. Se intenta ingresar antes de haber pasado el tiempo indicado por el programa	El programa muestra un mensaje que indica que aún no han transcurrido los minutos necesarios e indica aproximadamente cuantos minutos faltan para poder intentar ingresar nuevamente.	Sí
5	Tras haber intentado ingresar sin éxito en una sesión se espera el tiempo indicado y se ingresa con el nombre de usuario y contraseña correctos	El programa permite ingresar correctamente a la ventana principal del mismo.	Sí

6	Se intenta ingresar fallidamente en 2 sesiones consecutivas.	El programa muestra que se cerrará por 10 minutos.	Sí
7	Se intenta ingresar tras 2 sesiones fallidas para luego ingresar con el nombre de contraseña y usuario correctos.	El programa permite ingresar correctamente a la ventana principal del mismo.	Sí
<i>Correcciones: No existen correcciones</i>			

Control de Pruebas N° 16			
Caso de prueba revisado		Cambio de nombre de usuario y contraseña	
Versión del sistema		Alfa	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se intenta cambiar el nombre de usuario por uno vacío	El programa notifica que no se puede asignar nombre de usuario vacío	Sí
2	Se cambia el nombre de usuario por “agrícola Mollendo”	El programa permite el cambio de usuario correctamente.	Sí
3	Una vez cambiado el nombre de usuario, se sale del programa y se vuelve a ingresar con el nuevo nombre de usuario.	El programa deja entrar a la pantalla principal con el nuevo nombre de usuario.	Sí
4	Se intenta cambiar la contraseña poniendo los campos vacíos.	El programa no permite asignar una contraseña vacía al programa y lo notifica.	Sí
5	Se intenta cambiar la contraseña poniendo en los 2 campos contraseñas diferentes.	El programa no permite cambiar la contraseña e indica que ambos campos no son idénticos.	Sí
6	Se cambia la contraseña poniendo “mollendo2” en ambos campos.	El programa permite cambiar la contraseña e indica que se ha cambiado correctamente	Sí
7	Tras haber cambiado la contraseña, se sale del programa y se ingresa nuevamente con la última contraseña asignada.	El programa permite ingresar correctamente a la ventana principal del mismo.	Sí
Correcciones: No existen correcciones			

Anexo F “Ejecución de los casos de prueba (versión Beta)”

Control de Pruebas N° 1			
Caso de prueba revisado		Agregar etapa	
Versión del sistema		Beta	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se selecciona una etapa anterior, bajo la cual se desea ingresar una nueva etapa.	Se crea una nueva fila bajo la etapa seleccionada por el usuario, con los campos: etapas, peso de salida, consumo de peso vivo y cantidad de animales en color rojo.	Sí
2	Se agrega una nueva etapa con un nombre ya existente.	Marca en rojo la celda donde se ingresó el nombre de la etapa.	Sí
3	Se agrega una nueva etapa con un nombre nuevo.	La celda se muestra de un color normal, es decir no presenta errores.	Sí
4	Se agrega un peso de ingreso - 500 y salida "234dadf".	Marca en rojo las celdas donde se ingreso un número negativo y letras.	Sí
5	Se agrega un peso de ingreso de 300 y un peso de salida de 200.	Marca en rojo la celda peso de salida e indica mediante un tooltiptext "el peso de ingreso no puede ser superior al peso de salida".	Sí
6	Se agrega un peso de ingreso 250 y un peso de salida de 500.	La celda se muestra de un color normal, es decir no presenta errores.	Sí
7	Se agrega una cantidad de animales -700.	Marca en rojo la celda donde se ingreso un número negativo.	Sí
8	Se agrega una cantidad de	La celda se muestra de un color	Sí

	animales 700.	normal, es decir no presenta errores.	
9	Se ingresa un porcentaje de consumo de peso vivo -4.	Marca en rojo la celda porcentaje consumo peso vivo e indica mediante un tooltip "el porcentaje debe estar entre 0 y 100".	Sí
10	Se ingresa un porcentaje de consumo de peso vivo igual a 200.	Marca en rojo la celda porcentaje consumo peso vivo e indica mediante un tooltip "el porcentaje debe estar entre 0 y 100".	Sí
11	Se ingresa un porcentaje de consumo de peso vivo igual a 4	La celda se muestra de un color normal, es decir no presenta errores.	Sí
12	Se agrega una nueva etapa "etapa4"	La etapa queda correctamente guardada en la base de datos	Sí
13	Se presiona la pestaña "Gestión específica de raciones y restricciones" y se comprueba que en la sección etapas, esté la etapa ingresada correctamente en la pestaña "Gestión general de etapas".	En la pestaña "Gestión específica de raciones y restricciones" se encuentra la etapa ingresada en la pestaña "Gestión general de etapas".	Sí
14	Se intenta salir de la pestaña cuando existen celdas marcadas en rojo, es decir con errores.	Entrega un mensaje de error "Existen errores, por favor ingrese datos válidos en las celdas marcadas con rojo".	Sí
Correcciones: No se debe llevar a cabo ninguna corrección.			

Control de Pruebas N° 2			
Caso de prueba revisado		Eliminar etapa	
Versión del sistema		Beta	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se selecciona una etapa y se presiona el botón eliminar	Se muestra un mensaje de confirmación “Esta seguro que desea eliminar las siguientes etapas” y a continuación muestra el nombre de la etapa que se desea eliminar.	Sí
2	Se presiona si después del mensaje presentado en 1.	La etapa ya no se muestra en la interfaz.	Sí
3	Se presiona si después del mensaje presentado en 1.	La etapa es marcada como no vigente en la base de datos.	Sí
4	Se selecciona una etapa que tiene errores para ser eliminada.	Se muestra un mensaje de confirmación “Esta seguro que desea eliminar las siguientes etapas” y a continuación muestra el nombre de la etapa que se desea eliminar.	Sí
5	Se presiona si después del mensaje presentado en 4.	La etapa seleccionada es eliminada de la interfaz y es posible cambiarse de pestaña.	Sí
6	Se presiona la pestaña “Gestión específica de raciones y restricciones” y se comprueba que en la sección etapas, no esté la etapa eliminada en la pestaña “Gestión general de etapas”.	En la pestaña “Gestión específica de raciones y restricciones” no se encuentra la etapa ingresada en la pestaña “Gestión general de etapas”.	Sí

7	Se realiza una optimización luego de eliminar una etapa.	La etapa eliminada no está en la optimización.	Sí
<i>Correcciones: No se debe llevar a cabo ninguna corrección.</i>			

Control de Pruebas N° 3			
Caso de prueba revisado		Eliminar composición	
Versión del sistema		Beta	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se selecciona una compuesto y se presiona el botón eliminar.	Se muestra un mensaje de confirmación “Esta seguro que desea eliminar” y a continuación muestra el nombre del compuesto seleccionado.	Sí
2	Se presiona si después del mensaje presentado en 1.	El compuesto ya no se muestra en la interfaz.	Sí
3	Se presiona si después del mensaje presentado en 1.	El compuesto es marcado como no vigente en la base de datos.	Sí
4	Se presiona la pestaña “Gestión de precio y stock” y se comprueba que en la sección compuestos, no esté el compuesto eliminado.	El compuesto eliminado no se encuentra en la pantalla “Gestión de precio y stock”, y luego permite cambiarse de pestaña.	Sí
5	Se presiona la pestaña “Gestión específica de raciones y restricciones” y se comprueba que en la sección compuestos, no esté el compuesto eliminado.	El compuesto eliminado no se encuentra en la pantalla “Gestión específica de raciones y restricciones”.	Sí
Correcciones: No se debe llevar a cabo ninguna corrección.			

Control de Pruebas N° 4			
Caso de prueba revisado		Agregar nueva composición.	
Versión del sistema		Beta	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se selecciona un compuesto y se presiona el botón agregar composición.	Se crea una nueva fila, con los campos en color rojo, debido a que no existen valores.	Sí
2	Se agrega una nueva composición con un nombre ya existente.	Marca en rojo la celda donde se ingresó el nombre de la composición.	Sí
3	Se agrega una nueva composición con un nombre nuevo.	La celda se muestra de un color normal, es decir no presenta errores.	Sí
4	Se ingresan los siguientes valores para los aportes nutricionales: 1, -6, 100, 0.8, 2, -7.8, 0 y “aporte”.	Las celdas donde se ingresaron los números 1, 100, 0.8, 2 y 0 se muestra de un color normal, es decir no presentar errores. Sin embargo, en las celdas donde se ingresaron los datos -6,-7.8 y “aporte” se muestran en color rojo, puesto que son números negativos.	Sí
5	Se presiona la pestaña “Gestión de precio y stock” y se comprueba que en la sección compuestos, esté el compuesto agregado.	El compuesto agregado se encuentra en la pantalla “Gestión de precio y stock”.	Sí
6	Se presiona la pestaña “Gestión específica de raciones y	El compuesto agregado se encuentra en la pantalla	Sí

	restricciones” y se comprueba que en la sección compuestos, esté el compuesto agregado.	“Gestión específica de raciones y restricciones”.	
7	Se verifica luego de realizada la optimización y de haber apretado el botón guardar, que el compuesto agregado esté en la base de datos.	El compuesto agregado si se encuentra almacenado en la base de datos en un estado vigente.	Sí
8	Se verifica si la nueva composición es tomada como entrada en el módulo de optimización, con sus respectivos aportes, siempre y cuando haya sido seleccionado en la pestaña “Gestión específica de raciones y restricciones”.	El compuesto sí es recibido como parámetro de entrada al módulo de optimización, cuando la composición es seleccionada en la pestaña “Gestión específica de raciones y restricciones”.	Sí
9	Se ingresan -1 en todos los aportes nutricionales para un nuevo compuesto y se intenta cambiar de pestaña.	El programa no permite cambiar de pestaña, muestra el mensaje: “Existen errores, por favor ingrese datos válidos en las celdas marcadas con rojo”.	Sí
<i>Correcciones: No se debe llevar a cabo ninguna corrección.</i>			

Control de Pruebas N° 5			
Caso de prueba revisado		Modificar aporte nutricional	
Versión del sistema		Beta	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se modifican distintos aportes nutricionales, los valores ingresados son: 1, -6, 100, 0.8, 2, -7.8, 0 y un string “prueba”.	Las celdas donde se ingresaron los números 1, 100, 0.8, 2 y 0 se muestra de un color normal, es decir no presentar errores. Sin embargo, en las celdas donde se ingresaron los valores -6,-7.8 y “prueba” se muestran en color rojo, puesto que son números negativos y letras.	Sí
2	Se verifica luego de realizada la optimización y de haber apretado el botón guardar que los aportes modificados estén en la base de datos.	Los aportes nutricionales modificados quedan registrados en la base de datos.	Sí
3	Se intenta salir de la pestaña cuando existen celdas marcadas en rojo, es decir con errores.	Entrega un mensaje de error “Existen errores, por favor ingrese datos válidos en las celdas marcadas con rojo” y es imposible cambiarse de pestaña.	Sí
Correcciones: No se debe llevar a cabo ninguna corrección.			

Control de Pruebas N° 6			
Caso de prueba revisado		Modificar precio	
Versión del sistema		Beta	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se modifican distintos precios, los valores ingresados son: 30, -6, 100, 0.8, 2, -7.8, 0 y un string "prueba".	Las celdas donde se ingresaron los números 30, 100, 0.8 y 2 se muestran de un color normal, es decir no presentan errores. Sin embargo, en las celdas donde se ingresaron los valores -6,-7.8, 0 y "prueba" se muestran en color rojo, puesto que son números negativos y letras.	Sí
2	Se verifica luego de realizada la optimización y de haber apretado el botón guardar que los precios modificados estén en la base de datos.	El precio modificado queda registrado en la base de datos.	Sí
3	Se intenta salir de la pestaña cuando existen celdas marcadas en rojo, es decir con errores.	Entrega un mensaje de error "Existen errores, por favor ingrese datos válidos en las celdas marcadas con rojo" y es imposible cambiarse de pestaña.	Sí
4	Se verifica que el nuevo precio modificado sea utilizado en la optimización	El precio es utilizado en la optimización.	Sí
Correcciones: No se debe llevar a cabo ninguna corrección.			

Control de Pruebas N° 7			
Caso de prueba revisado		Modificar stock	
Versión del sistema		Beta	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se modifican distintos stock, los valores ingresados son: 30000, -6, 100, 0.8, 2, -7.8, 0, un string "prueba" y "".	Las celdas donde se ingresaron los valores 30000, 100, 0.8, 2, 0 y "" se muestran de un color normal, es decir no presentan errores. Sin embargo, en las celdas donde se ingresaron los valores -6,-7.8 y "prueba" se muestran en color rojo, puesto que son números negativos y letras.	Sí
2	Se verifica que el stock sea considerado a la hora de realizar el cálculo de la duración de la ración, siempre y cuando, el compuesto, al cual se le aplicó la restricción de stock, participe en el cálculo de la ración.	El stock es considerado para el cálculo de la duración de la ración.	Sí
3	Se intenta salir de la pestaña cuando existen celdas marcadas en rojo, es decir con errores.	Entrega un mensaje de error "Existen errores, por favor ingrese datos válidos en las celdas marcadas con rojo" y es imposible cambiarse de pestaña.	Sí
4	Se verifica luego de realizada la optimización y de haber apretado el botón guardar que el stock	El stock es almacenado en la base de datos.	Sí

	modificado esté almacenado en la base de datos.		
<i>Correcciones: No se debe llevar a cabo ninguna corrección.</i>			

Control de Pruebas N° 8			
Caso de prueba revisado		Modificar restricciones de cantidad de compuesto.	
Versión del sistema		Beta	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Para el compuesto vinaza se le asignó la restricción mínima -6 y máxima -7.8	Las celdas fueron marcadas con error.	Sí
2	Para el compuesto e. maíz se le asignó la restricción “prueba” y “”	La celda con el string “prueba” fue marcada en rojo, sin embargo la celda con el string “” fue marcada como correcta.	Sí
3	Para el compuesto Maíz grano húmedo se le asignó la restricción mínima 5 y máxima 4	La celda con la restricción máxima 4 es marcada en rojo y muestra un tooltip con el mensaje “La restricción mínima no puede ser mayor que la máxima”	Sí
4	Se verifica que al modificar una restricción de valor 6 por una de valor 3 esto se vea reflejado en la interfaz.	Al modificar el valor de una restricción se ve reflejado en la interfaz.	Sí
5	Se intenta salir de la pestaña cuando existen celdas marcadas en rojo, es decir con errores.	Entrega un mensaje de error “Existen errores, por favor ingrese datos válidos en las celdas marcadas con rojo” y es imposible cambiarse de pestaña.	Sí
6	Se verifica luego de realizada la optimización y de haber apretado	Las restricciones son almacenadas en la base de datos.	Sí

	el botón guardar que la restricción modificada esté almacenado en la base de datos.		
7	Se agrega para vinaza la restricción mínima 6 y máximo 9.	El programa muestra las restricciones y no marca ninguna con error.	Sí
8	Se agrega para e. maíz una restricción mínima de 4 y se deja en blanco la restricción máxima.	El programa muestra las restricciones y no marca ninguna con error.	Sí
9	Se agrega para Coseta Orafti la restricción máxima de 5.3 y se deja en blanco la restricción mínima.	El programa muestra las restricciones y no marca ninguna con error.	Sí
<i>Correcciones: No se debe llevar a cabo ninguna corrección.</i>			

Control de Pruebas N° 9			
Caso de prueba revisado		Eliminar restricciones de cantidad de compuesto.	
Versión del sistema		Beta	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se ingresan los valores para la restricción mínima 5 y máxima 4 y se eliminan estas restricciones con error	El programa si permite eliminar la restricción	Sí
2	Se selecciona un compuesto y se le asignan restricciones mínima 5 y máxima 4 y luego se elimina haciendo clic en el checkbox del compuesto.	El programa permite la eliminación de la fila de restricciones y permite el cambio de pestaña correctamente.	Sí
3	Se agrega una restricción al compuesto E.maiz mínima de 4 kg y luego se elimina.	El programa no considera la restricción al optimizar	Sí
Correcciones: No se debe llevar a cabo ninguna corrección.			

Control de Pruebas N° 10			
Caso de prueba revisado		Agregar restricciones de cantidad de compuesto.	
Versión del sistema		Beta	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se hace clic en el pestaña “Gestión específica de raciones y restricciones” para verificar la existencia del espacio para agregar una restricción	El programa muestra el espacio (tabla) para ingresar la restricción.	Sí
2	Se selecciona un compuesto, haciendo clic en su checkbox correspondiente.	El programa agrega el nombre el compuesto en la tabla inferior para ingresarle las respectivas restricciones. También marca el checkbox dentro de la casilla de “compuestos a utilizar”	Sí
3	Se le hace clic al compuesto vinaza y se le asigna la restricción mínima -6 y máxima -7.8	Las celdas fueron marcadas con error.	Sí
4	Para el compuesto e. maíz se le asignó la restricción “prueba” y “”	La celda con el string “prueba” fue marcada en rojo, sin embargo la celda con el string “” fue marcada como correcta.	Sí
5	Para el compuesto Maíz grano húmedo se le asignó la restricción mínima 5 y máxima 4	La celda con la restricción máxima 4 es marcada en rojo y muestra un tooltip con el mensaje “La restricción mínima no puede ser mayor que la	Sí

		máxima”	
6	Se verificará que las restricciones sean consideradas al realizar una optimización. A al compuesto coseta orafti se le impone una restricción mínima de 5 kg	El programa la considera y aparece el valor señalado como parte de la ración	Sí
7	Se selecciona un compuesto para agregarle una restricción y éstas son dejadas en blanco	El programa no lo marca como error pero tampoco las considera en la optimización.	Sí
8	Se asigna una restricción de -5 al compuesto vinaza y luego se hace clic en la pestaña de “optimización y reportes”	El programa indica que existen errores y deben arreglarse antes de poder cambiar de pestaña.	Sí
<i>Correcciones: No existen correcciones</i>			

Control de Pruebas N° 11			
Caso de prueba revisado		Seleccionar los compuestos que participan en cada etapa	
Versión del sistema		Beta	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se hace clic en la pestaña “Gestión específica de raciones y restricciones”.	El programa muestra en “ <i>compuestos disponibles</i> ” todos los compuestos agregados y cargados en la pestaña “Gestión de compuestos y aportes”	Sí
2	Se eliminan en la pestaña “Gestión específica de raciones y restricciones” los compuestos: Mezcla Soprodu y Coseta Orafti	El programa no muestra en “ <i>compuestos disponibles</i> ” los compuestos que fueron eliminados en “Gestión de compuestos y aportes”.	Sí
3	Se selecciona Sal Mollendo y Ensilaje de pradera. Luego se deselecta Ensilaje de pradera. Después se seleccionan todos los compuestos. Finalmente se deselectan todos.	El programa muestra, al seleccionar uno o muchos compuestos, los nombres de los compuestos en “ <i>compuestos a utilizar</i> ” y los quita de “ <i>compuestos disponibles</i> ” al deselectarlos.	Sí
4	No se selecciona ningún compuesto en la etapa 1.	El programa marca con un recuadro rojo la sección de “ <i>compuestos a utilizar</i> ” y no deja cambiarse de pestaña.	Sí
Correcciones: No existen correcciones			

Control de Pruebas N° 12			
Caso de prueba revisado		Modificar restricciones mínimas nutricionales por etapa.	
Versión del sistema		Beta	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se ingresan los siguientes requisitos nutricionales mínimos: MS: -8 PT: "asdlfkj" TDN:"" (vacío) MCAL:1.0 FDN: ""(vacío) CA: 0.8 P: 11000	El programa marca en rojo las celdas de: MS,PT,TDN,FDN. El resto de las celdas son consideradas sin error y no permite cambiar de pestaña.	Sí
2	Se le asignará como requisito mínimo 0.5 a todos los campos de "requisitos nutricionales mínimos" para la etapa de recría, luego se procederá a optimizar y a guardar la ración creada.	El programa permite optimizar la ración y guardarla. La suma de los aportes nutricionales de la ración, para cada nutriente es efectivamente mayor o igual que el señalado como restricción mínima nutricional. Al revisar la base de datos los requisitos mínimos ingresados se encuentran correctamente guardados.	Sí
Correcciones: No existen correcciones			

Control de Pruebas N° 13			
Caso de prueba revisado		Determinar las razones por cada etapa que minimice el costo total.	
Versión del sistema		Beta	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se ingresaron los datos mostrados en el Anexo G	El programa retornó lo que se muestra en el Anexo G Los resultados obtenidos fueron comparados con un programa específico de optimización llamado LINGO y concuerdan con lo que éste último entrega. Los datos se almacenan correctamente en la base de datos.	Sí
2	Se ingresaron datos insuficientes (se seleccionó un compuesto por etapa únicamente) para la realización de una optimización.	El programa indica, al intentar realizar la optimización que el problema ingresado no tiene solución.	Sí
Correcciones: No existen correcciones			

Control de Pruebas N° 14			
Caso de prueba revisado		Obtener reportes históricos de raciones anteriores.	
Versión del sistema		Beta	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se crean 4 planes de alimentación diferentes, con los nombres: Plan 1, Plan 2, Plan 3, Plan4. Después se hace clic en la pestaña reportes históricos y se verifica la existencia de los 4 planes agregados por nombre y por fecha.	El programa muestra los 4 planes de alimentación ingresados previamente por nombre y por fecha.	Sí
2	Con los planes anteriores, se le pide al programa crear los reportes en Excel.	El programa crea los reportes y los guarda en el lugar que el usuario indica. Todos los reportes tienen los compuestos seleccionados para participar en las raciones en las distintas etapas. Las cantidades de los compuestos concuerdan con los de la optimización. Los aportes nutricionales concuerdan con los ingresados para la generación de la ración. Los precios mostrados son correctos y la suma de éstos entrega el precio total de la ración correctamente.	Sí

		<p>La ración generada cumple con los requisitos nutricionales mínimos necesarios.</p> <p>La fecha de creación concuerda con la fecha actual.</p> <p>La cantidad de pestañas del archivo Excel concuerda con la cantidad de etapas y con sus respectivos nombres.</p> <p>Los requerimientos mínimos de las etapas ingresadas corresponden al nombre de la etapa que se muestra en la planilla.</p> <p>Los pesos de entrada y salida corresponden a los que se ingresaron en el programa.</p> <p>La suma de los aportes nutricionales por cada requisito nutricional es correcta.</p>	
<p><i>Correcciones: No existen correcciones</i></p>			

Control de Pruebas N° 15			
Caso de prueba revisado		Control de acceso al iniciar el programa	
Versión del sistema		Beta	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se ingresa con el nombre de usuario “mollendo” y contraseña “wagyu” (ambos correctos)	El programa deja ingresar a la ventana principal y se inicia correctamente.	Sí
2	Se ingresa el nombre de usuario incorrecto y la contraseña incorrecta una vez u luego se ingresan el nombre de usuario y contraseña correctamente.	El programa muestra en pantalla un mensaje que dice: “nombre de usuario y contraseña incorrectos” la primera vez. La segunda vez si deja ingresar a la ventana principal.	Sí
3	Se ingresan incorrectamente el nombre de usuario y contraseña 3 veces (en la primera sesión)	El programa indica que se cerrará por 5 minutos	Sí
4	Tras haber intentado ingresar 3 veces en una sesión y cerrarse el programa. Se intenta ingresar antes de haber pasado el tiempo indicado por el programa	El programa muestra un mensaje que indica que aún no han transcurrido los minutos necesarios e indica aproximadamente cuantos minutos faltan para poder intentar ingresar nuevamente.	Sí
5	Tras haber intentado ingresar sin éxito en una sesión se espera el tiempo indicado y se ingresa con el nombre de usuario y contraseña correctos	El programa permite ingresar correctamente a la ventana principal del mismo.	Sí

6	Se intenta ingresar fallidamente en 2 sesiones consecutivas.	El programa debe mostrar que se cerrará por 10 minutos.	Sí
7	Se intenta ingresar tras 2 sesiones fallidas para luego ingresar con el nombre de contraseña y usuario correctos.	El programa permite ingresar correctamente a la ventana principal del mismo.	Sí
<i>Correcciones: No existen correcciones</i>			

Control de Pruebas N° 16			
Caso de prueba revisado		Cambio de nombre de usuario y contraseña	
Versión del sistema		Beta	
Cumple con compilación		Sí	
Pruebas:			
N°	Entradas/Acciones realizadas	Salida/Comportamiento del Programa	Se ajusta a caso de uso
1	Se intenta cambiar el nombre de usuario por uno vacío	El programa notifica que no se puede asignar nombre de usuario vacío	Sí
2	Se cambia el nombre de usuario por “agrícola Mollendo”	El programa permite el cambio de usuario correctamente.	Sí
3	Una vez cambiado el nombre de usuario, se sale del programa y se vuelve a ingresar con el nuevo nombre de usuario.	El programa deja entrar a la pantalla principal con el nuevo nombre de usuario.	Sí
4	Se intenta cambiar la contraseña poniendo los campos vacíos.	El programa no permite asignar una contraseña vacía al programa y lo notifica.	Sí
5	Se intenta cambiar la contraseña poniendo en los 2 campos contraseñas diferentes.	El programa no permite cambiar la contraseña e indica que ambos campos no son idénticos.	Sí
6	Se cambia la contraseña poniendo “mollendo2” en ambos campos.	El programa permite cambiar la contraseña e indica que se ha cambiado correctamente	Sí
7	Tras haber cambiado la contraseña, se sale del programa y se ingresa nuevamente con la última contraseña asignada.	El programa permite ingresar correctamente a la ventana principal del mismo.	Sí
Correcciones: No existen correcciones			

Anexo G “Raciones de Prueba y Resultados de Optimización

Aportes nutricionales

Composición	Stock en Kilos	MS	PT	TDN	MCAL	FDN	CA	P	\$
e. maíz	5.000.000	0,3289	0,0700	0,6990	2,6700	0,4073	0,2500	0,1500	13,20
coqueta orafti	1.500.000	0,2300	0,0800	0,6800	2,4700	0,4600	2,7200	0,1600	7,70
vinaza	sin limite	0,5900	0,7700	0,7800	2,8300	0,1700	0,0000	0,0000	60,00
bagazo cebada	0	0,3300							
maíz grano húmedo	sin limite	0,7300	0,0800	0,8800	3,1900	0,0900	0,0600	0,3400	105,00
harina Trigo	sin limite	0,9200	0,1450	0,6400	3,1500	0,2410	0,2700	3,0000	130,00
paja de trigo	sin limite	0,9000	0,0360	0,4490	1,6200	0,8500	0,1800	0,0500	32,00
sal mollendo	sin limite	0,9000	0,0000	0,0000	0,0000	0,0000	16,0000	8,0000	315,00
bicarbonato	sin limite								186,00
sal mollendo E	sin limite								560,00
gluten Feed	sin limite	0,8900	0,2584		3,3708	0,4200	0,0800	0,0230	117,00
urea	sin limite	0,9930	2,8700						205,00
expeler mani granel	sin limite	0,9200	0,4500	0,7700	3,0200				203,00
ensilaje de pradera	0	0,3472	0,0946	0,6000	2,2700	0,5000	0,6700	0,1600	14,60
hominy feed de maiz	sin limite	0,8600	0,2750		3,3000	0,1900	0,0005	0,0069	138,00

Tabla 17. Aportes nutricionales.

Restricciones

Vinaza con restricción de 400 gramos por animal máximo.

Bicarbonato sólo 100 gramos por animal.

Sal Mollendo E sólo 100 gramos por animal.

Urea con restricción máximo 100 gramos por animal

Comprobación de las raciones para cada etapa

Racion para recria 200-300 kgs		ganancia diaria			1,2	kgs/dia		% pV		
RECRIA	250 Kilos prom				Energia		3,2%			
Alimento	cantidad T.C.O	Mat.Seca	PT	TDN	MCAL	FDN	CA	P	\$	
e. maíz	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
coseta orafti	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
vinaza	0,40	0,24	0,18	0,18	0,67	0,04	0,00	0,00	0,00	24,00
bagazo Cebada	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
maíz grano húmedo	4,74	3,46	0,28	3,05	11,05	0,31	0,21	1,18	1,18	498,04
harina Trigo	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
paja de trigo	4,54	4,08	0,15	1,83	6,61	3,47	0,73	0,20	0,20	145,15
sal molido	0,20	0,18	0,00	0,00	0,00	0,00	2,86	1,43	1,43	62,51
bicarbonato	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
sal molido E	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
gluten Feed	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
urea	0,10	0,10	0,28	0,00	0,00	0,00	0,00	0,00	0,00	20,50
expeler mani granel	0,02	0,02	0,01	0,02	0,06	0,00	0,00	0,00	0,00	4,57
ensilaje de pradera	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
hominy feed de maiz	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
aportes totales	9,98	8,08	0,90	5,08	18,39	3,82	3,80	2,81	2,81	754,76

Tabla 18. Ración para etapa recria

Racion para etapa1 200-300 kgs		ganancia diaria			1,2	kgs/dia		% pV		
RECRIA	250 Kilos prom				Energia		5,2%			
Alimento	cantidad T.C.O	Mat.Seca	PT	TDN	MCAL	FDN	CA	P	\$	
e. maíz	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
coseta orafti	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
vinaza	0,40	0,24	0,18	0,18	0,67	0,04	0,00	0,00	0,00	24,00
bagazo Cebada	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
maíz grano húmedo	2,75	2,01	0,16	1,77	6,40	0,18	0,12	0,68		288,74
harina Trigo	0,03	0,03	0,00	0,02	0,10	0,01	0,01	0,09		4,31
paja de trigo	11,50	10,35	0,37	4,65	16,77	8,80	1,86	0,52		368,15
sal molido	0,15	0,13	0,00	0,00	0,00	0,00	2,11	1,05		46,08
bicarbonato	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00		0,00
sal molido E	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00		0,00
gluten Feed	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00		0,00
urea	0,10	0,10	0,28	0,00	0,00	0,00	0,00	0,00		20,50
expeler mani granel	0,09	0,08	0,04	0,06	0,24	0,00	0,00	0,00		17,42
ensilaje de pradera	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00		0,00
hominy feed de maiz	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00		0,00
Aportes totales	14,93	12,94	1,04	6,68	24,18	9,03	4,10	2,35		769,20

Tabla 19. Ración para la etapa 1

Racion para E2 451-600 kgs		ganancia diaria				1	kgs/dia				% P.V.
Etapa 2		501 Kilos promedio				Energia				0,0%	
Alimento	cantidad T.C.O	Mat.Seca	PT	TDN	MCAL	FDN	CA	P	\$		
e. maíz	5,20	1,71	0,12	1,19	4,56	0,70	0,43	0,26	68,57		
coseta orafti	1,38	0,32	0,03	0,22	0,78	0,15	0,86	0,05	10,62		
vinaza	0,40	0,24	0,18	0,18	0,67	0,04	0,00	0,00	24,00		
bagazo Cebada	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00		
maíz grano húmedo	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00		
harina Trigo	0,50	0,46	0,07	0,30	1,46	0,11	0,12	1,39	65,29		
paja de trigo	13,49	12,14	0,44	5,45	19,66	10,32	2,18	0,61	431,58		
sal molido	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00		
bicarbonato	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00		
sal molido E	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00		
gluten Feed	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00		
urea	0,06	0,06	0,16	0,00	0,00	0,00	0,00	0,00	11,46		
expeler mani granel	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00		
ensilaje de pradera	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00		
hominy feed de maiz	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00		
Aportes totales	21,02	14,92	0,99	7,34	27,13	11,31	3,60	2,30	611,52		

Tabla 20. Ración para la etapa 2

Racion para E3 600 kgs o mas		ganancia diaria			0,8	kgs/dia	% P.V.			
Etapa 3	701 Kg Peso promedio				Energia		0,0%			
Alimento	cantidad T.C.O	Mat.Seca	PT	TDN	MCAL	FDN	CA	P	\$	
e. maíz	18,03	5,93	0,42	4,15	15,83	2,42	1,48	0,89	238,01	
coseta orafti	0,50	0,11	0,01	0,08	0,28	0,05	0,31	0,02	3,81	
vinaza	0,33	0,19	0,15	0,15	0,55	0,03	0,00	0,00	19,67	
bagazo cebada	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
maíz grano húmedo	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
harina Trigo	0,54	0,50	0,07	0,32	1,57	0,12	0,13	1,49	70,37	
paja de trigo	6,62	5,96	0,21	2,68	9,66	5,07	1,07	0,30	211,97	
sal molido	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
bicarbonato	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
sal molido E	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
gluten Feed	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
urea	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
expeler mani granel	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
ensilaje de pradera	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
hominy feed de maiz	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
Aportes totales	26,02	12,70	0,86	7,37	27,89	7,69	3,00	2,70	543,83	

Tabla 21. Ración para la etapa 3

Reportes entregados por el programa de optimización

RACIONES GANADO

WAGYU

30 nov 2010

Racion para recria 200-300 kgs

Alimento	Energia								
	cantidad T.C.O	Mat.Seca	PT	TDN	MCAL	FDN	CA	P	\$
Expeler mani granel	0,022524192	0,020722	0,009325	0,015956	0,062581	0	0	0	4,572411
paja de trigo	4,535849774	4,082265	0,146962	1,832937	6,613269	3,469925	0,734808	0,204113	145,1472
sal mollendo	0,198433368	0,17859	0	0	0	0	2,85744	1,42872	62,50651
vinaza	0,4	0,236	0,18172	0,18408	0,66788	0,04012	0	0	24
urea	0,1	0,0993	0,284991	0	0	0	0	0	20,5
maiz grano humedo	4,743192666	3,462531	0,277002	3,047027	11,04547	0,311628	0,207752	1,17726	498,0352
Total Aportes	10	8,079408	0,9	5,08	18,3892	3,821673	3,8	2,810094	754,7613
requerimientos	10	6,74	0,9	5,08	18,38	0	3,8	1,8	

Tabla 22. Ración entregada por el programa de optimización para la etapa de recria.

RACIONES GANADO
WAGYU

30 nov 2010

Racion para etapa 1 301-450

kgs

Alimento	cantidad T.C.O	Mat.Seca	PT	TDN	Energia		CA	P	\$
					MCAL	FDN			
vinaza	0,4	0,236	0,18172	0,18408	0,66788	0,04012	0	0	24
Expeler mani granel	0,085791513	0,078928	0,035518	0,060775	0,238363	0	0	0	17,41568
maiz grano humedo	2,749909165	2,007434	0,160595	1,766542	6,403713	0,180669	0,120446	0,682527	288,7405
urea	0,1	0,0993	0,284991	0	0	0	0	0	20,5
sal mollendo	0,146357217	0,131721	0	0	0	0	2,107544	1,053772	46,10252
harina trigo	0,033141527	0,03049	0,004421	0,019514	0,096044	0,007348	0,008232	0,091471	4,308399
paja de trigo	11,50480058	10,35432	0,372756	4,64909	16,774	8,801172	1,863778	0,517716	368,1536
Total Aportes	15,02	12,93819	1,04	6,68	24,18	9,02931	4,1	2,345486	769,2207
requerimientos	15,02	8,65	1,04	6,68	24,18	0	4,1	2,1	

Tabla 23. Ración entregada por el programa de optimización para la etapa 1.

RACIONES GANADO
WAGYU

30 nov 2010

Racion para etapa 2 451-600

kgs

Alimento	Energia									\$
	cantidad T.C.O	Mat.Seca	PT	TDN	MCAL	FDN	CA	P		
vinaza	0,4	0,236	0,18172	0,18408	0,66788	0,04012	0	0	0	24
paja de trigo	13,48687132	12,13818	0,436975	5,450045	19,66386	10,31746	2,184873	0,606909		431,5799
harina trigo	0,502176297	0,462002	0,06699	0,295681	1,455307	0,111343	0,124741	1,386007		65,28292
coseta orafti	1,379807244	0,317356	0,025388	0,215802	0,783868	0,145984	0,863207	0,050777		10,62452
urea	0,055902299	0,055511	0,159317	0	0	0	0	0		11,45997
E.maiz	5,195242843	1,708715	0,11961	1,194392	4,56227	0,69596	0,427179	0,256307		68,57721
Total Aportes	21,02	14,91777	0,99	7,34	27,13318	11,31086	3,6	2,3		611,5245
requerimientos	21,02	9,75	0,99	7,34	26,57	0	3,6	2,3		

Tabla 24. Ración entregada por el programa de optimización para la etapa 2.

RACIONES GANADO
WAGYU

30 nov 2010

Racion para etapa 3 601-700

kgs

Alimento	Energia									\$
	cantidad	Mat.Seca	PT	TDN	MCAL	FDN	CA	P		
	T.C.O									
paja de trigo	6,624033984	5,961631	0,214619	2,676772	9,657842	5,067386	1,073094	0,298082		211,9691
E.maiz	18,03166193	5,930614	0,415143	4,145499	15,83474	2,415539	1,482653	0,889592		238,0179
harina trigo	0,541341977	0,498035	0,072215	0,318742	1,568809	0,120026	0,134469	1,494104		70,37446
vinaza	0,327783491	0,193392	0,148912	0,150846	0,5473	0,032877		0	0	19,66701
coseta orafti	0,495178622	0,113891	0,009111	0,077446	0,281311	0,05239	0,309784	0,018223		3,812875
Total Aportes	26,02	12,69756	0,86	7,369305	27,89	7,688218		3	2,7	543,8414
requerimientos	26,02	9,91	0,86	7,01	27,89	0		3	2,7	

Tabla 25. Ración entregada por el programa de optimización para la etapa 2.