

UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA
FACULTAD DE INGENIERÍA DE MINAS, GEOLOGÍA Y CIVIL
ESCUELA DE FORMACIÓN PROFESIONAL DE INGENIERÍA DE SISTEMAS



**APLICACIÓN WEB QUE BRINDE INFORMACIÓN DE LA REPRODUCCIÓN
DE GANADO BOVINO PARA MONITOREAR LA PRODUCCIÓN LÁCTEA EN EL
CENTRO DE PRODUCCIÓN DE ALLPACHAKA - UNSCH, 2016**

Tesis presentado por:

Bach. NÚÑEZ CHÁVEZ NILO JUAN

Para optar el Título Profesional de:

INGENIERO DE SISTEMAS

AYACUCHO - PERÚ

2016

DEDICATORIA

A Dios, por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos, además de su infinita bondad y amor.

Con cariño para mis padres, Sr. Nilo Núñez Quispe y Rosa María Chávez Arones. A quienes debo toda la persona que soy. Gracias por guiarme durante todos los años de mi vida. Los quiero.

A la Universidad Nacional de San Cristóbal de Huamanga y en especial a la Escuela de Formación Profesional de Ingeniería de Sistemas por permitirme ser parte de una generación de triunfadores y gente productiva para el país.

AGRADECIMIENTO

Debo agradecer de manera especial y sincera al MSc. Ing. Efraín Elías Porras Flores por aceptarme para realizar esta tesis de grado bajo su dirección. Su apoyo y confianza en mi trabajo y su capacidad para guiar mis ideas ha sido un aporte invaluable, no solamente en el desarrollo de esta tesis, sino también en mi formación como investigador. Las ideas propias, siempre enmarcadas en su orientación y rigurosidad, han sido la clave del buen trabajo que hemos realizado juntos, el cual no se puede concebir sin su siempre oportuna participación. Le agradezco también el haberme facilitado siempre los medios suficientes para llevar a cabo todas las actividades propuestas durante el desarrollo de esta tesis.

CONTENIDO

	Pág.
DEDICATORIA -----	i
AGRADECIMIENTO -----	ii
CONTENIDO -----	iii
RESUMEN -----	v
INTRODUCCIÓN -----	vi

CAPÍTULO I

PLANTEAMIENTO DE LA INVESTIGACIÓN

1.1 DESCRIPCIÓN DEL PROBLEMA -----	1
1.2 FORMULACIÓN DEL PROBLEMA -----	2
1.3 OBJETIVOS DE LA INVESTIGACIÓN -----	2
1.4 JUSTIFICACIÓN Y DELIMITACIÓN DE LA INVESTIGACIÓN -----	3

CAPÍTULO II

REVISIÓN LITERARIA

2.1 ANTECEDENTES DE LA INVESTIGACIÓN -----	5
2.2 MARCO TEÓRICO -----	7
2.2.1 REPRODUCCIÓN DE GANADO BOVINO -----	7
2.2.2 PRODUCCIÓN LÁCTEA -----	10
2.2.3 PROGRAMACIÓN EXTREMA XP -----	13
2.2.4 SISTEMA ADMINISTRADOR DE BASE DE DATOS -----	25
2.2.5 LENGUAJE DE PROGRAMACIÓN ORIENTADA A OBJETOS -----	29
2.2.6 ARQUITECTURA MODELO VISTA CONTROLADOR (MVC) -----	34
2.2.7 APLICACIÓN WEB -----	36
2.2.8 TECNOLOGÍAS DE INTERNET -----	38
2.2.9 POBLACIÓN Y MUESTRA -----	39

CAPÍTULO III
METODOLOGÍA DE LA INVESTIGACIÓN

3.1	TIPO DE LA INVESTIGACIÓN	42
3.2	DISEÑO DE LA INVESTIGACIÓN	42
3.3	POBLACIÓN Y MUESTRA	43
3.4	VARIABLE E INDICADORES	43
3.4.1	DEFINICIÓN CONCEPTUAL DE LAS VARIABLES	43
3.4.2	DEFINICIÓN OPERACIONAL DE LAS VARIABLES	46
3.5	TÉCNICAS E INSTRUMENTOS	46
3.5.1	TÉCNICAS PARA RECOLECTAR INFORMACIÓN	46
3.5.2	INSTRUMENTO PARA RECOLECTAR INFORMACIÓN	47
3.5.3	HERRAMIENTAS PARA EL TRATAMIENTO DE DATOS E INFORMACIÓN	47
3.5.4	TÉCNICAS PARA APLICAR XP	48

CAPÍTULO IV
RESULTADOS DE LA INVESTIGACIÓN

4.1	ARTEFACTOS DEL SOFTWARE APLICANDO EL PROCESO XP	53
4.1.1	FASE DE EXPLORACIÓN	53
4.1.2	FASE DE PLANEACIÓN	56
4.1.3	FASE DE ITERACIÓN	67

CAPÍTULO V
CONCLUSIONES Y RECOMENDACIONES

5.1	CONCLUSIONES	214
5.2	RECOMENDACIONES	215
	BIBLIOGRAFÍA	216
	ANEXO A	223
	ANEXO B	230
	ANEXO C	232
	ANEXO D	236

RESUMEN

La información que se documenta y registra en cada evento ocurrido en la reproducción de ganado bovino y su relación con la producción láctea en el Centro de Producción de Allapachaka - UNSCH, se viene dando de modo tradicional, dificultando el cálculo de los índices reproductivos que son; días abierto, intervalo entre partos, etc. y productivos; producción individual y duración de lactancia.

Debido a la necesidad, del Centro de Producción, de contar con una herramienta tecnológica, que brinde información sobre la reproducción de ganado bovino, a fin de monitorear la producción láctea. Se propone implementar una aplicación web, haciendo uso de la metodología de desarrollo ágil XP, un administrador de base de datos relacional, un lenguaje de programación orientada a objetos.

El tipo de estudio es descriptivo y el diseño es no experimental, puesto que se observa y describe el comportamiento de la reproducción de ganado bovino y producción láctea sin influir sobre el.

Los principales resultados logrados son; planificación del proyecto, diseño, codificación y pruebas de la aplicación web respecto a la información operativa de la reproducción de ganado bovino y producción láctea, se detallan en el capítulo IV.

Palabras clave: reproducción de ganado bovino, producción láctea, aplicación web y programación extrema.

INTRODUCCIÓN

La reproducción de ganado bovino, de acuerdo a Almeyda y Parreño (2011), consiste en un conjunto de eventos que son; celo, servicio, parto, etc. y dan lugar a la generación de nuevos terneros. La producción láctea se inicia después de cada parto, y su persistencia generalmente depende del siguiente parto. Según Pérez (2014), señala que la programación extrema es una metodología de desarrollo ágil que tiene como principal objetivo aumentar la productividad a la hora de desarrollar un proyecto software. Da prioridad a los trabajos que dan un resultado directo.

En la actualidad, en el Centro de Producción de Allpachaka, los datos referentes a la reproducción de ganado bovino y producción láctea, solo es registrado en una hoja de cálculo, más no se realiza su debido procesamiento para determinar los índices reproductivos y productivos, que permita señalar el estado o situación de la reproducción bovina, y asimismo, controlar o supervisar la producción láctea.

El estudio se realizó con el interés de contribuir con una herramienta tecnológica, a la institución que me apoyó en mi formación profesional, que permita brindar información sobre la reproducción de ganado bovino y facilitar el trabajo en el monitoreo de la producción láctea en el Centro de Producción Allapachaka.

Se tiene como objetivos específicos; explorar, planificar e iterar la información necesaria referente a la reproducción de ganado bovino, con la finalidad de monitorear la producción individual y la duración de lactancia.

CAPÍTULO I

PLANTEAMIENTO DE LA INVESTIGACIÓN

1.1 DESCRIPCIÓN DEL PROBLEMA

El Centro de Producción Allpachaka de la Universidad Nacional de San Cristóbal de Huamanga, alberga alrededor de 255 bovinos (216 brown swiss y 39 criollo mejorado). La información de cada ganado vacuno se registra en tarjetas individuales o en un libro de hato. Los registros que llevan no son exactos y esto hace que todo evento de su explotación no pueda ser dirigida de forma eficiente e indicar el nivel de rentabilidad de cada vaca en producción.

En el Centro de Producción al completar las planillas de registros estos son archivados y esta acción es inapropiada. Los registros que manejan deben ser resumidos para proveer de información útil. Y cada animal debe ser identificado adecuadamente y cada evento debe ser registrado en forma correcta para obtener índices reproductivos y productivos que sean realmente representativos del desempeño del Centro de Producción. De acuerdo a lo señalado, en su proyecto de investigación Córdova, S. (2004), manifiesta que de acuerdo a las observaciones efectuadas en las visitas al Centro de Producción Allpachaka y versiones de parte de sus trabajadores, se pudieron obtener información preliminar de preocupantes índices productivos y reproductivos con una baja producción láctea por animal y establo, así como, problemas reproductivos que se traducen en bajos índices de fertilidad y abortos.

Por otra parte Córdova, A. (2008), afirma que no se ha podido alcanzar los niveles productivos esperados para animales de raza Brown Swiss. Las

razones habría que buscarlas en la calidad genética de los animales, en el sistema de manejo y alimentación, así como, en otras causales no bien especificadas las cuales deberían ser motivo de análisis, para su inmediata solución, con la finalidad de conseguir elevar los niveles productivos, acordes con la calidad genética de los animales y de la Institución Universitaria.

1.2 FORMULACIÓN DEL PROBLEMA

PROBLEMA PRINCIPAL

¿Qué información de la reproducción de ganado bovino permite monitorear la producción láctea en el Centro de Producción de Allpachaka – UNSCH, 2016?

PROBLEMA SECUNDARIO

- a) ¿Qué información de la reproducción de ganado bovino permite monitorear la producción individual?
- b) ¿Qué información de la reproducción de ganado bovino permite monitorear la duración de lactancia?

1.3 OBJETIVOS DE LA INVESTIGACIÓN

OBJETIVO GENERAL

Desarrollar una aplicación web que brinde información necesaria de la reproducción de ganado bovino mediante técnicas e instrumentos, el proceso ágil XP, un administrador de base de datos relacional, un lenguaje de programación orientado a objetos, tecnologías de internet, con la finalidad de monitorear la producción láctea en el Centro de Producción Allpachaka – UNSCH, 2016.

OBJETIVO ESPECÍFICO

- a) Explorar, planificar e iterar la información necesaria de la

reproducción de ganado bovino, con el fin de monitorear la producción individual.

- b) Explorar, planificar e iterar la información necesaria de la reproducción de ganado bovino, con el fin de monitorear la duración de lactancia.

1.4 JUSTIFICACIÓN Y DELIMITACIÓN DE LA INVESTIGACIÓN

IMPORTANCIA Y JUSTIFICACIÓN

El Centro de Producción de Allpachaka – UNSCH requiere de una aplicación web, para llevar el control de reproducción y producción láctea, de modo que, en el manejo de la reproducción, permitirá calcular la tasa de incremento del mismo y determinar la eficiencia reproductiva; mientras que, en el manejo de la producción, la información registrada les permitirá comparar y detectar a los animales con mayor potencial productivo; para lograr, avances en el mejoramiento del hato mediante la selección; conocer el nivel productivo de los animales; estimar la eficiencia de la producción de leche y evaluar la eficiencia de producción.

IMPORTANCIA TÉCNICA

Se plantea implementar una aplicación web, donde el productor pecuario podrá acceder a la información, de manera ordenada y continua; facilitando el análisis de la información y permitiendo identificar los problemas que limitan la productividad; tomar decisiones apropiadas para resolverlos y mejorar la eficiencia de su explotación.

IMPORTANCIA SOCIOECONÓMICA

Con el desarrollo de la aplicación web, el productor pecuario podrá contar con información en detalle, de cada una de los bovinos que conforman el hato, de modo que, podrán incrementar sus ingresos económicos. Dicha aplicación podrá ser usada por cualquier productor

de ganado bovino, puesto que, las tareas que realizan se asemejan.

DELIMITACIÓN

La investigación se realizó en el Centro de Producción de Alpachaka – UNSCH, durante el año 2016.

CAPÍTULO II

REVISIÓN LITERARIA

2.1 ANTECEDENTES DE LA INVESTIGACIÓN

Córdova, A. (2008), recomienda mejorar los registros de producción individual, registrando los datos diariamente para cumplir con la atención inmediata de los problemas sanitarios y aquello de índole reproductiva. Se puede preparar o adquirir programas de computación acordes con la naturaleza de la producción lechera, para el adecuado conocimiento de los responsables del manejo sanitario, reproductivos, de selección y mejoramiento genético.

Córdova, S. (2004), manifiesta que el propósito principal es producir altos niveles de leche. Con esta finalidad, las ganaderías modernas realizan una exigente selección genética, una alimentación cada vez más eficiente, un mejor control de las enfermedades y la introducción de novedosos sistemas de manejo; logrando así, conseguir sobresalientes producciones diarias que se incrementan año tras año.

Para cumplir el objetivo del proyecto de investigación se realizó un estudio de las soluciones existentes para brindar información de la reproducción y poder monitorear la producción láctea. Entre los sistemas estudiados se pueden mencionar.

Terrero y Morejón (2014), manifiestan que el control de los animales (bovinos) en la actualidad se lleva a cabo de forma manual y de una manera engorrosa, grandes cantidades de papel por el personal y otros documentos al personal destinado a tener el registro de la masa ganadera existente en cada una de los diferentes Centros, Productores

Independientes y otras organizaciones que realizan esta actividad. Este procedimiento de forma manual trae consigo varias dificultades. Y surge la necesidad de implementar un sistema informático cuyo objetivo principal sea garantizar la Gestión y Control del Ganado Vacuno y la Inseminación Artificial en cada una de las diferentes áreas destinadas a la cría o mejoramiento del mismo. En el trabajo que desarrollaron describen la propuesta de la aplicación informática Sistema para la Gestión y Control del Ganado Vacuno y la Inseminación Artificial (GAVIAC).

Marquez (2009), menciona sobre el software que implementó, es un software desarrollado para la plataforma Windows que sirve para manejar ganado bovino. Este fue desarrollado por un grupo internacional de consultores, veterinarios, ingenieros, productores de ganado vacuno, y es usado por algunas de las mayores granjas del mundo para la administración de su ganado. Incluye una lista extensiva de informes para reproducción, novillos y manejo financiero. El asistente de la base de datos lo ayuda a hacer consultas para hacer un análisis profundo de los datos. Además presente una serie de informes entre los que están; Informes productivos, Manejo de novillos y vacas, Gestión de la alimentación y Gestión económica.

Balkenende (s.f.), menciona sobre el software que desarrollo, este programa funciona en entorno Windows. El mismo puede ser operado por productores y técnicos del sector lechero. Está desarrollado en Visual Basic, aprovechando la tecnología de este lenguaje para el manejo en Windows. Este software es adecuado tanto para productores lecheros como para veterinarios, ya que puede manejar tanto uno como varios rodeos lecheros, y al ser configurable el usuario puede personalizar los datos que desea llevar. Entre las funcionalidades vitales se mencionan; Vacas a parir, Vacas a secar, Vacas a tacto, Vacas repetidoras, Resumen de controles, Cambios de lote y Existencia de

semen.

2.2 MARCO TEÓRICO

2.2.1 REPRODUCCIÓN DE GANADO BOVINO

De acuerdo a Gélvez (2007), sostiene que teóricamente una vaca debería producir una cría por año, esto depende de factores internos como la genética del animal y de factores externos como la nutrición, el manejo sanitario y el mismo manejo reproductivo. Las vacas presentan celo cada 21 días, son hembras poliéstricas continuas. Los machos servirán a las hembras solo durante el tiempo que el celo esté presente, esto es por un lapso de 2 a 4 días. En las vacas la gestación dura 9 meses aproximadamente.

Por su parte, Gutiérrez, P. (2009), opina que la vaca es una reproductora poliéstrica continua, es decir presenta ciclos a lo largo de todo el año. Los ciclos tienen una duración de 21 días. Dentro del ciclo el celo es muy corto con una duración media de 18 horas. La gestación tiene una duración de 270-290 días lo que equivale a 9 meses. La pubertad la alcanzan cuando han conseguido entre un 30-40% de su peso adulto y esto sucede a los 10-12 meses. Se cubren por primera vez cuando tienen 15 meses y por lo tanto tienen el primer parto a los 2 años. La cubrición en el vacuno de leche se suele realizar por inseminación artificial ya que se han eliminado los machos de las explotaciones. La inseminación artificial consiste en depositar en el aparato genital de la hembra el semen obtenido del macho por medios artificiales.

Según Roa (2006), señala que la reproducción afecta directamente el número de crías por año, la producción de leche por vaca, y la selección y progreso genético. Es decir, para obtener un desempeño productivo rentable en la finca es necesario tener eficiencia reproductiva en la misma. En consecuencia, el manejo reproductivo de cualquier rebaño bovino se fundamenta en un programa de

diagnóstico, control reproductivo y buenos registros.

A. PUBERTAD

Según Castro (2002), señala que el comienzo de la pubertad está influenciado por el nivel de alimentación, ya que a mejor alimentación más rápido sería el crecimiento y más temprano el inicio de la actividad sexual. Una alimentación escasa o insuficiente retarda la pubertad. La edad y peso en el momento de la pubertad tienen una amplia variación entre los bovinos. Los primeros alcanzan la pubertad entre los siete y diez meses de edad en tanto que los segundos entre los catorce y veinticuatro meses.

De acuerdo a Serrano (1999), menciona que la edad de la pubertad de novillas y consecuentemente su desarrollo sexual y endocrino, está marcadamente influenciado por el nivel nutricional. Se ha determinado que animales mantenidos en un plan bajo de energía llegan a la pubertad tardíamente (10 – 12 meses de retraso) en comparación con novillas o toretes mantenidos en un plan nutricional intermedio o alto respectivamente.

De acuerdo a Rosemberg (2000), asegura que la pubertad es el periodo de vida del animal donde se inician los ciclos reproductivos, desarrollan los órganos sexuales primario (ovario) y aparecen los caracteres sexuales secundarios. El inicio de la pubertad ocurre en promedio a los 9 meses de edad, presentándose en un rango de 5 a 15 meses.

B. PRIMER SERVICIO

Según Castro (2002), sostiene que para el primer servicio de la novilla se debe considerar el peso y la edad. Cuando la novilla tenga el peso adecuado, puede servirse (fecundarse) por primera vez, no importa que la edad no sea la indicada, pero no debe servirse una novilla que tenga la edad adecuada y no el peso indicado.

De acuerdo a Rosemberg (2000), asegura que la edad de la vaquilla tiene que relacionarse con la talla y el peso para efectuar el primer servicio. Estos criterios permiten lograr partos normales. Sin afectar el desarrollo y la producción animal. En nuestro medio la edad para el primer servicio oscila entre los 15 y 18 meses. La alimentación se considera un factor determinante para el primer servicio y en consecuencia, el primer parto. La concepción tardía puede causar importantes pérdidas económicas al propietario, reduciendo la producción diaria a lo largo de su vida útil.

C. DÍAS ABIERTOS

De acuerdo a Rosemberg (2000), sostiene que constituye el intervalo comprendido entre el parto y la siguiente concepción. se recomienda que la primera inseminación o servicio se realiza como mínimo a los 60 días después del parto, señalándose que las vacas conciben con mayor facilidad entre los 60 y 70 días después del parto y esas producen más leche de por vida.

Según el Instituto Interamericano de Cooperación para la Agricultura (2008), señala que es un parámetro que cubre un periodo menor que el intervalo entre partos y es igualmente eficiente para medir el potencial reproductivo si se considera que el periodo de gestación es constante (285 días, generalmente). Los riesgos y criterios para la estimación del intervalo entre partos también son válidos para el cálculo de los días abiertos.

Según Caballero y Hervas (2005), menciona que es el periodo transcurrido desde el parto a la concepción, y es el factor determinante de la duración del intervalo entre partos, ya que la duración de la gestación puede considerarse como constante.

D. SERVICIOS POR CONCEPCIÓN

Según De Alba (2004), asegura que esta medida está asociada al intervalo entre partos, ya que mientras más servicios se requieren, más tiempo transcurre después del parto y el servicio efectivo.

Según Bazán (2008), sostiene que el número de servicios por concepción puede estar influenciado por la fertilidad del toro y la vaca, la raza, estación del año y manejo de los animales, se ha obtenido un rango que va de 1,2 a 1,8 servicios por concepción en diferentes razas de ganado, requiriendo menor número de servicios por concepción las vacas servidas 60 días después del parto.

E. INTERVALO ENTRE PARTO

De acuerdo a Morales, Pérez y Botero, R. (2009), afirman que es el número de días que transcurren entre un parto y el siguiente, en un mismo vientre. El valor considerado como ideal es de 365 días, el óptimo va de 380 - 395 días y se considera un problema cuando este intervalo es mayor de 456 días (15 meses, equivalentes a 80% de natalidad).

Según Castro (2002), sostiene que el intervalo entre parto es el espacio de tiempo, expresado en días, en que tiene lugar un parto promedio por cada hembra adulta en existencia promedio en la lechería, y se calcula en existencia promedio en la lechería.

Según Caballero y Hervas (2005), asegura que este parámetro mide en tiempo la frecuencia de la reproducción del hato; es decir, es el periodo comprendido entre dos partos sucesivos de cada vaca.

2.2.2 PRODUCCIÓN LÁCTEA

Según Bretschneider, Salado, Cuatrin y Arias (2015), informan que debido a que la lactancia se inicia con el parto, la producción de leche depende exclusivamente de la gestación. Para ganar vida útil o productiva, la vaca es preñada mientras está en producción. De esta

manera, en algún momento del ciclo productivo, la gestación se va a superponer con la lactancia en curso hasta que la vaca se seque (cese de la lactancia), en general, dos meses previo al parto y, en consecuencia, al inicio de la siguiente lactancia.

De acuerdo al Grupo Sol (s.f.), señalan que éste comienza con un parto de la vaca, la cual durante los primeros cinco o seis días da leche con calostro, motivo por el cuál esta leche no es utilizada para la producción lechera diaria; durante este período no se le realiza control al animal (si es que coincide con la visita del inspector), siendo el primer control el inmediato posterior. Luego de finalizado el calostro se le hacen controles mensuales hasta su secado, luego de once meses de producción aproximadamente. Por "secado" se entiende cuando el personal del tambo considera que ya no es conveniente ordeñar al animal, sea porque no está dando la cantidad apreciable de leche o para prepararla para el nuevo parto.

A. PRODUCCIÓN INDIVIDUAL

Según Castro (2002), asegura que se define como la producción de leche que se obtiene de una vaca en el periodo de lactancia en un año y está determinado por el intervalo entre partos. Si calculamos el promedio diario de producción para cada día del intervalo entre partos, éste será representativo de su producción diaria en cada día de vida en ese periodo y en un año será ese promedio por 365 días.

De acuerdo a Almeyda y Parreño (2011), sostienen que la campaña productiva de una vaca especializada para producción de leche tiene tres diferentes etapas; primer tercio, segundo tercio y tercer tercio.

- a. El primer tercio (desde el parto hasta los 90 días después del parto) es la etapa más exigente en alimentación, donde el productor debe hacer el mayor esfuerzo con el objeto de satisfacer los requerimientos nutricionales principalmente de

energía. En este periodo se espera que una vaca de raza grande como la Holstein consuma niveles de 3,6 a 4,0 % de materia seca respecto a su peso corporal para lograr promedios de producción esperados de 35 a 40 kg de leche por vaca /día.

- b. El segundo tercio comprendido entre los 91 días post parto hasta los 210 días de la campaña se espera que la vaca consuma una ración alimenticia que le permita satisfacer los requerimientos nutricionales e incluso pueda recuperar su estado corporal afectado durante el primer tercio. En este periodo se espera que la vaca consuma niveles de 3,0 a 3,3 % de materia seca respecto a su peso corporal para lograr una producción esperada de 25 a 28 kg de leche por vaca/día en promedio.
- c. Finalmente en el último tercio comprendida entre los 211 días de la campaña hasta la fecha de seca, la vaca debe restablecerse totalmente e incluso ganar reservas corporales para que cuando llegue a la seca esté en una condición corporal de 3,25 a 3,75 grados. Tener presente que una vaca seca debe estar preñada. Para este periodo se espera que la vaca consuma niveles de 2,5% de materia seca respecto a su peso corporal y que logre una producción esperada de 15 a 18 kg de leche por vaca/día en promedio. De acuerdo a los datos indicados se estima que las vacas pueden producir entre 7500 a 8500 kg de leche por campaña de 305 días, bajo sistemas de crianza intensiva.

B. DURACIÓN DE LACTANCIA

Según Castro (2002), sostiene que es el tiempo promedio de duración de una lactancia de la existencia promedio de hembras adultas en una lechería. Conforme más se aproximen a los 300 días de lactancia, mejor será la persistencia de producción lechera.

De acuerdo a Caballero y Hervas (2005), aseguran que un periodo de 10 meses (305 días) de ordeño permite un parto/año y un intervalo entre

partos de 12 a 13 meses. Esta ciclicidad asegura una óptima productividad por vaca en rebaños bajo condiciones de clima templado. La prolongación de la lactancia por largos intervalos entre partos, a pesar de aumentar la producción de leche/lactancia disminuye la producción por vida de la vaca.

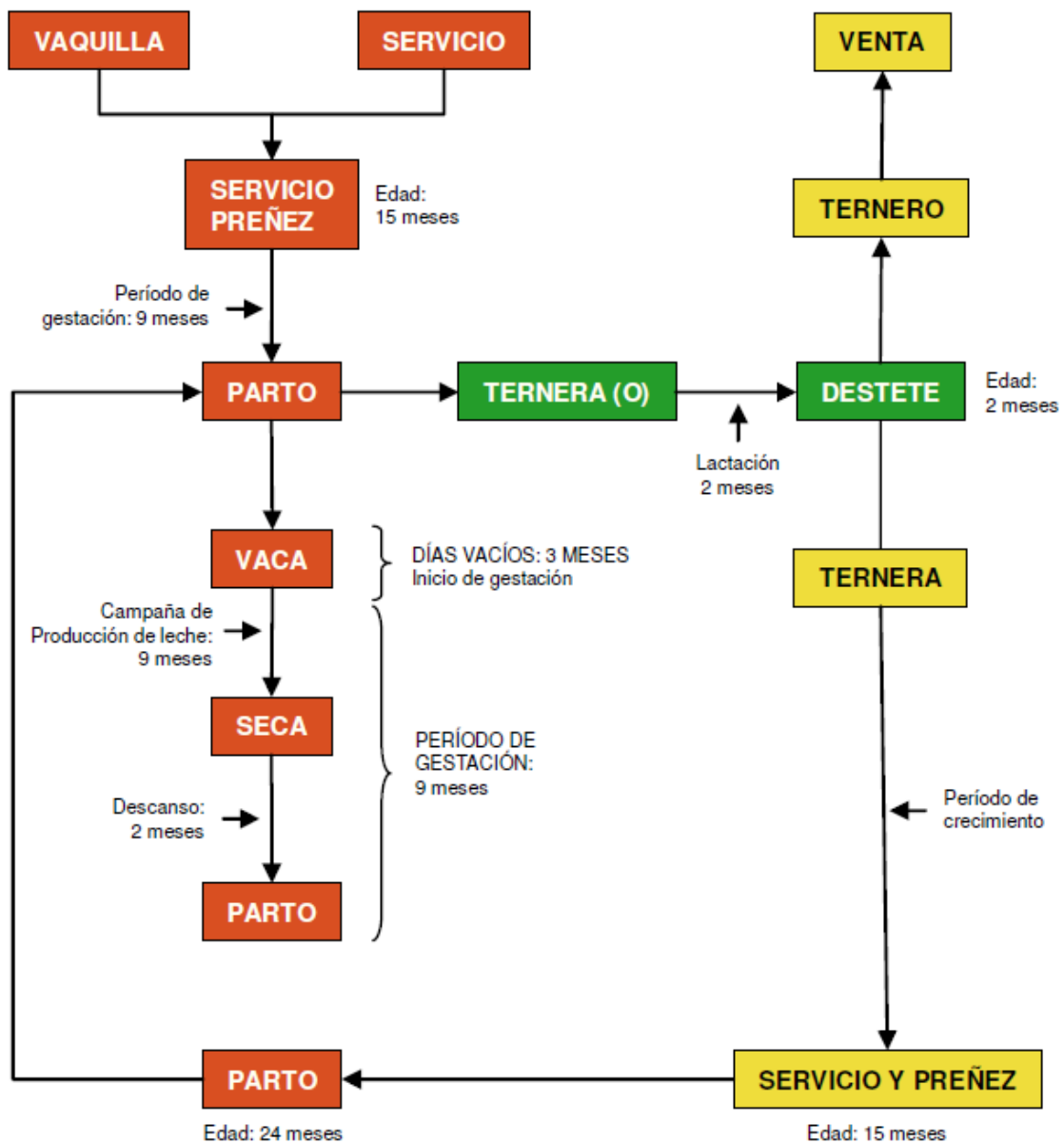


Figura N° 2.1: Flujograma de manejo productivo y reproductivo recomendable para vacunos de raza Holstein o Brown Swiss (Almeyda y Parreño, 2011).

2.2.3 PROGRAMACION EXTREMA XP

Según weitzenfeld (2005), afirma que la programación extrema es un modelo de proceso de software que toma los principios y practicas

aceptadas, y las lleva a niveles extremos. Tiene como objetivo reducir el riesgo en el ciclo de vida del software mediante grupos pequeños de desarrollo. Aunque XP define varias prácticas a seguir, quizá la más representativa del proceso de XP es la programación en pares, donde todo desarrollo requiere de dos programadores que trabajen juntos.

Según Anderson (2000), señala que “la programación extrema es una metodología de desarrollo ligera (o ágil) basada en una serie de valores y de prácticas de buenas maneras que persigue el objetivo de aumentar la productividad a la hora de desarrollar programas”.

Según Kendall (2005), asegura que la programación extrema (XP, Extreme Programming) es un enfoque para el desarrollo de software que utiliza buenas prácticas de desarrollo y las lleva a los extremos. Se basa en valores, principios y prácticas esenciales. Los cuatro valores son la comunicación, la simplicidad, la retroalimentación y la valentía. Y recomienda a los analistas de sistemas que adopten estos valores en todos los proyectos que emprendan, no sólo cuando recurran a medidas de programación extrema.

“La Programación Extrema es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado “(Beck y Andres, 2004).

A. VALORES DE LA PROGRAMACIÓN EXTREMA (XP)

La Programación Extrema “Extreme Programming” (XP) no es un conjunto de reglas a seguir, sino una forma de trabajar en armonía con los valores personales y organizacionales, que tiene su punto de partida en cinco valores fundamentales (Rodriguez, R., 2012).

COMUNICACIÓN.- Es muy importante que haya una comunicación

constante con el cliente y dentro de todo el equipo de trabajo, de esto dependerá que el desarrollo se lleve a cabo de una manera sencilla, entendible y que se entregue al cliente lo que necesita. (Universo el periódico de los universitarios, 2012).

SIMPLICIDAD.- En la programación extrema se refiere que ante todo y sin importar qué funcionalidad requiera el usuario en su sistema, éste debe ser fácil. El diseño debe ser sencillo y amigable al usuario, el código debe ser simple y entendible, programando sólo lo necesario y lo que se utilizará. (Universo el periódico de los universitarios, 2012).

RETROALIMENTACIÓN (Feedback).- Tomaremos seriamente los compromisos con el usuario establecidos en todas las iteraciones, entregando software en funcionamiento en cada una. Mostraremos al usuario nuestro software frecuentemente y de forma temprana, escuchando cuidadosamente sus observaciones y realizando los cambios que sean necesarios. Adaptaremos nuestros procesos al proyecto y no al contrario. (Rodríguez, R., 2012).

CORAJE.- Se refiere a la valentía que se debe tener al modificar o eliminar el código que se realizó con tanto esfuerzo; el desarrollador debe saber cuándo el código que desarrolló no es útil en el sistema y, por lo mismo, debe ser eliminado. También se refiere a tener la persistencia para resolver los errores en la programación. (Universo el periódico de los universitarios, 2012).

RESPECTO.- Todos en el equipo dan y reciben el respeto que merecen como integrantes del equipo y los aportes de cada integrante son valorados por todos. Todos contribuyen, así sea simplemente con entusiasmo. Los desarrolladores respetan la experticia de los clientes y viceversa. La Gerencia respeta el derecho del equipo de asumir responsabilidad y tener autoridad sobre su trabajo. (Rodríguez, R., 2012).

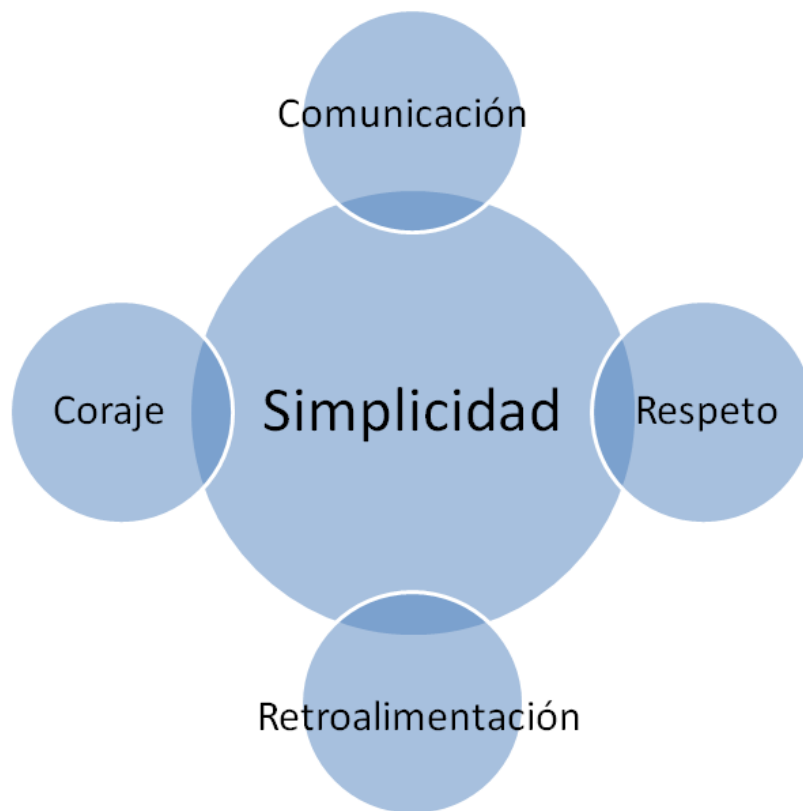


Figura N° 2.2: Los valores de XP (Rodríguez, R., 2012).

B. PRÁCTICAS DE LA PROGRAMACIÓN EXTREMA (XP)

La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. XP apuesta por un crecimiento lento del costo del cambio y con un comportamiento asintótico. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las prácticas que se describe a continuación. (Letelier y Penadés, 2006).

Principio de pruebas.- lo primero que se debe hacer es establecer un período de pruebas de aceptación del programa, en el cual se definirán las entradas y salidas del sistema. Básicamente se define lo que debe hacer el software desarrollado. Como si fuese una caja negra.

(Pérez, 2015).

Planificación.- el cliente (o su representante) escribirá sus necesidades para definir concretamente las actividades que el sistema debe realizar. En esta fase se creará un documento que contendrá historias de usuario que forman el plan de liberación, el cual define los tiempos de entrega de la aplicación para poder recibir feedback por parte del cliente. (Pérez, 2015).

Cliente in-situ.- el cliente (o su representante) deberá formar parte del equipo de desarrollo. Se le dará poder para determinar los requisitos de la aplicación, definir la funcionalidad y dar prioridad a determinadas cosas. Gracias a esto, habrá una fuerte interacción con los programadores, disminuyendo así el tiempo de comunicación y la cantidad de documentación a redactar. El cliente estará con el equipo durante todo el proceso de desarrollo del proyecto. (Pérez, 2015).

Pair-programming.- este punto junto con el anterior son los más radicales de esta metodología. Consiste en escribir código en parejas compartiendo una sola máquina. Según los experimentos ya realizados sobre este método, se producen mejores y más consistentes aplicaciones a igual o menor coste. (Pérez, 2015).

Integración continua.- consiste en implementar progresivamente las nuevas características del software. En lugar de crear versiones estables en función de una planificación previamente realizada, los programadores reúnen su código y reconstruyen el proyecto varias veces al día si hace falta. (Pérez, 2015).

Refactorización.- mediante la constante eliminación de código duplicado y/o ineficiente los equipos de programación mejoran el diseño del sistema. El código se evalúa continuamente para ofrecer la

mayor calidad posible.

Entregas pequeñas.- el producto es evaluado en un ambiente real mediante la colocación de un sistema sencillo en producción el cual se actualizará rápidamente, es decir, cada 2 semanas (3 como máximo) el software será puesto en producción. (Pérez, 2015).

Diseño simple.- el mejor programa será aquel que cumpla con los requisitos y sea más simple. Es importante proporcionar un software que cubra las necesidades de un cliente. Ni más ni menos. (Pérez, 2015).

Metáfora.- El objetivo de la metáfora del sistema es proporcionar a todo el equipo una misma visión del fin del sistema y de su arquitectura general. Con ello se facilita que todos los desarrolladores hablen un mismo idioma y que nuevos desarrolladores lo adquieran más rápido e integrarse en el proyecto sin dificultades. Este segundo aspecto es el que puede hacerla interesante para el software libre. La posibilidad de conseguir el código en muchas ocasiones no rebaja suficientemente la barrera de entrada que existe para nuevas incorporaciones. De hecho, estoy seguro de que la información que se puede plasmar de manera concisa en la metáfora se encuentra dispersada en el código y las listas de correo (Robles y Ferrer, 2002).

Propiedad colectiva del código.- el código tiene propiedad compartida. Nadie es propietario de nada, ni siquiera de lo que ha desarrollado. Todos los programadores son "dueños" de todo el código. Según esta metodología, cuantos más programadores haya trabajado en una parte de código, menos errores tendrá. (Pérez, 2015).

Estándar de programación.- define las reglas para escribir y documentar código, además de cómo se comunican las diferentes piezas de código desarrolladas por diferentes equipos. El objetivo de esto es que parezca

que el código ha sido escrito por una única persona. (Pérez, 2015).

Semana de 40 horas.- los programadores cansados escriben peor código. Es importante minimizar las horas extras y mantener a los programadores frescos y descansados. De esta manera, se generará mejor código. Si es necesario hacer horas extras, quiere decir que el proyecto está mal planificado. (Pérez, 2015).

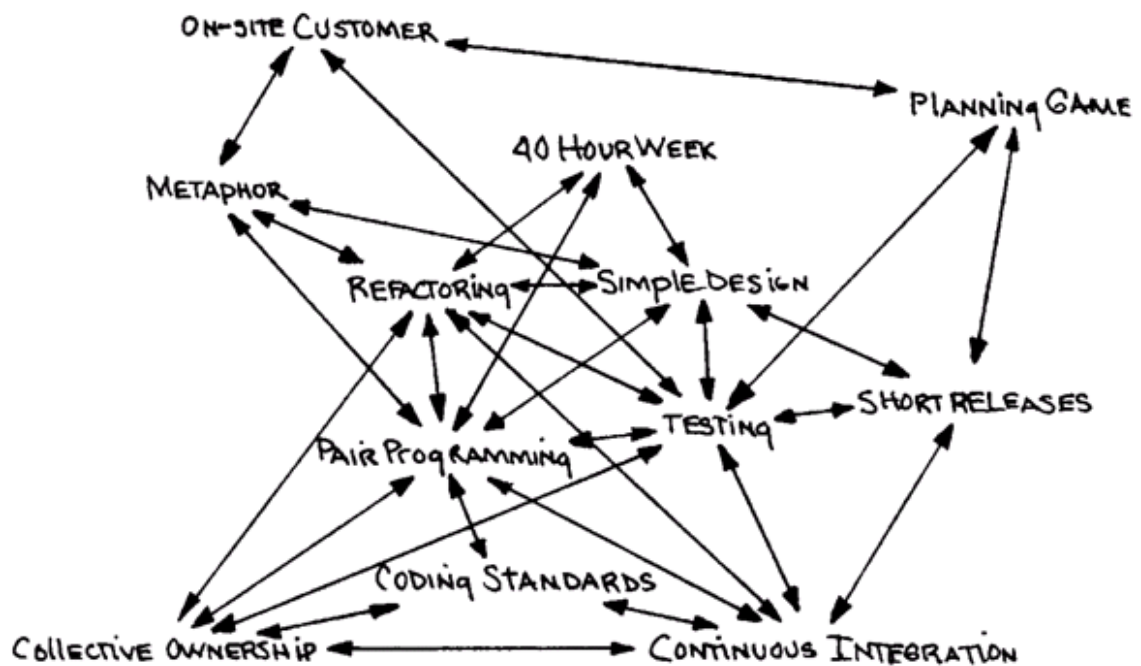


Figura N° 2.3: Extreme Programming Explained: Embrace Change (Beck y Andres, 2004).

C. ARTEFACTOS XP

A continuación describimos los artefactos de XP, entre los que se encuentran: Historias de Usuario, Tareas de Ingeniería, Tarjetas CRC y Caso de pruebas de aceptación.

HISTORIAS DEL USUARIO

Según Beck y Andres (2004), sostienen que una historia de usuario es una representación de un requerimiento de software escrito en una o dos frases utilizando el lenguaje común del usuario. Las historias de usuario

son utilizadas en las metodologías de desarrollo ágiles para la especificación de requerimientos (acompañadas de las discusiones con los usuarios y las pruebas de validación). Cada historia de usuario debe ser limitada, esta debería poderse escribir sobre una nota adhesiva pequeña. Dentro de la metodología XP las historias de usuario deben ser escritas por los clientes.

Según (Kendall, 2005), afirma que las historias de usuario son una forma rápida de administrar los requerimientos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos. Las historias de usuario permiten responder rápidamente a los requerimientos cambiantes:

- a. Las historias de usuario tienen el mismo propósito que los casos de uso.
- b. Las escriben los propios clientes, tal y como ven ellos las necesidades del sistema.
- c. Las historias de usuario son similares al empleo de escenarios, con la excepción de que no se limitan a la descripción de la interfaz de usuario. También conducirán el proceso de creación de los test de aceptación (empleados para verificar que las historias de usuario han sido implementadas correctamente).
- d. Existen diferencias entre estas y la tradicional especificación de requisitos. La principal diferencia es el nivel de detalle. Las historias de usuario solamente proporcionaran los detalles sobre la estimación del riesgo y cuánto tiempo conllevará la implementación de dicha historia de usuario.

Historia de Usuario	
Número:	Nombre Historia de Usuario:
Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):	
Usuario:	Iteración Asignada:

Prioridad en Negocio: (Alta / Media / Baja)	Puntos Estimados:
Riesgo en Desarrollo: (Alto / Medio / Bajo)	Puntos Reales:
Descripción:	
Observaciones:	

Tabla N° 2.1: Modelo propuesto para una historia de usuario, (Porras, 2010).

TAREAS DE INGENIERÍA

“Son tarjetas que se elaboran para ayudar y simplificar la programación de una historia de usuario”. (Universo el periódico de los universitarios, 2012).

Tarea de Ingeniería	
Número Tarea:	Historia de Usuario (Nro. y Nombre):
Nombre Tarea:	
Tipo de Tarea : Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados:
Fecha Inicio:	Fecha Fin:
Programador Responsable:	
Descripción:	

Tabla N° 2.2: Modelo propuesto para una tarea de ingeniería, (Porras, 2010).

TARJETAS CRC (CLASE - RESPONSABILIDAD – COLABORADOR)

Estas tarjetas dividen en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores.

“Cada tarjeta representa una clase en la programación orientada a objetos y define sus responsabilidades (lo que ha de hacer) y las

colaboraciones con las otras clases (cómo se comunica con ellas)” (Beck y Andres, 2004).

Nombre del clase	
Responsabilidades	Colaboradores

Tabla N° 2.3: Modelo de tarjeta CRC, (Porrás, 2010).

CASO DE PRUEBAS DE ACEPTACIÓN

De acuerdo a Sommerville (2005), sostiene que es la etapa final en el proceso de pruebas antes de que se acepte que el sistema se ponga en funcionamiento. Éste se prueba con los datos proporcionados por el cliente más que con datos de prueba simulados. Debido a la diferencia existente entre los datos reales y los de prueba, la prueba de aceptación puede revelar errores y omisiones en la definición de requerimientos del sistema. También puede revelar problemas en los requerimientos donde los recursos del sistema no cumplen las necesidades del usuario o donde el desempeño del sistema es inaceptable.

“La Prueba de Aceptación es una prueba formal conducida para determinar si un sistema satisface los criterios de aceptación y permite al cliente determinar si acepta o no el sistema” (Beck y Andres, 2004).

Caso de Prueba de Aceptación	
Código:	Historia de Usuario (Nro. y Nombre):
Nombre:	

Descripción:
Condiciones de Ejecución:
Entrada / Pasos de ejecución:
Resultado Esperado:
Evaluación de la Prueba:

Tabla N° 2.4: Modelo propuesto para una prueba de aceptación (Porras, 2010).

D. ETAPAS DEL PROCESO DE DESARROLLO DE XP

Existen cinco etapas; exploración, planeación, iteraciones a la primera versión, puesta en producción y mantenimiento.

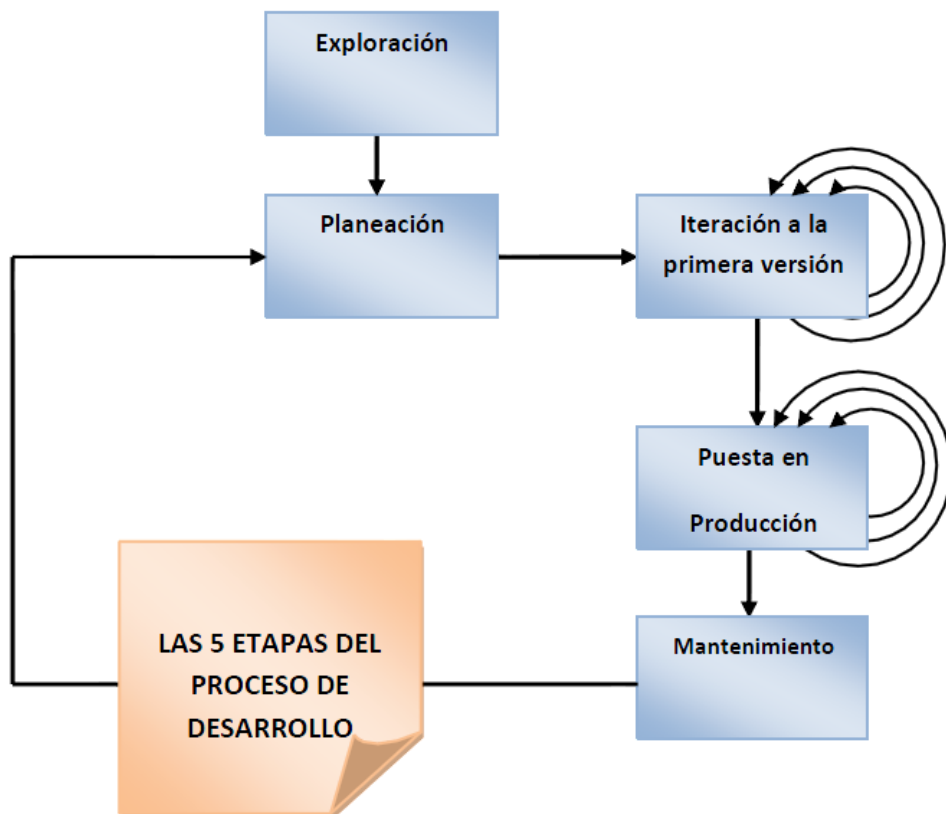


Figura N° 2.4: Etapas del proceso de desarrollo de XP (Kendall, 2005).

EXPLORACIÓN.- Se examinará su entorno, sosteniendo su convicción de que el problema puede y debe enfrentarse mediante programación

extrema, conformará el equipo y valorará las habilidades de los miembros del mismo. Esta etapa durará desde unas cuantas semanas hasta algunos meses y También se ocupará de examinar las tecnologías potenciales que requerirá para construir el nuevo sistema. Durante esta etapa se debe practicar el cálculo de tiempo que tomarán diversas tareas. Los clientes también experimentarán con la escritura de relatos del usuario. El objetivo es lograr que el cliente refine lo suficiente un relato para que usted pueda calcular con eficiencia la cantidad de tiempo que tomará construir la solución en el sistema que está planeando (Kendall, 2005).

LA PLANEACIÓN.- La planeación podría tomar sólo algunos días. En esta etapa usted y sus clientes establecen una fecha de común acuerdo, que puede ir de dos meses a medio año a partir de la fecha actual, para la entrega de soluciones a los problemas de negocios más urgentes de los clientes (Kendall, 2005).

ITERACIONES A LA PRIMERA VERSIÓN.- Por lo general, estas iteraciones (ciclos de pruebas, retroalimentación y cambios) duran aproximadamente tres semanas. Tendrá que bosquejar toda la arquitectura del sistema, aunque sólo sea un diseño preliminar. Una meta es realizar pruebas de funcionamiento escritas por el cliente al final de cada iteración. Al finalizar todas las iteraciones, el sistema está listo para pasar a la siguiente etapa (Kendall, 2005).

LA PUESTA EN PRODUCCIÓN.- Durante esta etapa se realiza diversas actividades. El ciclo de retroalimentación se acelera, de tal manera que en lugar de recibir retroalimentación para una iteración cada tres semanas, las revisiones del software se realizan en una semana. El producto se libera en esta etapa, aunque se puede mejorar incorporándole otras características (Kendall, 2005).

MANTENIMIENTO.- Una vez que se ha liberado el sistema, es necesario mantenerlo funcionando sin problemas. Se pueden agregar nuevas características, se pueden tomar en cuenta las sugerencias más arriesgadas del cliente y se pueden cambiar o incorporar nuevos miembros del equipo (Kendall, 2005).

2.2.4 SISTEMA ADMINISTRADOR DE BASE DE DATOS

BASE DE DATOS

De acuerdo a Ramos y Ramos (2007), sostienen que una base de datos (BD) es un conjunto de datos relacionados entre sí, organizados y estructurados, con información referente a algo, por su parte, Nevado (2010), manifiesta que, una base de datos es la información almacenada, que cumple una serie de características y restricciones.

SISTEMA GESTOR DE BASE DE DATOS

“El sistema gestor de base de datos (SGBD) es una aplicación que permite a los usuarios definir, crear y mantener la BD y proporciona un acceso controlado a la misma” (Ramos y Ramos, 2007, p: 3).

Según Cobo (2008), asegura que un sistema de gestión de base de datos es un software o conjunto de programas que permite crear y mantener una base de datos. El SGBD actúa como interfaz entre los programas de aplicación (Usuarios) y el sistema operativo. El objetivo principal de un SGBD es proporcionar un entorno eficiente a la hora de almacenar y recuperar la información de la base de datos.

Los SGBD, son aplicaciones que permiten a los usuarios definir, crear y mantener la base de datos y proporcionar un acceso controlado a la misma. Y que interactúa con los usuarios de los programas de aplicación y la base de datos.

Las características de un Sistema Gestor de Base de Datos (SGBD), son:

Abstracción de la información

Según Gutiérrez, G. (2010), opina que los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.

Según Cobo (2008), indica que el SGBD proporciona a los usuarios una visión abstracta de la información, es decir, el sistema ahorra al usuario la necesidad de conocer los detalles de cómo se almacena los datos. Para ocultar esos detalles, se definen varios niveles de abstracción.

Independencia

Según Taboada y Cotos (2005), aseguran que la información no depende de las aplicaciones cliente, estas accederán a la base de datos mediante alguna interfaz estándar.

Según Cobo (2008), sostiene que es la capacidad para modificar un esquema de definición sin afectar a los programas de aplicación.

De acuerdo a Pons, Marín, Medina, Acid y Vila (2005), afirman que el principio de independencia de datos establece el cumplimiento de la siguiente regla fundamental. Los datos deben organizarse independientemente de las aplicaciones que los vayan a usar y de los ficheros en los que vayan a almacenarse dichos datos.

Redundancia mínima

Según Cobo (2008), afirma que consiste en evitar el almacenamiento múltiple de una misma información para uso de distintas aplicaciones. En principio puede parecer que lo más conveniente es una redundancia nula, pero, en la práctica es mejor mantener ciertos duplicados a efectos de realizar búsqueda más rápida.

De acuerdo a Pons et al. (2005), indican que la redundancia no solo ocasiona el desaprovechamiento del espacio de almacenamiento, sino también acarrea problemas de inconsistencia. Como un SGBD provee un fondo de datos común que puede ser compartido por varias aplicaciones, no es preciso, por tanto, duplicar datos. Sin embargo, puesto que varias aplicaciones pueden acceder a los mismos datos al mismo tiempo, el SGBD debe disponer de los mecanismos adecuados para gestionar esas concurrencias.

Según Taboada y Cotos (2005), señalan que la existencia de redundancia puede dar lugar a dificultades para decidir qué información repetida es la correcta.

Consistencia

Según Gutiérrez, G. (2010), opina que en aquellos casos en los que no se ha logrado eliminar la redundancia, será necesario vigilar que aquella información que aparece repetida se actualice de forma simultánea. Por otra parte, la base de datos representa una realidad determinada que tiene determinadas condiciones.

Según Cobo (2008), asegura que consiste en impedir que exista información inconsistente o contradictoria en la base de datos. La inconsistencia surge cuando existe varias copias del mismo dato y tras la modificación de una de ellas, las demás no son actualizadas, o si lo son pero de forma incorrecta. Si existen datos duplicados, en la actualización de esos datos, el SGBD debe garantizar la adecuada actualización de los datos en todos los ficheros donde se encuentre.

Seguridad de los datos

Según Desongles (2005), sostiene que la seguridad de la base de datos es la protección de la base de datos frente a usuarios no autorizados. Sin unas buenas medidas de seguridad, la integración de datos en los

sistemas de base de datos hace que éstos sean más vulnerables que en los sistemas de ficheros. Sin embargo, los SGBD permiten mantener la seguridad mediante el establecimiento de claves para identificar al personal autorizado a utilizar la base de datos. Las autorizaciones se pueden realizar a nivel de operaciones.

De acuerdo a Pons et al. (2005), afirman que el sistema debe articular mecanismos que eviten que algún usuario pueda acceder a la información no autorizada. Se utilizaran mecanismos de identificación y gracias a ellos se podrán definir que usuario accede a que recurso. Todo ello gestionado por el SGBD.

Integridad de los datos

Según De Pablos, López, Santiago y Medina (2004), indican que lo deseable para la organización es que solo se almacene la información que es correcta. Una correcta estructuración de una base de datos permite que se den redundancias en la información con la consecuente materialización en la integridad de datos.

Según Nevado (2010), señala que garantiza la integridad de los datos, gestionando que los datos que se almacenan en la base de datos satisfagan las restricciones definidas en el esquema de la misma.

Según Ramos y Ramos (2007), opinan que permite mantener la integridad y consistencia de los datos utilizando mecanismos para evitar que los datos sean perjudicados por cambios no autorizados.

Respaldo y recuperación

Según Menéndez-Barzanallana (2012), asegura que los SGBD deben proporcionar una forma eficiente de realizar copias de seguridad de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.

Según Ramos y Ramos (2007), sostiene que permite mecanismos de copias de respaldo y recuperación para restablecer la información en caso de fallos en el sistema.

Control de la concurrencia

Según Nevado (2010), afirma que garantiza el acceso concurrente a la base de datos de forma que varios usuarios puedan acceder al mismo o distinto dato sin que se provoque pérdida en la integridad de los datos.

Según Ramos y Ramos (2007), indica que permite el acceso compartido a la base de datos, controlando la interacción entre usuarios concurrentes.

LENGUAJE DE CONSULTA ESTRUCTURADO (SQL)

De acuerdo a Kroenke (2003), señala que el lenguaje de consulta estructurado, o SQL (Structured Query Language) es el lenguaje de manejo de datos relacionales actual más importante. Ha recibido el respaldo del American National Standards Institute (ANSI) como el lenguaje seleccionado para el manejo de base de datos relacionales, y es el lenguaje de acceso a datos que usan muchos productos DBMS comerciales..., SQL ha sido el lenguaje estándar para el intercambio de información entre computadoras. Puesto que hay una versión SQL que puede funcionar en casi todas las computadoras y sistemas operativos, los sistemas de cómputo pueden intercambiar datos, consultas y respuestas.

Según Rob y Coronel (2003), opinan que el lenguaje se compone de comandos que les permite a los usuarios crear base de datos y estructuras de tabla, realizar varios tipos de manipulación y administración de datos y consultar la base de datos para extraer información útil. Todo el software de RDMS soporta SQL y muchos

vendedores de software han desarrollado extensiones del conjunto de comandos SQL básico.

2.2.5 LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS

A. LENGUAJE DE PROGRAMACIÓN

“Un lenguaje de programación puede definirse como una notación para escribir instrucciones u órdenes útiles para el ordenador y necesarias para la realización de un determinado proceso” (García, E., 2007, p: 62).

“Los lenguajes de programación son como idiomas que constituyen el sistema de comunicación entre el hombre y el ordenador, mediante el cual se transmiten a éste las instrucciones e información en un formato comprensible para la máquina” (Heredero, Romo y Medina, 2004, p: 110).

B. PROGRAMACIÓN ORIENTADA A OBJETOS (POO)

“La programación orientada a objetos trata de utilizar una visión real del mundo dentro de nuestros programas. La visión que se tiene del mundo dentro de la POO es que se encuentra formado por objetos” (Noriega, 2007).

Según Weitzenfeld (2005), asegura que la programación orientada a objetos define una estructura de más alto nivel llamada objeto, que ofrece dos ventajas sobre la programación tradicional:

- a) Permite al programador que organice su programa de acuerdo con abstracciones de más alto nivel, siendo éstas más cercanas a la manera de pensar de la gente.
- b) Los datos globales desaparecen, siendo éstos junto con las funciones parte interna de los objetos. Por lo tanto, cualquier cambio en la estructura de alguno de los datos sólo debiera afectar las funciones definidas en ese mismo objeto y no en los

demás.

B.1 ELEMENTOS DE LA POO

CLASE

“Una clase describe a un conjunto de objetos que comparten una estructura y un comportamiento común” (García, L., 2010, p: 7).

Según Noriega (2007), sostiene que una clase es un modelo o prototipo que define un tipo de objeto determinado. Una clase define los atributos y métodos que va a poseer un objeto. Mediante las clases se podrá crear o instanciar objetos de un mismo tipo, es decir, las clases ofrecen el beneficio de la reutilización.

OBJETO

“Un objeto de software es cualquier cosa real o abstracta acerca de la cual almacenamos datos y los métodos que controlan dichos datos” (Flores, 2007, p: 535).

Según Duran, Gutiérrez y Pimentel (2007), afirman que un objeto se puede considerar como una representación de un objeto real o aun no siendo tan real podemos hacernos una idea de que es un elemento existente con ciertas propiedades. Estos objetos no tienen que representar fielmente los objetos reales del mundo; solamente deben representar aquella información que sea necesaria para solucionar nuestro problema.

ATRIBUTO

Según Flores (2007), asegura que los atributos en principio, sirven para almacenar valores de los objetos que se instancian a partir de una clase. Y se clasifica en; atributos de instancia, son variables que almacenan valores distintos para todos los objetos de una misma clase, y atributos de clase, son variables que almacenan el mismo valor para todos los objetos de una misma clase.

“Un atributo es una propiedad que ayuda a describir un objeto” (García, L., 2010, p: 8).

MENSAJES

“Los mensajes son la forma que tienen de comunicarse distintos objetos entre sí. Puesto que un objeto por sí solo no es demasiado útil, sino que se suele utilizar dentro de una aplicación o programa que utiliza otros objetos” (Noriega, 2007, p: 23).

“Un mensaje está asociado con un método, de tal manera que cuando un objeto recibe un mensaje la respuesta a ese mensaje es ejecutar el método asociado” (Carrasco, 2004, p: 100).

MÉTODOS

“Abstracción de una acción, servicio, comportamiento o tarea que puede ser realizado por un objeto. Generalmente, un método manipula la información registrada en los atributos a través de una o más instrucciones” (García, L., 2010, p: 26).

“Son acciones que se pueden realizar con los objetos. También se define un método como la implementación de un mensaje, al fin acabo, un mensaje es la llamada o invocación de un método de un objeto” (Noriega, 2007, p: 25).

B.2 CARACTERÍSTICAS DE LA POO

ENCAPSULAMIENTO

“El encapsulamiento hace referencia a ocultar los detalles de implementación internos del objeto a los demás. Esta propiedad permite asegurar que el contenido de la información de un objeto se encuentra seguro del mundo exterior” (García, L., 2010, p: 40).

“El encapsulamiento nos permite combinar los datos y las operaciones en un mismo objeto, de forma que éste ofrece una interfaz de comunicación para relacionarse con otros objetos, haciendo inaccesibles todos los detalles internos de su funcionamiento.” (Carrillo, 2006, p: 353).

HERENCIA

Según Badenas, Llopis y Coltell (2001), indican que la herencia es el mecanismo más importante de la POO. En realidad es lo que le separa de la programación con tipos abstractos de datos. Los humanos solemos hacer abstracciones en dos dimensiones: *parte-de* y *tipo-de*. Un coche es un *tipo-de* vehículo y *tiene* ruedas, motor, etc. La relación *parte-de* ha sido utilizada y explotada por la programación durante tiempo, mientras que la herencia añade la otra relación.

Por su parte Deitel y Deitel (2004), señalan que la herencia es una forma de reutilización de software, en la que las nuevas clases se desarrollan rápidamente y absorben fácilmente las capacidades de las clases existentes y agregan de manera adecuada nuevas capacidades.

POLIMORFISMO

“El polimorfismo se refiere a la capacidad de adquirir muchas formas. En el contexto de POO, esto significa que el mismo método denota muchos métodos que son miembros de diferentes objetos” (Drozdek, 2007, p: 21).

“En POO, el polimorfismo permite que diferentes objetos respondan de modo diferente al mismo mensaje. El polimorfismo adquiere su máxima potencia cuando se utiliza en unión de herencia” (Joyanes, 2003, p: 686).

ABSTRACCIÓN

Según Noriega (2007), opina que la abstracción es la capacidad de ignorar algunos aspectos de la realidad con el fin de facilitar la realización de una nueva tarea. Nos permite ignorar aquellos aspectos de la realidad que no intervienen en el problema que deseamos abordar, y también nos permite ignorar los aspectos de implementación de los objetos en los pasos iniciales, con lo cual sólo necesitamos conocer qué es lo que hace un objeto, y no cómo lo hace, para definir un objeto y establecer las relaciones de éste con otros objetos.

De acuerdo a Xhafa, Vásquez, Gómez, Molinero y Martin (2006), aseguran que la abstracción tiene que ver con el conocimiento de las similitudes entre los objetos, situaciones o procesos del mundo real, el estudio de estas similitudes y prescindiendo de las diferencias.

2.2.6 ARQUITECTURA MODELO VISTA CONTROLADOR (MVC)

De acuerdo a Berzal, Cubero y Cortijo (2005), sostienen que el modelo MVC ayuda a modularizar correctamente una aplicación en la cual el usuario manipula datos a través de una interfaz. Si el usuario puede trabajar con los mismos datos de distintas formas, lo habitual es encapsular el código compartido en un módulo aparte con el fin de evitar la existencia de código duplicado. Se puede decir que el modelo contiene el comportamiento común a las distintas formas que tiene el usuario de manipular los datos. De hecho, la existencia de un modelo independiente facilita enormemente la construcción de sistemas que han de ofrecer varios interfaces.

“Fue creada en 1979 por Trygve Reenskaung. Es un patrón que permite separar la GUI, de los datos y de la lógica apoyándose en tres componentes. A saber: modelo, vista y controlador” (Cardona, Jaramillo y Villegas, 2008, p: 120).

Modelo

“La capa de modelo describe las entidades propias del dominio del problema que resuelve la aplicación. En esta capa aparecen los llamados objetos de negocio que abstraen las características del dominio del problema” (Katrib, del Valle, Sierra y Hernández, 2009, p: 59).

“El modelo, que es la aplicación del dominio, no tiene conocimiento específico de la vista, que es la interfaz de usuario, ni de los controladores, que son los manejadores de eventos” (Alonso, Martínez y Segovia, 2005, p: 473).

Vista

“La capa de vista describe la apariencia y funcionalidad de la interfaz de usuario. Las monumentales capacidades de Windows Presentation Foundation sirven de soporte a esta capa” (Katrib et al., 2009, p: 59).

“La vista (interfaz de usuario) tiene conocimiento de los controladores (los manejadores de eventos) porque los crea y especifica qué tipo de evento, emitido por cada componente, debe atender cada manejador. Sin embargo la vista no tiene conocimiento del modelo” (Alonso et al., 2005, p: 473).

Controlador

Según Alonso et al. (2005), afirman que los controladores tienen conocimiento del modelo porque crean objetos del dominio al que mandan mensajes, derivados fundamentalmente de los eventos producidos en la interfaz de usuarios; y tienen conocimiento de la vista porque mandan mensajes a sus componentes para recibir o enviarles información.

“El propósito de la capa de controlador es crear un puente entre las otras capas y depende de la descripción de las otras dos capas”

(Katrib et al., 2009, p: 59).

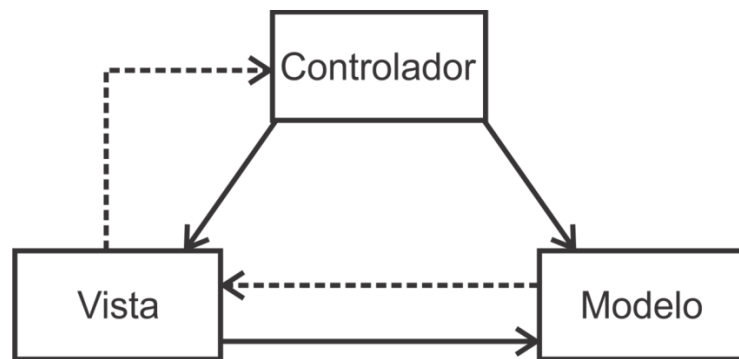


Figura N° 2.5: Programación Avanzada en Java (Jaramillo, Cardona y Villa, 2008).

2.2.7 APLICACIÓN WEB

Una aplicación web es un conjunto de recursos web que participan en el funcionamiento de la propia aplicación web. Una aplicación web está compuesta (Aumaille, 2002):

- De componentes de servidor dinámicos; Servlets y JSPs.
- De bibliotecas de clases java utilitarias.
- De elementos web estáticos; paginas HTML, imágenes y sonidos.
- De componentes de cliente dinámicos; Applets, JavaBeans y clases.
- De un descriptor de desarrollo y de configuración de la aplicación web en formato de uno o múltiples archivos XML (fundamentalmente web.xml). Este archivo permite definir el entorno de ejecución de la aplicación web así como relacionar entre si los componentes.

Una aplicación web es un software que se consulta por medio de un navegador web. En lugar de funcionar en forma local, gran parte del código del programa de software se ejecuta en una computadora remota conectada a internet u otra red. Existen aplicaciones web para muchas de las aplicaciones locales, como correo electrónico, calendarios, base de datos, compartir fotografías, administración de

proyectos, juegos, mapas e incluso procesadores de texto (Jamrich y Oja, 2008).

SERVLET

La interacción en la web se logra mediante programas en un equipo servidor que interactúan con el software, por lo general un navegador, en un equipo cliente. Una alternativa para CGI es utilizar Servlet de java en el servido. Un Servlet es un programa en Java codificado de acuerdo con ciertas convenciones, con clases de utilería, por ejemplo para tener acceso a los datos del cliente. Los beneficios de los Servlets son (Bell y Parr, 2003):

- a. Cada vez que se utiliza la página web y se envían los datos al servidor, el Servlet crea un nuevo subproceso;
- b. Un Servlet puede ejecutarse en una caja de arena, ofreciendo mayor seguridad;
- c. Un Servlet es independiente de la plataforma.

Un Servlet de Java es otro lenguaje de alto nivel para el desarrollo de aplicaciones cliente/servidor. Un Servlet es un programa en Java que se ejecuta en un servidor Web y los procesos de páginas Web utilizando el Protocolo de transferencia de hipertexto (HTTP). En una aplicación Web, el navegador actúa como cliente (Morelli y Walde, 2006).

JSP

JSP es una tecnología que permite añadir contenido dinámico a las páginas web convencionales. Sin JSP siempre se tenía que actualizar la apariencia o el contenido de las páginas HTML planas o estáticas a mano. Incluso cuando todo lo que se quería hacer era cambiar una fecha o una imagen, se necesitaba editar el archivo HTML y escribir el nuevo código. Nadie más lo haría por usted, y ciertamente no se haría solo, mientras que con JSP puede hacerse el contenido dependiente de muchos factores, incluso dependiendo de la hora del día, la

información brindada por un usuario, la historia de interacción con el sitio web, e incluso por el tipo de navegador (Zambon y Sekler, 2007).

2.2.8 TECNOLOGÍAS DE INTERNET

A. INTERNET

Internet no es una simple red de ordenadores, sino una red de redes, es decir, un conjunto de redes interconectada a escala mundial con la particularidad de que cada una de ellas es independiente y autónoma. El funcionamiento de internet se basa en tres factores fundamentales (Rodríguez, 2006):

- a. Protocolos de comunicación; un protocolo es un conjunto consensuada de normas que determinan como debe funcionar algo. Estos hacen posible que distintos ordenadores repartidos por todo el mundo puedan intercambiar datos.
- b. Dirección IP; a cada ordenador se le asigna una dirección o un nombre que se le conoce como dirección IP, y que es única para cada uno de ellos.
- c. Servidores; es necesario que exista algún ordenador que organice un poco la comunicación entre unos equipos y otros, para garantizar de este modo el funcionamiento de la red.

Internet es una red de computadores mundial, esto es, una red que conecta millones de dispositivos de cómputo a través del mundo. La mayoría de estos dispositivos de computo son los PC tradicionales de sobremesa, las estaciones de trabajo basados en UNIX y los llamados servidores, que almacenan y transmiten información de tipo de páginas web y mensajes de correo electrónico (Kurose y Ross, 2004).

B. HTTP

El protocolo HTTP es el responsable del servicio World Wide Web. Una de sus vulnerabilidades más conocidas procede de la posibilidad de entrega de información por parte de los usuarios del servicio. Esta

entrega de información desde el cliente de HTTP es posible mediante la ejecución remota de código en la parte del servidor. La ejecución de este código por parte del servidor suele utilizarse para dar el formato adecuado tanto a la información entregada por el usuario como a los resultados devueltos (para que el navegador del cliente la pueda visualizar correctamente). Si este código que se ejecuta presenta deficiencias de programación, la seguridad del equipo en el que esté funcionando el servidor se podrá poner en peligro (Stallings, 2004).

C. PROTOCOLO TCP/IP

El protocolo es el elemento que hace posible que los distintos ordenadores repartidos por el mundo y conectados a la red intercambien información. El protocolo que utiliza internet es el TCP/IP (Kurose y Ross, 2004).

El protocolo TCP/IP, es el protocolo que utiliza internet para encaminar la información entre dispositivos, por este motivo toda máquina que se conecte a Internet debe implementar este dispositivo. IP es el protocolo de capa de red de la familia de protocolos TCP/IP, interactúa en esta capa con un conjunto de protocolos subsidiarios que aportan mensajes de control y resolución de direcciones físicas a direcciones IP. Además el conjunto de protocolos TCP/IP incluye dos protocolos en la capa de transporte; el Protocolo de Control de Transmisión (TCP, Transmission Control Protocol), que es un protocolo orientado a conexión, y el Protocolo Datagrama de Usuario, (UDP, User Datagram Protocol), que es no orientado a conexión. (Stallings, 2004).

2.2.9 POBLACIÓN Y MUESTRA

De acuerdo a Díaz (2006), indica que se denomina población o universo a la totalidad o conjunto de elementos (personas, animales, números, etc.) susceptibles de presentar una o varias características en común que estén bien definidos, llamándose muestra a cualquier

subconjunto de la población o universo.

A. POBLACIÓN

De acuerdo a Tamayo (2004), señala que la totalidad de un fenómeno de estudio, incluye la totalidad de unidades de análisis o entidades de población que integran dicho fenómeno y que debe cuantificarse para un determinado estudio integrando un conjunto N de entidades que participan de una determinada característica, y se le denomina población.

B. MUESTRA

Según Martel y Díez (1997), opina que la muestra es un subconjunto de individuos pertenecientes una población, y representativos de la misma. Existen diversas formas de obtención de la muestra en función del análisis que se pretenda efectuar (aleatorio, por conglomeración, etc.).

Muestreo.-es un instrumento de gran validez, en la investigación, con el cual el investigador selecciona las unidades representativas a partir de las cuales obtendrá los datos que le permitirán extraer inferencias acerca de la población sobre la cual se investiga (Tamayo, 2004).

Muestreo por conveniencia.- consiste en la decisión que toma el investigador, según los objetivos y propósitos del estudio, sobre los elementos que a su juicio sean pertinentes o representantes apropiados de la población para integrar la muestra. En este caso, el investigador conoce la población y las características que pueden ser utilizadas para seleccionar la muestra (Borda, Tiesca y Navarro, 2014).

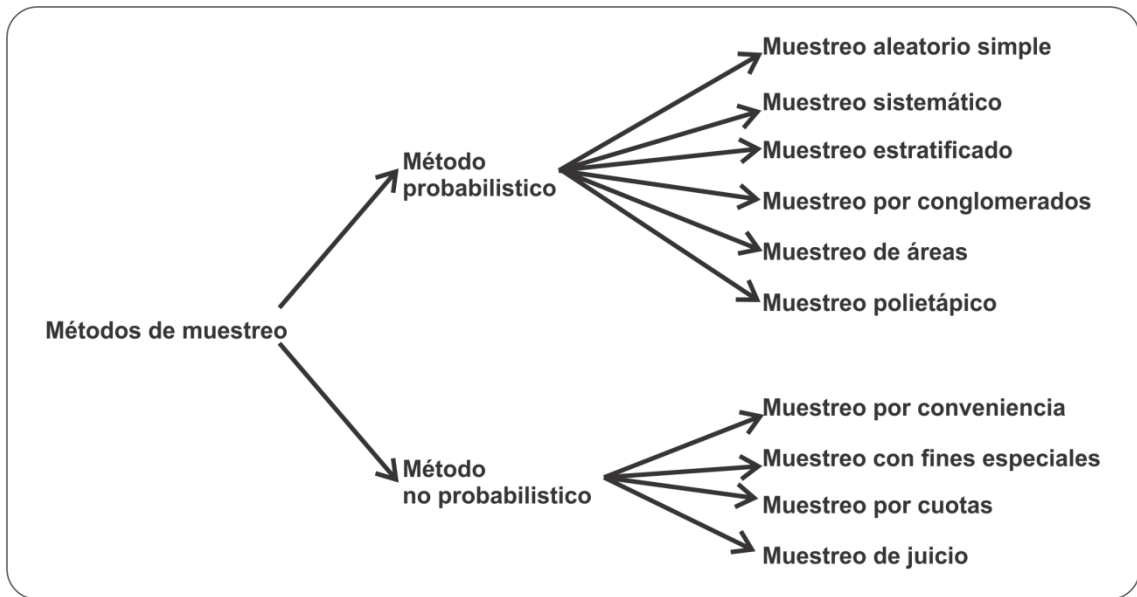


Figura N° 2.6: Métodos de Muestreo (Bernal, 2006).

CAPÍTULO III

METODOLOGÍA DE LA INVESTIGACIÓN

3.1. TIPO DE INVESTIGACIÓN

De acuerdo a Bernal (2006), asegura que la investigación descriptiva es uno de los tipos o procedimientos investigativos más populares y utilizados por los principiantes en la actividad investigativa. Los trabajos de grado, en los pregrados y en muchas de las maestrías, son estudios de carácter eminentemente descriptivo. En tales estudios se muestran, narran, reseñan o identifican hechos, situaciones, rasgos, características de un objeto de estudio, o se diseñan productos, modelos, prototipos, guías, etcétera, pero no se dan explicaciones o razones de las situaciones, los hechos, los fenómenos, etcétera.

Por lo mencionado, el tipo de investigación es descriptiva, porque en la práctica se desarrolló un software que describe las características del Centro de Producción de Allpachaka.

3.2. DISEÑO DE LA INVESTIGACIÓN

De acuerdo a Hernández, Fernández y Baptista (2006), opinan que la investigación no experimental cuantitativa se realiza sin manipular deliberadamente la variable. Es decir, se trata de estudios, donde no se hace variar en forma intencional la variable independiente para ver su efecto sobre otras variables. Lo que se hace en la investigación no experimental, es observar fenómenos tal como se dan en su contexto natural, para después analizarlos.

El diseño de la presente investigación es no experimental, pues no se manipulan las variables, sino, que más bien se estudian y describen tal

como se presentan.

En la presente investigación, la recolección de datos se realizó para un periodo de tiempo establecido, en un solo momento, por lo tanto, se considera que el diseño de la investigación es no experimental y transversal.

3.3. POBLACIÓN Y MUESTRA

POBLACIÓN.- La población estuvo compuesta por; 24 terneros, 12 terneras, 67 vaquillas, 17 vaquillonas, 132 vacas y 3 toros del Centro de Producción Allpachaka - UNSCH, 2016.

MUESTRA.- En esta investigación todos los bovinos cumplen con ciertas características físicas y, cualquiera de ellas es representativo de la población y un bovino que no se tome en cuenta, no afectará el resultado del estudio, es decir; la "conveniencia" en la selección realmente no va a representar algún sesgo. Por tanto, se tomó una muestra por conveniencia conformada por; 2 terneras, 2 vaquillas, 2 vaquillonas, 14 vacas y 3 toros del Centro de Producción Allpachaka - UNSCH, 2016.

3.4. VARIABLES E INDICADORES

3.4.1. DEFINICIÓN CONCEPTUAL DE LAS VARIABLES

PRIMERA VARIABLE

Reproducción de ganado bovino.- La reproducción de ganado bovino es un conjunto de procesos relacionados a la generación de nuevos terneros, que incluyen; la gametogénesis, fecundación, gestación y parto, y es la actividad esencial para iniciar la producción de leche. En el caso de las vacas la edad de la madurez sexual o capacidad reproductiva, con un buen crecimiento y desarrollo, se puede dar a los

14 o 15 meses y pueden parir a los 24 meses, mientras que en los toros se da a los 12 o 13 meses. El celo tiene una duración de 20 horas aproximadamente, si la vaca no es servida (por monta natural o inseminación artificial) en el momento ideal, el celo retornará nuevamente a los 21 días en promedio. El periodo de gestación en las vacas es aproximadamente 280 días (con un rango de 10 días más o menos) y se restablece su ciclo estral a los 30 o 35 días después del parto, y se recomienda tener un periodo de espera voluntario de 40 a 60 días, generalmente se sirve en el celo que aparece después de los 50 días del parto.

INDICADORES DE LA PRIMERA VARIABLE

Pubertad.- La pubertad en los terneros dependerá del nivel nutricional, es decir, los terneros adecuadamente alimentados crecerán más rápido e iniciaran su actividad sexual a una edad temprana. Con miras a lograr una preñez pronta, se considera contar con terneras de alrededor de 10 meses de edad con un peso aproximado de 277 kg.

Primer servicio.- Para servir a una novilla e incorporarlo cuanto antes a la producción, se tiene en cuenta su adecuada edad y peso. Cuando la novilla entre en celos a los seis meses de edad, no se recomienda servirla, porque su órgano reproductor no se ha desarrollado plenamente para albergar un feto y tendría un parto complicado. Para servirla deberá tener un peso que oscile entre los 370 y 414 kg. Cuando la novilla llega a la edad necesaria y alcanza este peso, estará apta para quedar preñada. Los toros pueden empezar a montar vacas desde los 12 meses de edad, sin embargo se recomienda utilizarlos cuando tengan 18 meses. El toro no tendrá que engordar demasiado porque se volverá sexualmente perezoso.

Días abiertos.- Es el tiempo transcurrido desde el parto hasta que se da el "primer servicio", lo ideal es que este indicador no sea mayor de 140

días. Las causas más comunes por las que se alarga, son las infecciones uterinas que ocasionan retraso en la involución uterina y por la mala detección del éstro.

Servicios por concepción.- Es el número de servicios que en promedio se necesitan para que una vaca quede preñada. Se obtiene de sumar todos los servicios que se hayan realizado en el hato durante un tiempo determinado, y dividirlos entre el número de vacas diagnosticadas preñadas a la palpación. Resultados superiores a 2.5 nos advierte problemas en el proceso de celo o servicio.

Intervalo entre partos.- Es el periodo transcurrido entre un parto y otro en la misma vaca. Se calcula contando los días a partir de la fecha del último parto a la fecha del parto inmediato anterior, lo óptimo es tener un periodo interparto de 365 días, este indicador influye en el número de partos en la vida productiva.

SEGUNDA VARIABLE

Producción láctea.- la producción láctea se inicia únicamente cuando nace el ternero, durante las primeras 8 semanas la producción de leche se incrementa y luego disminuye gradualmente hasta el momento del secado, exactamente a los 7 meses de gestación o cuando baja la producción diaria de leche. El secado permite la reposición y regeneración de la glándula mamaria para que esté nuevamente en condiciones de producir leche en cantidad y calidad después del parto. El proceso de ordeño se realiza de manera manual o mecánica de dos a tres veces al día.

INDICADORES DE LA SEGUNDA VARIABLE

Producción individual.- es la producción de leche que se obtiene de una vaca en el periodo de lactancia en un año y está determinado por el descanso entre partos.

Duración de lactancia.- la duración de lactancia, está determinada por el tiempo que transcurre desde el inicio del parto hasta 2 meses antes del próximo parto. Una duración de 305 días es considerada óptima, y se divide en tercios, el primero va desde el día 1 hasta el 100; el segundo del día 101 hasta el 200 y el último tercio desde el día 201 hasta el 305.

3.4.2. DEFINICIÓN OPERACIONAL DE LAS VARIABLES

PRIMERA VARIABLE

X: Reproducción de ganado bovino.

INDICADORES DE LA PRIMERA VARIABLE

X1: Pubertad.

X2: Primer servicio.

X3: Días abiertos.

X4: Servicios por concepción.

X5: Intervalo entre parto.

SEGUNDA VARIABLE

Y: Producción láctea.

INDICADORES DE LA SEGUNDA VARIABLE

Y1: Producción individual.

Y2: Duración de lactancia.

OPERACIONALIZACIÓN DE LAS VARIABLES

Se muestra en el anexo D.

3.5. TÉCNICAS E INSTRUMENTOS

3.5.1. TÉCNICAS PARA RECOLECTAR INFORMACIÓN

Se empleó técnicas como; entrevista, encuesta y análisis documental para el levantamiento de información, sobre el que se analizará la información para el desarrollo de la aplicación web.

3.5.2. INSTRUMENTO PARA RECOLECTAR INFORMACIÓN

Se utilizó cuestionarios para entrevistar a los productores pecuarios (técnico pecuario, veterinario, biólogo, agrónomo) del Centro de Producción, para la recolección de datos e información referente a la reproducción de ganado bovino, producción láctea, edad a la pubertad, edad al primer servicio, días abiertos, servicio por concepción e intervalo entre partos. A partir de los cuales se determinó los requerimientos funcionales para la aplicación web. (Ver Anexo A).

Para reforzar la entrevista se encuestó a los productores pecuarios, con el propósito de levantar información necesaria en relación a la reproducción de ganado bovino, producción láctea, edad a la pubertad, edad al primer servicio, días abiertos, servicio por concepción e intervalo entre partos. (Ver Anexo B).

Para determinar cómo son los procesos en la reproducción de ganado bovino y producción láctea, se empleó fichas para análisis documental (tarjetas individuales, libros de hato, registro de sanidad, etc). (Ver Anexo C).

3.5.3. HERRAMIENTAS PARA EL TRATAMIENTO DE DATOS E INFORMACIÓN

Software	Versión	Descripción
WINDOWS 7	Ultimate	Windows 7 es una versión de Microsoft Windows, línea de sistemas operativos producida por Microsoft Corporation. Esta versión está diseñada para uso en PC, incluyendo equipos de escritorio en hogares y oficinas

SQL SERVER	11.0.2100.60	SQL Server es un sistema de gestión de bases de datos relacionales (RDBMS) de Microsoft que está diseñado para el entorno empresarial. SQL Server se ejecuta en T-SQL (Transact-SQL), un conjunto de extensiones de programación de Sybase y Microsoft que añaden varias características de SQL estándar, incluyendo control de transacciones, excepción y manejo de errores, procesamiento de fila, así como variables declaradas.
J2EE	1.4	Java Platform, Enterprise Edition, es una plataforma de programación—parte de la Plataforma Java—para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java. Permite utilizar arquitecturas de N capas distribuidas y se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.
Netbeans IDE	8.0.2	Es un entorno de desarrollo - una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.
Jasperreport	4.0.2	Es una herramienta para generar reportes en Java, con la habilidad de producir contenido completo para la pantalla, directo para impresora o en diferentes formatos de archivo.

Tabla N° 3.1 Herramientas tecnológicas que se usarán.

3.5.4. TÉCNICAS PARA APLICAR XP

Revisado el marco teórico desarrollado en el capítulo II, sección 2.2.3, formulamos el proceso con apoyo de la referencia, que considera las fases para desarrollar la aplicación web usando XP, como se indica en las tablas 3.2 a 3.4.

TAREA	ARTEFACTO	TÉCNICA	RESPONSABLES
-------	-----------	---------	--------------

Escribir historias de usuario	Historia de usuario	Describir brevemente la historia de usuario con la regla del negocio (lo que el sistema debe hacer) Dividir historias de usuario grandes	Cliente
Probar las tecnologías a utilizar	Arquitectura técnica inicial	Explorar posibilidades de uso de tecnologías Probar el rendimiento de las tecnologías Definir las tecnologías a usar	Cliente Programador Entrenador
Estimar esfuerzo para historia de usuario	Plan de alto nivel	Conocer previamente la historia de usuario Hacer una implementación rápida de historia de usuario Estimar esfuerzo (semana) para desarrollar la historia de usuario	Programador

Tabla N° 3.2: Fase de Exploración (Porras, 2010).

TAREA	ARTEFACTO	TÉCNICA	RESPONSABLES
Rescribir las historias de usuario	Historia de usuario	Describir detalladamente la historia de usuario con la regla de negocio	Cliente
Formular el plan de versiones	Plan de versión (una iteración)	Introducir nuevos requisitos del software. Definir prioridad para cada historia de usuario por necesidad del negocio.	Cliente
		Utilizar técnicas de elaboración del plan de alto nivel. Estimar y asignar esfuerzo (semana) para cada historia de usuario en función a tiempo para planear, diseñar, implementar y probar.	Programador

		<p>Estimar y asignar riesgo a cada historia de usuario en función a situación que afecta la estimación del esfuerzo.</p> <p>Actualizar tarjeta de historia de usuario.</p>	
--	--	--	--

Tabla N° 3.3: Fase de planificación (Porras, 2010).

TAREA	ARTEFACTO	TÉCNICA	RESPONSABLES
Definir la arquitectura técnica	Arquitectura técnica	<p>Actualizar la arquitectura técnica inicial.</p> <p>Usar características del negocio.</p> <p>Utilizar arquitectura por capas.</p> <p>Integrar frameworks</p>	<p>Cliente</p> <p>Programador</p> <p>Entrenador</p>
Escribir tareas de ingeniería	Tarea de ingeniería	Dividir cada historia de usuario en tareas, describir usando reglas del negocio para cada tarea de ingeniería.	<p>Cliente</p> <p>programador</p>
Formular el plan de iteraciones	Plan de iteración	Estimar y asignar esfuerzo para desarrollar una tarea de ingeniería.	Programador
		Asignar una tarea de ingeniería al programador.	Entrenador Programador
		Utilizar el plan de versión.	
		<p>Actualizar el plan con tareas de ingeniería de la siguiente iteración.</p> <p>Actualizar el plan cuándo fallo prueba de aceptación.</p> <p>Actualizar el plan con tareas no concluidas.</p> <p>Actualizar las tarjetas de tarea de ingeniería.</p>	

Crear pruebas de aceptación	Caso de prueba de aceptación	Escribir pruebas de aceptación para cada historia de usuario por iteración.	Cliente Encargado de pruebas
Implementar las interfaces	GUI	Diseñar con precisión la GUI relacionada a cada historia de usuario. Generar código para la interface usando herramienta.	Cliente Programador
Escribir tarjetas CRC para cada tarea de ingeniería	Tarjeta CRC	Diseñar para una tarea de ingeniería de forma simple. Rediseñar por falla de prueba de aceptación una tarea Identificar responsabilidades Identificar colaboración. Identificar Atributos	Cliente Programador
Implementar la base de datos física	Base de datos física	Escribir script usando tarjeta CRC. Ejecutar script usando DBMS.	Programador
Implementar código para clases entidad	Código fuente	Escribir código fuente o generar con una herramienta usando tarjetas CRC.	Programador
Crear pruebas unitarias para las clases control	Prueba unitaria	Escribir código fuente para una prueba unitaria, usando una herramienta.	Programador
Implementar código fuente	Código fuente	Codificar una tarea de ingeniería. Hacer refactoring. Mover programadores.	Programador Supervisor
Ejecutar pruebas unitarias	Reporte de prueba unitaria	Ejecutar el módulo de cada prueba unitaria. Modificar código fuente si la prueba unitaria muestra resultado incorrecto.	Programador

Realizar integración continua	Código fuente	Integrar las tareas para una historia de usuario. Mantener sistema integrado todo el tiempo.	Programador
Ejecutar pruebas de integración para una historia de usuario	Reporte pruebas de integración	Integrar continuamente al concluir las tareas de una historia de usuario. Verificar que las pruebas de integración pasan al 100%.	Programador
Ejecutar pruebas de aceptación	Reporte de pruebas de aceptación	Correr la última versión de una iteración. Utilizar los casos de prueba de aceptación.	Cliente Encargado de pruebas

Tabla N° 3.4: Fase de iteración (Porras, 2010).

CAPÍTULO IV

ANÁLISIS Y RESULTADOS DE LA INVESTIGACIÓN

4.1. ARTEFACTOS DEL SOFTWARE APLICANDO EL PROCESO XP

4.1.1. FASE DE EXPLORACIÓN

Según el procedimiento desarrollado en la tabla 3.2 para el proceso XP, descrito en el capítulo II, sección 2.2.3, fase de exploración. Obtenemos las historias de usuario, arquitectura técnica inicial y el plan de alto nivel.

N°	HISTORIA DE USUARIO	DESCRIPCIÓN
1	Registrar alta de bovino.	El usuario (productor pecuario), podrá registrar el alta de los ganados bovinos.
2	Registrar calor o celo.	El usuario (productor pecuario), podrá registra el calor cuando la vaca entra en celo.
3	Registrar servicio de monta o empadre.	El usuario (productor pecuario), podrá registrar el empadre de la vaca y el toro utilizado.
4	Registrar diagnóstico de preñez.	El usuario (productor pecuario), podrá registrar el resultado del diagnóstico de preñez realizada a los 35 días después del servicio de monta.
5	Registrar datos del bovino.	El usuario (productor pecuario), podrá registrar los datos del ternero nacido en el ható lechero.
6	Registrar parto.	El usuario (productor pecuario), podrá registrar las condiciones del parto.
7	Registrar aborto.	El usuario (productor pecuario), podrá registrar el aborto sufrido por una de las vacas.
8	Emitir reporte servicio concepción.	El usuario (productor pecuario), podrá visualizar el registro de reproducción individual de la vaca.
9	Registrar producción de leche.	El usuario (productor pecuario), podrá registrar la cantidad de leche recolectada en un ordeño.
10	Registrar propiedades de la leche.	El usuario (productor pecuario), podrá registrar las propiedades de la leche que produce la vaca en cada campaña.
11	Registrar secado	El usuario (productor pecuario), podrá registrar la

	de la vaca.	fecha en que la vaca es secada.
12	Registrar estado nutricional de la vaca.	El usuario (productor pecuario), podrá registrar el estado nutricional de la vaca reproductora en cada tercio de su ciclo de producción láctea.
13	Registrar características morfológicas de la vaca.	El usuario (productor pecuario), podrá registrar los resultados de cada parámetro de valoración lineal obtenidos a partir de la evaluación morfológica de la vaca a una determinada edad.
14	Registrar característica morfológica del semental.	El usuario (productor pecuario), podrá registrar los resultados de cada parámetro de valoración lineal obtenidos a partir de la evaluación morfológica del semental a una determinada edad.
15	Registrar característica física del aparato reproductor del bovino macho.	El usuario (productor pecuario), podrá registrar la característica física del aparato reproductor del bovino macho durante su crecimiento y desarrollo.
16	Registrar evaluación de libido.	El usuario (productor pecuario), podrá registrar los resultados de cada parámetro de valoración lineal respecto al libido del semental.
17	Registrar cualidades de semen.	El usuario (productor pecuario), podrá registrar las cualidades del semen producidas por el semental, a partir de su madurez sexual o capacidad para reproducirse.
18	Registrar colección de semen.	El usuario (veterinario), podrá registrar la colección del semen.
19	Registrar tratamiento sanitario del bovino.	El usuario (veterinario), podrá registrar el tratamiento sanitario del bovino en cada etapa de su vida.
20	Registrar crecimiento y desarrollo del bovino.	El usuario (productor pecuario), podrá registrar los parámetros de crecimiento y desarrollo del bovino en cada periodo del mismo.
21	Registrar saca de bovinos.	El usuario (productor pecuario), podrá registrar los bovinos que son separados del hato para beneficio (consumo) o reproducción (re-cría).
22	Registrar mortandad de bovinos.	El usuario (productor pecuario), podrá registrar el resultado del diagnóstico sobre la causa de muerte súbita de bovino.
23	Crear cuenta de usuario.	El usuario (administrador), podrá administrar cuentas de usuario.
24	Iniciar sesión.	El usuario (productor pecuario), podrá autenticarse para iniciar sesión en la aplicación web.

Tabla N° 4.1: Historias de usuario.

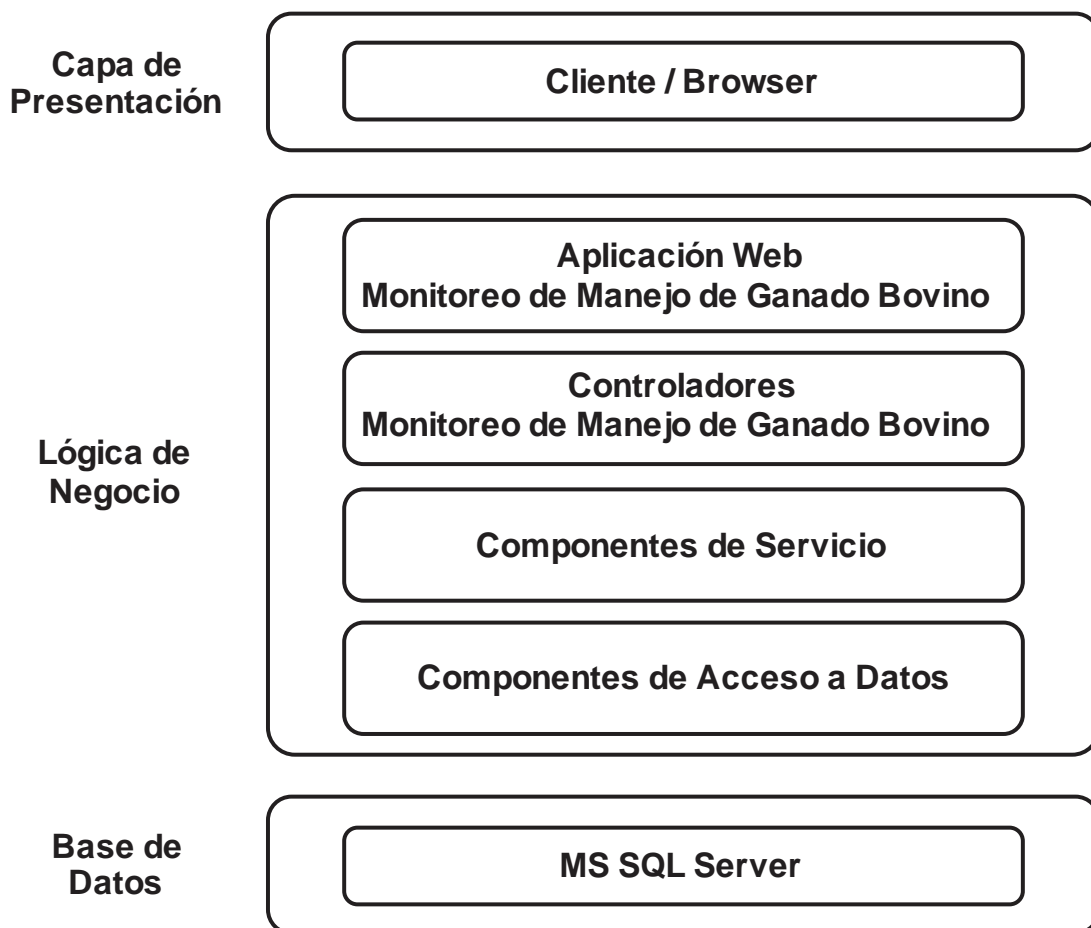


Figura N° 4.1: Arquitectura técnica inicial

N°	HISTORIAS DE USUARIO	ESFUERZO (días)
1	Registrar alta de bovino.	4
2	Registrar calor o celo.	2
3	Registrar servicio de monta o empadre.	3
4	Registrar diagnóstico de preñez.	2
5	Registrar datos del bovino.	4
6	Registrar parto.	2
7	Registrar aborto.	2
8	Emitir reporte servicio concepción.	5
9	Registrar producción de leche.	3
10	Registrar propiedades de la leche.	3
11	Registrar secado de la vaca.	3
12	Registrar estado nutricional de la vaca.	3
13	Registrar características morfológicas de la vaca.	3

14	Registrar características morfológicas del semental.	3
15	Registrar características físicas del aparato reproductor del bovino macho.	3
16	Registrar evaluación de libido.	3
17	Registrar cualidades de semen.	3
18	Registrar colección de semen.	4
19	Registrar tratamiento sanitario del bovino.	3
20	Registrar crecimiento y desarrollo del bovino.	3
21	Registrar saca de bovinos.	3
22	Registrar mortalidad de bovinos.	3
23	Crear cuentas de usuario.	3
24	Iniciar sesión	2

Tabla N° 4.2: Plan de alto nivel.

4.1.2. FASE DE PLANEAMIENTO

Aplicamos la técnica para la fase de planificación, presentada en la tabla 3.3, según la teoría del capítulo II, sección 2.2.3, obteniendo los artefactos; historias de usuario en detalle y el plan de versión (primera iteración).

Historia de Usuario	
Número: 1	Usuario: Productor pecuario.
Nombre historia: Registrar alta de bovino.	
Prioridad en negocio: Alto.	Riesgo en desarrollo: Alto.
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (Productor pecuario), una vez situada en la página principal hace clic en el menú “ control de movimientos ”, y la aplicación carga la página “ controlMovimientos.jsp ”; el usuario llena los datos en el formulario referente a la alta de bovino y hace clic en el botón “ Guardar ”; la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de datos el nuevo registro de alta y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	
Observaciones: El sistema podrá sugerir el código de bovino para un determinado año.	

Tabla N° 4.3: Historia usuario. Registrar alta de bovino.

Historia de Usuario	
Número: 2	Usuario: Técnico pecuario.
Nombre historia: Registrar calor o celo.	
Prioridad en negocio: Alto.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (técnico pecuario), una vez situada en la página principal hace clic en el menú “reproducción” , y la aplicación carga la página “reproducción.jsp” ; el usuario llena los datos en el formulario respecto al celo y hace clic en el botón “Guardar” ; la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	
Observaciones: Ninguno.	

Tabla N° 4.4: Historia usuario. Registrar calor o celo.

Historia de Usuario	
Número: 3	Usuario: Veterinario.
Nombre historia: Registrar servicio de monta o empadre.	
Prioridad en negocio: Alto.	Riesgo en desarrollo: Alto.
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (veterinario), una vez situada en la página principal hace clic en el menú “reproducción” , y la aplicación carga la página “reproducción.jsp” ; el usuario hace clic en el menú “Servicio” , y la aplicación muestra el formulario respecto al servicio de monta; el usuario llena los datos en el formulario cuando se sirve a una vaca con determinado semental, luego hace clic en el botón “Guardar” ; la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	
Observaciones: Ninguno.	

Tabla N° 4.5: Historia usuario. Registrar servicio de monta o empadre.

Historia de Usuario	
Número: 4	Usuario: Veterinario.
Nombre historia: Registrar diagnóstico de preñez.	
Prioridad en negocio: Alto.	Riesgo en desarrollo: Alto.
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (veterinario), una vez situada en la página principal hace clic en el menú “reproducción” , y la aplicación carga la página “reproducción.jsp” ; el usuario hace clic en el menú “Diagnostico de preñez” , y la aplicación	

muestra el formulario respecto al diagnóstico de preñez; el usuario llena los datos en el formulario el resultado del diagnóstico, luego hace clic en el botón “Guardar” ; la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.
Observaciones: Ninguno.

Tabla N° 4.6: Historia usuario. Registrar diagnóstico de preñez.

Historia de Usuario	
Número: 5	Usuario: Administrador.
Nombre historia: Registrar datos de bovino.	
Prioridad en negocio: Alto.	Riesgo en desarrollo: Alto.
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (administrador), una vez situada en la página principal hace clic en el menú “reproducción” , y la aplicación carga la página “reproducción.jsp” ; el usuario hace clic en el menú “Parto” , y la aplicación muestra el formulario respecto a los datos del bovino; el usuario llena los datos en el formulario y hace clic en el botón “Guardar” ; la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	
Observaciones: Ninguno.	

Tabla N° 4.7: Historia usuario. Registrar bovino.

Historia de Usuario	
Número: 6	Usuario: Productor pecuario.
Nombre historia: Registrar parto.	
Prioridad en negocio: Alto.	Riesgo en desarrollo: Alto.
Puntos estimados: 1	Iteración asignada: 4
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (productor pecuario), una vez situada en la página principal hace clic en el menú “reproducción” , y la aplicación carga la página “reproducción.jsp” ; el usuario hace clic en el menú “Parto” , y la aplicación muestra el formulario respecto a la condición del parto; el usuario llena los datos en el formulario y hace clic en el botón “Guardar” ; la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	
Observaciones: Ninguno.	

Tabla N° 4.8: Historia usuario. Registrar parto.

Historia de Usuario	
Número: 7	Usuario: Veterinario.

Nombre historia: Registrar aborto.	
Prioridad en negocio: Medio.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 4
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (veterinario), una vez situada en la página principal hace clic en el menú “ reproducción ”, y la aplicación carga la página “ reproducción.jsp ”; el usuario hace clic en el menú “ Aborto ”, y la aplicación muestra el formulario respecto al aborto; el usuario llena los datos en el formulario las causas del aborto sufrido por alguna vaca y hace clic en el botón “ Guardar ”; la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	
Observaciones: Ninguno.	

Tabla N° 4.9: Historia usuario. Registrar aborto.

Historia de Usuario	
Número: 8	Usuario: Productor pecuario.
Nombre historia: Emitir reporte Servicio Concepción.	
Prioridad en negocio: Alto.	Riesgo en desarrollo: Alto.
Puntos estimados: 1	Iteración asignada: 5
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (productor pecuario), una vez situada en la página principal hace clic en el menú “ reproducción ”, y la aplicación carga la página “ reproducción.jsp ”; el usuario hace clic en el menú “ Reporte de eventos reproductivos ”, y la aplicación muestra una caja de texto en donde ingresará el código del bovino y podrá visualizar el registro de servicio concepción y los índices reproductivos de la vaca.	
Observaciones: Ninguno.	

Tabla N° 4.10: Historia usuario. Emitir reporte servicio concepción.

Historia de Usuario	
Número: 9	Usuario: Técnico pecuario.
Nombre historia: Registrar ordeño.	
Prioridad en negocio: Alto.	Riesgo en desarrollo: Alto.
Puntos estimados: 1	Iteración asignada: 5
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (técnico pecuario), una vez situada en la página principal hace clic en el menú “ producción ”, y la aplicación carga la página “ producción.jsp ”; el usuario hace clic en el menú “ Registrar ordeño ”, y la aplicación muestra el formulario respecto al ordeño diario; el usuario llena los datos en el formulario y hace clic en el botón “ Guardar ”; la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	

Observaciones: Ninguno.

Tabla N° 4.11: Historia usuario. Registrar leche ordeñada.

Historia de Usuario	
Número: 10	Usuario: Biólogo.
Nombre historia: Registrar propiedades de la leche.	
Prioridad en negocio: Medio.	Riesgo en desarrollo: Bajo.
Puntos estimados: 1	Iteración asignada: 5
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (biólogo), una vez situada en la página principal hace clic en el menú “ producción ”, y la aplicación carga la página “ producción.jsp ”; el usuario hace clic en el menú “ Propiedades de la leche ”, y la aplicación muestra el formulario respecto a las propiedades de la leche; el usuario llena los datos en el formulario y hace clic en el botón “ Guardar ”; la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	
Observaciones: Ninguno.	

Tabla N° 4.12: Historia usuario. Registrar propiedades de la leche.

Historia de Usuario	
Número: 11	Usuario: Productor pecuario.
Nombre historia: Registrar secado de la vaca.	
Prioridad en negocio: Alto.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 5
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (productor pecuario), una vez situada en la página principal hace clic en el menú “ producción ”, y la aplicación carga la página “ producción.jsp ”; el usuario hace clic en el menú “ Asentar seca ”, y la aplicación muestra el formulario respecto al secado de la vaca en producción; el usuario llena los datos en el formulario y hace clic en el botón “ Guardar ”; la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	
Observaciones: Ninguno.	

Tabla N° 4.13: Historia usuario. Registrar secado de la vaca.

Historia de Usuario	
Número: 12	Usuario: Veterinario.
Nombre historia: Registrar estado nutricional de la vaca.	
Prioridad en negocio: Alto.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Nilo Juan Núñez Chávez.	

<p>Descripción: El usuario (veterinario), una vez situada en la página principal hace clic en el menú “producción”, y la aplicación carga la página “producción.jsp”; el usuario hace clic en el menú “Estado nutricional”, y la aplicación muestra el formulario respecto al estado nutricional de la vaca que está produciendo de leche; el usuario llena los datos en el formulario y hace clic en el botón “Guardar”; la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>
Observaciones: Ninguno.

Tabla N° 4.14: Historia usuario. Registrar estado nutricional.

Historia de Usuario	
Número: 13	Usuario: Veterinario.
Nombre historia: Registrar características morfológicas de la vaca.	
Prioridad en negocio: Medio.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (veterinario), una vez situada en la página principal hace clic en el menú “Reproductora”, y la aplicación carga la página “reproductoras.jsp”; el usuario hace clic en el menú “Características físicas”, y la aplicación muestra el formulario respecto a las características físicas de la vaca; el usuario llena los datos en el formulario los resultados de cada parámetro de valoración lineal obtenidos a partir de la evaluación morfológica de la vaca y hace clic en el botón “Guardar”; la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>	
Observaciones: Ninguno.	

Tabla N° 4.15: Historia usuario. Registrar características morfológicas de la vaca.

Historia de Usuario	
Número: 14	Usuario: Veterinario.
Nombre historia: Registrar características morfológicas del semental.	
Prioridad en negocio: Medio.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (veterinario), una vez situada en la página principal hace clic en el menú “Sementales”, y la aplicación carga la página “sementales.jsp”; el usuario hace clic en el menú “Características físicas”, y la aplicación muestra el formulario respecto a las características físicas del semental; el usuario llena los datos en el formulario los resultados de cada parámetro de valoración lineal obtenidos a partir de la evaluación morfológica del semental y hace clic en el botón “Guardar”; la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de</p>	

datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.
Observaciones: Ninguno.

Tabla N° 4.16: Historia usuario. Registrar características morfológicas del semental.

Historia de Usuario	
Número: 15	Usuario: Veterinario.
Nombre historia: Registrar características físicas del aparato reproductor del bovino macho.	
Prioridad en negocio: Alto.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 5
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (veterinario), una vez situada en la página principal hace clic en el menú “Sementales” , y la aplicación carga la página “sementales.jsp” ; el usuario hace clic en el menú “Aparato reproductor” , y la aplicación muestra el formulario respecto a las características físicas de aparato reproductor del bovino macho; el usuario llena los datos en el formulario y hace clic en el botón “Guardar” ; la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	
Observaciones: Ninguno.	

Tabla N° 4.17: Historia usuario. Registrar características físicas del aparato reproductor del bovino macho.

Historia de Usuario	
Número: 16	Usuario: Técnico pecuario.
Nombre historia: Registrar evaluación de libido.	
Prioridad en negocio: Medio.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 5
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (técnico pecuario), una vez situada en la página principal hace clic en el menú “Sementales” , y la aplicación carga la página “sementales.jsp” ; el usuario hace clic en el menú “Evaluación de libido” , y la aplicación muestra el formulario respecto a los parámetros de valoración lineal del libido en bovinos; el usuario llena los datos en el formulario y hace clic en el botón “Guardar” ; la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	
Observaciones: Ninguno.	

Tabla N° 4.18: Historia usuario. Registrar resultado de evaluación de libido.

Historia de Usuario	
Número: 17	Usuario: Biólogo.

Nombre historia: Registrar cualidades de semen.	
Prioridad en negocio: Alto.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 5
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (biólogo), una vez situada en la página principal hace clic en el menú “Sementales” , y la aplicación carga la página “sementales.jsp” ; el usuario hace clic en el menú “Cualidades del semen” , y la aplicación muestra el formulario respecto a las cualidades del semen; el usuario llena los datos en el formulario y hace clic en el botón “Guardar” ; la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	
Observaciones: Ninguno.	

Tabla N° 4.19: Historia usuario. Registrar cualidades de semen.

Historia de Usuario	
Número: 18	Usuario: Biólogo.
Nombre historia: Registrar colección de semen.	
Prioridad en negocio: Alto.	Riesgo en desarrollo: Alto.
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (biólogo), una vez situada en la página principal hace clic en el menú “Sementales” , y la aplicación carga la página “sementales.jsp” ; el usuario hace clic en el menú “Colección de semen” , y la aplicación muestra el formulario respecto a la colección de semen; el usuario llena los datos en el formulario y hace clic en el botón “Guardar” ; la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	
Observaciones: Ninguno.	

Tabla N° 4.20: Historia usuario. Registrar colección de semen.

Historia de Usuario	
Número: 19	Usuario: Veterinario.
Nombre historia: Registrar tratamiento sanitario del bovino.	
Prioridad en negocio: Alto.	Riesgo en desarrollo: Alto.
Puntos estimados: 1	Iteración asignada: 5
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (veterinario), una vez situada en la página principal hace clic en el menú “Sanidad” , y la aplicación carga la página “sanidad.jsp” ; el usuario hace clic en el menú “Inmunización” , y la aplicación muestra el formulario respecto a la inmunización de bovino; el usuario llena los datos en el formulario y hace clic en el botón “Guardar” ; la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de	

datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.
Observaciones: Ninguno.

Tabla N° 4.21: Historia usuario. Registrar tratamiento sanitario del bovino.

Historia de Usuario	
Número: 20	Usuario: Productor pecuario.
Nombre historia: Registrar crecimiento y desarrollo del bovino.	
Prioridad en negocio: Alto.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 5
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (productor pecuario), una vez situada en la página principal hace clic en el menú " Crecimiento y desarrollo ", y la aplicación carga la página " crecimientoDesarrollo.jsp "; la aplicación muestra el formulario respecto a los parámetros de crecimiento y desarrollo del bovino; el usuario llena los datos en el formulario y hace clic en el botón " Guardar "; la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	
Observaciones: Ninguno.	

Tabla N° 4.22: Historia usuario. Registrar crecimiento y desarrollo del bovino.

Historia de Usuario	
Número: 21	Usuario: Administrador.
Nombre historia: Registrar saca de bovinos.	
Prioridad en negocio: Alto.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 5
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (administrador), una vez situada en la página principal hace clic en el menú " Control de movimientos ", y la aplicación carga la página " controlMovimientos.jsp "; el usuario hace clic en el menú " Saca ", y la aplicación muestra el formulario respecto a la saca de bovino que son separados del hato para beneficio (consumo) o reproducción (re-cría) y hace clic en el botón " Guardar "; la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	
Observaciones: Ninguno.	

Tabla N° 4.23: Historia usuario. Registrar saca de bovinos.

Historia de Usuario	
Número: 22	Usuario: Administrador.
Nombre historia: Registrar mortalidad de bovinos.	

Prioridad en negocio: Alto.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 5
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (administrador), una vez situada en la página principal hace clic en el menú " Mortalidad "; la aplicación carga la página " Mortalidad.jsp " conteniendo el formulario respecto a la mortandad de bovinos; el usuario llena los datos en el formulario el resultado del diagnóstico sobre la causa de muerte súbita y hace clic en el botón " Guardar "; la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	
Observaciones: Ninguno.	

Tabla N° 4.24: Historia usuario. Registrar mortalidad de bovinos.

Historia de Usuario	
Número: 23	Usuario: Administrador.
Nombre historia: Crear cuenta de usuario.	
Prioridad en negocio: Alto.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (administrador), ingresa la dirección de la aplicación web en uno de los navegadores web, la aplicación carga la página " index.jsp "; el usuario hace clic en el enlace administrar cuenta y la aplicación carga una venta " cuenta de usuario " y muestra el formulario respecto a los datos personales y cuenta de usuario del productor pecuario; el usuario administrador llena los datos en el formulario y hace clic en el botón " Guardar "; la aplicación realiza la validación de los campos, en caso de ser correcto, lanza una ventana emergente solicitando el usuario y clave del administrador de sistemas y almacena en la base de datos el nuevo registro, caso contrario muestra un mensaje de error.	
Observaciones: Ninguno.	

Tabla N° 4.25: Historia usuario. Crear cuenta de usuario.

Historia de Usuario	
Número: 24	Usuario: Productor pecuario.
Nombre historia: Iniciar sesión.	
Prioridad en negocio: Alto.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (productor pecuario), ingresa la dirección de la aplicación web en uno de los navegadores web, el sistema carga la página " index.jsp ". El usuario ingresa el usuario y contraseña luego hace clic en el botón " Iniciar sesión ", en caso de que los datos ingresados sean los correctos, entonces, el sistema permite el acceso a la página " principal.jsp ", caso contrario el	

sistema cargará la página "index.jsp" con el siguiente mensaje de error "El usuario o la contraseña son incorrectos".

Observaciones: Ninguno.

Tabla N° 4.26: Historia usuario. Iniciar sesión.

PLAN DE VERSIÓN

Nº	HISTORIA DE USUARIO	PRIORIDAD	RIESGO	ESFUERZO (DÍAS)	ITERACIÓN
1	Registrar alta de bovino.	Alto	Alto	4	1
2	Registrar calor o celo.	Alto	Medio	2	2
3	Registrar servicio de monta o empadre.	Alto	Alto	3	3
4	Registrar diagnóstico de preñez.	Alto	Alto	2	3
5	Registrar datos del bovino.	Alto	Alto	4	1
6	Registrar parto.	Alto	Alto	2	4
7	Registrar aborto.	Medio	Medio	3	4
8	Emitir reporte Servicio Concepción.	Alto	Alto	5	5
9	Registrar producción de leche.	Alto	Alto	3	5
10	Registrar propiedades de la leche ordeñada.	Medio	Bajo	3	5
11	Registrar secado de la vaca.	Alto	Medio	3	5
12	Registrar estado nutricional de la vaca.	Alto	Medio	3	3
13	Registrar características morfológicas de la vaca.	Medio	Medio	3	3
14	Registrar características morfológicas del semental.	Medio	Medio	3	3
15	Registrar características físicas del aparato reproductor del semental.	Alto	Medio	3	5
16	Registrar evaluación de libido.	Medio	Medio	3	5
17	Registrar cualidades de semen.	Alto	Medio	3	5
18	Registrar colección de semen	Alto	Alto	4	2
19	Registrar tratamiento sanitario del bovino.	Alto	Alto	3	5
20	Registrar crecimiento y	Alto	Medio	3	5

	desarrollo del bovino.				
21	Registrar saca de bovinos.	Alto	Medio	3	5
22	Registrar mortalidad de bovinos.	Alto	Medio	3	5
23	Administrar cuentas de usuario.	Alto	Medio	5	1
24	Iniciar sesión	Alto	Medio	2	2

Tabla N° 4.27: Plan de versión.

4.1.3. FASE DE ITERACIÓN

La fase de iteración presentado en el capítulo II, sección 2.2.3, y la técnica referenciada en la tabla 3.4, permite obtener los entregables; arquitectura técnica, tareas de ingeniería, plan de iteración, casos de prueba de aceptación, GUI, tarjetas CRC, base de datos física, código fuente para clases entidad, pruebas unitarias, código fuente para tarea de ingeniería, reporte de pruebas unitarias, reporte de pruebas de integración y de aceptación.

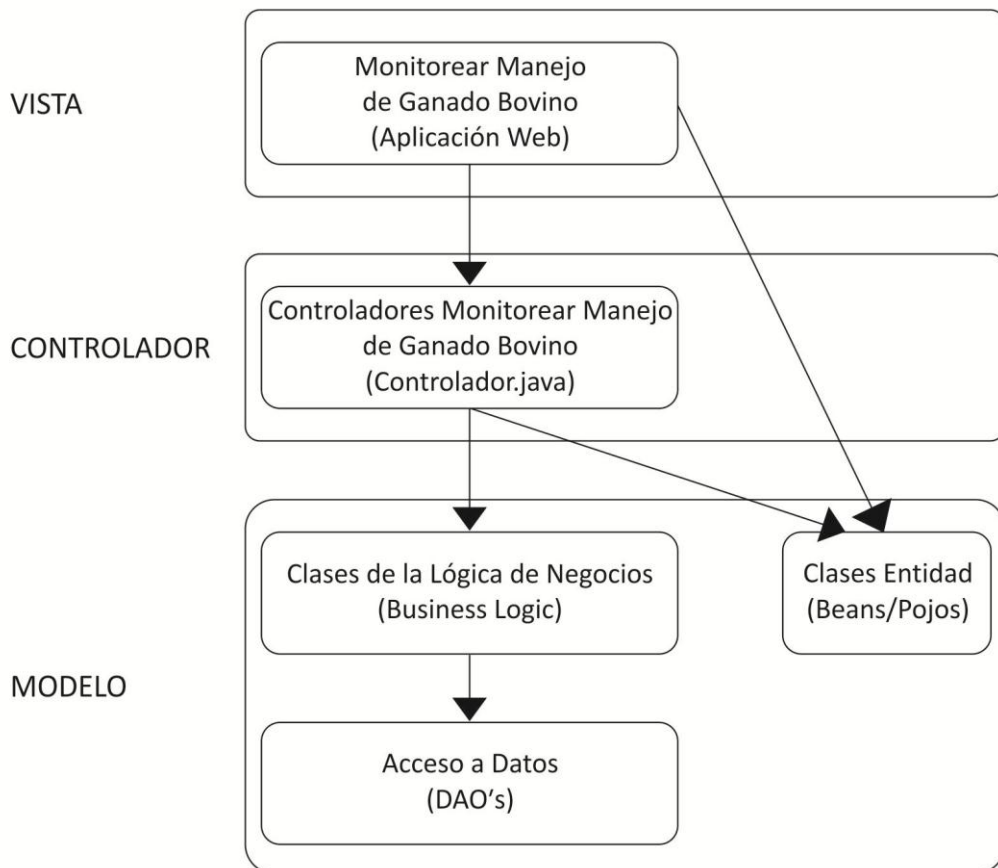


Figura N° 4.2: Arquitectura técnica final. Diagrama de componente

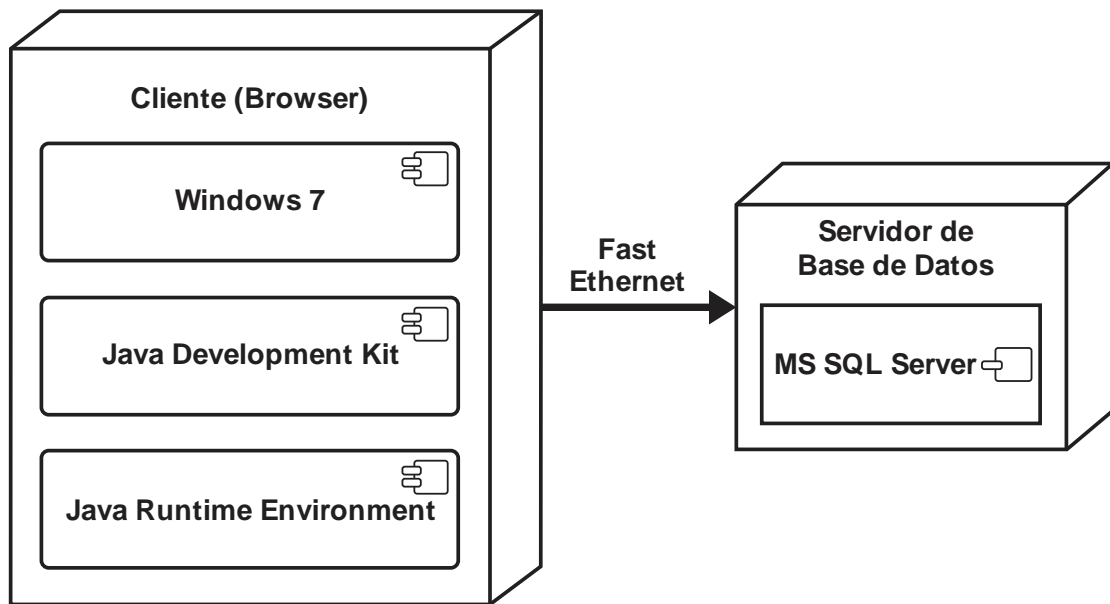


Figura N° 4.3: Arquitectura técnica final. Diagrama de despliegue.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Número historia de usuario: 1
Nombre tarea: Registrar alta de bovino.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 01/07/2016	Fecha fin: 01/07/2016
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (administrador) una vez situada en la página principal hace clic en el menú " Control de movimientos ", y la aplicación carga la página " controlMovimientos.jsp " mostrando un formulario con los campos (número de registro, procedencia, certificado de procedencia, fecha de alta, condición, código del bovino, nombre, fecha de nacimiento, raza, sexo y fotografía); el usuario al hacer clic en el icono " ayuda " la aplicación muestra una caja emergente (Prompt box), si el usuario ingresa un determinado año, entonces la aplicación sugiere el "ultimo" código para ese año puesto que los códigos tienen un orden secuencial por año; el usuario hace clic en el botón " Guardar " y la aplicación realiza la validación de los campos, en caso de ser correcto almacena en la base de datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	

Tabla N° 4.28: Tarea de ingeniería. Registrar alta de bovino.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 2	Número historia de usuario: 1
Nombre tarea: Mantener alta de bovino.	

Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 02/07/2016	Fecha fin: 02/07/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>El usuario (administrador) una vez situada en la página principal hace clic en el menú “Control de movimientos”, y la aplicación carga la página “controlMovimientos.jsp”; el usuario hace clic en el menú “Mantener alta” y la aplicación muestra un caja de texto; el usuario mientras ingresa el código del bovino en la caja de texto el sistema ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación carga los datos de la alta en los respectivos cajas de entrada; el usuario podrá editar los datos que cree conveniente haciendo clic en sus respectivos iconos “Editar”; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los datos, en caso de ser correcto actualiza los datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>	

Tabla N° 4.29: Tarea de ingeniería. Mantener alta de bovino.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 3	Número historia de usuario: 1
Nombre tarea: Emitir reporte de alta de bovinos.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 04/07/2016	Fecha fin: 05/07/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>El usuario (administrador) una vez situada en la página principal hace clic en el menú “Control de movimientos” y la aplicación carga la página “controlMovimientos.jsp”; el usuario hace clic en el menú “Listar alta” y la aplicación muestra una lista de opciones para visualizar los registro sobre alta de bovinos (Listar todas, Filtrar por código de bovino, Filtrar por rango de fechas); si el usuario hace clic en la opción listar todas, entonces la aplicación listará todos los registros de alta; si el usuario hace clic en la opción filtrar por código, entonces la aplicación mostrara una caja de texto. El usuario mientras ingresa el código del bovino en la caja de texto la aplicación ira sugiriendo una lista de códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación muestra el registro de alta correspondiente al código seleccionado; si el usuario hace clic en la opción filtrar por rango de fechas, entonces la aplicación mostrará dos cajas de fecha; el usuario ingresa la fecha de inicio y la fecha final y la aplicación muestra los registros de alta correspondiente al rango de fecha; el usuario hace clic en el botón “Imprimir” y la aplicación imprime el reporte.</p>	

Tabla N° 4.30: Tarea de ingeniería. Emitir reporte de alta de bovinos.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Número historia de usuario: 2

Nombre tarea: Registrar calor o celo.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 06/07/2016	Fecha fin: 06/07/2016
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (técnico pecuario) una vez situada en la página principal hace clic en el menú “Reproducción” y la aplicación carga la página “reproducción.jsp” mostrando un formulario con los campos (código de la vaca, fecha y hora) ; el usuario mientras ingresa el código de la vaca la aplicación ira sugiriendo una lista códigos; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	

Tabla N° 4.31: Tarea de ingeniería. Registrar calor o celo.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 2	Número historia de usuario: 2
Nombre tarea: Mantener calor o celo.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 07/07/2016	Fecha fin: 07/07/2016
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (técnico pecuario) una vez situada en la página principal hace clic en el menú “reproducción” , y la aplicación carga la página “reproducción.jsp” ; el usuario hace clic en el menú “Mantener eventos reproductivos” y la aplicación muestra un menú de navegación por pestañas; El usuario mientras ingresa el código de la vaca en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación lista los celos correspondiente al código seleccionado; el usuario modifica los datos (fecha, hora) haciendo clic en cada icono “Editar” ; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	

Tabla N° 4.32: Tarea de ingeniería. Mantener registro de calor o celo.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Número historia de usuario: 3
Nombre tarea: Registrar servicio de monta.	

Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 08/07/2016	Fecha fin: 08/07/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (veterinario) una vez situada en la página principal hace clic en el menú “reproducción”, y la aplicación carga la página “reproducción.jsp”; el usuario hace clic en la pestaña “Servicio” y la aplicación muestra el formulario correspondiente; El usuario mientras ingresa el código de la vaca en la caja de texto la aplicación ira sugiriendo una lista códigos; en caso de que el método de servicio sea monta natural el usuario llena los datos (código de la vaca, código del semental, fecha, hora), mientras que si el método de servicio sea inseminación artificial el usuario llena los datos (código de la vaca, código de la pajilla, dni del inseminador, fecha y hora); el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>	

Tabla N° 4.33: Tarea de ingeniería. Registrar servicio de monta.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 2	Número historia de usuario: 3
Nombre tarea: Mantener servicio de monta.	
Tipo de tarea : Desarrollo	Puntos estimados: 5
Fecha inicio: 09/07/2016	Fecha fin: 11/07/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (veterinario) una vez situada en la página principal hace clic en el menú “reproducción” y la aplicación carga la página “reproducción.jsp”; el usuario hace clic en el menú “Mantener eventos reproductivos” y la aplicación muestra un menú de navegación por pestañas; el usuario hace clic en la pestaña “Servicio” y la aplicación muestra una caja de texto; el usuario mientras ingresa el código de la vaca en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación lista los servicios correspondiente al código seleccionado; el usuario modifica los datos que cree conveniente haciendo clic en cada icono “Editar”; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>	

Tabla N° 4.34: Tarea de ingeniería. Mantener servicio de monta.

TAREA DE INGENIERÍA

Número tarea de ingeniería: 1	Número historia de usuario: 4
Nombre tarea: Registrar diagnóstico de preñez.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 12/07/2016	Fecha fin: 12/07/2016
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (veterinario) una vez situada en la página principal hace clic en el menú “ reproducción ”, y la aplicación carga la página “ reproducción.jsp ”; el usuario hace clic en la pestaña “ Diagnostico de preñez ”, y la aplicación muestra el formulario respectivo con los campos (código de la vaca, fecha y resultado); el usuario mientras ingresa el código de la vaca en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación lista la fecha de los servicios que aún no fueron diagnosticadas; el usuario hace clic en el botón “ Guardar ” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	

Tabla N° 4.35: Tarea de ingeniería. Registrar diagnóstico de preñez.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Número historia de usuario: 5
Nombre tarea: Registrar datos de bovino.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 14/07/2016	Fecha fin: 14/07/2016
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (administrador) una vez situada en la página principal hace clic en el menú “ reproducción ”, y la aplicación carga la página “ reproducción.jsp ”; el usuario hace clic en la pestaña “ Parto ”, y la aplicación muestra el formulario respectivo con los campos (código de la vaca, código del semental, código del ternero, nombre, fecha nacimiento, raza, sexo y fotografía) el usuario mientras ingresa el código de la vaca en la caja de texto la aplicación ira sugiriendo una lista de códigos; el usuario hace clic en el icono de “ sugerencia ” y la aplicación lista los sementales que sirvieron a la vaca; el usuario hace clic en el botón “ Guardar ” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	

Tabla N° 4.36: Tarea de ingeniería. Registrar datos de bovino.

TAREA DE INGENIERÍA

Número tarea de ingeniería: 2	Número historia de usuario: 5
Nombre tarea: Mantener datos del bovino.	
Tipo de tarea : Desarrollo	Puntos estimados: 5
Fecha inicio: 15/07/2016	Fecha fin: 15/07/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (administrador) una vez situada en la página principal hace clic en el menú “reproducción” y la aplicación carga la página “reproducción.jsp”; el usuario hace clic en el menú “Mantener eventos reproductivos” y la aplicación muestra un menú de navegación por pestañas; el usuario hace clic en la pestaña “Parto” y la aplicación muestra una caja de texto; el usuario mientras ingresa el código de la vaca en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación lista los partos correspondiente al código seleccionado; el usuario hace clic en uno de los iconos “Editar” de la columna “CIB Temero” y la aplicación muestra una ventana emergente “Mantener Datos del Bovino” con datos del bovino seleccionado; el usuario modifica los datos que cree conveniente haciendo clic en cada icono “Editar” de la ventana emergente; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>	

Tabla N° 4.37: Tarea de ingeniería. Mantener datos de bovino.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 3	Número historia de usuario: 5
Nombre tarea: Emitir reporte de datos de la vaca.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 16/07/2016	Fecha fin: 16/07/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: una vez situada en la página principal hace clic en el menú “Reproductoras” y la aplicación carga la página “reproductoras.jsp” mostrando todas las vacas con los datos (nombre, cib, edad, estado reproductivo, estado productivo, estado sanitario); el usuario hace clic en la imagen de la vaca y la aplicación muestra ventana emergente conteniendo los datos completos de la vaca y las tablas características morfológicas, crecimiento y desarrollo, sanidad, producción láctea y una gráfica lineal crecimiento y desarrollo - estado nutricional (Peso y Cc, año); el usuario hace clic en el botón “Imprimir” y la aplicación imprime el reporte.</p>	

Tabla N° 4.38: Tarea de ingeniería. Emitir reporte de datos de la vaca.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 4	Número historia de usuario: 5
Nombre tarea: Emitir reporte de datos del toro.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 18/07/2016	Fecha fin: 18/07/2016
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: una vez situada en la página principal hace clic en el menú “ Sementales ” y la aplicación carga la página “ sementales.jsp ” mostrando todos los sementales con los datos relevantes (nombre, cib, edad, servicios, servicios infértiles, preñadas, abortos, partos problemáticos, partos); el usuario hace clic en la imagen del semental y la aplicación muestra ventana emergente conteniendo los datos completos del semental y las tablas características morfológicas, características del aparato reproductor, inventario de semen, características del semen y tratamiento sanitario ; el usuario hace clic en el botón “ Imprimir ” y la aplicación imprime el reporte.	

Tabla N° 4.39: Tarea de ingeniería. Emitir reporte de datos del toro.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Número historia de usuario: 6
Nombre tarea: Registrar parto.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 19/07/2016	Fecha fin: 19/07/2016
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (productor pecuario) una vez situada en la página principal hace clic en el menú “ reproducción ” y la aplicación carga la página “ reproducción.jsp ”; el usuario hace clic en la pestaña “ Parto ” y la aplicación muestra el formulario respectivo con los campos (código de la madre, código del ternero, dificultad, hipocalcemia, placenta retenida, edema ubre, observación); el usuario mientras ingresa el código de la vaca en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación lista los códigos de los terneros que aún no fueron vinculados con algún registro de parto; el usuario hace clic en el botón “ Guardar ” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	

Tabla N° 4.40: Tarea de ingeniería. Registrar parto.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 2	Número historia de usuario: 6
Nombre tarea: Mantener parto.	
Tipo de tarea : Desarrollo	Puntos estimados: 5
Fecha inicio: 20/07/2016	Fecha fin: 20/07/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (productor pecuario) una vez situada en la página principal hace clic en el menú “reproducción” y la aplicación carga la página “reproducción.jsp”; el usuario hace clic en el menú “Mantener eventos reproductivos” y la aplicación muestra un menú de navegación por pestañas; el usuario hace clic en la pestaña “Parto” y la aplicación muestra una caja de texto; el usuario mientras ingresa el código de la vaca en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación lista los partos correspondiente al código seleccionado; el usuario modifica los datos que cree conveniente haciendo clic en cada icono “Editar”; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>	

Tabla N° 4.41: Tarea de ingeniería. Mantener parto.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Número historia de usuario: 7
Nombre tarea: Registrar aborto.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 21/07/2016	Fecha fin: 21/07/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (veterinario) una vez situada en la página principal hace clic en el menú “reproducción” y la aplicación carga la página “reproducción.jsp”; el usuario hace clic en la pestaña “Aborto” y la aplicación muestra el formulario respectivo con los campos (código de la vaca, fecha de servicio, fecha de aborto, causa, agente, observación); el usuario mientras ingresa el código de la vaca en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación lista la fecha de los servicios que resultaron positivos y que aún no fueron vinculados con algún registro de parto o aborto; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena en la base de datos el nuevo registro y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>	

Tabla N° 4.42: Tarea de ingeniería. Registrar aborto.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 2	Número historia de usuario: 7
Nombre tarea: Mantener aborto.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 22/07/2016	Fecha fin: 22/07/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (veterinario) una vez situada en la página principal hace clic en el menú “reproducción” y la aplicación carga la página “reproducción.jsp”; el usuario hace clic en el menú “Mantener eventos reproductivos” y la aplicación muestra un menú de navegación por pestañas; el usuario hace clic en la pestaña “Aborto” y la aplicación muestra una caja de texto; el usuario mientras ingresa el código de la vaca en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación lista los abortos sufridos correspondiente al código seleccionado; el usuario modifica los datos que cree conveniente haciendo clic en cada icono “Editar”; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>	

Tabla N° 4.43: Tarea de ingeniería. Mantener aborto.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 3	Número historia de usuario: 7
Nombre tarea: Emitir reporte de aborto.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 23/07/2016	Fecha fin: 23/07/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (productor pecuario) una vez situada en la página principal hace clic en el menú “Reproducción” y la aplicación carga la página “reproduccion.jsp”; el usuario hace clic en el submenú “Aborto” del menú “Reporte de eventos reproductivos” y la aplicación muestra una lista de opciones para filtrar los registro sobre abortos sufridos (Listar todas, Filtrar por año, Filtrar por mes, Filtrar por vaca); si el usuario hace clic en la opción listar todas, entonces la aplicación listará todos los registros de aborto; si el usuario hace clic en la opción filtrar por año, entonces la aplicación muestra una caja de tipo numérico; el usuario ingresa el año y la aplicación muestra los registros de aborto correspondiente al año; si el usuario hace clic en la</p>	

opción filtrar por mes, entonces la aplicación muestra una caja de tipo numérico y una lista de meses. El usuario ingresa el año y selecciona el mes y la aplicación muestra los registros de aborto correspondiente al mes en un determinado año; si el usuario hace clic en la opción filtrar por vaca, entonces la aplicación muestra una caja de texto. El usuario mientras ingresa el código de la vaca en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación muestra el registro de la aborto correspondiente al código seleccionado; el usuario hace clic en el botón **“Imprimir”** y la aplicación imprime el reporte.

Tabla N° 4.44: Tarea de ingeniería. Emitir reporte de aborto.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Número historia de usuario: 8
Nombre tarea: Emitir reporte Servicio Concepción.	
Tipo de tarea : Desarrollo	Puntos estimados: 5
Fecha inicio: 25/07/2016	Fecha fin: 29/07/2016
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (productor pecuario) una vez situada en la página principal hace clic en el menú “Reproducción” y la aplicación carga la página “reproduccion.jsp” ; el usuario hace clic en el submenú “Servicio Concepción” del menú “Reporte de eventos reproductivos” y la aplicación muestra una caja de texto; el usuario mientras ingresa el código de la vaca en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación muestra una tabla con los campos (número del ciclo reproductivo, fecha de parto, sexo del ternero, nombre del ternero, fecha de los celos sin servicio, fechas de los celos servidos, días al primer celo, días al primer servicio, días de vacía, número de servicios por concepción, intervalo entre parto) ; el usuario hace clic en el botón “Imprimir” y la aplicación imprime el reporte.	

Tabla N° 4.45: Tarea de ingeniería. Emitir reporte Servicio Concepción.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Número historia de usuario: 9
Nombre tarea: Registrar producción de leche.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 30/07/2016	Fecha fin: 30/07/2016
Programador responsable: Nilo Juan Núñez Chávez.	

Descripción:

El usuario (técnico pecuario) una vez situada en la página principal hace clic en el menú **“Producción”** y la aplicación carga la página **“produccion.jsp”** y muestra un formulario con los campos **(código de la vaca, fecha de ordeño, horario y cantidad)**; el usuario mientras ingresa el código de la vaca en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en el botón **“Guardar”** y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.

Tabla N° 4.46: Tarea de ingeniería. Registrar producción de leche.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 2	Número historia de usuario: 9
Nombre tarea: Mantener producción de leche.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 01/08/2016	Fecha fin: 01/08/2016
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (técnico pecuario) una vez situada en la página principal hace clic en el menú “Producción” y la aplicación carga la página “produccion.jsp” ; el usuario hace clic en el submenú “Mantener producción de leche” del menú “Producción de leche” y la aplicación muestra una caja de texto para el ingreso del código de la vaca y una lista de opciones para filtrar los registro sobre producción de leche (filtrar por campaña, filtrar por fecha) ; si el usuario hace clic en la opción filtrar por campaña, entonces la aplicación listará todos los registros de producción láctea de una determinada campaña; si el usuario hace clic en la opción filtrar por fecha, entonces el sistema listara el registro de producción láctea para una determinada fecha; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	

Tabla N° 4.47: Tarea de ingeniería. Mantener producción de leche.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 3	Número historia de usuario: 9
Nombre tarea: Emitir reporte de producción de leche.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 02/08/2016	Fecha fin: 02/08/2016
Programador responsable: Nilo Juan Núñez Chávez.	

Descripción:

El usuario (técnico pecuario) una vez situada en la página principal hace clic en el menú **“Producción”** y la aplicación carga la página **“produccion.jsp”**; el usuario hace clic en el submenú **“Reporte de producción de leche”** del menú **“Producción Láctea”** y la aplicación muestra una lista de filtros (**filtrar por producción individual, filtrar por producción grupal**); si el usuario selecciona la primera opción, entonces la aplicación muestra una caja de texto; el usuario mientras ingresa el código de la vaca en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación muestra una gráfica de línea (**litros, años**); si el usuario selecciona la segunda opción, entonces la aplicación muestra una gráfica de líneas (**promedio de litros producidos, años**); el usuario hace clic en el botón **“Imprimir”** y la aplicación imprime el reporte.

Tabla N° 4.47: Tarea de ingeniería. Emitir reporte de producción de leche.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Número historia de usuario: 10
Nombre tarea: Registrar propiedades de la leche.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 03/08/2016	Fecha fin: 03/08/2016
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (biólogo) una vez situada en la página principal hace clic en el menú “Producción” y la aplicación carga la página “produccion.jsp” ; el usuario hace clic en el submenú “Registrar propiedades de la leche” del menú “Propiedades de la leche” y la aplicación muestra el formulario correspondiente con los campos (código de la vaca, número de campaña, proteína, grasa, unidades formadoras de colonias, recuento de células somáticas y lactosa); el usuario mientras ingresa el código de la vaca en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	

Tabla N° 4.48: Tarea de ingeniería. Registrar propiedades de la leche.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 2	Número historia de usuario: 10
Nombre tarea: Mantener las propiedades de la leche.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 04/08/2016	Fecha fin: 04/08/2016

Programador responsable: Nilo Juan Núñez Chávez.
<p>Descripción: El usuario (biólogo) una vez situada en la página principal hace clic en el menú “Producción” y la aplicación carga la página “produccion.jsp”; el usuario hace clic en el submenú “Mantener propiedades de la leche” del menú “Propiedades de la leche” y la aplicación muestra una caja de texto; el usuario mientras ingresa el código de la vaca en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación muestra los registros de propiedades de la leche; el usuario modifica los datos que cree conveniente haciendo clic en cada icono “Editar”; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>

Tabla N° 4.50: Tarea de ingeniería. Mantener las propiedades de la leche.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 3	Número historia de usuario: 10
Nombre tarea: Emitir reporte de propiedades de la leche.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 05/08/2016	Fecha fin: 05/08/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (biólogo) una vez situada en la página principal hace clic en el menú “Producción” y la aplicación carga la página “produccion.jsp”; el usuario hace clic en el submenú “Reporte de las propiedades de la leche” del menú “Propiedades de la leche” y la aplicación muestra una caja de texto; el usuario mientras ingresa el código de la vaca en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación muestra una gráfica de líneas (Campaña, %); el usuario hace clic en el botón “Imprimir” y la aplicación imprime el reporte.</p>	

Tabla N° 4.49: Tarea de ingeniería. Emitir reporte de propiedades de la leche.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Número historia de usuario: 11
Nombre tarea: Registrar secado de la vaca.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 06/08/2016	Fecha fin: 06/08/2016
Programador responsable: Nilo Juan Núñez Chávez.	

Descripción:

El usuario (productor pecuario) una vez situada en la página principal hace clic en el menú **“Producción”** y la aplicación carga la página **“produccion.jsp”**; el usuario hace clic en el submenú **“Registrar seca”** del menú **“Seca”** y la aplicación muestra el formulario correspondiente; el usuario llena los datos (**código de la vaca, número de campaña y fecha**); el usuario mientras ingresa el código de la vaca en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en el botón **“Guardar”** y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.

Tabla N° 4.50: Tarea de ingeniería. Registrar secado de la vaca.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 2	Número historia de usuario: 11
Nombre tarea: Mantener secado de la vaca.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 08/08/2016	Fecha fin: 08/08/2016
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (productor pecuario) una vez situada en la página principal hace clic en el menú “Producción” y la aplicación carga la página “produccion.jsp” ; el usuario hace clic en el submenú “Mantener seca” del menú “Seca” y la aplicación muestra una caja de texto; el usuario mientras ingresa el código de la vaca en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación muestra los registros de seca respectivo; el usuario modifica los datos que cree conveniente haciendo clic en cada icono “Editar” ; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	

Tabla N° 4.51: Tarea de ingeniería. Mantener secado de la vaca.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 3	Número historia de usuario: 11
Nombre tarea: Emitir reporte sobre secado de la vaca.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 09/08/2016	Fecha fin: 09/08/2016
Programador responsable: Nilo Juan Núñez Chávez.	

Descripción:
 El usuario (técnico pecuario) una vez situada en la página principal hace clic en el menú **“Producción”** y la aplicación carga la página **“produccion.jsp”**; el usuario hace clic en el submenú **“Reporte de seca”** del menú **“Seca”** y la aplicación muestra una lista de opciones (**listar vacas que secarán esta semana, listar vacas que secarán este mes, listar seca por vaca, listar vacas que no secaron a tiempo**); el usuario al seleccionar cualquiera de las opciones la aplicación lista el registro correspondiente; el usuario hace clic en el botón **“Imprimir”** y la aplicación imprime el reporte.

Tabla N° 4.52: Tarea de ingeniería. Emitir reporte sobre secado de la vaca.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Número historia de usuario: 12
Nombre tarea: Registrar estado nutricional de la vaca.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 10/08/2016	Fecha fin: 10/08/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (veterinario) una vez situada en la página principal hace clic en el cuadro “Reproductoras” y la aplicación carga la página “reproductoras.jsp”; el usuario hace clic en el submenú “Registrar estado nutricional” del menú “Estado nutricional” y la aplicación muestra el formulario correspondiente con los campos (código de la vaca, fecha, estado, condición corporal, peso); el usuario mientras ingresa el código de la vaca en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>	

Tabla N° 4.53: Tarea de ingeniería. Registrar estado nutricional de la vaca.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 2	Número historia de usuario: 12
Nombre tarea: Mantener el estado nutricional de la vaca.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 11/08/2016	Fecha fin: 11/08/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (veterinario) una vez situada en la página principal hace clic en el menú “Reproductoras” y la aplicación carga la página “reproductoras.jsp”; el usuario hace clic en el submenú “Mantener estado nutricional” del menú</p>	

“Estado nutricional” y la aplicación muestra una caja de texto; el usuario mientras ingresa el código de la vaca en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación muestra el registro estado nutricional de la vaca correspondiente al código seleccionado; el usuario edita los datos que cree conveniente haciendo clic en sus respectivos iconos “**Editar**”; el usuario hace clic en el botón “**Guardar**” y la aplicación realiza la validación de los datos, en caso de ser correcto actualiza los datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.

Tabla N° 4.54: Tarea de ingeniería. Mantener el estado nutricional de la vaca.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Número historia de usuario: 13
Nombre tarea: Registrar características morfológicas de la vaca.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 12/08/2016	Fecha fin: 12/08/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (veterinario) una vez situada en la página principal hace clic en el menú “Reproductora” y la aplicación carga la página “reproductoras.jsp”; el usuario hace clic en el submenú “Registrar morfología” del menú “Morfología” y la aplicación muestra el formulario correspondiente con los campos (código de la vaca, fecha, estatura, profundidad corporal, angulosidad, ancho del pecho, ancho de la grupa, ángulo de la grupa, vista lateral de las patas traseras, vista posterior de las patas traseras, ángulo podal, ligamento delantero, altura de la ubre, ancho de la ubre, profundidad de la ubre); el usuario mientras ingresa el código de la vaca en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>	

Tabla N° 4.55: Tarea de ingeniería. Registrar características morfológicas de la vaca.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 2	Número historia de usuario: 13
Nombre tarea: Mantener características morfológicas de la vaca.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 13/08/2016	Fecha fin: 13/08/2016
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción:	

El usuario (veterinario) una vez situada en la página principal hace clic en el menú **“Reproductora”** y la aplicación carga la página **“reproductoras.jsp”**; el usuario hace clic en el submenú **“Mantener morfología”** del menú **“Morfología”** y la aplicación muestra una caja de texto; el usuario mientras ingresa el código de la vaca en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación muestra los respectivos registros de morfología; el usuario modifica los datos que cree conveniente haciendo clic en cada icono **“Editar”**; el usuario hace clic en el botón **“Guardar”** y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.

Tabla N° 4.56: Tarea de ingeniería. Mantener características morfológicas de la vaca.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Número historia de usuario: 14
Nombre tarea: Registrar características morfológicas del semental.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 15/08/2016	Fecha fin: 15/08/2016
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (veterinario) una vez situada en la página principal hace clic en el menú “Sementales” y la aplicación carga la página “sementales.jsp” ; el usuario hace clic en el submenú “Registrar Morfología” del menú “Morfología del Semental” y la aplicación muestra el formulario correspondiente con los campos (código de semental, cabeza, ojo, pezuña, hocico, cuello, pecho, hombros, línea dorsal, musculo, miembro posterior vista detrás, miembro posterior vista lado, miembro anterior vista lateral, miembro anterior vista frente); el usuario mientras ingresa el código del semental en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	

Tabla N° 4.57: Tarea de ingeniería. Registrar características morfológicas del semental.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 2	Número historia de usuario: 14
Nombre tarea: Mantener las características morfológicas del semental.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 16/08/2016	Fecha fin: 16/08/2016

Programador responsable: Nilo Juan Núñez Chávez.
<p>Descripción: El usuario (veterinario) una vez situada en la página principal hace clic en el menú “Sementales” y la aplicación carga la página “sementales.jsp”; el usuario hace clic en el submenú “Mantener Morfología” del menú “Morfología del Semental” y la aplicación muestra una caja de texto; el usuario mientras ingresa el código del semental en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación lista las características morfológicas del semental correspondiente al código seleccionado; el usuario modifica los datos que cree conveniente haciendo clic en cada icono “Editar”; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>

Tabla N° 4.58: Tarea de ingeniería. Mantener las características morfológicas del semental.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Número historia de usuario: 15
Nombre tarea: Registrar las características físicas del aparato reproductor del bovino macho.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 17/08/2016	Fecha fin: 17/08/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (veterinario) una vez situada en la página principal hace clic en el menú “Sementales” y la aplicación carga la página “sementales.jsp”; el usuario hace clic en el submenú “Registrar Características Físicas” del menú “Aparato Reproductor” y la aplicación muestra el formulario correspondiente con los campos (código de semental, fecha, circunferencia escrotal, forma del testículo, posición del escroto, consistencia del escroto); el usuario mientras ingresa el código del semental en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>	

Tabla N° 4.59: Tarea de ingeniería. Registrar las características físicas del aparato reproductor del bovino macho.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 2	Número historia de usuario: 15
Nombre tarea: Mantener las características físicas del aparato reproductor del bovino macho.	

Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 18/08/2016	Fecha fin: 18/08/2016
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (veterinario) una vez situada en la página principal hace clic en el menú “Sementales” y la aplicación carga la página “sementales.jsp” ; el usuario hace clic en el submenú “Mantener Características Físicas” del menú “Aparato Reproductor” y la aplicación muestra una caja de texto; el usuario mientras ingresa el código del semental en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación lista las características físicas del aparato reproductor del semental correspondiente al código seleccionado; el usuario modifica los datos que cree conveniente haciendo clic en cada icono “Editar” ; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	

Tabla N° 4.60: Tarea de ingeniería. Mantener las características físicas del aparato reproductor del bovino macho.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Número historia de usuario: 16
Nombre tarea: Registrar evaluación de libido.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 19/08/2016	Fecha fin: 19/08/2016
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (técnico pecuario) una vez situada en la página principal hace clic en el menú “Sementales” y la aplicación carga la página “sementales.jsp” ; el usuario hace clic en el submenú “Registrar Evaluación del Libido” del menú “Libido” y la aplicación muestra el formulario correspondiente con los siguientes campos (código de semental, fecha, olfateo genital, lameteo genital, lameteo social, apoyo de cabeza, aparejo, flehmen, intento de monta, monta incompleta y servicio); el usuario mientras ingresa el código del semental en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	

Tabla N° 4.61: Tarea de ingeniería. Registrar los resultados de evaluación de libido.

TAREA DE INGENIERÍA

Número tarea de ingeniería: 2	Número historia de usuario: 16
Nombre tarea: Mantener evaluación de libido.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 20/08/2016	Fecha fin: 20/08/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (veterinario) una vez situada en la página principal hace clic en el menú “Sementales” y la aplicación carga la página “sementales.jsp”; el usuario hace clic en el submenú “Mantener Evaluación del Libido” del menú “Libido” y la aplicación muestra una caja de texto; el usuario mientras ingresa el código del semental en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación lista los registros de evaluación de libido del semental correspondiente al código seleccionado; el usuario modifica los datos que cree conveniente haciendo clic en cada icono “Editar”; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>	

Tabla N° 4.62: Tarea de ingeniería. Mantener los resultados de evaluación de libido.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Número historia de usuario: 17
Nombre tarea: Registrar las cualidades del semen.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 22/08/2016	Fecha fin: 22/08/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (biólogo) una vez situada en la página principal hace clic en el menú “Sementales” y la aplicación carga la página “sementales.jsp”; el usuario hace clic en el submenú “Registrar Cualidades del Semen” del menú “Semen” y la aplicación muestra el formulario correspondiente con los siguientes campos (código de semental, fecha, volumen, pH, motilidad masal, motilidad individual, concentración espermática, anormalidades, viabilidad); el usuario mientras ingresa el código del semental en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>	

Tabla N° 4.63: Tarea de ingeniería. Registrar las cualidades del semen.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 2	Número historia de usuario: 17
Nombre tarea: Mantener las cualidades del semen.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 23/08/2016	Fecha fin: 23/08/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (veterinario) una vez situada en la página principal hace clic en el menú “Sementales” y la aplicación carga la página “sementales.jsp”; el usuario hace clic en el submenú “Registrar Cualidades del Semen” del menú “Semen” y la aplicación muestra una caja de texto; el usuario mientras ingresa el código del semental en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación lista las cualidades de semen del semental correspondiente al código seleccionado; el usuario modifica los datos que cree conveniente haciendo clic en cada icono “Editar”; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>	

Tabla N° 4.64: Tarea de ingeniería. Mantener las cualidades del semen.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Número historia de usuario: 18
Nombre tarea: Registrar colección de semen.	
Tipo de tarea : Desarrollo	Puntos estimados: 5
Fecha inicio: 24/08/2016	Fecha fin: 24/08/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (veterinario) una vez situada en la página principal hace clic en el menú “Sementales” y la aplicación carga la página “sementales.jsp”; el usuario hace clic en el submenú “Inventario del Semen” del menú “Semen” y la aplicación muestra un menú de navegación por pestañas y el formulario correspondiente con los siguientes campos (código de bovino, fecha, glosa, cantidad de pajillas, tanque, número de canastilla, procedencia, precio); el usuario mientras ingresa el código del semental en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>	

Tabla N° 4.65: Tarea de ingeniería. Registrar colección de semen.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 2	Número historia de usuario: 18
Nombre tarea: Registrar salida de pajilla.	
Tipo de tarea : Desarrollo	Puntos estimados: 5
Fecha inicio: 25/08/2016	Fecha fin: 25/08/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (veterinario) una vez situada en la página principal hace clic en el menú “Sementales” y la aplicación carga la página “sementales.jsp”; el usuario hace clic en el submenú “Inventario del semen” del menú “Semen” y la aplicación muestra un menú de navegación por pestañas; el usuario hace clic en la pestaña “Registrar salida” y la aplicación muestra el formulario correspondiente con los siguientes campos (código del semental, fecha, glosa, cantidad de pajillas, tanque, número de canastilla, destino, precio); el usuario mientras ingresa el código del semental en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>	

Tabla N° 4.66: Tarea de ingeniería. Registrar salida de pajilla.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 3	Número historia de usuario: 18
Nombre tarea: Emitir reporte de inventario de pajillas.	
Tipo de tarea : Desarrollo	Puntos estimados: 5
Fecha inicio: 26/08/2016	Fecha fin: 26/08/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (veterinario) una vez situada en la página principal hace clic en el menú “Sementales” y la aplicación carga la página “sementales.jsp”; el usuario hace clic en el submenú “Inventario del semen” del menú “Semen” y la aplicación muestra un menú de navegación por pestañas; el usuario hace clic en la pestaña “Ver inventario” y la aplicación muestra una caja de texto; El usuario mientras ingresa el código del bovino en la caja de texto la aplicación ira sugiriendo una lista de códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación muestra el inventario de pajillas correspondiente al código seleccionado; el usuario hace clic en el botón “Imprimir” y la aplicación imprime el reporte.</p>	

Tabla N° 4.67: Tarea de ingeniería. Emitir reporte de inventario de pajillas.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 4	Número historia de usuario: 18
Nombre tarea: Mantener colección de semen.	
Tipo de tarea : Desarrollo	Puntos estimados: 5
Fecha inicio: 27/08/2016	Fecha fin: 27/08/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (veterinario) una vez situada en la página principal hace clic en el menú “Sementales” y la aplicación carga la página “sementales.jsp”; el usuario hace clic en el submenú “Inventario del semen” del menú “Semen” y la aplicación muestra un menú de navegación por pestañas ; el usuario hace clic en la pestaña “Mantener inventario” y la aplicación muestra una caja de texto; el usuario mientras ingresa el código del semental en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación lista los registros de colección de semen del semental correspondiente al código seleccionado; el usuario modifica los datos que cree conveniente haciendo clic en cada icono “Editar”; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>	

Tabla N° 4.68: Tarea de ingeniería. Mantener colección de semen.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Número historia de usuario: 19
Nombre tarea: Registrar tratamiento sanitario del bovino.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 29/08/2016	Fecha fin: 29/08/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (veterinario) una vez situada en la página principal hace clic en el menú “Sanidad Animal” y la aplicación carga la página “sanidad.jsp” conteniendo un formulario con los siguientes campos (código del bovino, fecha de inicio, fecha de finalización, tipo de registro, nombre del tipo de registro, tratamiento utilizado, días, comentario y personal responsable); el usuario mientras ingresa el código del semental en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>	

Tabla N° 4.69: Tarea de ingeniería. Registrar tratamiento sanitario del bovino.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 2	Número historia de usuario: 19
Nombre tarea: Mantener tratamiento sanitario del bovino.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 30/08/2016	Fecha fin: 30/08/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (veterinario) una vez situada en la página principal hace clic en el menú “Sanidad Animal” y la aplicación carga la página “sanidad.jsp”; el usuario hace clic en el menú “Mantener Tratamiento Sanitario” y la aplicación muestra una caja de texto; el usuario mientras ingresa el código del semental en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación muestra los registros de tratamiento sanitario correspondiente al código seleccionado; el usuario modifica los datos que cree conveniente haciendo clic en cada icono “Editar”; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>	

Tabla N° 4.70: Tarea de ingeniería. Mantener inmunización a bovinos.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Número historia de usuario: 20
Nombre tarea: Registrar crecimiento y desarrollo del bovino.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 31/08/2016	Fecha fin: 31/08/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (productor pecuario) una vez situada en la página principal hace clic en el menú “Crecimiento y Desarrollo” y la aplicación carga la página “crecimiento-desarrollo.jsp” conteniendo un formulario con los siguientes campos (código del bovino, fecha, peso, perímetro torácico, altura de cruz, condición corporal y observación); el usuario mientras ingresa el código del bovino en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>	

Tabla N° 4.71: Tarea de ingeniería. Registrar crecimiento y desarrollo del bovino.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 2	Número historia de usuario: 20
Nombre tarea: Mantener crecimiento y desarrollo del bovino.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 01/09/2016	Fecha fin: 01/09/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (productor pecuario) El usuario (productor pecuario) una vez situada en la página principal hace clic en el menú “Crecimiento y Desarrollo” y la aplicación carga la página “crecimiento-desarrollo.jsp”; el usuario hace clic en el menú “Mantener Crecimiento y Desarrollo” y la aplicación muestra una caja de texto; el usuario mientras ingresa el código del semental en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación muestra los registros de crecimiento y desarrollo correspondiente al código seleccionado; el usuario modifica los datos que cree conveniente haciendo clic en cada icono “Editar”; el usuario hace clic en el botón “Guardar” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>	

Tabla N° 4.72: Tarea de ingeniería. Mantener crecimiento y desarrollo del bovino.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 3	Número historia de usuario: 20
Nombre tarea: Emitir reporte sobre crecimiento y desarrollo del bovino.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 02/09/2016	Fecha fin: 02/09/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (productor pecuario) El usuario (productor pecuario) una vez situada en la página principal hace clic en el menú “Crecimiento y Desarrollo” y la aplicación carga la página “crecimiento-desarrollo.jsp”; el usuario hace clic en el menú “Reporte de Crecimiento y Desarrollo” y la aplicación muestra una caja de texto; el usuario mientras ingresa el código del semental en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación muestra una gráfica lineal correspondiente al código seleccionado; el usuario hace clic en el botón “Imprimir” y la aplicación imprime el reporte.</p>	

Tabla N° 4.73: Tarea de ingeniería. Emitir reporte sobre crecimiento y desarrollo del bovino.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Número historia de usuario: 21
Nombre tarea: Registrar saca de bovinos.	
Tipo de tarea : Desarrollo	Puntos estimados: 5
Fecha inicio: 03/09/2016	Fecha fin: 03/09/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (administrador) una vez situada en la página principal hace clic en el menú "Control de movimientos", y la aplicación carga la página "control-movimientos.jsp"; el usuario hace clic en el submenú "Registrar Saca" del menú "Saca", y la aplicación muestra el formulario correspondiente con los siguientes campos (Número de registro, motivo, destino, fecha, código de vaca, precio, responsable) ; el usuario mientras ingresa el código del bovino en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en el botón "Guardar" y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>	

Tabla N° 4.74: Tarea de ingeniería. Registrar saca de bovinos.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 2	Número historia de usuario: 21
Nombre tarea: Mantener saca de bovinos.	
Tipo de tarea : Desarrollo	Puntos estimados: 5
Fecha inicio: 05/09/2016	Fecha fin: 05/09/2016
Programador responsable: Nilo Juan Núñez Chávez.	
<p>Descripción: El usuario (administrador) una vez situada en la página principal hace clic en el menú "Control de movimientos", y la aplicación carga la página "control-movimientos.jsp"; el usuario hace clic en el submenú "Mantener Saca" del menú "Saca" y la aplicación muestra una caja de texto; el usuario mientras ingresa el código del semental en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación muestra el registro de saca correspondiente al código seleccionado; el usuario modifica los datos que cree conveniente haciendo clic en cada icono "Editar"; el usuario hace clic en el botón "Guardar" y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.</p>	

Tabla N° 4.75: Tarea de ingeniería. Mantener saca de bovinos.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 3	Número historia de usuario: 21
Nombre tarea: Emitir reporte sobre saca de bovinos.	
Tipo de tarea : Desarrollo	Puntos estimados: 5
Fecha inicio: 06/09/2016	Fecha fin: 06/09/2016
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (administrador) una vez situada en la página principal hace clic en el menú " Control de movimientos ", y la aplicación carga la página " control-movimientos.jsp "; el usuario hace clic en el submenú " Mantener Saca " del menú " Saca ", y la aplicación muestra una lista de opciones; el usuario selecciona una de las opciones y la aplicación muestra el respectivo gráfico de barras; el usuario hace clic en el botón " Imprimir " y la aplicación imprime el reporte.	

Tabla N° 4.76: Tarea de ingeniería. Emitir reporte sobre saca de bovinos.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Número historia de usuario: 22
Nombre tarea: Registrar mortalidad de bovinos.	
Tipo de tarea : Desarrollo	Puntos estimados: 5
Fecha inicio: 07/09/2016	Fecha fin: 07/09/2016
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (administrador) una vez situada en la página principal hace clic en el menú " Mortalidad "; la aplicación carga la página " mortalidad.jsp " con el respectivo formulario con los campos (código del bovino, número de registro, causa, fecha, observación, responsable); el usuario mientras ingresa el código del bovino en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en el botón " Guardar " y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	

Tabla N° 4.77: Tarea de ingeniería. Registrar mortalidad de bovinos.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 2	Número historia de usuario: 22
Nombre tarea: Mantener mortalidad de bovinos.	
Tipo de tarea : Desarrollo	Puntos estimados: 5

Fecha inicio: 08/09/2016	Fecha fin: 08/09/2016
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (administrador) una vez situada en la página principal hace clic en el menú "Mortalidad" , y la aplicación carga la página "mortalidad.jsp" ; el usuario hace clic en el menú "Mantener Muertes" y la aplicación muestra una caja de texto; el usuario mientras ingresa el código del semental en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación muestra el registro de mortandad correspondiente al código seleccionado; el usuario modifica los datos que cree conveniente haciendo clic en cada icono "Editar" ; el usuario hace clic en el botón "Guardar" y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	

Tabla N° 4.78: Tarea de ingeniería. Mantener mortalidad de bovinos.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 3	Número historia de usuario: 22
Nombre tarea: Emitir reporte sobre mortalidad de bovinos.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 09/09/2016	Fecha fin: 09/09/2016
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (administrador) una vez situada en la página principal hace clic en el menú "Mortalidad" , y la aplicación carga la página "mortalidad.jsp" ; el usuario hace clic en el menú "Mantener Muertes" y la aplicación muestra una lista de opciones; el usuario selecciona una de las opciones y la aplicación muestra el respectivo gráfico de barras; el usuario hace clic en el botón "Imprimir" y la aplicación imprime el reporte.	

Tabla N° 4.79: Tarea de ingeniería. Emitir reporte sobre mortalidad de bovinos.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Número historia de usuario: 23
Nombre tarea: Procesos de cuenta de usuario.	
Tipo de tarea : Desarrollo	Puntos estimados: 5
Fecha inicio: 10/09/2016	Fecha fin: 10/09/2016
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (administrador) ingresa la dirección de la aplicación web en algún	

navegador web y la aplicación carga la página “**index.jsp**”; el usuario hace clic en el enlace administrar cuenta y la aplicación solicita al usuario que se autentique con la cuenta de administrador; el usuario inicia sesión y el sistema carga una venta “**cuenta de usuario**” mostrando tres grupos de campo “**Cambiar estado de usuario**”, “**Cambiar rol de usuario**” y “**Reseteo clave**”; el usuario hace clic en el botón “**Guardar**” o “**Reseteo**” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.

Tabla N° 4.80: Tarea de ingeniería. Procesos de cuenta de usuario.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 2	Número historia de usuario: 23
Nombre tarea: Crear cuenta de usuario.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 12/09/2016	Fecha fin: 12/09/2016
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (administrador) ingresa la dirección de la aplicación web en algún navegador web y la aplicación carga la página “ index.jsp ”; el usuario hace clic en el enlace administrar cuenta y la aplicación solicita al usuario que se autentique con la cuenta de administrador; el usuario inicia sesión y el sistema carga una venta “ cuenta de usuario ” y muestra un menú de navegación por pestañas; el usuario hace clic en la pestaña “ crear cuenta ” y la aplicación muestra un formulario respecto a los datos personales y cuenta de usuario del productor pecuario con los campos (dni, nombre, apellido paterno, apellido materno, teléfono, e-mail y cargo) y (usuario, clave, fecha, estado y rol) respectivamente; el usuario hace clic en el botón “ Guardar ” y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.	

Tabla N° 4.81: Tarea de ingeniería. Crear cuenta de usuario.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 3	Número historia de usuario: 23
Nombre tarea: Mantener cuenta de usuario.	
Tipo de tarea : Desarrollo	Puntos estimados: 5
Fecha inicio: 13/09/2016	Fecha fin: 13/09/2016
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (administrador) ingresa la dirección de la aplicación web en algún	

navegador web y la aplicación carga la página **“index.jsp”**; el usuario hace clic en el enlace administrar cuenta y la aplicación solicita al usuario que se autentique con la cuenta de administrador; el usuario inicia sesión y el sistema carga una venta **“cuenta de usuario”** y muestra un menú de navegación por pestañas; el usuario hace clic en la pestaña **“Mantener cuenta”** y la aplicación muestra una caja de texto; el usuario mientras ingresa el código del semental en la caja de texto la aplicación ira sugiriendo una lista códigos; el usuario hace clic en uno de los códigos sugeridos y la aplicación carga en los respectivos cajas de texto los datos del usuario correspondiente al código seleccionado; el usuario modifica los datos que cree conveniente haciendo clic en cada icono **“Editar”**; el usuario hace clic en el botón **“Guardar”** y la aplicación realiza la validación de los campos, en caso de ser correcto, almacena los datos modificados en la base de datos y muestra un mensaje de éxito, caso contrario muestra un mensaje de error.

Tabla N° 4.82: Tarea de ingeniería. Mantener cuenta de usuario.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 4	Número historia de usuario: 23
Nombre tarea: Emitir reporte de cuentas de usuario.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 14/09/2016	Fecha fin: 14/09/2016
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción: El usuario (administrador) ingresa la dirección de la aplicación web en algún navegador web y la aplicación carga la página “index.jsp” ; el usuario hace clic en el enlace administrar cuenta y la aplicación solicita al usuario que se autentique con la cuenta de administrador; el usuario inicia sesión y el sistema carga una venta “cuenta de usuario” y muestra un menú de navegación por pestañas; el usuario hace clic en la pestaña “Listar cuenta” y la aplicación lista todas las cuentas de usuario del más reciente al más antiguo.	

Tabla N° 4.83: Tarea de ingeniería. Emitir reporte de cuentas de usuario.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Número historia de usuario: 24
Nombre tarea: Iniciar Sesión.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 15/09/2016	Fecha fin: 15/09/2016
Programador responsable: Nilo Juan Núñez Chávez.	
Descripción:	

El usuario (productor pecuario) ingresa la dirección de la aplicación web en algún navegador web y la aplicación carga la página “**index.jsp**”; el usuario ingresa el **usuario y contraseña** luego hace clic en el botón “**Iniciar sesión**” y la aplicación verifica los datos ingresados, en caso de ser correctos la aplicación permite el acceso a la página “**principal.jsp**” de lo contrario, la aplicación muestra el mensaje de error “El usuario o la contraseña son incorrectos”.

Tabla N° 4.84: Tarea de ingeniería. Iniciar Sesión.

PLAN DE ITERACIÓN (Primera)

N°	HISTORIA DE USUARIO	TAREA DE INGENIERÍA
1	Registrar alta de bovinos	Registrar alta de bovino
		Mantener alta de bovino
		Emitir reporte de alta de bovinos
5	Registrar datos de bovino	Registrar datos de bovino
		Mantener datos del bovino
		Emitir reporte de datos de la vaca
		Emitir reporte de datos del toro
23	crear cuenta de usuario	Procesos de cuenta de usuario
		Crear cuenta de usuario
		Mantener cuenta de usuario
		Emitir reporte de cuentas de usuario

Tabla N° 4.85: Plan de iteración (Primera).

PLAN DE ITERACIÓN (Segunda)

N°	HISTORIA DE USUARIO	TAREA DE INGENIERÍA
2	Registrar calor o celo	Registrar calor o celo
		Mantener calor o celo
12	Registrar estado nutricional de la vaca	Registrar estado nutricional de la vaca
		Mantener el estado nutricional de la vaca
13	Registrar características morfológicas de la vaca	Registrar características morfológicas de la vaca

		Mantener características morfológicas de la vaca
14	Registrar características morfológicas del semental	Registrar características morfológicas del semental
		Mantener las características morfológicas del semental
17	Registrar cualidades de semen	Registrar las cualidades del semen
		Mantener las cualidades del semen
18	Registrar colección de semen	Registrar colección de semen
		Registrar salida de pajilla
		Emitir reporte de inventario de pajillas
		Mantener colección de semen
19	Registrar tratamiento sanitario del bovino	Registrar tratamiento sanitario del bovino
		Mantener tratamiento sanitario del bovino
22	Registrar mortalidad de bovinos	Registrar mortalidad de bovinos
		Mantener mortalidad de bovinos
		Emitir reporte de mortalidad de bovinos
24	Iniciar sesión	Iniciar Sesión

Tabla N° 4.86: Plan de iteración (Segunda).

PLAN DE ITERACIÓN (Tercera)

N°	HISTORIA DE USUARIO	TAREA DE INGENIERÍA
3	Registrar servicio de monta o empadre	Registrar servicio de monta

		Mantener servicio de monta
--	--	----------------------------

Tabla N° 4.87: Plan de iteración (Tercera).

PLAN DE ITERACIÓN (Cuarta)

N°	HISTORIA DE USUARIO	TAREA DE INGENIERÍA
4	Registrar diagnóstico de preñez	Registrar diagnóstico de preñez

Tabla N° 4.88: Plan de iteración (Cuarta).

PLAN DE ITERACIÓN (Quinta)

N°	HISTORIA DE USUARIO	TAREA DE INGENIERÍA
6	Registrar parto	Registrar parto
		Mantener parto
7	Registrar aborto	Registrar aborto
		Mantener aborto
		Emitir reporte de aborto

Tabla N° 4.89: Plan de iteración (Quinta).

PLAN DE ITERACIÓN (Sexta)

N°	HISTORIA DE USUARIO	TAREA DE INGENIERÍA
8	Emitir reporte servicio concepción	Emitir reporte Servicio Concepción
9	Registrar producción de leche	Registrar producción de leche
		Mantener producción de leche
		Emitir reporte de producción de leche
10	Registrar propiedades de la leche	Registrar propiedades de la leche

		Mantener las propiedades de la leche
		Emitir reporte de propiedades de la leche
11	Registrar secado de la vaca	Registrar secado de la vaca
		Mantener secado de la vaca
		Emitir reporte sobre secado de la vaca
15	Registrar características físicas del aparato reproductor del semental	Registrar las características físicas del aparato reproductor del semental
		Mantener las características físicas del aparato reproductor del semental
16	Registrar evaluación de libido	Registrar evaluación de libido
		Mantener evaluación de libido
20	Registrar crecimiento y desarrollo del bovino	Registrar crecimiento y desarrollo del bovino
		Mantener crecimiento y desarrollo del bovino
		Emitir reporte sobre crecimiento y desarrollo del bovino
21	Registrar saca de bovinos	Registrar saca de bovinos
		Mantener saca de bovinos
		Emitir reporte sobre saca de bovinos

Tabla N° 4.90: Plan de iteración (Sexta).

HISTORIA DE USUARIO	TAREA DE INGENIERÍA	FECHA INICIO	FECHA FIN	PROGRAMADOR
Registrar alta de bovinos	Registrar alta de bovino	01/07/2016	01/07/2016	Nilo Juan Núñez Chávez
	Mantener alta de bovino	02/07/2016	02/07/2016	Nilo Juan Núñez Chávez
	Emitir reporte de alta de bovinos	04/07/2016	05/07/2016	Nilo Juan Núñez Chávez

Registrar calor o celo	Registrar calor o celo	06/07/2016	06/07/2016	Nilo Juan Núñez Chávez
	Mantener calor o celo	07/07/2016	07/07/2016	Nilo Juan Núñez Chávez
Registrar servicio de monta o empadre	Registrar servicio de monta	08/07/2016	08/07/2016	Nilo Juan Núñez Chávez
	Mantener servicio de monta	09/07/2016	11/07/2016	Nilo Juan Núñez Chávez
Registrar diagnóstico de preñez	Registrar diagnóstico de preñez	12/07/2016	12/07/2016	Nilo Juan Núñez Chávez
Registrar datos de bovino	Registrar datos de bovino	14/07/2016	14/07/2016	Nilo Juan Núñez Chávez
	Mantener datos del bovino	15/07/2016	15/07/2016	Nilo Juan Núñez Chávez
	Emitir reporte de datos de la vaca	16/07/2016	16/07/2016	Nilo Juan Núñez Chávez
	Emitir reporte de datos del toro	18/07/2016	18/07/2016	Nilo Juan Núñez Chávez
Registrar parto	Registrar parto	19/07/2016	19/07/2016	Nilo Juan Núñez Chávez
	Mantener parto	20/07/2016	20/07/2016	Nilo Juan Núñez Chávez
Registrar aborto	Registrar aborto	21/07/2016	21/07/2016	Nilo Juan Núñez Chávez
	Mantener aborto	22/07/2016	22/07/2016	Nilo Juan Núñez Chávez
	Emitir reporte de aborto	23/07/2016	23/07/2016	Nilo Juan Núñez Chávez
Emitir reporte servicio concepción	Emitir reporte Servicio Concepción	25/07/2016	29/07/2016	Nilo Juan Núñez Chávez
Registrar producción de leche	Registrar producción de leche	30/07/2016	30/07/2016	Nilo Juan Núñez Chávez
	Mantener producción de leche	01/08/2016	01/08/2016	Nilo Juan Núñez Chávez
	Emitir reporte de producción de leche	02/08/2016	02/08/2016	Nilo Juan Núñez Chávez
Registrar propiedades de la leche	Registrar propiedades de la leche	03/08/2016	03/08/2016	Nilo Juan Núñez Chávez
	Mantener las propiedades de la leche	04/08/2016	04/08/2016	Nilo Juan Núñez Chávez
	Emitir reporte de propiedades de la leche	05/08/2016	05/08/2016	Nilo Juan Núñez Chávez

Registrar secado de la vaca	Registrar secado de la vaca	06/08/2016	06/08/2016	Nilo Juan Núñez Chávez
	Mantener secado de la vaca	08/08/2016	08/08/2016	Nilo Juan Núñez Chávez
	Emitir reporte sobre secado de la vaca	09/08/2016	09/08/2016	Nilo Juan Núñez Chávez
Registrar estado nutricional de la vaca	Registrar estado nutricional de la vaca	10/08/2016	10/08/2016	Nilo Juan Núñez Chávez
	Mantener el estado nutricional de la vaca	11/08/2016	11/08/2016	Nilo Juan Núñez Chávez
Registrar características morfológicas de la vaca	Registrar características morfológicas de la vaca	12/08/2016	12/08/2016	Nilo Juan Núñez Chávez
	Mantener características morfológicas de la vaca	13/08/2016	13/08/2016	Nilo Juan Núñez Chávez
Registrar característica morfológica del semental	Registrar características morfológicas del semental	15/08/2016	15/08/2016	Nilo Juan Núñez Chávez
	Mantener las características morfológicas del semental	16/08/2016	16/08/2016	Nilo Juan Núñez Chávez
Registrar característica física del aparato reproductor del semental	Registrar las características físicas del aparato reproductor del semental	17/08/2016	17/08/2016	Nilo Juan Núñez Chávez
	Mantener las características físicas del aparato reproductor del semental	18/08/2016	18/08/2016	Nilo Juan Núñez Chávez
Registrar evaluación de libido	Registrar evaluación de libido	19/08/2016	19/08/2016	Nilo Juan Núñez Chávez
	Mantener evaluación de libido	20/08/2016	20/08/2016	Nilo Juan Núñez Chávez
Registrar cualidades de semen	Registrar las cualidades del semen	22/08/2016	22/08/2016	Nilo Juan Núñez Chávez
	Mantener las cualidades del semen	23/08/2016	23/08/2016	Nilo Juan Núñez Chávez

Registrar colección de semen	Registrar colección de semen	24/08/2016	24/08/2016	Nilo Juan Núñez Chávez
	Registrar salida de pajilla	25/08/2016	25/08/2016	Nilo Juan Núñez Chávez
	Emitir reporte de inventario de pajillas	26/08/2016	26/08/2016	Nilo Juan Núñez Chávez
	Mantener colección de semen	27/08/2016	27/08/2016	Nilo Juan Núñez Chávez
Registrar tratamiento sanitario del bovino	Registrar tratamiento sanitario del bovino	29/08/2016	29/08/2016	Nilo Juan Núñez Chávez
	Mantener tratamiento sanitario del bovino	30/08/2016	30/08/2016	Nilo Juan Núñez Chávez
Registrar crecimiento y desarrollo del bovino	Registrar crecimiento y desarrollo del bovino	31/08/2016	31/08/2016	Nilo Juan Núñez Chávez
	Mantener crecimiento y desarrollo del bovino	01/09/2016	01/09/2016	Nilo Juan Núñez Chávez
	Emitir reporte sobre crecimiento y desarrollo del bovino	02/09/2016	02/09/2016	Nilo Juan Núñez Chávez
Registrar saca de bovinos	Registrar saca de bovinos	03/09/2016	03/09/2016	Nilo Juan Núñez Chávez
	Mantener saca de bovinos	05/09/2016	05/09/2016	Nilo Juan Núñez Chávez
	Emitir reporte sobre saca de bovinos	06/09/2016	06/09/2016	Nilo Juan Núñez Chávez
Registrar mortalidad de bovinos	Registrar mortalidad de bovinos	07/09/2016	07/09/2016	Nilo Juan Núñez Chávez
	Mantener mortalidad de bovinos	08/09/2016	08/09/2016	Nilo Juan Núñez Chávez
	Emitir reporte de mortalidad de bovinos	09/09/2016	09/09/2016	Nilo Juan Núñez Chávez
Crear cuenta de usuario	Procesos de cuenta de usuario	10/09/2016	10/09/2016	Nilo Juan Núñez Chávez
	Crear cuenta de usuario	12/09/2016	12/09/2016	Nilo Juan Núñez Chávez
	Mantener cuenta de usuario	13/09/2016	13/09/2016	Nilo Juan Núñez Chávez

	Emitir reporte de cuentas de usuario	14/09/2016	14/09/2016	Nilo Juan Núñez Chávez
Iniciar sesión	Iniciar Sesión	15/09/2016	15/09/2016	Nilo Juan Núñez Chávez

Tabla N° 4.91: Plan de iteraciones clasificadas en fechas de desarrollo.

N° H.U.	REQUISITO	N° C.P.	CASO DE PRUEBA (C.P.)
1	Registrar alta de bovino	1	Validar campos del formulario registrar alta
		2	Sugerir código para el bovino
		3	Registrar satisfactoriamente el alta de bovino
5	Registrar datos del bovino	1	Validar campos del formulario registrar bovino
		2	Filtrar búsqueda de código de la vaca de acuerdo a su edad y sexo
		3	Sugerir una lista de códigos de sementales que sirvieron y fecundaron a la vaca
		4	Sugerir código para el bovino
		5	Registrar satisfactoriamente datos del bovino
2	Registrar calor o celo	1	Validar campos del formulario registrar calor o celo
		2	Filtrar búsqueda de código de la vaca de acuerdo a su edad y sexo
		3	Registrar satisfactoriamente el calor o celo
3	Registrar servicio de monta	1	Validar campos del formulario servicio o monta
		2	Filtrar búsqueda de código de la vaca de acuerdo a su edad y sexo
		3	Verificar que la fecha de servicio corresponde a la fecha de algún registro de celo
		4	Registrar satisfactoriamente el servicio o monta
4	Registrar diagnóstico de preñez	1	Validar campos del formulario diagnóstico de preñez
		2	Filtrar búsqueda de código de la vaca de acuerdo a su edad y sexo

		3	Para una determinada vaca listar las fecha de los servicios que se le practico y que aún no fueron diagnosticados
		4	Registrar satisfactoriamente el diagnóstico de preñez
6	Registrar parto	1	Validar campos del formulario parto
		2	Filtrar búsqueda de código de la vaca de acuerdo a su edad y sexo
		3	Listar códigos de terneros que aún no fueron relacionados con algún registro de parto o aborto
		4	Registrar satisfactoriamente el parto
7	Registrar aborto	1	Validar campos del formulario aborto
		2	Filtrar búsqueda de código de la vaca de acuerdo a su edad y sexo
		3	Listar fechas de servicios con resultado de diagnóstico positivo que aún no fueron relacionados con algún registro de aborto o parto
		4	Registrar satisfactoriamente el aborto
8	Emitir reporte Servicio Concepción	1	Filtrar búsqueda de código de la vaca de acuerdo su edad y sexo
		2	Cálculo de número de días al primer celo, número de días al primer servicio, número de días de vacía, numero de servicios por concepción e intervalo entre partos
24	Iniciar Sesión	1	Carga la página principal.
		2	No permitir el acceso a la página principal.

Tabla N° 4.92: Casos de prueba de aceptación.

INTERFAZ DE USUARIO Y CODIFICACIÓN

Las interfaces se han diseñado considerando estándares de usabilidad, según los objetivos de la tesis no considera su evaluación, los factores para estándares utilizados son; botones, títulos de páginas y menús, fuentes y colores, tamaños de fuentes, etc., que permiten tener interfaces amigables y, fáciles de usar.

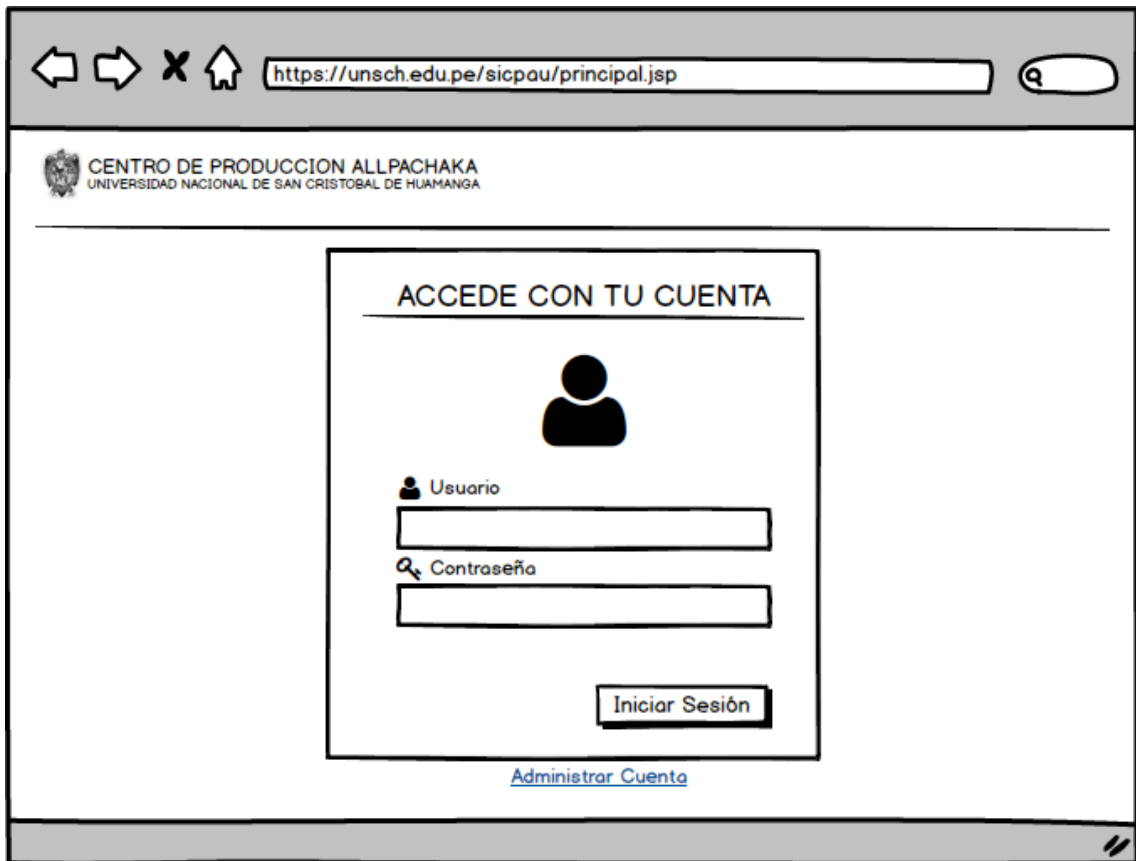


Figura N° 4.5: Interfaz Iniciar Sesión.

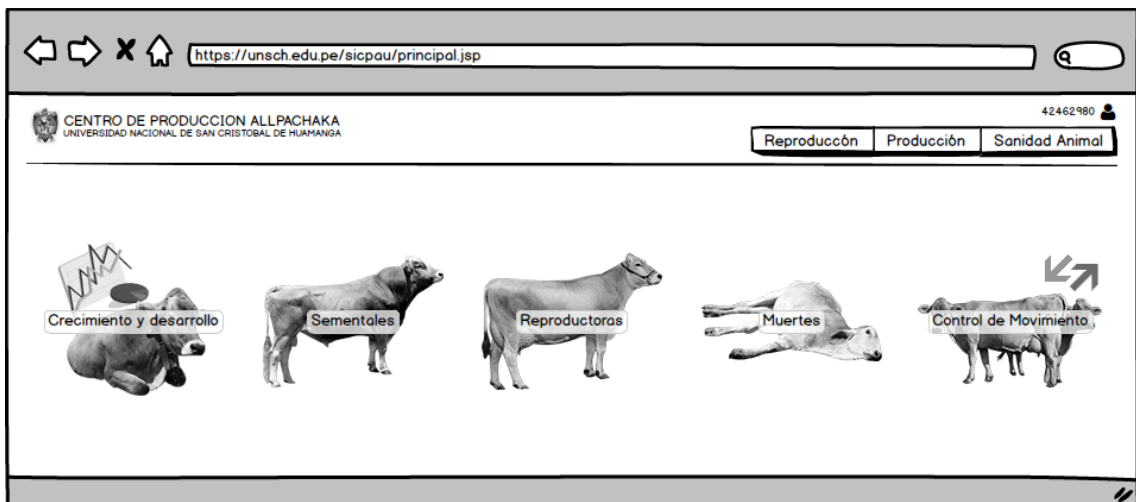


Figura N° 4.6: Interfaz Principal.

https://unsch.edu.pe/sicpau/controlMovimientos.jsp

CENTRO DE PRODUCCION ALLPACHAKA
 UNIVERSIDAD NACIONAL DE SAN CRISTOBAL DE HUAMANGA

42462980

Reproducción Producción Sanidad Animal

CONTROL DE MOVIMIENTO - ALTA DE BOVINO

Alta

Registrar Alta

Mantener Alta

Listar Alta

Saca

Registrar Alta

Numero de registro:

Procedencia:

Certificado de procedencia:

Fecha de alta:

Condición: ▼

Código:

Nombre:

Fecha de nacimiento:

Raza: ▼


Sexo: ▼

Fotografía:

Precio:

Figura N° 4.7: Interfaz Registrar Alta.

← → × 🏠 🔍

 **CENTRO DE PRODUCCION ALLPACHAKA**
UNIVERSIDAD NACIONAL DE SAN CRISTOBAL DE HUAMANGA 42462980 👤

CONTROL DE MOVIMIENTO - ALTA DE BOVINO

Alta

Registrar Alta


Mantener Alta


Listar Alta


Saca



Mantener Alta


Código del bovino:



Numero de registro: 


Procedencia: 



Certificado de procedencia: 


Fecha de alta:  


Condición: 

Código:  

Nombre: 

Fecha de nacimiento:  

Raza: 

Sexo: 


Fotografía: 

Figura N° 4.8: Interfaz Mantener Alta.

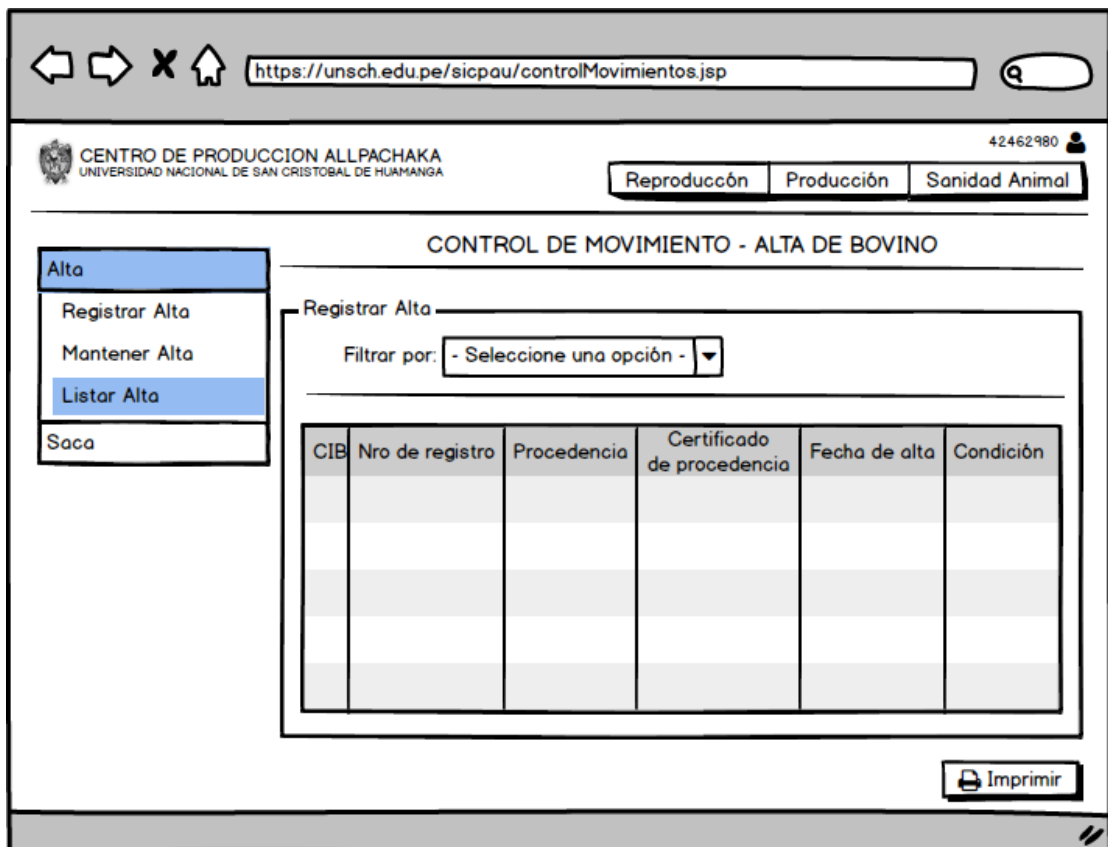


Figura N° 4.9: Interfaz Listar Alta.

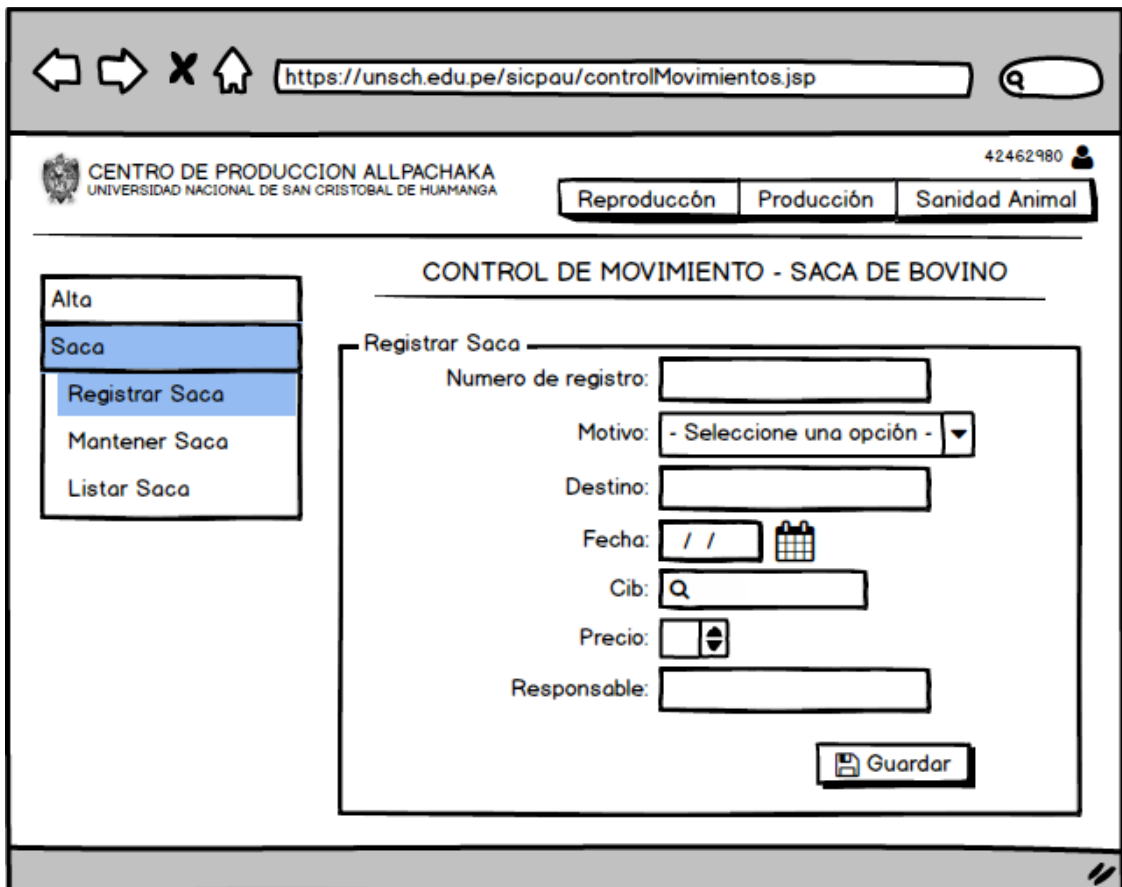


Figura N° 4.10: Interfaz Registrar Saca.

https://unsch.edu.pe/sicpau/control-movimientos.jsp

CENTRO DE PRODUCCION ALLPACHAKA
 UNIVERSIDAD NACIONAL DE SAN CRISTOBAL DE HUAMANGA

42462980

Reproducción Producción Sanidad Animal

CONTROL DE MOVIMIENTO - SACAS DE BOVINO

Mantener Saca

Código del bovino:

Numero de registro:

Motivo:

Destino:

Fecha:

Cib:

Precio:

Responsable:

Figura N° 4.11: Interfaz Mantener Saca.

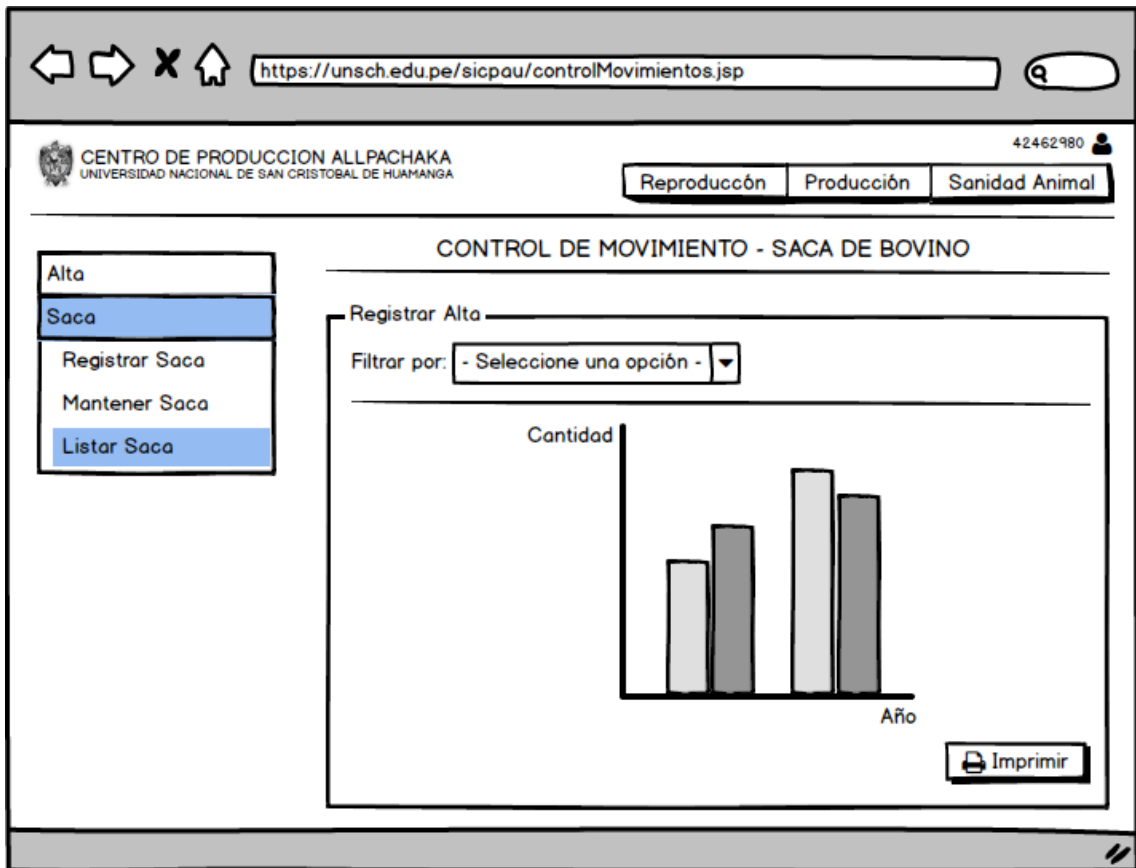


Figura N° 4.12: Interfaz Listar Saca.

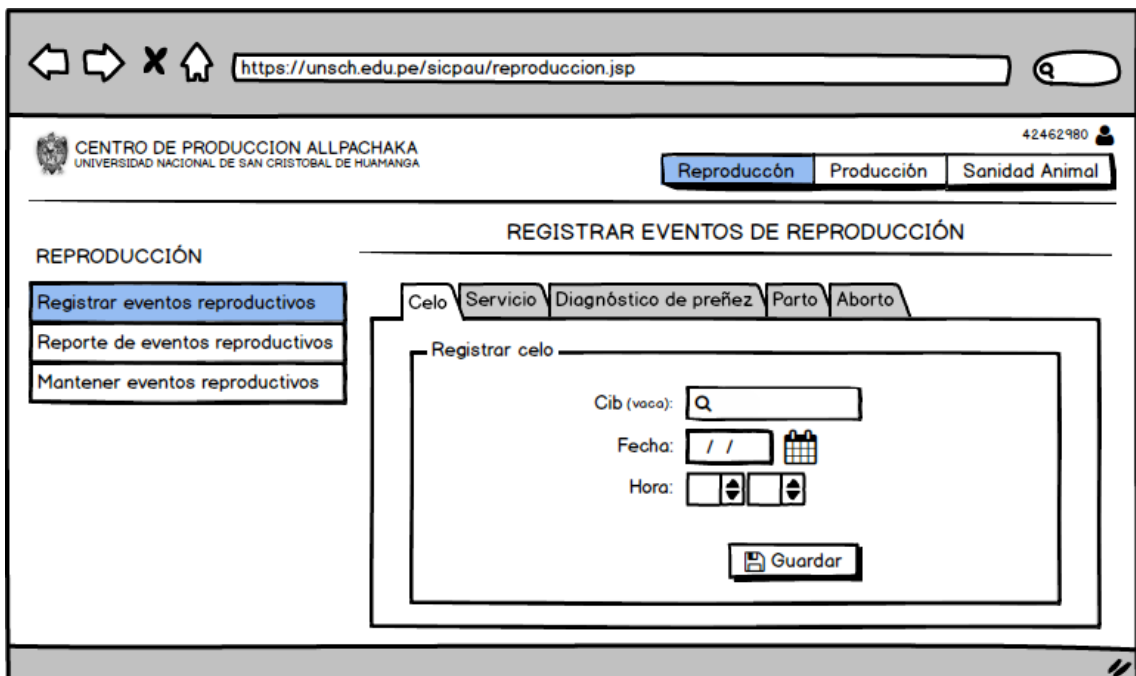


Figura N° 4.13: Interfaz Registrar Celo.

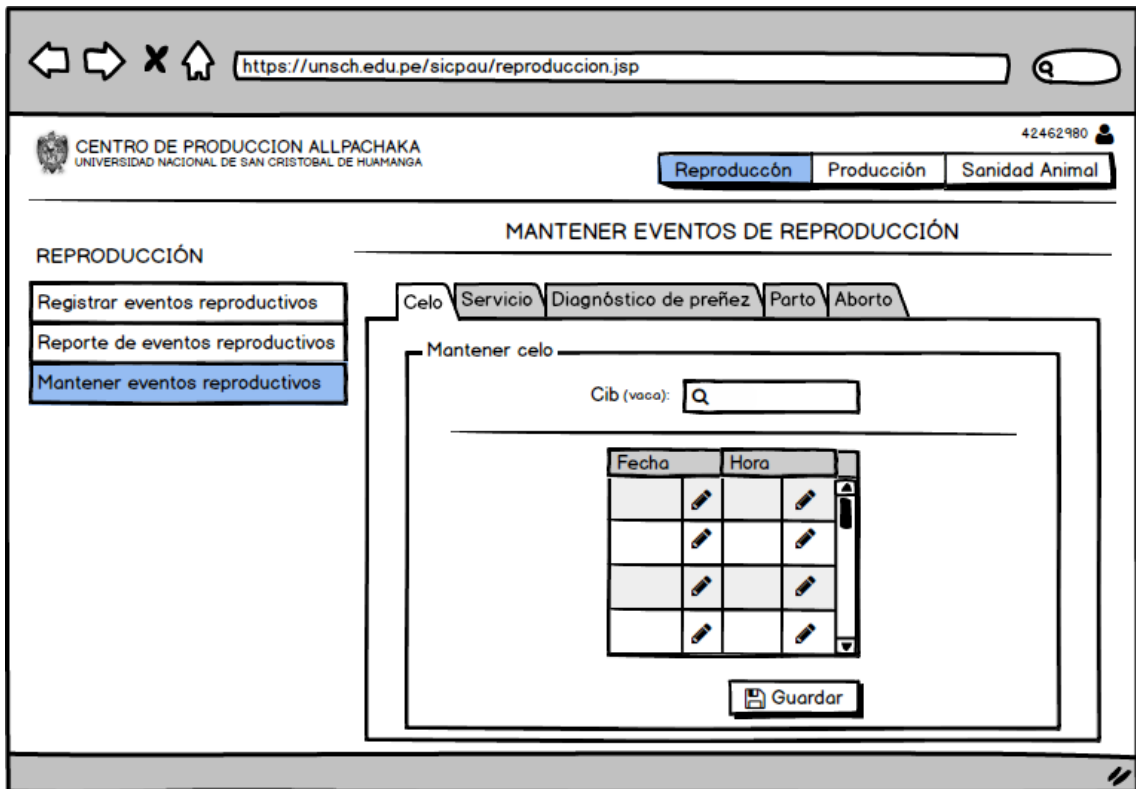


Figura N° 4.14: Mantener Celo.

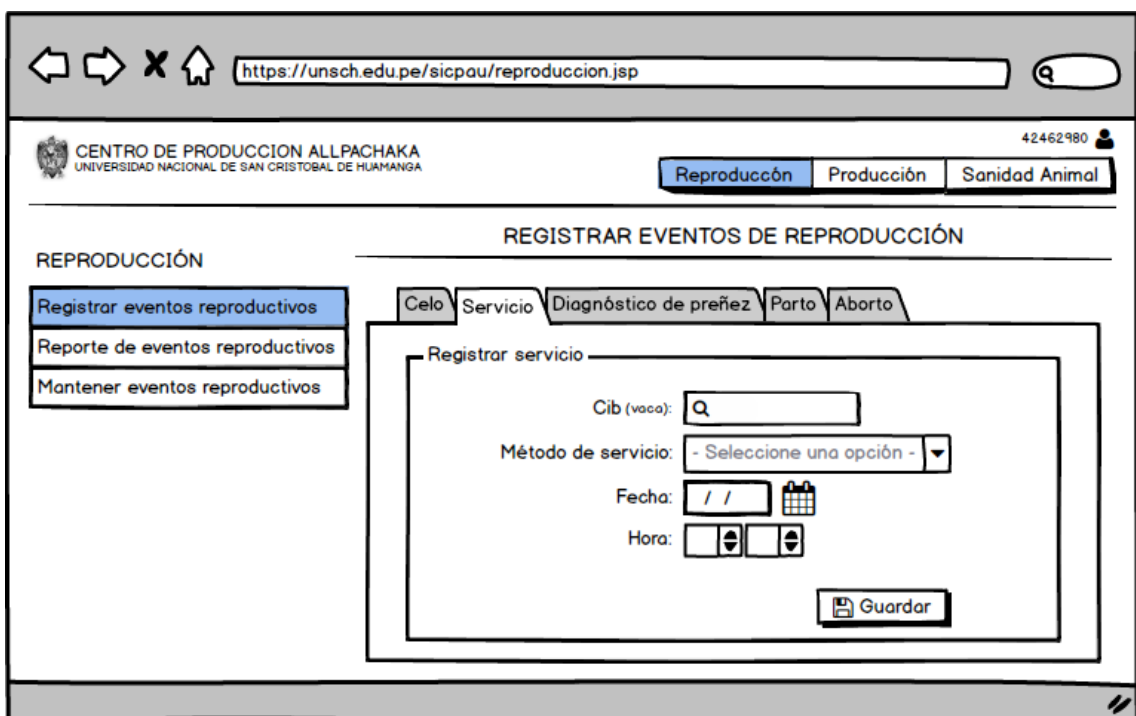


Figura N° 4.15: Interfaz Registrar Servicio.

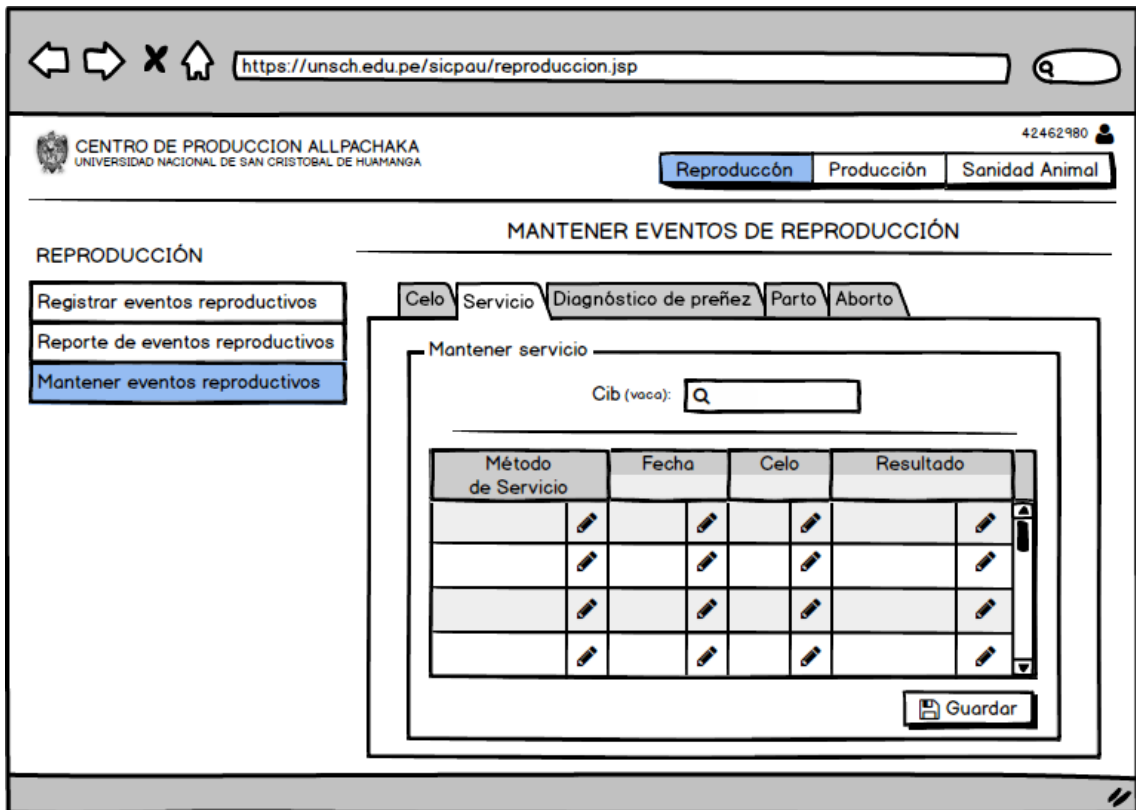


Figura N° 4.16: Interfaz Mantener Servicio.

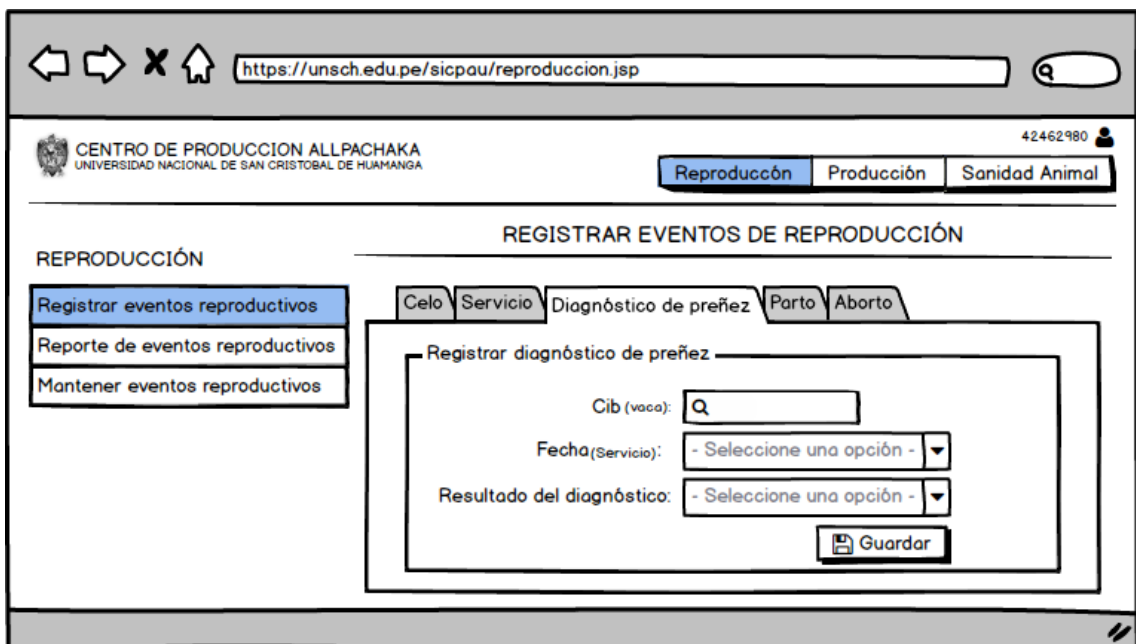


Figura N° 4.17: Interfaz Registrar Diagnóstico de Preñez.

https://unsch.edu.pe/sicpau/reproduccion.jsp

CENTRO DE PRODUCCIÓN ALLPACHAKA
 UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA

42462980

Reproducción Producción Sanidad Animal

REGISTRAR EVENTOS DE REPRODUCCIÓN

Celo Servicio Diagnóstico de preñez Parto Aborto

REPRODUCCIÓN

- Registrar eventos reproductivos
- Reporte de eventos reproductivos
- Mantener eventos reproductivos

Registrar datos del ternero

Cib(madre):

Cib(padre):

Cib(ternero):

Nombre:

Fecha de nacimiento:

Raza:

Sexo:

Fotografía:

Registrar datos del parto

Cib(madre):

Cib(ternero):

Dificultad:

Hipocalcemia:

Placenta retenida:

Edema de la ubre:

Observación(es):

Figura N° 4.18: Interfaz Registrar Parto.

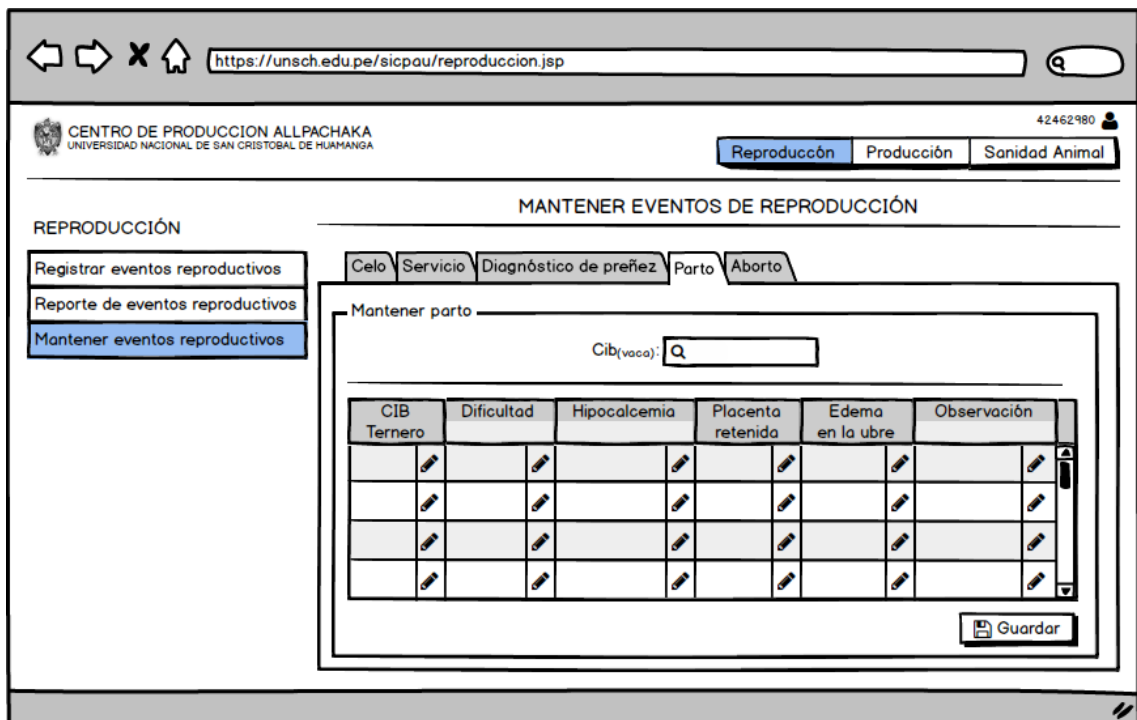


Figura N° 4.19: Interfaz Mantener Parto.



Figura N° 4.20: Interfaz Mantener Datos del Bovino.

<https://unsch.edu.pe/sicpau/reproduccion.jsp>

CENTRO DE PRODUCCION ALLPACHAKA
 UNIVERSIDAD NACIONAL DE SAN CRISTOBAL DE HUAMANGA

42462980

[Reproducción](#) [Producción](#) [Sanidad Animal](#)

REGISTRAR EVENTOS DE REPRODUCCIÓN

[Celo](#) [Servicio](#) [Diagnóstico de preñez](#) [Parto](#) [Aborto](#)

REPRODUCCIÓN

- Registrar eventos reproductivos
- Reporte de eventos reproductivos
- Mantener eventos reproductivos

Registrar aborto

Cib_(madre):

Fecha(Servicio):

Fecha(Aborto):

Causa:

Agente:

Observación(es):

Guardar

Figura N° 4.21: Interfaz Registrar Aborto.

<https://unsch.edu.pe/sicpau/reproduccion.jsp>

CENTRO DE PRODUCCION ALLPACHAKA
 UNIVERSIDAD NACIONAL DE SAN CRISTOBAL DE HUAMANGA

42462980

[Reproducción](#) [Producción](#) [Sanidad Animal](#)

MANTENER EVENTOS DE REPRODUCCIÓN

[Celo](#) [Servicio](#) [Diagnóstico de preñez](#) [Parto](#) [Aborto](#)

REPRODUCCIÓN

- Registrar eventos reproductivos
- Reporte de eventos reproductivos
- Mantener eventos reproductivos

Mantener aborto

Cib_(vaca):

Fecha	Causa	Agente	Observación

Guardar

Figura N° 4.22: Interfaz Mantener Aborto.



Figura N° 4.25: Interfaz Registrar Leche.

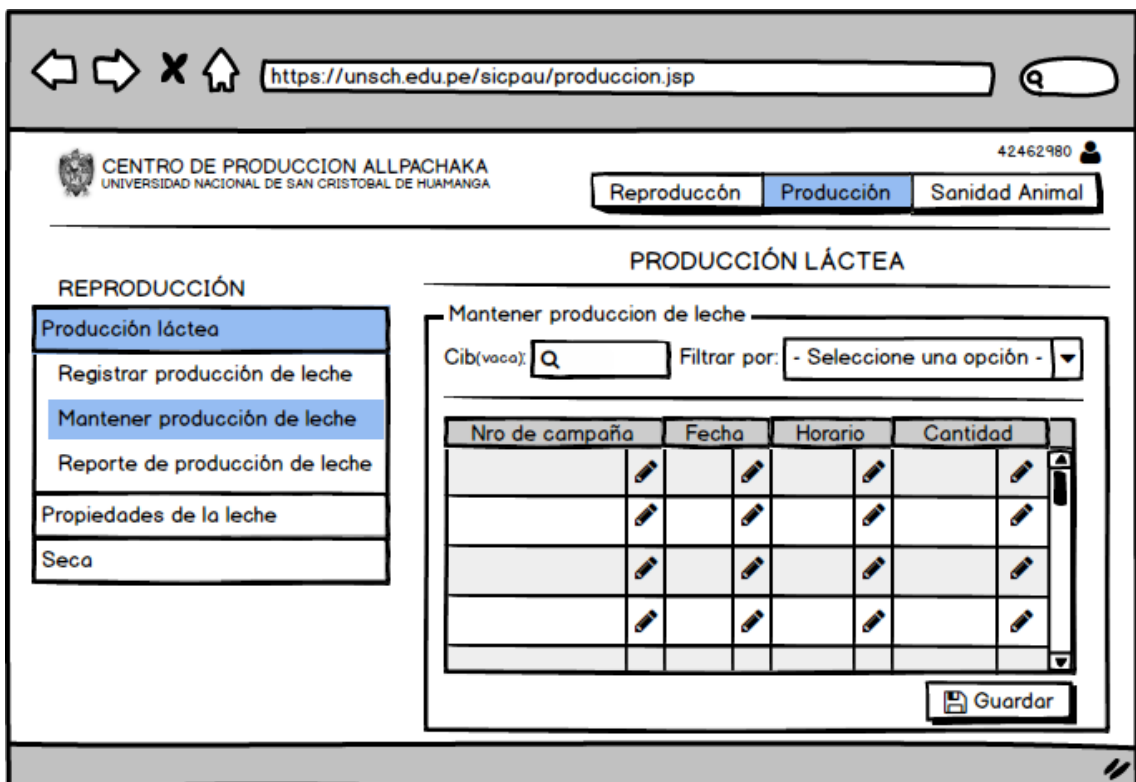


Figura N° 4.26: Interfaz Mantener Producción de Leche.

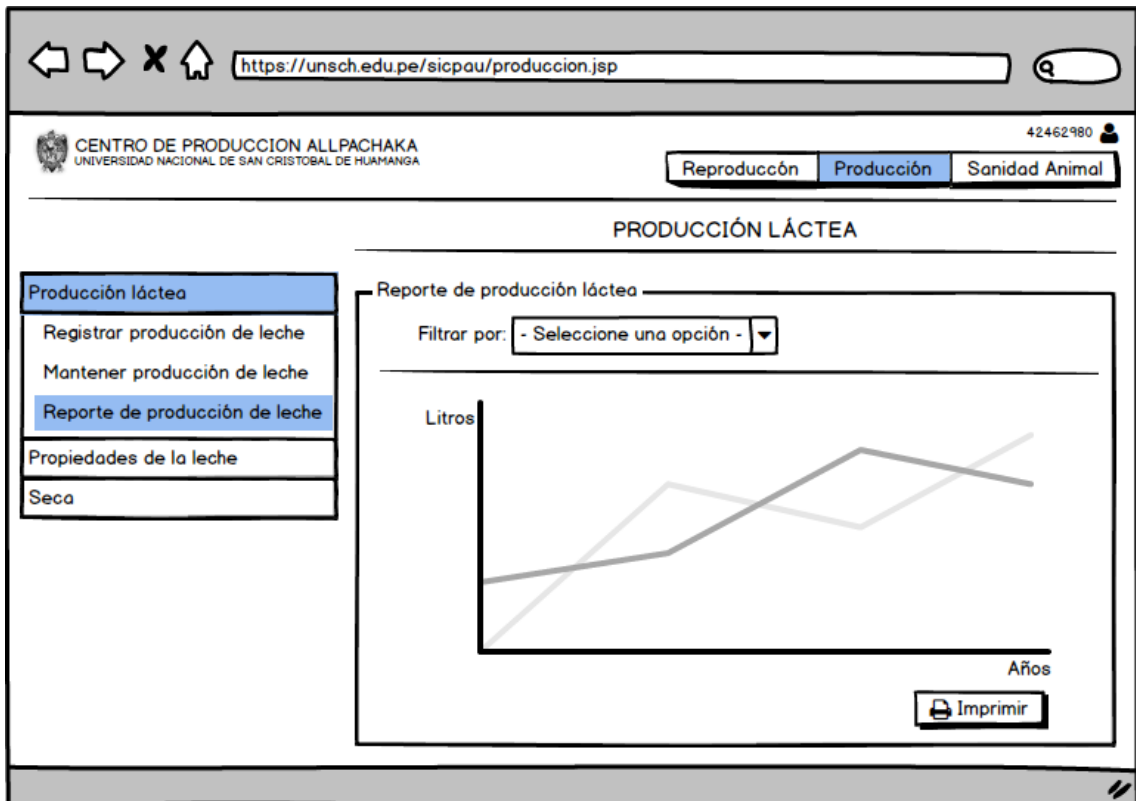


Figura N° 4.27: Interfaz Reporte de Producción de Leche.

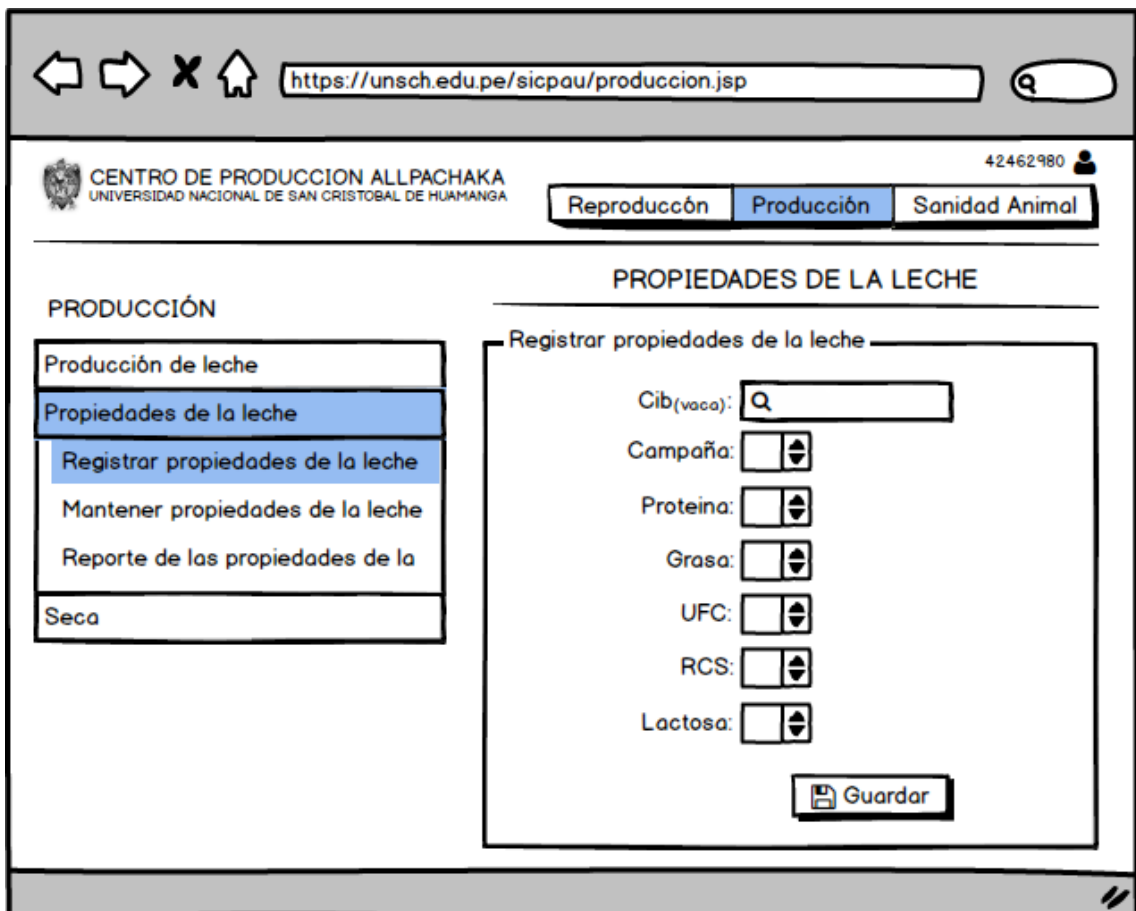


Figura N° 4.28: Interfaz Registrar Propiedades de la Leche.

https://unsch.edu.pe/sicpau/produccion.jsp

CENTRO DE PRODUCCION ALLPACHAKA
UNIVERSIDAD NACIONAL DE SAN CRISTOBAL DE HUAMANGA

42462980

Reproducción Producción Sanidad Animal

PRODUCCIÓN

Producción de leche

Propiedades de la leche

Registrar propiedades de la leche

Mantener propiedades de la leche

Reporte de las propiedades de la leche

Seca

PROPIEDADES DE LA LECHE

Mantener propiedades de la leche

Cib(vaca):

Nro de campaña	Proteina	Grasa	UFC	RCS	Lactosa

Guardar

Figura N° 4.29: Interfaz Mantener Registro de Propiedades de la Leche.

https://unsch.edu.pe/sicpau/produccion.jsp

CENTRO DE PRODUCCION ALLPACHAKA
UNIVERSIDAD NACIONAL DE SAN CRISTOBAL DE HUAMANGA

42462980

Reproducción Producción Sanidad Animal

PROPIEDADES DE LA LECHE

Reporte de propiedades de la leche

Cib(vaca):

Imprimir

Figura N° 4.30: Interfaz Reporte de Propiedades de la Leche.



Figura N° 4.31: Interfaz Registrar Seca.

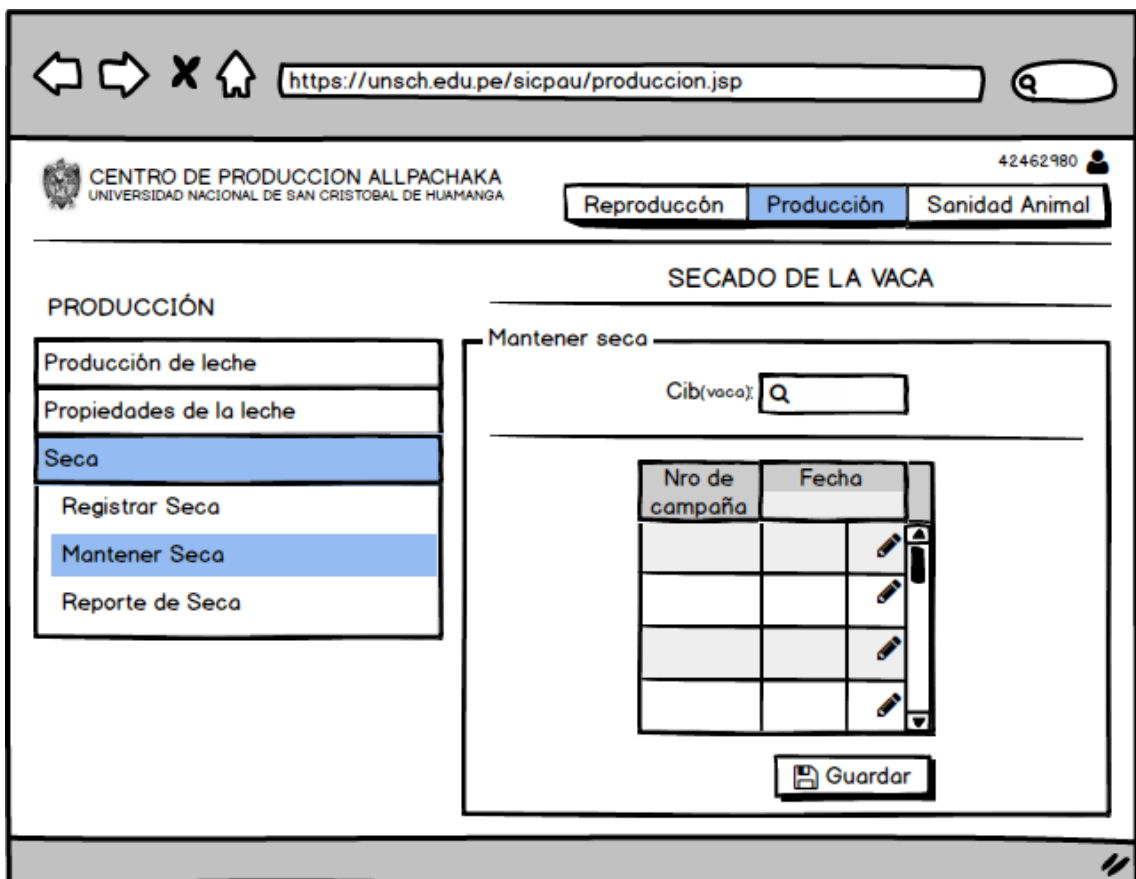


Figura N° 4.32: Interfaz Mantener Seca.

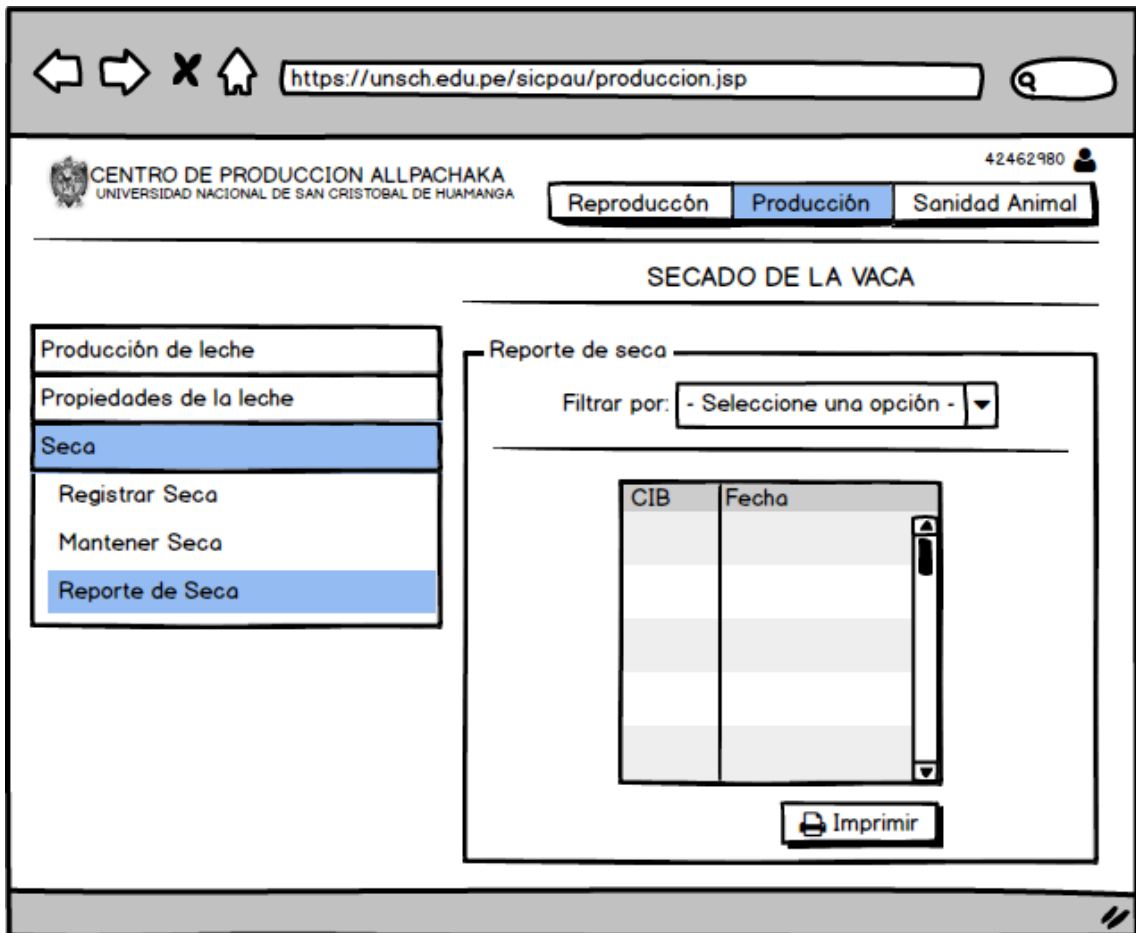


Figura N° 4.33: Interfaz Reporte de Seca.

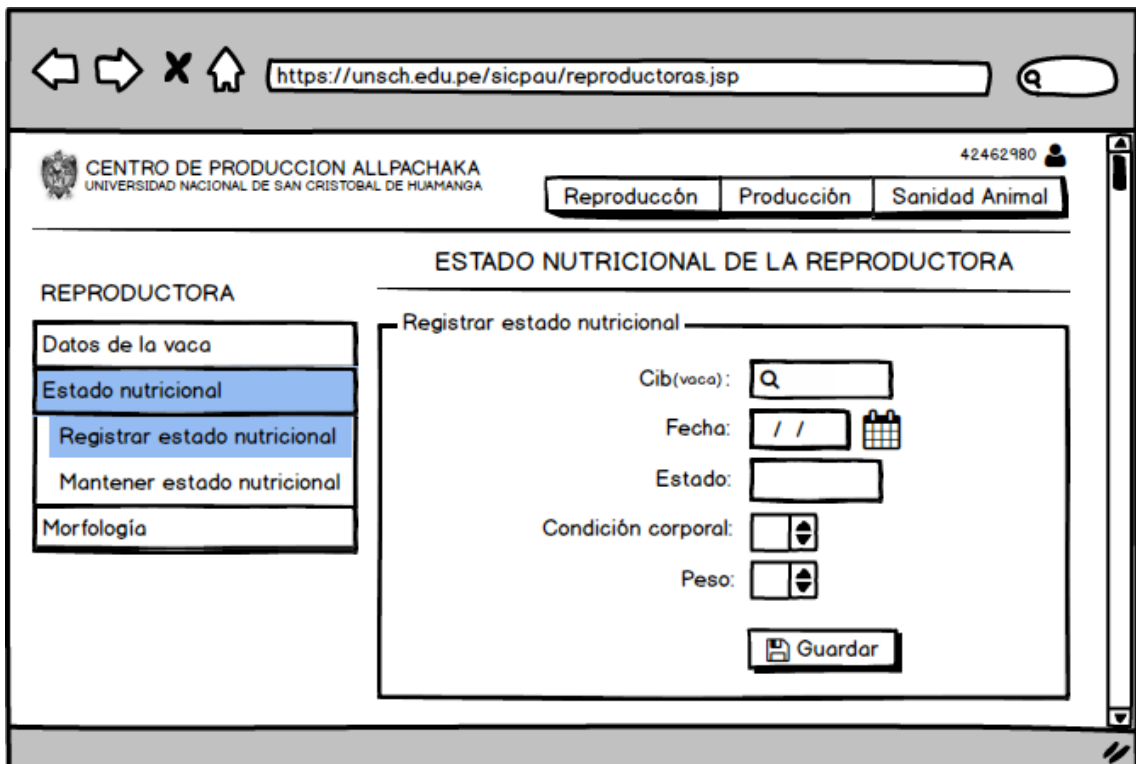


Figura N° 4.34: Interfaz Registrar Estado Nutricional de la Reproductora.

https://unsch.edu.pe/sicpau/reproductoras.jsp

CENTRO DE PRODUCCION ALLPACHAKA
 UNIVERSIDAD NACIONAL DE SAN CRISTOBAL DE HUAMANGA

42462980

Reproducción Producción Sanidad Animal

REPRODUCTORA

Datos de la vaca
 Estado nutricional
 Registrar estado nutricional
 Mantener estado nutricional
 Morfología



ESTADO NUTRICIONAL DE LA REPRODUCTORA

Mantener estado nutricional

Cib(vaca):

Fecha	Estado	Condición Corporal	Peso

Figura N° 4. 35: Interfaz Mantener Estado Nutricional de la Reproductora.


CENTRO DE PRODUCCION ALLPACHAKA
 UNIVERSIDAD NACIONAL DE SAN CRISTOBAL DE HUAMANGA
 42462980 

MORFOLOGIA DE LA REPRODUCTORA

REPRODUCTORA

Datos de la vaca

Estado nutricional


Morfología

Registrar Morfología

Mantener Morfología

Registrar Morfología

Cib(vaca):

Fecha: 

Estatura:

Profundidad Corporal:

Angulosidad:

Ancho del Pecho:

Ancho de la Grupa:

Angulo de la Grupa:

Vista Lateral de las Patas Traseras:

Vista Posterior de las Patas Traseras:

Angulo Podal:

Ligamento Delantero:

Altura de la Ubre:

Ancho de la Ubre:

Profundidad de la Ubre:

Figura N° 4.36: Interfaz Registrar Morfología de la Reproductora.

https://unsch.edu.pe/sicpau/reproductoras.jsp

CENTRO DE PRODUCCION ALLPACHAKA
UNIVERSIDAD NACIONAL DE SAN CRISTOBAL DE HUAMANGA

42462980

Reproducción Producción Sanidad Animal

MORFOLOGIA LA REPRODUCTORA

Mantener morfología de la reproductora _____

Cib(vaca):

CARACTERISTICAS MORFOLOGICAS														
Edad	Estatura	Profundidad Corporal	Angulosidad	Ancho del Pecho	Ancho de la Grupa	Angulo de la Grupa	Vista Lateral de las Patas Traseras	Vista Posterior de las Patas traseras	Angulo Podal	Ligamento Delantero	Altura de la Ubre	Ancho de la Ubre	Profundidad de la Ubre	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Guardar

Figura N° 4.37: Interfaz Mantener Morfología de la Reproductora.

<https://unsch.edu.pe/sicpau/reproductoras.jsp>

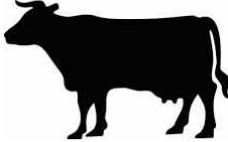
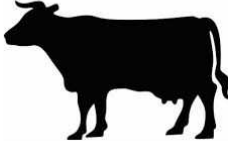
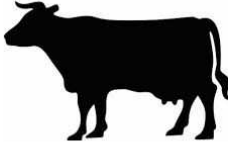
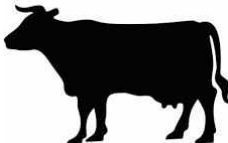
CENTRO DE PRODUCCION ALLPACHAKA
 UNIVERSIDAD NACIONAL DE SAN CRISTOBAL DE HUAMANGA

42462780

Reproducción Producción Sanidad Animal

REPRODUCTORA

DATOS SOBRE LA REPRODUCTORA

Datos de la vaca			
Estado nutricional	Nombre: CIB: Edad: Estado Reproductivo: Estado Productivo: Estado Sanitario:		Nombre: CIB: Edad: Estado Reproductivo: Estado Productivo: Estado Sanitario:
Marfología	Nombre: CIB: Edad: Estado Reproductivo: Estado Productivo: Estado Sanitario:		Nombre: CIB: Edad: Estado Reproductivo: Estado Productivo: Estado Sanitario:
	Nombre: CIB: Edad: Estado Reproductivo: Estado Productivo: Estado Sanitario:		Nombre: CIB: Edad: Estado Reproductivo: Estado Productivo: Estado Sanitario:
	Nombre: CIB: Edad: Estado Reproductivo: Estado Productivo: Estado Sanitario:		Nombre: CIB: Edad: Estado Reproductivo: Estado Productivo: Estado Sanitario:

Primero < Atrás 1 2 3 4 5 6 7 8 9 10 Siguiente > Último

Figura N° 4.38: Interfaz Reporte de Datos Generales de la Reproductora.

REPRODUCTORA
Salir

Datos de la vaca

Nombre: _____		Padre: _____	Padre: _____	Estado Reproductivo: _____
CIB: _____		Madre: _____	Madre: _____	
Edad: _____				Estado Productivo: _____
Fecha de nacimiento: _____				Estado Sanitario: _____
Raza: _____		Padre: _____	Padre: _____	
Sexo: _____		Madre: _____	Madre: _____	

CARACTERISTICAS MORFOLOGICAS													
Edad	Estatura	Profundidad Corporal	Angulosidad	Ancho del Pecho	Ancho de la Grupa	Angulo de la Grupa	Vista Lateral de las Patas Traseras	Vista Posterior de las Patas Traseras	Angulo Podal	Ligamento Delantero	Altura de la Ubre	Ancho de la Ubre	Profundidad de la Ubre

CRECIMIENTO Y DESARROLLO						
Fecha	Peso	Perimetro Toracico	Altura Cruz	CC	RCS	Observación

SANIDAD							
Fecha de inicio	Fecha de finalización	Tipo	Denominación	Tratamiento	Días	Comentario	Responsable

PRODUCCION LACTEA							
Nro de campaña	Kg	Proteina	Grasa	UFC	RCS	Lactosa	Nro de días

Crecimiento y Desarrollo - Estado Nutricional

Imprimir

Figura N° 4.39: Interfaz Reporte de Datos Detallados de la Reproductora.

<https://unsch.edu.pe/sicpau/semntales.jsp>
42462780

CENTRO DE PRODUCCION ALLPACHAKA
 UNIVERSIDAD NACIONAL DE SAN CRISTOBAL DE HUAMANGA

Reproucción Producción Sanidad Animal

DATOS SOBRE EL SEMENTAL

SEMENTAL

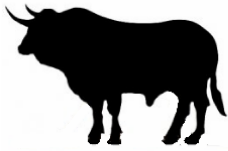
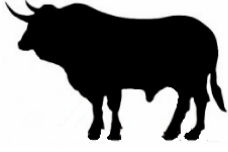


Datos del Semental

Morfología del Semental

Aparato Reproductor

Libido

Semen

Nombre: _____ CIB: _____ Edad: _____ Servicios: _____ Servicios Infértiles: _____	Preñadas: _____ Abortos: _____ Partos Problemáticos: _____ Partos: _____	
Nombre: _____ CIB: _____ Edad: _____ Servicios: _____ Servicios Infértiles: _____	Preñadas: _____ Abortos: _____ Partos Problemáticos: _____ Partos: _____	
Nombre: _____ CIB: _____ Edad: _____ Servicios: _____ Servicios Infértiles: _____	Preñadas: _____ Abortos: _____ Partos Problemáticos: _____ Partos: _____	
Nombre: _____ CIB: _____ Edad: _____ Servicios: _____ Servicios Infértiles: _____	Preñadas: _____ Abortos: _____ Partos Problemáticos: _____ Partos: _____	

Primero < Atrás 1 2 3 4 5 6 7 8 9 10 Siguiete > Último


Figura N° 4.40: Interfaz Reporte de Datos Generales del Semental.

SEMENTAL

Datos de la vaca

Nombre: _____
 CIB: _____ Padre: _____ Padre: _____
 Edad: _____ Madre: _____
 Fecha de nacimiento: _____ Padre: _____
 Raza: _____ Madre: _____
 Sexo: _____

Servicios: _____ Preñadas: _____ Partos Problematicos: _____
 Servicios Infértiles: _____ Abortos: _____ Partos: _____



CARACTERISTICAS MORFOLOGICAS

Edad	Estatura	Profundidad Corporal	Angulosidad	Ancho del Pecho	Ancho de la Grupa	Angulo de la Grupa	Vista Lateral de las Patas Traseras	Vista Posterior de las Patas Traseras	Angulo Podal

CARACTERISTICAS FISICAS DEL APARATO REPRODUCTOR DEL SEMENTAL

Edad	Circunferencia Escrotal	Circunferencia del Escroto	Forma del Testículo	Posición del Escroto	Consistencia del Escroto

EVALUACION DEL LIBIDO

Edad	Olfateo Genital	Lameteo Genital	Apoyo de Cabeza	Aparejo	Flehmen	Intento de Monta	Monta Incompleta	Servicio

INVENTARIO DE PAJILLA

Fecha	Glosa	Cantidad Entrada	Cantidad Salida	Tanque	Canastilla	Procedencia	Destino	Precio	Stock

CUALIDADES DEL SEMEN

Fecha Colectada	Volumen	pH	Motilidad masal	Motilidad Individual	Concentración Espermática	Anormalidades	Viabilidad Espermática

SANIDAD

Fecha de inicio	Fecha de finalización	Tipo	Denominación	Tratamiento	Días	Comentario	Responsable

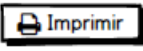


Figura Nº 4.41: Interfaz Reporte de Datos Detallados del Semental.

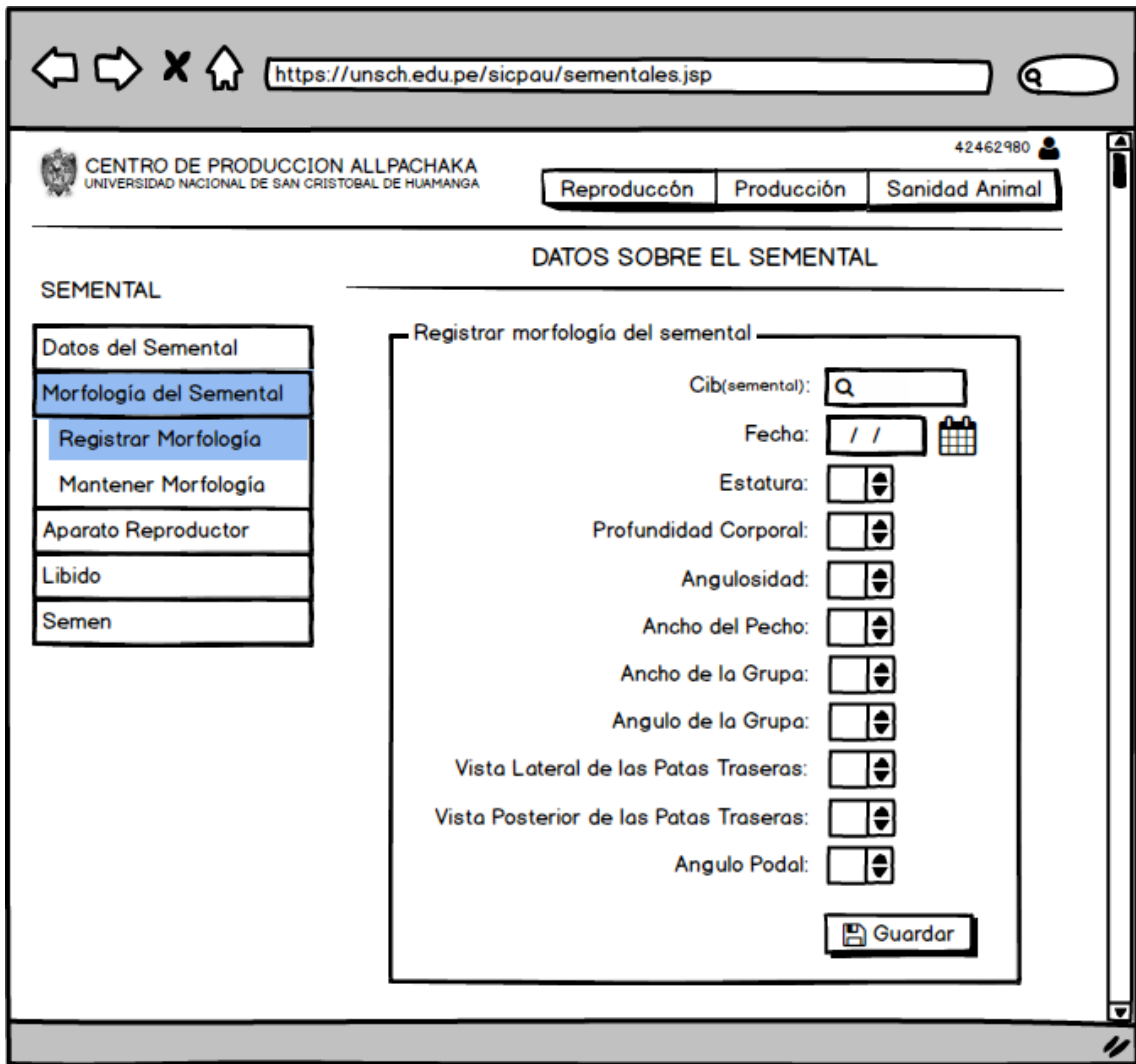


Figura N° 4.42: Interfaz Registrar Morfología del Semental.

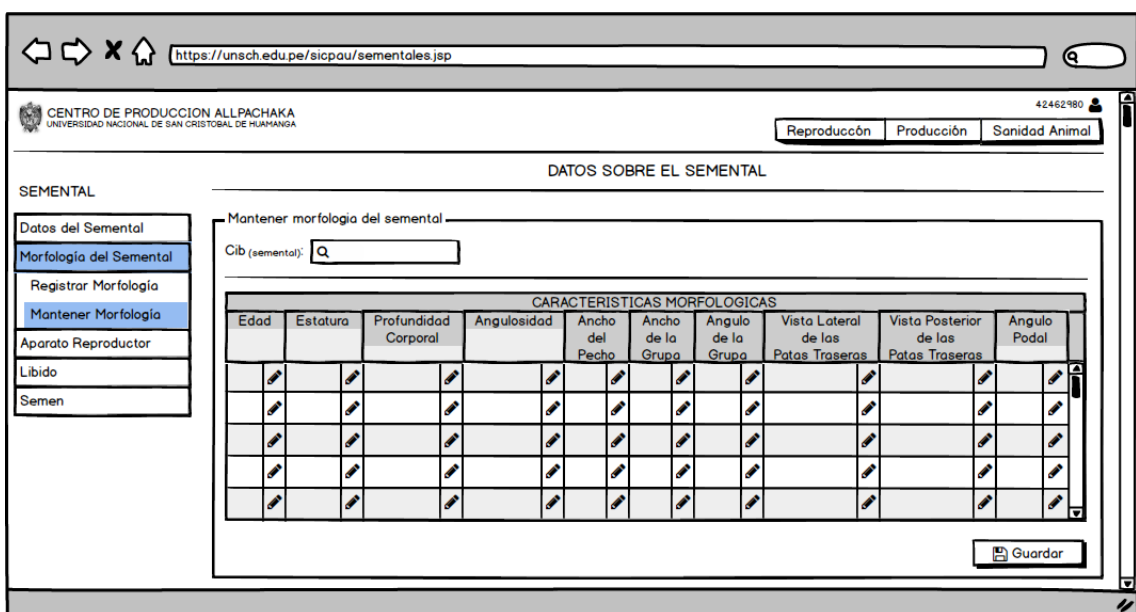


Figura N° 4.43: Interfaz Mantener Morfología del Semental.

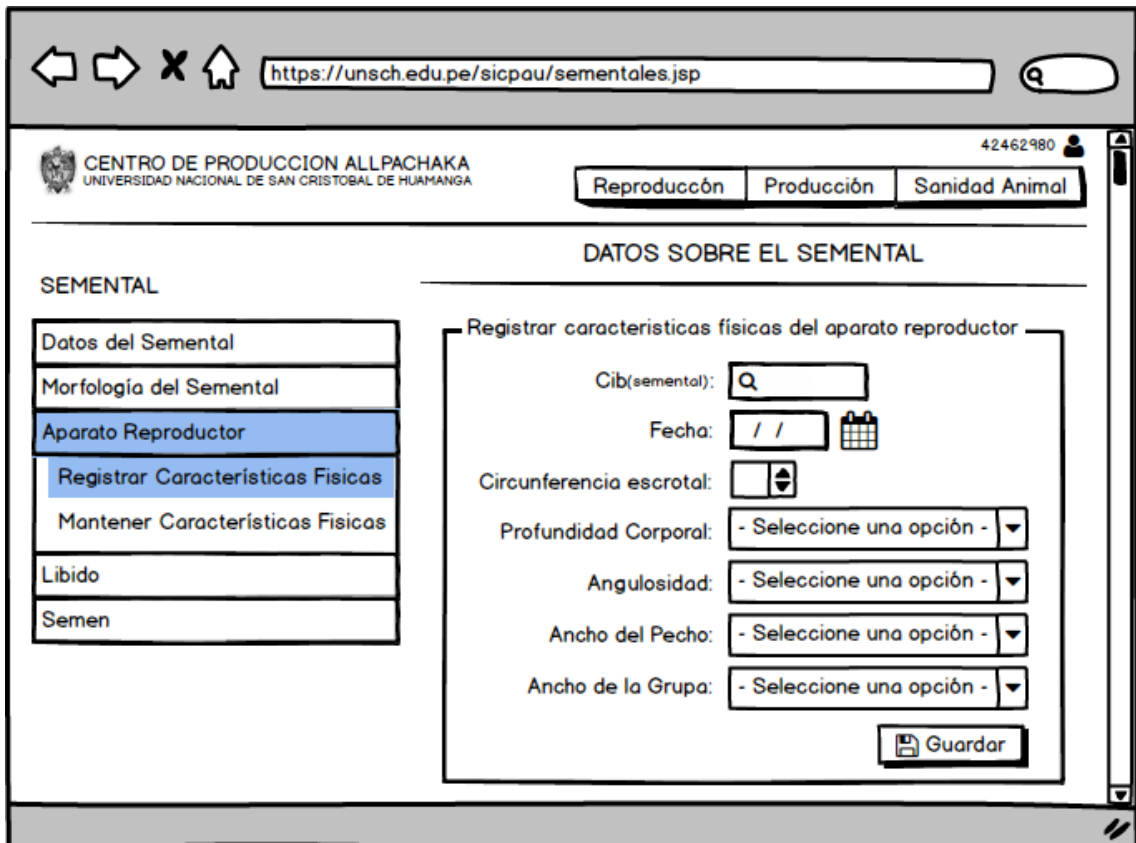


Figura N° 4.44: Interfaz Registrar Características Físicas del Aparato Reproductor del Semental.

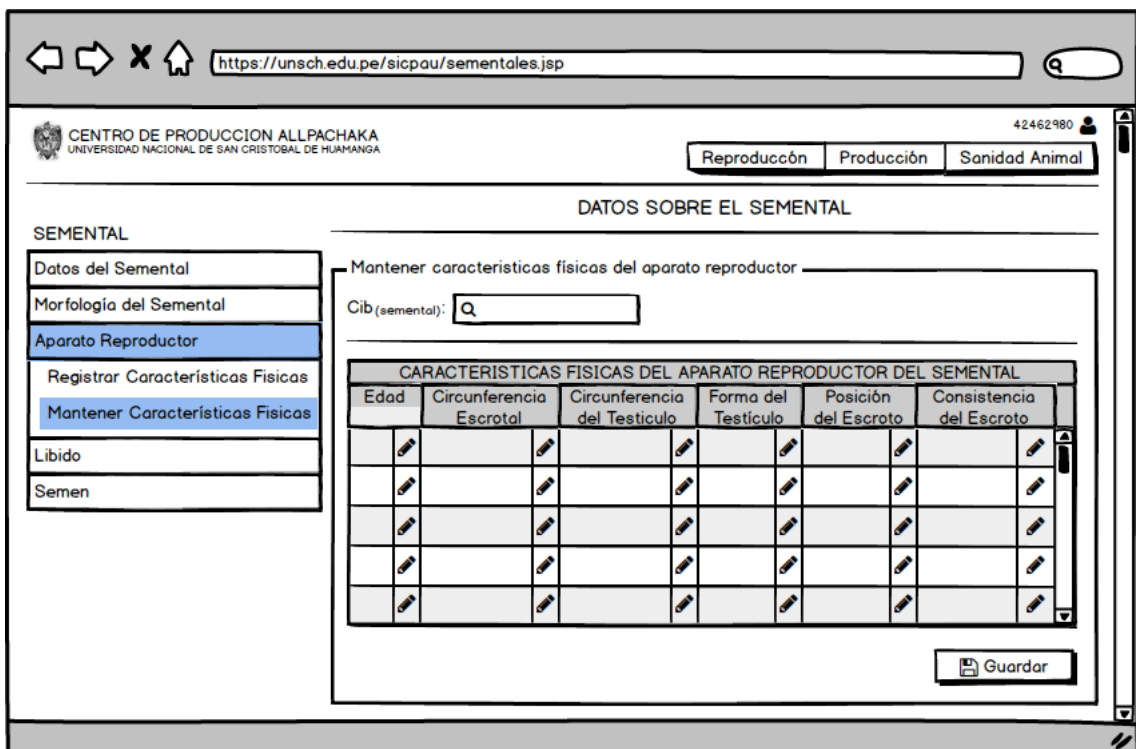


Figura N° 4.45: Interfaz Mantener Características Físicas del Aparato Reproductor del Semental.

← → × 🏠 <https://unsch.edu.pe/sicpau/semntales.jsp> 🔍

CENTRO DE PRODUCCION ALLPACHAKA
UNIVERSIDAD NACIONAL DE SAN CRISTOBAL DE HUAMANGA 42462980

Reproducción Producción Sanidad Animal

SEMENTAL

DATOS SOBRE EL SEMENTAL

SEMENTAL

- Datos del Semental
- Morfología del Semental
- Aparato Reproductor
- Libido
- Registrar Evaluación del Libido
- Mantener Evaluación del Libido
- Semen

Registrar evaluación de libido

Cib(amental):

Fecha: / / 📅

Olfateo genital: ⬆️ ⬆️

Lameteo genital: ⬆️ ⬆️

Lameteo social: ⬆️ ⬆️

Apoyo de la cabeza: ⬆️ ⬆️

Aparejo: ⬆️ ⬆️

Flehmen: ⬆️ ⬆️

Intento de monta: ⬆️ ⬆️

Monta incompleta: ⬆️ ⬆️

Servicio: ⬆️ ⬆️

Figura N° 4.46: Interfaz Registrar Evaluación del Libido.

← → × 🏠 <https://unsch.edu.pe/sicpau/semntales.jsp> 🔍

CENTRO DE PRODUCCION ALLPACHAKA
UNIVERSIDAD NACIONAL DE SAN CRISTOBAL DE HUAMANGA 42462980

Reproducción Producción Sanidad Animal

SEMENTAL

DATOS SOBRE EL SEMENTAL

SEMENTAL

- Datos del Semental
- Morfología del Semental
- Aparato Reproductor
- Libido
- Registrar Evaluación del Libido
- Mantener Evaluación del Libido
- Semen

Mantener evaluación de libido

Cib(amental):

EVALUACION DEL LIBIDO								
Edad	Olfateo Genital	Lameteo Genital	Apoyo de Cabeza	Aparejo	Flehmen	Intento de Monta	Monta Incompleta	Servicio
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura N° 4.47: Interfaz Mantener Evaluación del Libido.

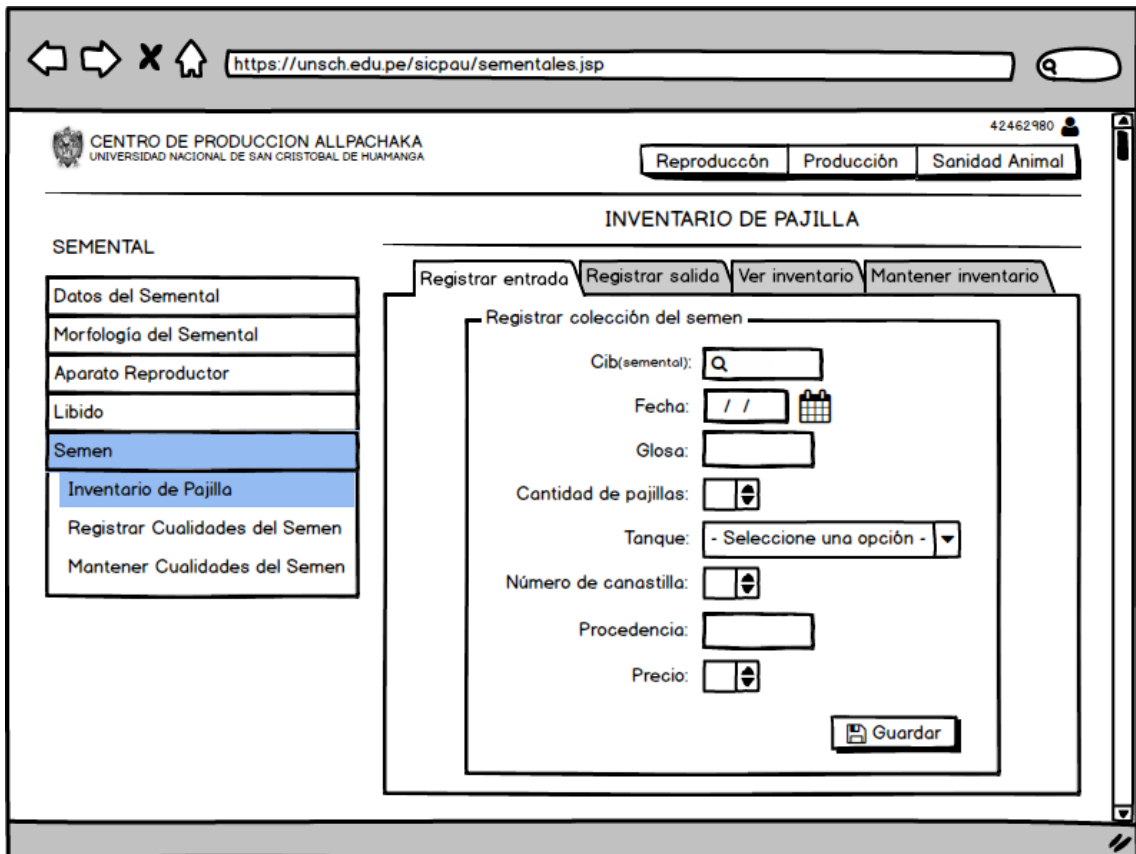


Figura N° 4.48: Interfaz Registrar Colección del Semen.

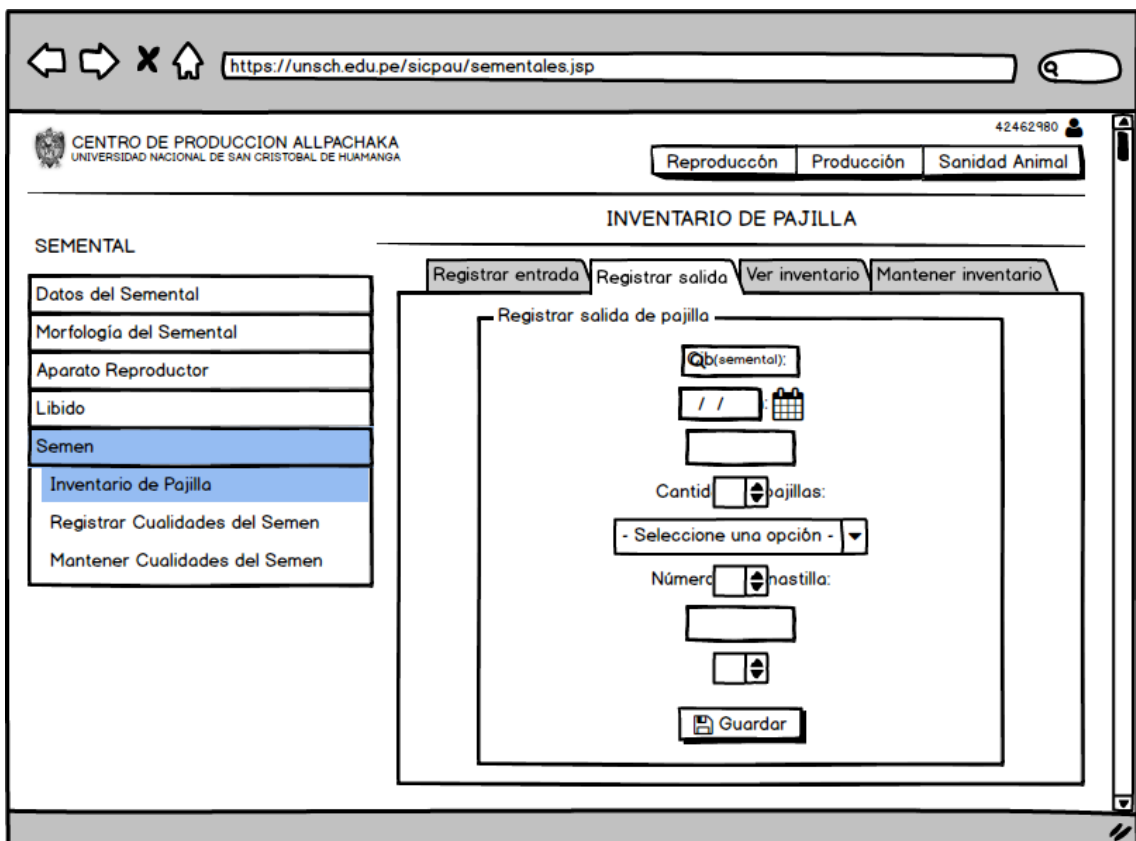


Figura N° 4.49: Interfaz Registrar Salida de Pajilla.

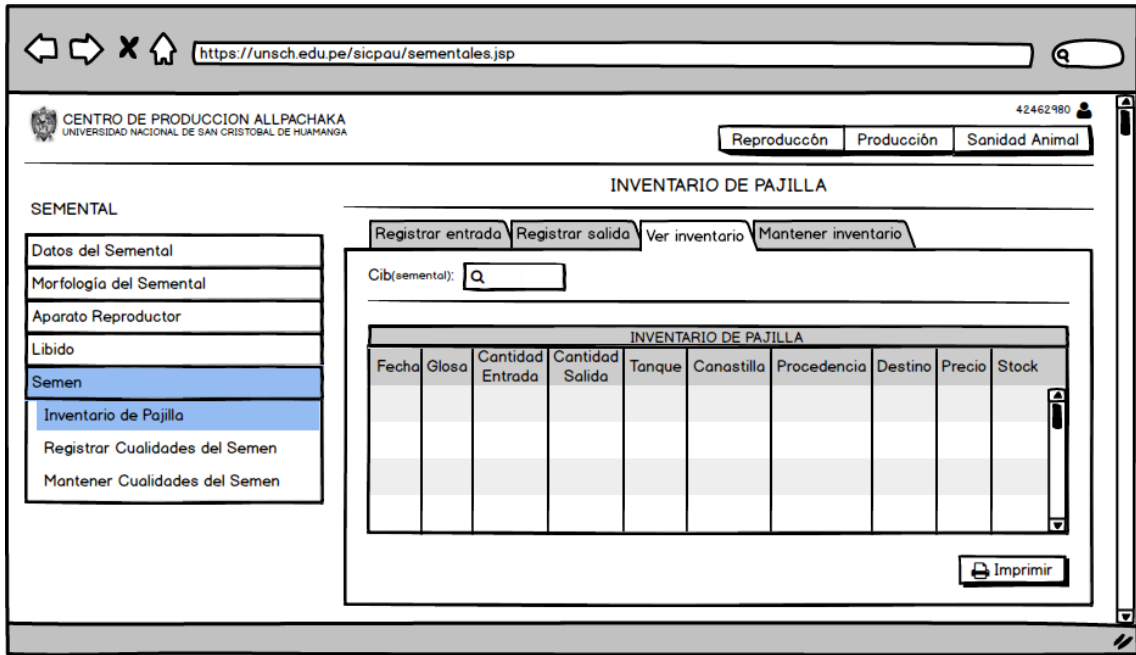


Figura N° 4.50: Interfaz Reporte de Inventario de Pajilla.

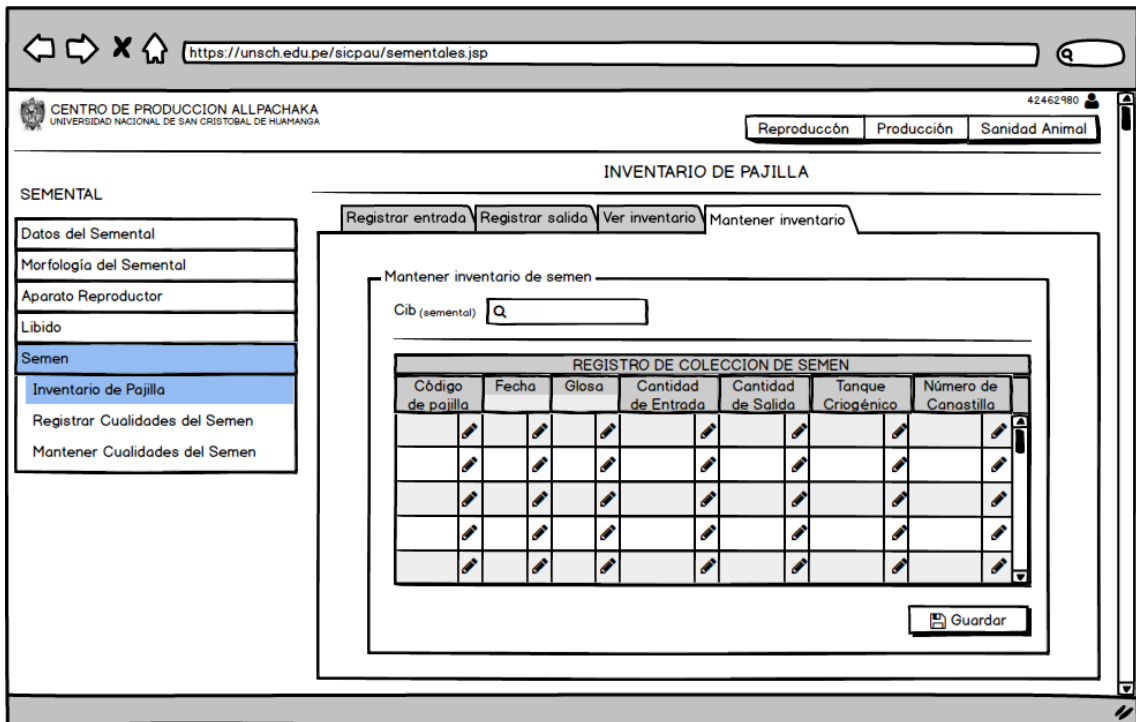


Figura N° 4.51: Interfaz Mantener Inventario de Pajilla.

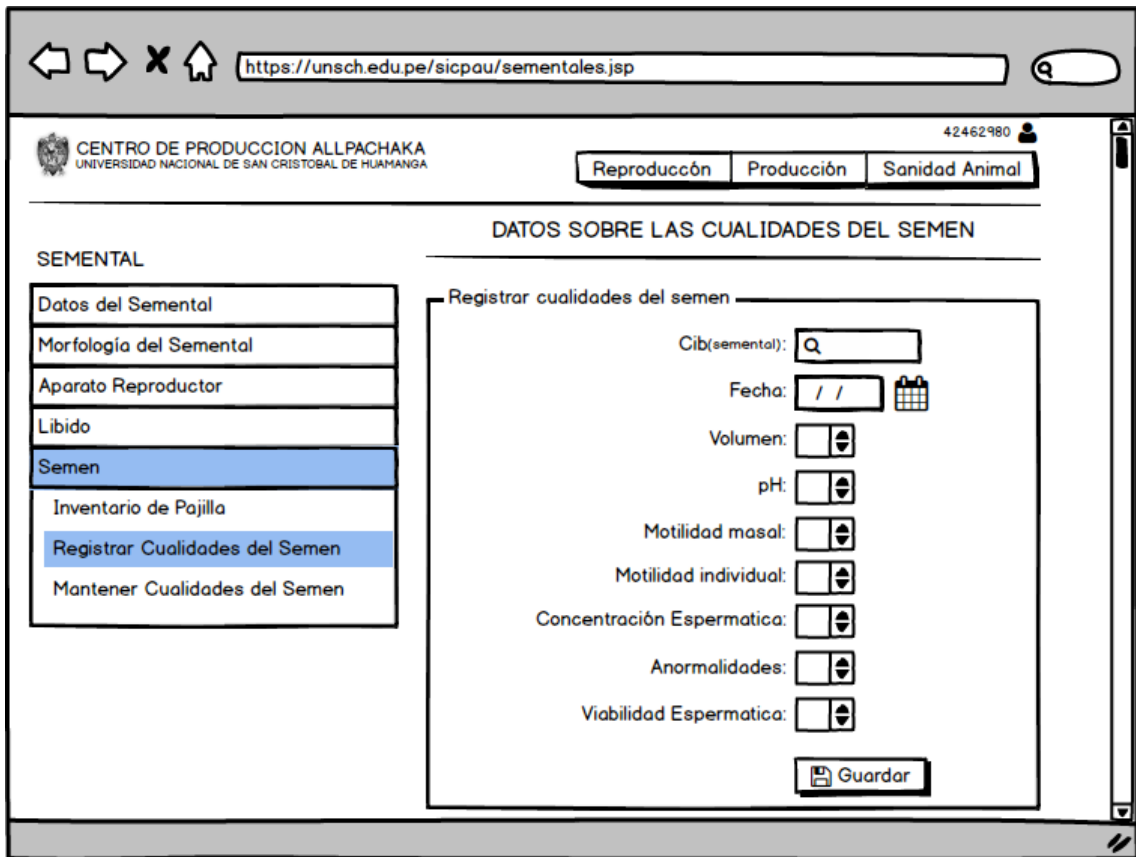


Figura N° 4.52: Interfaz Registrar Cualidades del Semen.

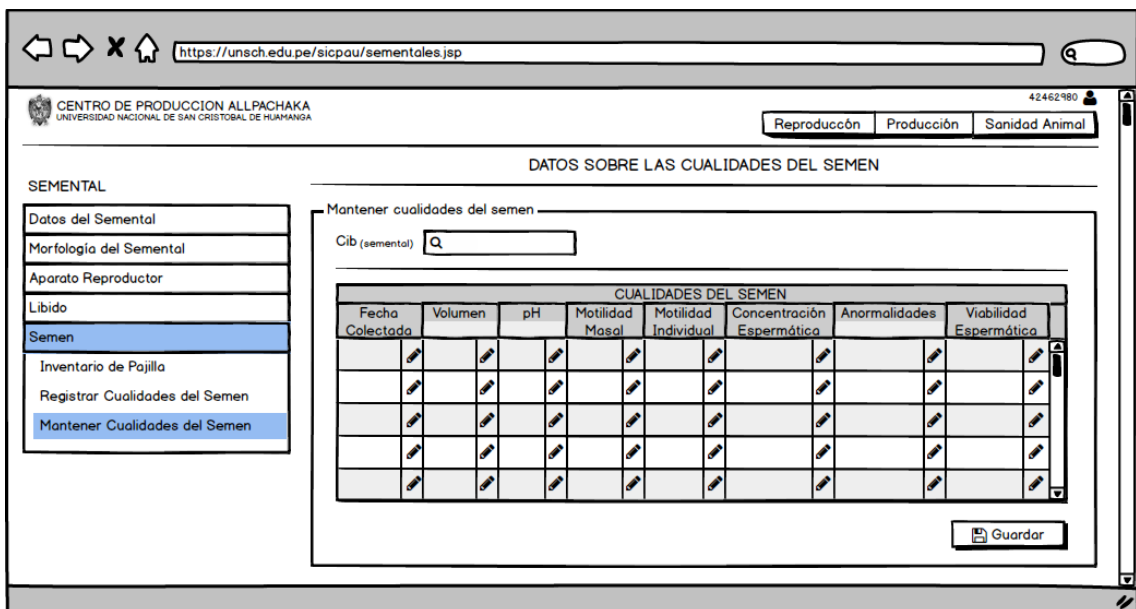


Figura N° 4.53: Interfaz Mantener Cualidades del Semen.

Figura N° 4.54: Interfaz Registrar Tratamiento Sanitario.

Fecha Inicio	Fecha Finalización	Tipo Registro	Nombre Registro	Tratamiento Utilizado	Días	Comentarios	Responsable

Figura N° 4.55: Interfaz Registrar Tratamiento Sanitario.

<https://unsch.edu.pe/sicpau/crecimiento-desarrollo.jsp>

CENTRO DE PRODUCCION ALLPACHAKA
 UNIVERSIDAD NACIONAL DE SAN CRISTOBAL DE HUAMANGA

42462980

Reproducción | Producción | **Sanidad Animal**

CRECIMIENTO Y DESARROLLO DEL TERNERO

SANIDAD

- Registrar Crecimiento y Desarrollo
- Mantener Crecimiento y Desarrollo
- Reporte de Crecimiento y Desarrollo

Registrar crecimiento y desarrollo del ternero

Cib:

Fecha:

Peso:

Perimetro Toracico:

Altura de cruz:

Condición corporal:

Observación:

Guardar

Figura N° 4.56: Interfaz Registrar Crecimiento y Desarrollo del Ternero.

<https://unsch.edu.pe/sicpau/crecimiento-desarrollo.jsp>

CENTRO DE PRODUCCION ALLPACHAKA
 UNIVERSIDAD NACIONAL DE SAN CRISTOBAL DE HUAMANGA

42462980

Reproducción | Producción | **Sanidad Animal**

CRECIMIENTO Y DESARROLLO DEL TERNERO

CRECIMIENTO Y DESARROLLO

- Registrar Crecimiento y Desarrollo
- Mantener Crecimiento y Desarrollo**
- Reporte de Crecimiento y Desarrollo

Cib:

REGISTRO DE CRECIMIENTO Y DESARROLLO						
Fecha	Peso	Perimetro Toracico	Altura de Cruz	Condición Corporal	Observación	

Guardar

Figura N° 4.57: Interfaz Mantener Crecimiento y Desarrollo del Ternero.

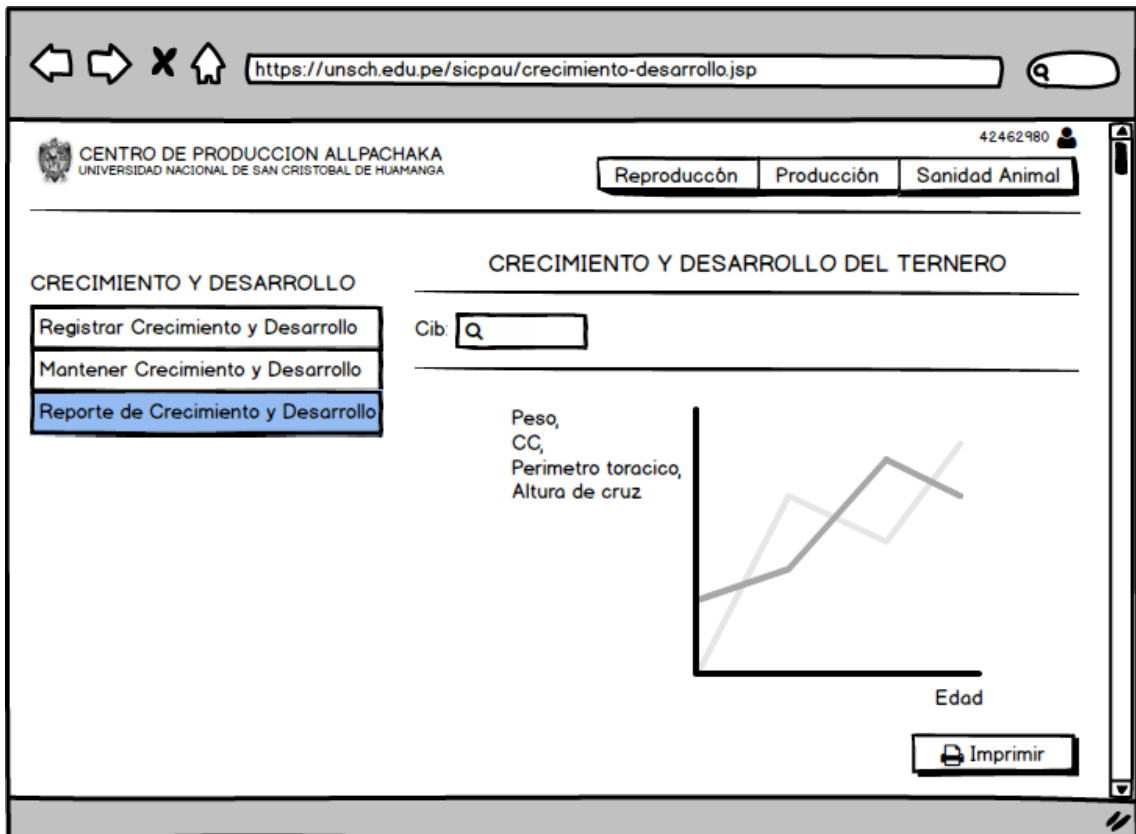


Figura N° 4.58: Interfaz Reporte de Crecimiento y Desarrollo del Ternero.

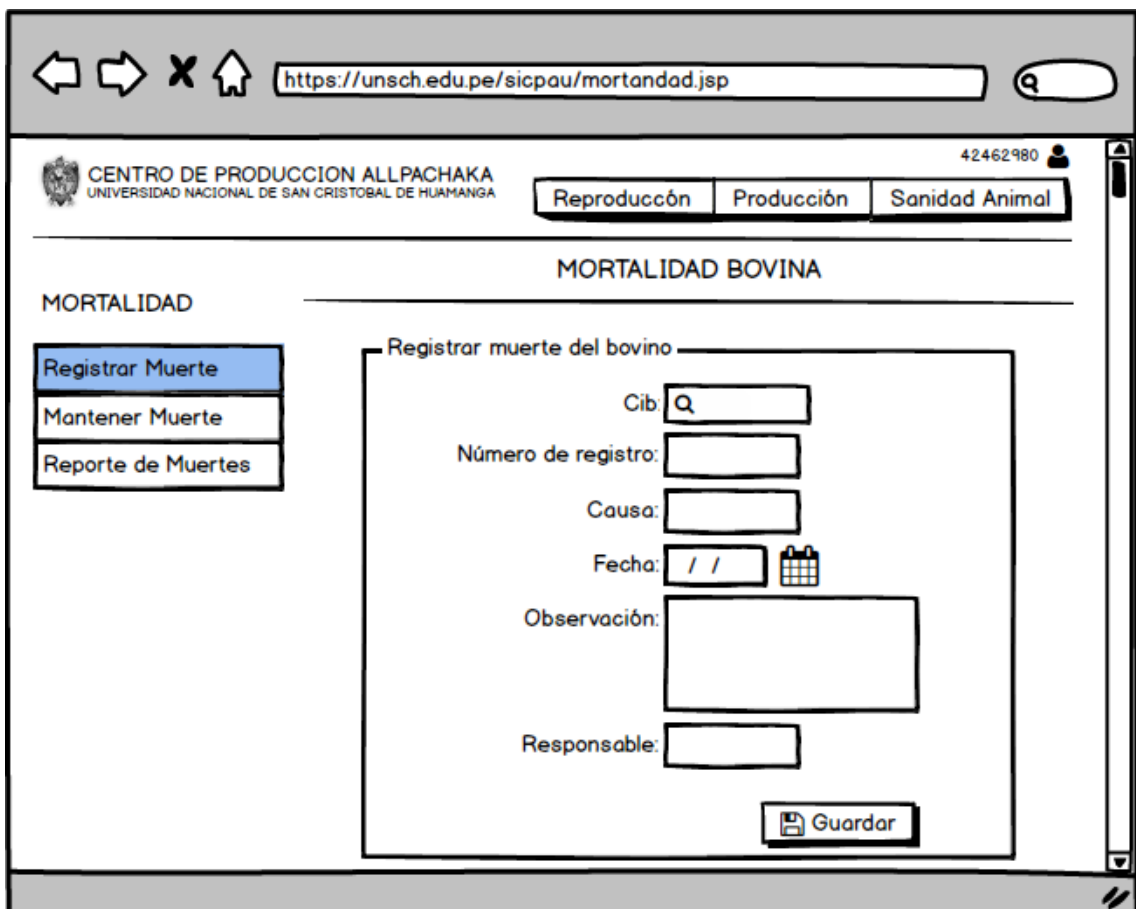


Figura N° 4.59: Interfaz Registrar Muerte del Bovino.

https://unsch.edu.pe/sicpau/mortalidad.jsp

CENTRO DE PRODUCCION ALLPACHAKA
UNIVERSIDAD NACIONAL DE SAN CRISTOBAL DE HUAMANGA

42462980

Reproducción Producción Sanidad Animal

MORTALIDAD

MORTALIDAD BOVINA

Registrar Muerte

Mantener Muerte

Reporte de Muertes

Cib. Q

REGISTRO DE MORTANDAD				
Número de registro	Causa	Fecha	Observación	Responsable

Guardar

Figura N° 4.60: Interfaz Mantener Muerte del Bovino.

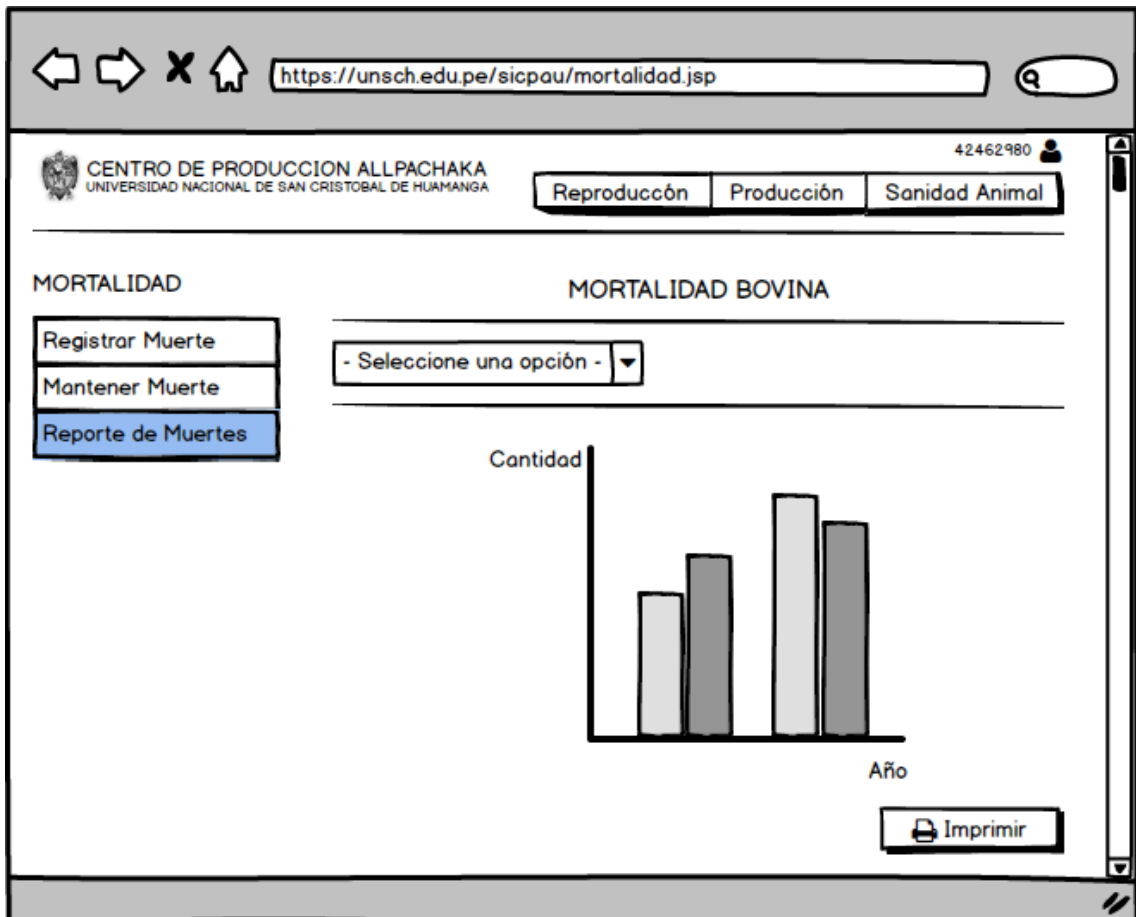


Figura N° 4.61: Interfaz Reporte de Muertes.

The screenshot shows a web form titled 'Administrar Cuenta de Usuario'. At the top center, there is an icon of a detective wearing a hat and a suit. Below the icon, there is a label 'Usuario Administrador' next to a small person icon. Underneath this label is a text input field. Below the first input field, there is a label 'Contraseña' next to a magnifying glass icon. Underneath this label is another text input field. At the bottom right of the form, there is an 'Acceder' button.

Figura N° 4.62: Interfaz Administrar Cuenta de Usuario.

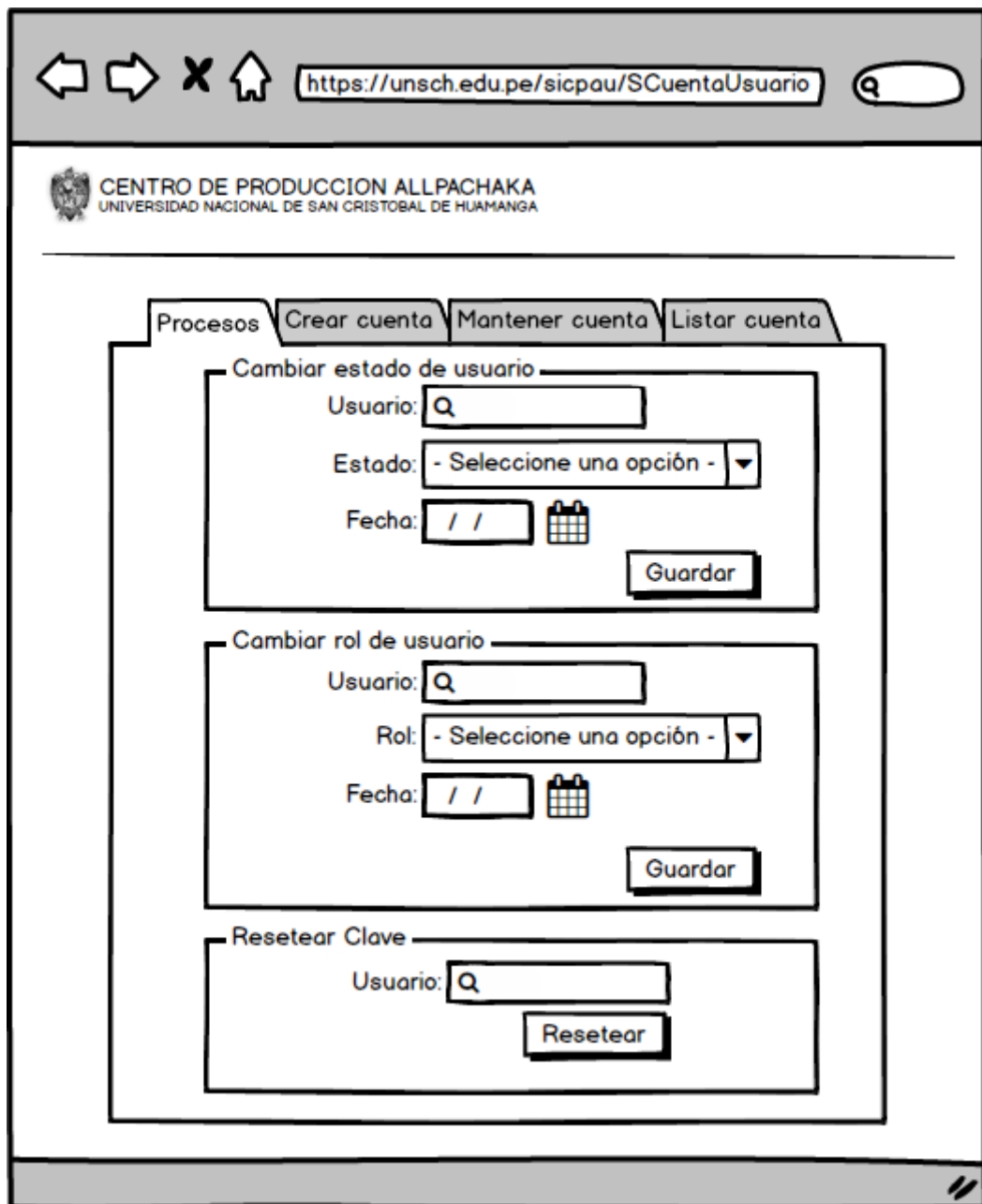


Figura N° 4.63: Interfaz Procesos de Cuenta de Usuario.

Procesos **Crear cuenta** Mantener cuenta Listar cuenta

Datos del usuario

DNI:

Nombre:

Apellido Paterno:

Apellido Materno:

Teléfono:

e-mail:

Cargo:

Datos de la cuenta de usuario

Usuario:

Clave:

Confirmar clave:

Fecha:

Estado:

Rol:

Guardar

Figura N° 4.64: Interfaz Crear Cuenta de Usuario.

Procesos | Crear cuenta | Mantener cuenta | Listar cuenta

Usuario:

DNI:

Nombre:

Apellido Paterno:

Apellido Materno:

Teléfono:

e-mail:

Cargo:

Figura N° 4.65: Interfaz Mantener Cuenta de Usuario.

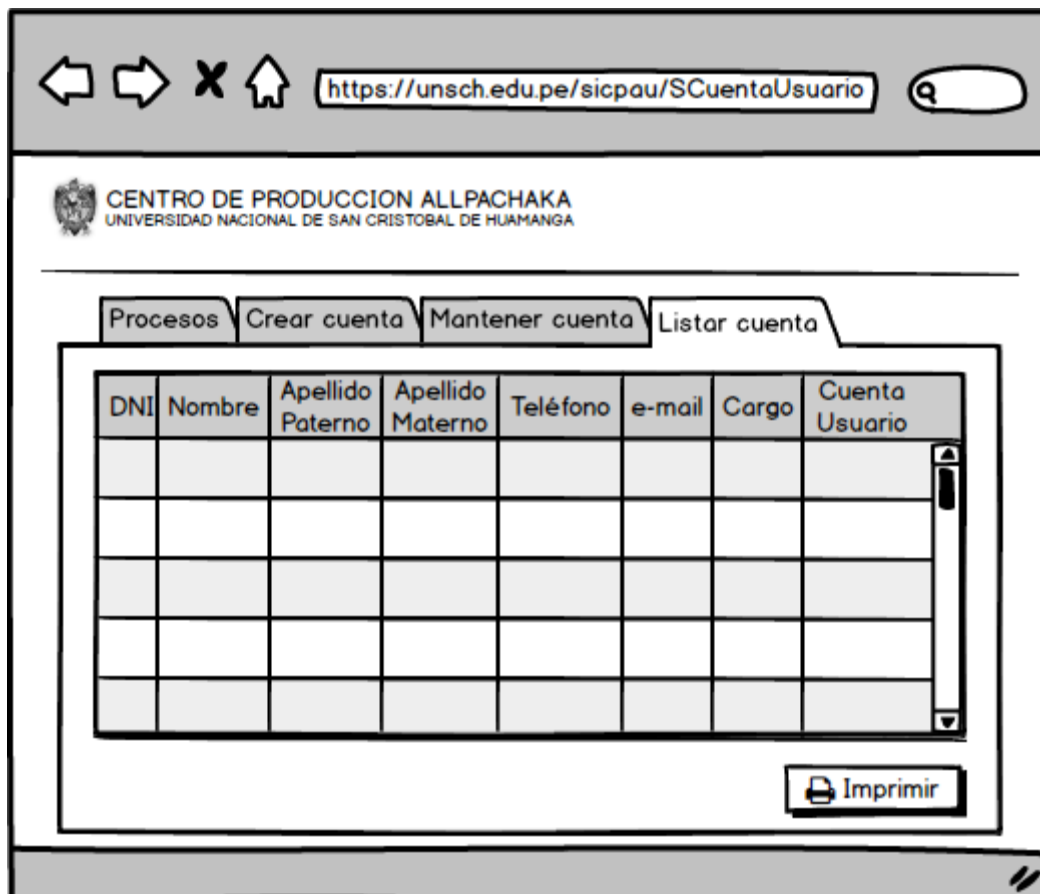


Figura N° 4.66: Interfaz Reporte de Cuentas de Usuario.

TARJETAS CLASE RESPONSABILIDAD Y COLABORACIÓN (CRC)

Para un diseño simple, se procede a definir tarjetas CRC, que permitan identificar las clases, sus responsabilidades y las colaboraciones que realiza para una historia de usuario y las tareas correspondientes, según se desarrolla en el capítulo II, sección 2.2.2

CLASE: ALTA	
<p>Responsabilidades: Registrar alta Mantener alta Emitir reporte de alta</p>	<p>Colaboradores: Bovino Productor_pecuario</p> <p>Atributos: numero_registro procedencia certificado fecha precio cib</p>

	dni
--	-----

Tabla N° 4.93: Clase Alta.

CLASE: BOVINO	
<p>Responsabilidades: Registrar bovino Mantener datos del bovino Emitir reporte sobre el bovino</p>	<p>Colaboradores: Bovino Celo Servicio Parto Aborto Crecimiento_desarrollo Destete Libido Morfologia_semental Morfologia_vaca Campania Estado_nutricional Semen Sanidad Alta Saca Muerte Aparato_reproductor</p> <p>Atributos: cib cib_padre cib_madre nombre fecha_nacimiento raza sexo fotografia</p>

Tabla N° 4. 94: Clase Bovino.

CLASE: CELO	
<p>Responsabilidades: Registrar celo Mantener celo</p>	<p>Colaboradores: Bovino Servicio</p> <p>Atributos: id_celo cib_vaca fecha hora</p>

Tabla N° 4.94: Clase Celo.

CLASE: SERVICIO	
<p>Responsabilidades: Registrar servicio Mantener servicio</p>	<p>Colaboradores: Bovino Celo Pajilla</p> <p>Atributos: id_servicio id_celo cib_vaca cib_semental código_pajilla fecha hora resultado método_servicio dni</p>

Tabla N° 4.95: Clase Servicio.

CLASE: PARTO	
<p>Responsabilidades: Registrar parto Mantener parto</p>	<p>Colaboradores: Bovino Servicio</p> <p>Atributos: id_parto id_servicio cib_madre cib_ternero dificultad placenta_retenida edema_ubre hipocalcemia observacion</p>

Tabla N° 4.96: Clase Parto.

CLASE: ABORTO	
<p>Responsabilidades: Registrar aborto Mantener parto Emitir reporte sobre parto</p>	<p>Colaboradores: Bovino Servicio</p> <p>Atributos: id_aborto cib_vaca id_servicio fecha</p>

	causa agente observacion
--	--------------------------------

Tabla N° 4.97: Clase Aborto.

CLASE: CUENTA_USUARIO	
Responsabilidades: Iniciar sesión Cerrar sesion	Colaboradores: Productor_pecuario Administra_cuenta_usuario Atributos: dni usuario clave

Tabla N° 4.98: Clase Cuenta de Usuario.

CLASE: MORFOLOGIA_VACA	
Responsabilidades: Registrar morfología de la vaca Mantener morfología de la vaca	Colaboradores: Bovino Atributos: id_morf_vaca cib fecha estatura profundidad_corporal angulosidad ancho_pecho ancho_grupa angulo_grupa vista_lateral_pt vista_posterior_pt angulo_podal ligamento_delantero altura_ubre ancho_ubre profundidad_ubre

Tabla N° 4.99: Clase Morfología de la Vaca.

CLASE: MORFOLOGIA_SEMENTAL	
Responsabilidades: Registrar morfología del semental Mantener morfología del semental	Colaboradores: Bovino Atributos:

	id_morf_sem cib fecha estatura profundidad_corporal angulosidad ancho_pecho ancho_grupa angulo_grupa vista_lateral_pt vista_posterior_pt angulo_podal
--	--

Tabla N° 4.100: Clase Morfología del Semental.

CLASE: LIBIDO	
Responsabilidades: Registrar libido Mantener libido	Colaboradores: Bovino Atributos: id_libido cib fecha olfateo_genital lameteo_genital lameteo_social apoyo_cabeza aparejo flehmen intento_monta monta_incompleta servicio

Tabla N° 4.101: Clase Libido.

CLASE: CRECIMIENTO_DESARROLLO	
Responsabilidades: Registrar crecimiento y desarrollo Mantener crecimiento y desarrollo	Colaboradores: Bovino Atributos: id_crec_des cib fecha peso perímetro_toracico altura_cruz condición_corporal destete observacion

Tabla N° 4.102: Clase Crecimiento y Desarrollo.

CLASE: ESTADO_NUTRICIONAL_REPRODUCTORA	
<p>Responsabilidades: Registrar estado nutricional de la reproductora Mantener estado nutricional de la reproductora</p>	<p>Colaboradores: Bovino</p> <p>Atributos: id_est_nut_rep cib fecha estado condición_corporal peso</p>

Tabla N° 4.103: Clase Estado Nutricional de la Reproductora.

CLASE: SECA	
<p>Responsabilidades: Registrar seca Mantener seca Emitir reporte de seca</p>	<p>Colaboradores: Produccion_leche</p> <p>Atributos: numero_campania cib fecha</p>

Tabla N° 4.104: Clase Seca.

CLASE: PRODUCCION_LECHE	
<p>Responsabilidades: Registrar produccionde leche Mantener producción de leche Reporte de producción de leche</p>	<p>Colaboradores: Bovino Seca</p> <p>Atributos: numero_campania cib fecha horario cantidad</p>

Tabla N° 4.105: Clase Producción de Leche.

CLASE: PROPIEDAD_LECHE	
<p>Responsabilidades: Registrar propiedades de la leche Mantener propiedades de la leche</p>	<p>Colaboradores: Produccion_leche</p>

Reporte de las propiedades de la leche	Atributos: id_propiedad_leche numero_campania cib proteína grasa ufc rcs lactosa
--	---

Tabla N° 4.106: Clase Propiedades de la Leche.

CLASE: APARATO_REPRODUCTOR	
Responsabilidades: Registrar características físicas del aparato reproductor Mantener características físicas del aparato reproductor	Colaboradores: Bovino Atributos: id_apa_rep cib fecha circunferencia_escrotal circunferencia_testiculo forma_testiculo posición_escroto consistencia_escroto

Tabla N° 4.107: Clase Aparato Reproductor del Ternero.

CLASE: CUALIDAD_SEMEN	
Responsabilidades: Registrar cualidades del semen Mantener cualidades del semen	Colaboradores: Bovino Pajilla Atributos: id_cualidad_semen fecha_colectada volumen pH motilidad_masal motilidad_individual concentración_espermatoca anormalidades viabilidad_espermatoca

Tabla N° 4.108: Clase Cualidades del Semen.

CLASE: PAJILLA

<p>Responsabilidades: Registrar colección de semen</p>	<p>Colaboradores: Inventario_pajilla Semen Servicio</p> <p>Atributos: codigo_pajilla id_cualidad_semen volumen codigo_partida numero_registro procedencia</p>
---	---

Tabla N° 4.109: Clase Pajilla.

CLASE: INVENTARIO_PAJILLA	
<p>Responsabilidades: Registrar salida de pajilla Ver inventario Mantener inventario</p>	<p>Colaboradores: Pajilla</p> <p>Atributos: id_inv_paj codigo_pajilla fecha glosa cantidad_entrada cantidad_salida tanque canastilla procedencia destino precio</p>

Tabla N° 4.110: Clase Inventario de Pajilla.

CLASE: MORTALIDAD	
<p>Responsabilidades: Registrar muerte de bovino Mantener muerte de bovino Reporte de mortalidad</p>	<p>Colaboradores: Bovino Productor_pecuario</p> <p>Atributos: id_mortalidad numero_registro causa fecha observación cib dni</p>

Tabla N° 4.111: Clase Mortalidad Bovina.

CLASE: SACA	
<p>Responsabilidades: Registrar saca Mantener saca Reporte de saca</p>	<p>Colaboradores: Bovino Productor_pecuario</p> <p>Atributos: id_saca numero_registro motivo destino fecha precio cib dni</p>

Tabla N° 4.112: Clase Saca.

CLASE: SANIDAD	
<p>Responsabilidades: Registrar tratamiento sanitario Mantener tratamiento sanitario</p>	<p>Colaboradores: Bovino Productor_pecuario</p> <p>Atributos: id_sanidad fecha_inicio fecha_finalizacion tipo_registro nombre_tipo_registro tratamiento_utilizado dias comentario cib dni</p>

Tabla N° 4.113: Clase Sanidad.

CLASE: PRODUCTOR_PECUARIO	
<p>Responsabilidades: Registrar datos del productor pecuario</p>	<p>Colaboradores: Alta Servicio Cuenta_usuario Sanidad Saca Mortalidad</p> <p>Atributos: dni nombre</p>

	ap_paterno ap_materno teléfono email cargo
--	--

Tabla N° 4.114: Clase Productor Pecuario.

CLASE: ADMINISTRAR_CUENTA_USUARIO	
Responsabilidades: Cambiar estado de cuenta de usuario Cambiar rol de usuario Resetar clave Crear cuenta de usuario Mantener cuenta de usuario Listar cuenta de usuario	Colaboradores: Cuenta_usuario Atributos: fecha estado rol responsable dni

Tabla N° 4.115: Clase Administrar Cuenta de Usuario.

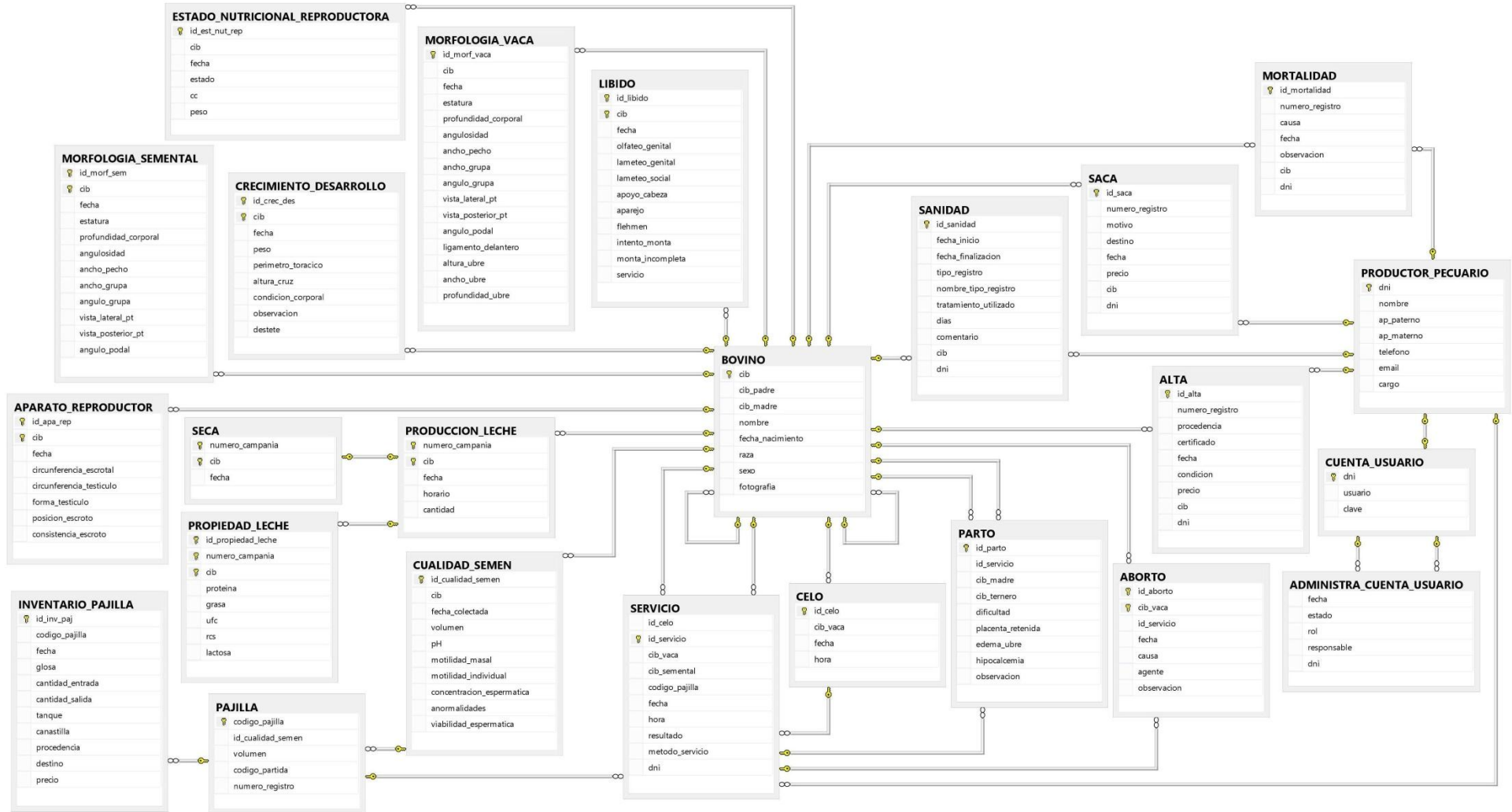


Figura N° 4.67: Esquema Físico de la Base de Datos sobre la Reproducción de Ganado Bovino para Monitorear la Producción Láctea.

CÓDIGO FUENTE

La tabla 4.116. Muestra el código fuente de la configuración del archivo XML y conexión a la base de datos para brindar información operativa sobre eventos reproductivos y productivos de ganado bovino; las tablas 4.117, 4.119, 4.121, 4.123, 4.125, 4.127, 4.129 y 4.131 muestran el código fuente para las clases servlet "registrar alta", "registrar celo", "registrar servicio", "registrar bovino", "registrar parto", "registrar aborto", "servicio concepción" y "login" respectivamente y las tablas 4.118, 4.120, 4.122, 4.124, 4.126, 4.128, 4.130 y 4.132 muestran el código fuente para las clases dao "alta", "celo", "servicio", "bovino", "parto", "aborto", "servicio concepcion" y "login" respectivamente.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd">
  <context-param>
    <param-name>protocolo</param-name>
    <param-value>jdbc:sqlserver://</param-value>
  </context-param>
  <context-param>
    <param-name>host</param-name>
    <param-value>127.0.0.1:1433</param-value>
  </context-param>
  <context-param>
    <param-name>sid</param-name>
    <param-value>databaseName=DBSICPAU</param-value>
  </context-param>
  <context-param>
    <param-name>user</param-name>
    <param-value>sa</param-value>
  </context-param>
  <context-param>
    <param-name>password</param-name>
    <param-value>pwsqserver</param-value>
  </context-param>
  <context-param>
    <param-name>driver</param-name>
    <param-value>com.microsoft.sqlserver.jdbc.SQLServerDriver</param-
value>
  </context-param>
</web-app>
package DS;
```

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import javax.servlet.ServletContext;

public class DBA {
    private Connection connection = null;
    private String driver = null;
    private String url = null;
    private String protocolo = null;
    private String host = null;
    private String sid = null;
    private String user = null;
    private String password = null;

    public DBA(ServletContext servletContext) {
        driver = servletContext.getInitParameter("driver");
        protocolo = servletContext.getInitParameter("protocolo");
        host = servletContext.getInitParameter("host");
        sid = servletContext.getInitParameter("sid");
        user = servletContext.getInitParameter("user");
        password = servletContext.getInitParameter("password");

        url = protocolo + host + ";" + sid;
    }

    public Connection getConexion(){
        try {
            Class.forName(driver).newInstance();
            connection = DriverManager.getConnection(url, user, password);
        } catch (ClassNotFoundException | InstantiationException |
IllegalAccessException | SQLException e) {
            System.out.println("Error en data source: " + e);
        }
        return connection;
    }

    public void closeConnection(Connection connection){
        try {
            if (connection != null) {
                connection.close();
                connection = null;
            }
            this.connection = connection;
        } catch (SQLException e) {
            System.out.println("Error FuenteDatos -> closeConnection" + e);
        }
    }
}

```

Tabla N° 4.116: Código fuente Configuración XML y Conexión a la Base de Datos.

```

package Servlets;

import Beans.BAlta;
import Beans.BBovino;
import DAO.DAlta;
import DAO.DBovino;
import DS.DBA;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.MultipartConfig;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author Nilo
 */
@WebServlet("/SRegAlta")
@MultipartConfig(maxFileSize = 16177215)
public class SRegistrarAlta extends HttpServlet {
    protected void processRequest(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        String msj, dni = (String) request.getSession().getAttribute("dni");
        if (dni == null) {
            msj = "No se registró, inicie sesión antes de guardar";
        } else {
            ServletContext servletContext =
this.getServletConfig().getServletContext();
            DBA dba = new DBA(servletContext);
            Connection connection = dba.getConexion();

            BBovino bBovino = new BBovino();
            bBovino.setCib(request.getParameter("codigo").substring(0, 8));

            DBovino dBovino = new DBovino(connection);
            if (dBovino.verificarCodigo(bBovino)) {
                bBovino.setCib(request.getParameter("codigo"));
                bBovino.setCibMadre(null);
                bBovino.setCibPadre(null);
                bBovino.setNombre(request.getParameter("nombre"));
                try {
                    java.util.Date fechaNacimiento = new SimpleDateFormat("yyyy-

```

```

MM-dd").parse(request.getParameter("fechaNacimiento"));
        bBovino.setFechaNacimiento(new
java.sql.Date(fechaNacimiento.getTime()));
    } catch (ParseException e) {
        System.out.println("Error en la fecha de nacimiento -->> " + e);
    }
    bBovino.setRaza(request.getParameter("raza"));
    bBovino.setSexo(request.getParameter("sexo"));

    bBovino.setFotografia(request.getPart("fotografia"));

    dba = new DBA(servletContext);
    connection = dba.getConnection();

    dBovino = new DBovino(connection);
    if (dBovino.registrarBovino(bBovino) == 0) {
        System.out.println("No se registró los datos del bovino");
    } else {
        System.out.println("Se registró los datos del bovino");
    }

    BAlta bAlta = new BAlta();
    bAlta.setNumero_registro(request.getParameter("numero_registro"));
    bAlta.setProcedencia(request.getParameter("procedencia"));
    bAlta.setCertificado(request.getPart("certificado"));
    try {
        java.util.Date fechaAlta = new SimpleDateFormat("yyyy-MM-
dd").parse(request.getParameter("fechaAlta"));
        bAlta.setFechaAlta(new java.sql.Date(fechaAlta.getTime()));
    } catch (ParseException e) {
        System.out.println("Error en la fecha de alta -->> " + e);
    }
    bAlta.setCondicion(request.getParameter("condicion"));
    bAlta.setCib(request.getParameter("codigo"));
    bAlta.setDni(dni);

    dba = new DBA(servletContext);
    connection = dba.getConnection();

    DAlta dAlta = new DAlta(connection);
    if (dAlta.registrarAlta(bAlta) == 0) {
        System.out.println("No se registró la alta del bovino");
    } else {
        System.out.println("Se registró la alta del bovino");
    }
    msj = "Se registró con éxito";
} else {
    msj = "[Error], el orden de inclusión del bovino en ese año ya fue
registrado";
}
}
}

```

```

        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            out.println(msj);
        }
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
    public String getServletInfo() {
        return "Short description";
    }
}

```

Tabla N° 4.117: Código Fuente Servlet Registrar Alta.

```

package DAO;

import Beans.BAlta;
import java.io.InputStream;
import java.sql.Connection;
import java.sql.PreparedStatement;

/**
 *
 * @author Nilo
 */
public class DAlta {

    private Connection connection = null;

    public DAlta(Connection connection) {
        this.connection = connection;
    }

    public int registrarAlta(BAlta bAlta) {

```

```

InputStream inputStream = null;
int nroRegistros = 0;
String sql = "INSERT INTO alta(numero_registro, procedencia, certificado,
fecha, condicion, cib, dni) VALUES (?, ?, ?, ?, ?, ?, ?)";
PreparedStatement preparedStatement;
try {
    preparedStatement = connection.prepareStatement(sql);
    preparedStatement.setString(1, bAlta.getNumero_registro());
    preparedStatement.setString(2, bAlta.getProcedencia());
    if (bAlta.getCertificado() != null) {
        inputStream = bAlta.getCertificado().getInputStream();
    }
    preparedStatement.setBlob(3, inputStream);
    preparedStatement.setDate(4, bAlta.getFechaAlta());
    preparedStatement.setString(5, bAlta.getCondicion());
    preparedStatement.setString(6, bAlta.getCib());
    preparedStatement.setString(7, bAlta.getDni());
    nroRegistros = preparedStatement.executeUpdate();

    preparedStatement.close();
    connection.close();
} catch (Exception e) {
    System.out.println("Error al registrar el alta: " + e);
}
return nroRegistros;
}
}

```

Tabla N° 4.118: Código Fuente DAO Alta.

```

package Servlets;

import Beans.BCelo;
import DAO.DCelo;
import DS.DBA;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.MultipartConfig;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

```



```

/**
 *
 * @author Nilo
 */
@WebServlet("/SRegCelo")
@MultipartConfig(maxFileSize = 16177215)
public class SRegistrarCelo extends HttpServlet {
    protected void processRequest(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        String msj, dni = (String) request.getSession().getAttribute("dni");
        if (dni == null) {
            msj = "¡No se registró, inicie sesión antes de guardar!";
        } else {
            BCelo bCelo = new BCelo();
            bCelo.setCib_vaca(request.getParameter("cibVaca"));
            try {
                java.util.Date fechaNacimiento = new SimpleDateFormat("yyyy-MM-
                dd").parse(request.getParameter("fechaCelo"));
                bCelo.setFecha(new java.sql.Date(fechaNacimiento.getTime()));
            } catch (ParseException e) {
                e.printStackTrace();
                msj = "[ERROR] [Error] verifique la fecha ingresada: " + e;
            }

            SimpleDateFormat simpleDateFormatHora = new
            SimpleDateFormat("HH:mm");
            String strHora = request.getParameter("horaCelo");
            Date dateHora = null;
            try {
                dateHora = simpleDateFormatHora.parse(strHora);
                bCelo.setHora(new java.sql.Time(dateHora.getTime()));
            } catch (ParseException e) {
                e.printStackTrace();
                msj = "[ERROR] verifique la hora ingresada: " + e;
            }

            ServletContext servletContext =
            this.getServletConfig().getServletContext();
            DBA dba = new DBA(servletContext);
            Connection connection = dba.getConexion();

            DCelo dCelo = new DCelo(connection);
            if (dCelo.registrarCelo(bCelo) == 1) {
                msj = "¡Se registró con éxito!";
            } else {
                msj = "¡No se registró, ocurrió un error!";
            }
        }
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {

```

```

        out.println(msj);
    }
}

@Override
protected void doGet(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    processRequest(request, response);
}

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    processRequest(request, response);
}

@Override
public String getServletInfo() {
    return "Short description";
}
}

```

Tabla N° 4.119: Código Fuente Servlet, Registrar Celo.

```

package DAO;

import Beans.BCelo;
import java.sql.Connection;
import java.sql.PreparedStatement;

/**
 *
 * @author Nilo
 */
public class DCelo {

    private Connection connection;

    public DCelo(Connection connection) {
        this.connection = connection;
    }

    public int registrarCelo(BCelo bCelo) {
        int nroRegistros = 0;
        String sql = "INSERT INTO CELO (cib_vaca, fecha, hora) VALUES (?, ?,

```

```

    ?)";
    PreparedStatement preparedStatement;
    try {
        preparedStatement = connection.prepareStatement(sql);

        preparedStatement.setString(1, bCelo.getCib_vaca());
        preparedStatement.setDate(2, bCelo.getFecha());
        preparedStatement.setTime(3, bCelo.getHora());

        nroRegistros = preparedStatement.executeUpdate();
        preparedStatement.close();
        connection.close();
    } catch (Exception e) {
        System.out.println("Error al registrar el celo: " + e);
    }
    return nroRegistros;
}
}

```

Tabla Nº 4.120: Código Fuente DAO Celo.

```

package Servlets;

import Beans.BServicio;
import DAO.DServicio;
import DS.DBA;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.MultipartConfig;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author Nilo
 */
@WebServlet("/SRegServicio")
@MultipartConfig(maxFileSize = 16177215)
public class SRegistrarServicio extends HttpServlet {

```

```

protected void processRequest(HttpServletRequest request,
HttpServletRequest response) throws ServletException, IOException {
    String msj, dni = (String) request.getSession().getAttribute("dni");
    if (dni == null) {
        msj = "[ERROR], ¡NO SE REGISTRÓ, inicie sesión antes de guardar!";
    } else {
        BServicio bServicio = new BServicio();
        bServicio.setCib_vaca(request.getParameter("cibVaca"));
        bServicio.setMetodo_servicio(request.getParameter("metodo"));
        if ("".equals(request.getParameter("cibSemental"))) {
            bServicio.setCib_semental(null);
        } else {
            bServicio.setCib_semental(request.getParameter("cibSemental"));
        }
        if ("".equals(request.getParameter("codigoPajilla"))) {
            bServicio.setCodigo_pajilla(null);
        } else {
            bServicio.setCodigo_pajilla(request.getParameter("codigoPajilla"));
        }
        if ("".equals(request.getParameter("dni"))) {
            bServicio.setDni(null);
        } else {
            bServicio.setDni(request.getParameter("dni"));
        }
        try {
            java.util.Date fechaNacimiento = new SimpleDateFormat("yyyy-MM-
dd").parse(request.getParameter("fechaServicio"));
            bServicio.setFecha(new java.sql.Date(fechaNacimiento.getTime()));
        } catch (ParseException e) {
            msj = "[ERROR], " + e;
        }
        SimpleDateFormat simpleDateFormatHora = new
SimpleDateFormat("HH:mm");
        String strHora = request.getParameter("horaServicio");
        Date dateHora = null;
        try {
            dateHora = simpleDateFormatHora.parse(strHora);
            bServicio.setHora(new java.sql.Time(dateHora.getTime()));
        } catch (ParseException e) {
            msj = "[ERROR], " + e;
        }

        ServletContext servletContext =
this.getServletConfig().getServletContext();
        DBA dba = new DBA(servletContext);
        Connection connection = dba.getConexion();

        DServicio dServicio = new DServicio(connection);
        if (dServicio.existeCelo(bServicio)) {
            if (dServicio.existeRegistroServicio(bServicio)) {

```

```

        msj = "[ERROR], Existe un registro de servicio con esta fecha";
    } else {
        if (dServicio.registrarServicio(bServicio) == 1) {
            msj = "¡Se registró con éxito!";
        } else {
            msj = "[ERROR], ¡No se registró!";
        }
    }
} else {
    msj = "[ERROR], no existe registro de celo que corresponda a este
servicio";
}
}
response.setContentType("text/html;charset=UTF-8");
try (PrintWriter out = response.getWriter()) {
    out.println(msj);
}
}
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    processRequest(request, response);
}

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    processRequest(request, response);
}

@Override
public String getServletInfo() {
    return "Short description";
}
}
}

```

Tabla Nº 4.121: Código Fuente Servlet Registrar Servicio.

```

package DAO;

import Beans.BBovino;
import Beans.BServicio;
import com.google.gson.Gson;
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.Date;
import java.sql.PreparedStatement;

```

```

import java.sql.ResultSet;
import java.util.Calendar;

import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.LinkedHashMap;
import java.util.Map;

/**
 *
 * @author Nilo
 */
public class DServicio {
    private Connection connection;

    public DServicio(Connection connection) {
        this.connection = connection;
    }

    public int registrarServicio(BServicio bServicio) {
        int nroRegistros = 0;
        String sql = "INSERT INTO SERVICIO (id_celo, cib_vaca, cib_semental,
codigo_pajilla, fecha, hora, metodo_servicio, dni) VALUES (?, ?, ?, ?, ?, ?, ?, ?)";
        PreparedStatement preparedStatement;
        try {
            preparedStatement = connection.prepareStatement(sql);

            preparedStatement.setInt(1, bServicio.getId_celo());
            preparedStatement.setString(2, bServicio.getCib_vaca());
            preparedStatement.setString(3, bServicio.getCib_semental());
            preparedStatement.setString(4, bServicio.getCodigo_pajilla());
            preparedStatement.setDate(5, bServicio.getFecha());
            preparedStatement.setTime(6, bServicio.getHora());
            preparedStatement.setString(7, bServicio.getMetodo_servicio());
            preparedStatement.setString(8, bServicio.getDni());

            nroRegistros = preparedStatement.executeUpdate();
            preparedStatement.close();
            connection.close();
        } catch (Exception e) {
            System.out.println("Error en DaoServicio -> al registrar servicio: " + e);
        }
        return nroRegistros;
    }

    public boolean existeCelo(BServicio bServicio){
        boolean existe = false;
        Calendar calendar = Calendar.getInstance();
        calendar.setTime(bServicio.getFecha()); // Configuramos la fecha que se
recibe
        calendar.add(Calendar.DAY_OF_YEAR, -1);
    }

```

```

String sql = "SELECT * FROM CELO WHERE cib_vaca = ? AND fecha
BETWEEN ? AND ?";
PreparedStatement preparedStatement;
ResultSet resultSet;
try {
    preparedStatement = connection.prepareStatement(sql);
    preparedStatement.setString(1, bServicio.getCib_vaca());
    preparedStatement.setDate(2, new
java.sql.Date(calendar.getTime().getTime()));
    preparedStatement.setDate(3, bServicio.getFecha());
    resultSet = preparedStatement.executeQuery();
    while (resultSet.next()) {
        bServicio.setId_celo(resultSet.getInt("id_celo"));
        existe = true;
    }
} catch (Exception e) {
    System.out.println("Error en DaoServicio -> método existeCelo()" + e);
}
return existe;
}

public boolean existeRegistroServicio(BServicio bServicio){
    boolean existe = false;
    Calendar calendar = Calendar.getInstance();
    calendar.setTime(bServicio.getFecha()); // Configuramos la fecha que se
recibe
    calendar.add(Calendar.DAY_OF_YEAR, -1);
    String sql = "SELECT * FROM SERVICIO WHERE cib_vaca = ? AND fecha
BETWEEN ? AND ?";
    PreparedStatement preparedStatement;
    ResultSet resultSet;
    try {
        preparedStatement = connection.prepareStatement(sql);
        preparedStatement.setString(1, bServicio.getCib_vaca());
        preparedStatement.setDate(2, new
java.sql.Date(calendar.getTime().getTime()));
        preparedStatement.setDate(3, bServicio.getFecha());
        resultSet = preparedStatement.executeQuery();
        while (resultSet.next()) {
            existe = true;
        }
    } catch (Exception e) {
        System.out.println("Error en DaoServicio -> método
existeRegistroServicio()" + e);
    }
    return existe;
}

public void loadFechaServicio(BBovino bBovino){
    String sp_sql = "{call sp_loadfechaservicio(?)}";

```

```

CallableStatement callableStatement;
ResultSet resultSet;
try {
    callableStatement = connection.prepareCall(sp_sql);
    callableStatement.setString(1, bBovino.getCibMadre());
    callableStatement.execute();
    resultSet = callableStatement.getResultSet();
    Map<String, String> list = new LinkedHashMap<String, String>();
    while (resultSet.next()) {
        Date date = resultSet.getDate("fecha");
        DateFormat fecha = new SimpleDateFormat("dd-MM-yyyy");
        String fechaConvertido = fecha.format(date);
        list.put(Integer.toString(resultSet.getInt("id_servicio")),
fechaConvertido);
    }
    String json = new Gson().toJson(list);
    bBovino.setJson(json);

    resultSet.close();
    callableStatement.close();
    connection.close();
} catch (Exception e) {
    System.out.println("Error en DaoServicio -> loadFechaServicio" + e);
}
}

public int registrarDiagnosticoPreniez(BServicio bServicio){
    int nroRegistros = 0;
    String sql = "UPDATE SERVICIO SET resultado = ? WHERE cib_vaca = ? AND
id_servicio = ?";
    PreparedStatement preparedStatement;
    try {
        preparedStatement = connection.prepareStatement(sql);
        preparedStatement.setString(1, bServicio.getResultado());
        preparedStatement.setString(2, bServicio.getCib_vaca());
        preparedStatement.setInt(3, bServicio.getId_servicio());

        nroRegistros = preparedStatement.executeUpdate();
        preparedStatement.close();
        connection.close();
    } catch (Exception e) {
        System.out.println("Error en DaoServicio -> al registrar servicio: " + e);
    }
    return nroRegistros;
}

public void loadFechaServicioPositivo(BBovino bBovino){
    String sp_sql = "{call sp_loadfechaserviciopositivo(?)}";
    CallableStatement callableStatement;
    ResultSet resultSet;
    try {

```



```

callableStatement = connection.prepareCall(sp_sql);
callableStatement.setString(1, bBovino.getCibMadre());
callableStatement.execute();
resultSet = callableStatement.getResultSet();
Map<String, String> list = new LinkedHashMap<String, String>();
while (resultSet.next()) {
    Date date = resultSet.getDate("fecha");
    DateFormat fecha = new SimpleDateFormat("dd-MM-yyyy");
    String fechaConvertido = fecha.format(date);
    list.put(Integer.toString(resultSet.getInt("id_servicio")),
fechaConvertido);
}
String json = new Gson().toJson(list);
bBovino.setJson(json);

resultSet.close();
callableStatement.close();
connection.close();
} catch (Exception e) {
    System.out.println("Error en DaoServicio -> loadFechaServicio" + e);
}
}
}

```

Tabla N° 4.122: Código Fuente DAO Servicio.

```

package Servlets;

import Beans.BBovino;
import DAO.DBovino;
import DS.DBA;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.MultipartConfig;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author Nilo
 */
@WebServlet("/SRegBovino")

```

```

@MultipartConfig(maxFileSize = 16177215)
public class SRegistrarBovino extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
        String msj, dni = (String) request.getSession().getAttribute("dni");
        if (dni == null) {
            msj = "No se registró, inicie sesión antes de guardar";
        } else {
            ServletContext servletContext =
this.getServletConfig().getServletContext();
            DBA dba = new DBA(servletContext);
            Connection connection = dba.getConexion();

            BBovino bBovino = new BBovino();
            bBovino.setCib(request.getParameter("cibTernero").substring(0, 8));

            DBovino dBovino = new DBovino(connection);
            if (dBovino.verificarCodigo(bBovino)) {
                bBovino.setCib(request.getParameter("cibTernero"));
                bBovino.setCibPadre(request.getParameter("cibPadre"));
                bBovino.setCibMadre(request.getParameter("cibMadre"));
                bBovino.setNombre(request.getParameter("nombre"));
                try {
                    java.util.Date fechaNacimiento = new SimpleDateFormat("yyyy-
MM-dd").parse(request.getParameter("fechaNacimiento"));
                    bBovino.setFechaNacimiento(new
java.sql.Date(fechaNacimiento.getTime()));
                } catch (ParseException e) {
                    System.out.println("Error en la fecha de nacimiento -->> " + e);
                }
                bBovino.setRaza(request.getParameter("raza"));
                bBovino.setSexo(request.getParameter("sexo"));
                // Obtiene la parte de archivo de carga en esta petición de partes
múltiples
                bBovino.setFotografia(request.getPart("fotografia"));

                dba = new DBA(servletContext);
                connection = dba.getConexion();

                dBovino = new DBovino(connection);
                if (dBovino.registrarBovino(bBovino) == 0) {
                    msj = "[Error], No se registró los datos del bovino";
                } else {
                    msj = "Se registró con éxito";
                }
            } else {
                msj = "[Error], el orden de inclusión del bovino en ese año ya fue
registrado";
            }
        }
    }
}

```

```

        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            out.println(msj);
        }
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
    public String getServletInfo() {
        return "Short description";
    }
}

```

Tabla N° 4.123: Código Fuente Servlet Registrar Bovino.

```

package DAO;

import Beans.BBovino;
import com.google.gson.Gson;
import java.io.InputStream;
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author Nilo
 */
public class DBovino {

    private Connection connection = null;

    public DBovino(Connection connection) {
        this.connection = connection;
    }
}

```

```

}

public int registrarBovino(BBovino bBovino) {
    InputStream inputStream = null;
    int nroRegistros = 0;
    String sql = "INSERT INTO bovino(cib, cib_padre, cib_madre, nombre,
fecha_nacimiento, raza, sexo, fotografia) VALUES (?, ?, ?, ?, ?, ?, ?, ?)";
    PreparedStatement preparedStatement;

    try {
        preparedStatement = connection.prepareStatement(sql);
        preparedStatement.setString(1, bBovino.getCib());
        preparedStatement.setString(2, bBovino.getCibPadre());
        preparedStatement.setString(3, bBovino.getCibMadre());
        preparedStatement.setString(4, bBovino.getNombre());
        preparedStatement.setDate(5, bBovino.getFechaNacimiento());
        preparedStatement.setString(6, bBovino.getRaza());
        preparedStatement.setString(7, bBovino.getSexo());
        if (bBovino.getFotografia() != null) {
            // Obtiene la secuencia de entrada del archivo de carga
            inputStream = bBovino.getFotografia().getInputStream();
        }
        preparedStatement.setBlob(8, inputStream);
        nroRegistros = preparedStatement.executeUpdate();

        preparedStatement.close();
        connection.close();
    } catch (Exception e) {
        System.out.println("Error al registrar el bovino: " + e);
    }
    return nroRegistros;
}

public void consultarCodigo(BBovino bBovino) {
    String sql = "SELECT TOP 1 cib FROM bovino WHERE cib LIKE '" +
bBovino.getCodAnio() + "%' ORDER BY cib DESC;";
    PreparedStatement preparedStatement;
    ResultSet resultSet;
    try {
        preparedStatement = connection.prepareStatement(sql);
        resultSet = preparedStatement.executeQuery();
        while (resultSet.next()) {
            bBovino.setCib(resultSet.getString("cib"));
        }

        resultSet.close();
        preparedStatement.close();
        connection.close();
    } catch (Exception e) {
        System.out.println("Error en DaoBovino -> consultarCodigo: " + e);
    }
}

```

```

}

public boolean verificarCodigo(BBovino bBovino) {
    String sql = "SELECT cib FROM bovino WHERE cib LIKE " + bBovino.getCib()
+ "_";
    PreparedStatement preparedStatement;
    ResultSet resultSet;
    boolean resultado = true;
    try {
        preparedStatement = connection.prepareStatement(sql);
        resultSet = preparedStatement.executeQuery();
        while (resultSet.next()) {
            resultado = false;
        }

        resultSet.close();
        preparedStatement.close();
        connection.close();
    } catch (Exception e) {
        System.out.println("Error en DaoBovino -> verificarCodigo" + e);
    }
    return resultado;
}

public void listarCib(BBovino bBovino) {
    String sp_sql = "{call sp_listarCib(?,?)}";
    CallableStatement callableStatement;
    ResultSet resultSet;
    try {
        callableStatement = connection.prepareCall(sp_sql);
        callableStatement.setString(1, bBovino.getKeyword());
        callableStatement.setString(2, bBovino.getSexo());
        callableStatement.execute();
        resultSet = callableStatement.getResultSet();

        List<String> list = new ArrayList<String>();
        while (resultSet.next()) {
            list.add(resultSet.getString("cib").trim());
        }
        String json = new Gson().toJson(list);
        bBovino.setJson(json);

        resultSet.close();
        callableStatement.close();
        connection.close();
    } catch (Exception e) {
        System.out.println("Error en DaoBovino -> listarCib" + e);
    }
}

public void listarCibCria(BBovino bBovino){

```

```

String sp_sql = "{call sp_loadcibcria(?)}";
CallableStatement callableStatement;
ResultSet resultSet;
try {
    callableStatement = connection.prepareCall(sp_sql);
    callableStatement.setString(1, bBovino.getCibMadre());
    callableStatement.execute();
    resultSet = callableStatement.getResultSet();
    List<String> list = new ArrayList<String>();
    while (resultSet.next()) {
        list.add(resultSet.getString("cib").trim());
    }
    String json = new Gson().toJson(list);
    bBovino.setJson(json);

    resultSet.close();
    callableStatement.close();
    connection.close();
} catch (Exception e) {
    System.out.println("Error en DaoBovino -> listarCibCria" + e);
}
}

public void listarCibSemental(BBovino bBovino){
String sp_sql = "{call sp_loadcibsemental(?)}";
CallableStatement callableStatement;
ResultSet resultSet;
try {
    callableStatement = connection.prepareCall(sp_sql);
    callableStatement.setString(1, bBovino.getCibMadre());
    callableStatement.execute();
    resultSet = callableStatement.getResultSet();
    List<String> list = new ArrayList<String>();
    while (resultSet.next()) {
        list.add(resultSet.getString("cib_padre").trim());
    }
    String json = new Gson().toJson(list);
    bBovino.setJson(json);

    resultSet.close();
    callableStatement.close();
    connection.close();
} catch (Exception e) {
    System.out.println("Error en DaoBovino -> listarCibSemental" + e);
}
}
}
}

```

Tabla N° 4.124: Código Fuente DAO Bovino.

```

package Servlets;

import Beans.BParto;
import DAO.DParto;
import DS.DBA;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.MultipartConfig;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author Nilo
 */
@WebServlet("/SRegParto")
@MultipartConfig(maxFileSize = 16177215)
public class SRegistrarParto extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        String msj, dni = (String) request.getSession().getAttribute("dni");
        if (dni == null) {
            msj = "No se registró, inicie sesión antes de guardar";
        } else {
            BParto bParto = new BParto();
            bParto.setCib_madre(request.getParameter("cibVaca"));
            bParto.setCib_ternero(request.getParameter("cibTernero"));
            bParto.setDificultad(Integer.parseInt(request.getParameter("dificultad")));
            bParto.setPlacenta_retenida(request.getParameter("placenta"));
            bParto.setEdema_ubre(request.getParameter("edema"));
            bParto.setHipocalcemia(request.getParameter("hipocalcemia"));
            if ("".equals(request.getParameter("observacion"))) {
                bParto.setObservacion(null);
            } else {
                bParto.setObservacion(request.getParameter("observacion"));
            }

            ServletContext servletContext =
                this.getServletContext().getServletContext();
            DBA dba = new DBA(servletContext);
            Connection connection = dba.getConexion();

            DParto dParto = new DParto(connection);
            dParto.registrarParto(bParto);
        }
    }
}

```

```

        msj = bParto.getMensaje();
    }
    response.setContentType("text/html;charset=UTF-8");
    try (PrintWriter out = response.getWriter()) {
        out.println(msj);
    }
}

@Override
protected void doGet(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    processRequest(request, response);
}

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    processRequest(request, response);
}

@Override
public String getServletInfo() {
    return "Short description";
}
}

```

Tabla N° 4.125: Código Fuente Servlet, Registrar Parto.

```

package DAO;

import Beans.BParto;
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.Types;

/**
 *
 * @author Nilo
 */
public class DParto {
    private Connection connection;

    public DParto(Connection connection) {
        this.connection = connection;
    }
}

```



```

public void registrarParto(BParto bParto){
    String sp_sql = "{call sp_registrarparto(?, ?, ?, ?, ?, ?, ?, ?)}";
    CallableStatement callableStatement;
    try {
        callableStatement = connection.prepareCall(sp_sql);
        callableStatement.setString(1, bParto.getCib_madre());
        callableStatement.setString(2, bParto.getCib_tenero());
        callableStatement.setInt(3, bParto.getDifficultad());
        callableStatement.setString(4, bParto.getPlacenta_retenida());
        callableStatement.setString(5, bParto.getEdema_ubre());
        callableStatement.setString(6, bParto.getHipocalcemia());
        callableStatement.setString(7, bParto.getObservacion());

        callableStatement.registerOutParameter(8, Types.NVARCHAR);

        callableStatement.execute();

        bParto.setMensaje(callableStatement.getString(8));

        callableStatement.close();
        connection.close();
    } catch (Exception e) {
        System.out.println("Error en DaoParto -> registrarParto" + e);
    }
}
}
}

```

Tabla N° 4.126: Código Fuente DAO Parto.

```

package Servlets;

import Beans.BAborto;
import DAO.DAborto;
import DS.DBA;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.MultipartConfig;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *

```

```

* @author Nilo
*/
@WebServlet("/SRegAborto")
@MultipartConfig(maxFileSize = 16177215)
public class SRegistrarAborto extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
        String msj, dni = (String) request.getSession().getAttribute("dni");
        if (dni == null) {
            msj = "[ERROR], ¡NO SE REGISTRÓ, inicie sesión antes de guardar!";
        } else {
            BAborto bAborto = new BAborto();
            bAborto.setCib_vaca(request.getParameter("cibVaca"));

            bAborto.setId_servicio(Integer.parseInt(request.getParameter("idServicio")));
            try {
                java.util.Date fechaNacimiento = new SimpleDateFormat("yyyy-MM-
                dd").parse(request.getParameter("fechaAborto"));
                bAborto.setFecha(new java.sql.Date(fechaNacimiento.getTime()));
            } catch (ParseException e) {
                msj = "[ERROR], " + e;
            }
            bAborto.setCausa(request.getParameter("causa"));
            bAborto.setAgente(request.getParameter("agente"));
            if ("".equals(request.getParameter("observacion"))) {
                bAborto.setObservacion(null);
            } else {
                bAborto.setObservacion(request.getParameter("observacion"));
            }

            ServletContext servletContext =
            this.getServletConfig().getServletContext();
            DBA dba = new DBA(servletContext);
            Connection connection = dba.getConexion();

            DAborto dAborto = new DAborto(connection);

            if (dAborto.registrarAborto(bAborto) == 1) {
                msj = "¡Se registró con éxito!";
            } else {
                msj = "[ERROR], ¡No se registró!";
            }
        }
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            out.println(msj);
        }
    }

    @Override

```

```

protected void doGet(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    processRequest(request, response);
}

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    processRequest(request, response);
}

@Override
public String getServletInfo() {
    return "Short description";
}
}

```

Tabla N° 4.127: Código Fuente Servlet Registrar Aborto.

```

package DAO;

import Beans.BAborto;
import java.sql.Connection;
import java.sql.PreparedStatement;

/**
 *
 * @author Nilo
 */
public class DAborto {
    private Connection connection;

    public DAborto(Connection connection) {
        this.connection = connection;
    }

    public int registrarAborto(BAborto bAborto){
        int nroRegistros = 0;
        String sql = "INSERT INTO ABORTO (cib_vaca, id_servicio, fecha, causa,
agente, observacion) VALUES (?, ?, ?, ?, ?, ?)";
        PreparedStatement preparedStatement;
        try {
            preparedStatement = connection.prepareStatement(sql);

            preparedStatement.setString(1, bAborto.getCib_vaca());
            preparedStatement.setInt(2, bAborto.getId_servicio());
            preparedStatement.setDate(3, bAborto.getFecha());

```

```

        preparedStatement.setString(4, bAborto.getCausa());
        preparedStatement.setString(5, bAborto.getAgente());
        preparedStatement.setString(6, bAborto.getObservacion());

        nroRegistros = preparedStatement.executeUpdate();

        preparedStatement.close();
        connection.close();
    } catch (Exception e) {
        System.out.println("Error al registrar el aborto: " + e);
    }
    return nroRegistros;
}
}
}

```

Tabla Nº 4.128: Código Fuente DAO Aborto.

```

package Servlets;

import Beans.BServicioConcepcion;
import DAO.DServicioConcepcion;
import DS.DBA;
import java.io.IOException;
import java.sql.Connection;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author Nilo
 */
public class SServicioConcepcion extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        String msj="", dni = (String) request.getSession().getAttribute("dni");
        if (dni == null) {
            msj = "[ERROR], ¡NO SE REGISTRÓ, inicie sesión antes de guardar!";
        } else {
            BServicioConcepcion bServicioConcepcion = new
            BServicioConcepcion();
            bServicioConcepcion.setCibVaca(request.getParameter("cibVaca"));
            bServicioConcepcion.setDni(dni);

            ServletContext servletContext =
            this.getServletConfig().getServletContext();

```

```

        DBA dba = new DBA(servletContext);
        Connection connection = dba.getConexion();

        DServicioConcepcion dServicioConcepcion = new
DServicioConcepcion(connection);

dServicioConcepcion.consultarServicioConcepcion(bServicioConcepcion);

        response.setContentType("application/json");
        response.setCharacterEncoding("UTF-8");
        response.getWriter().write(bServicioConcepcion.getJson());
        System.out.println("Lista de Maps: " + bServicioConcepcion.getJson());
    }
    System.out.println("Mensaje: " + msj);
}

@Override
protected void doGet(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    processRequest(request, response);
}

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    processRequest(request, response);
}

@Override
public String getServletInfo() {
    return "Short description";
}
}

```

Tabla N° 4.129: Código Fuente Servlet Servicio Concepción.

```

package DAO;

import Beans.BServicioConcepcion;
import com.google.gson.Gson;
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;

```

```

import java.util.LinkedHashMap;
import java.util.List;
import java.util.Map;

/**
 *
 * @author Nilo
 */
public class DServicioConcepcion {

    private Connection connection;

    public DServicioConcepcion(Connection connection) {
        this.connection = connection;
    }

    public void consultarServicioConcepcion(BServicioConcepcion
bServicioConcepcion) {
        String sp_sql = "{call sp_servicioConcepcion(?,?)}";
        CallableStatement callableStatement = null;
        PreparedStatement preparedStatement = null;
        Statement statement = null;
        ResultSet resultSet = null;
        try {
            callableStatement = connection.prepareCall(sp_sql);
            callableStatement.setString(1, bServicioConcepcion.getCibVaca());
            callableStatement.setString(2, bServicioConcepcion.getDni());
            callableStatement.execute();
            resultSet = callableStatement.getResultSet();

            String sql = "select fechaParto, sexoTernero, nombreTernero,
fechaCelo1, fechaCelo2, celoServicio1, celoServicio2, celoServicio3,
celoServicio4, dpc, dps, dv, nsc, ip from tempServicioConcepcion where
idTempServicioConcepcion = ?";

            preparedStatement = connection.prepareStatement(sql);
            preparedStatement.setString(1, bServicioConcepcion.getDni());
            resultSet = preparedStatement.executeQuery();

            List<Map> list = new ArrayList<Map>();
            Map<String, String> map = null;
            while (resultSet.next()) {
                map = new LinkedHashMap<String, String>();

                map.put("fechaParto", (resultSet.getString("fechaParto") == null) ? "-" :
resultSet.getString("fechaParto"));
                map.put("sexoTernero", (resultSet.getString("sexoTernero") == null) ? "-"
: resultSet.getString("sexoTernero"));
                map.put("nombreTernero", (resultSet.getString("nombreTernero") ==
null) ? "-" : resultSet.getString("nombreTernero"));
                map.put("fechaCelo1", (resultSet.getString("fechaCelo1") == null) ? "-"

```

```

: resultSet.getString("fechaCelo1"));
    map.put("fechaCelo2", (resultSet.getString("fechaCelo2") == null) ? "-"
: resultSet.getString("fechaCelo2"));
    map.put("celoServicio1", (resultSet.getString("celoServicio1") == null) ?
"-": resultSet.getString("celoServicio1"));
    map.put("celoServicio2", (resultSet.getString("celoServicio2") == null) ?
"-": resultSet.getString("celoServicio2"));
    map.put("celoServicio3", (resultSet.getString("celoServicio3") == null) ?
"-": resultSet.getString("celoServicio3"));
    map.put("celoServicio4", (resultSet.getString("celoServicio4") == null) ?
"-": resultSet.getString("celoServicio4"));
    map.put("dpc", (resultSet.getInt("dpc") == 0) ? "-" :
Integer.toString(resultSet.getInt("dpc")));
    map.put("dps", (resultSet.getInt("dps") == 0) ? "-" :
Integer.toString(resultSet.getInt("dps")));
    map.put("dv", (resultSet.getInt("dv") == 0) ? "-" :
Integer.toString(resultSet.getInt("dv")));
    map.put("nsc", (resultSet.getInt("nsc") == 0) ? "-" :
Integer.toString(resultSet.getInt("nsc")));
    map.put("ip", (resultSet.getInt("ip") == 0) ? "-" :
Integer.toString(resultSet.getInt("ip")));
    list.add(map);
}
String json = new Gson().toJson(list);
bServicioConcepcion.setJson(json);

sql = "delete tempCeloServicio where idTempCeloServicio = " +
bServicioConcepcion.getDni();
statement = null;
try {
    statement = connection.createStatement();
    statement.executeUpdate(sql);
} catch (Exception e) {
    System.out.println("[ERROR] limpiarCeloServicio -> " + e);
}

sql = "delete tempServicioConcepcion where
idTempServicioConcepcion = " + bServicioConcepcion.getDni();
statement = null;
try {
    statement = connection.createStatement();
    statement.executeUpdate(sql);
} catch (Exception e) {
    System.out.println("[ERROR] limpiarServicioConcepcion -> " + e);
}

} catch (Exception e) {
    System.out.println("Error en DaoServicioConcepcion ->
consultarServicioConcepcion" + e);
} finally {
    try {

```



```

import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 *
 * @author Nilo
 */
public class SLogin extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {

        HttpSession httpSession = request.getSession(true);

        BLogin bLogin = new BLogin();
        bLogin.setUsuario(request.getParameter("usuario"));
        bLogin.setClave(request.getParameter("clave"));

        ServletContext servletContext =
this.getServletContext().getServletContext();
        DBA dba = new DBA(servletContext);
        Connection connection = dba.getConexion();

        if (connection != null) {
            DLogin dLogin = new DLogin(connection);
            dLogin.login(bLogin);
            if (bLogin.isExiste_cuenta()) {
                httpSession.setAttribute("dni", bLogin.getDni());
                request.getRequestDispatcher("principal.jsp").forward(request,
response);

                // Creamos la carpeta temporal para el usuario
                String path = "D:/AppWeb/SICPAU/web/img/temporal";

                File directorio = new File(path + "/" + bLogin.getDni());
                httpSession.setAttribute("directorio", directorio);
                if (directorio.exists()) {
                    File[] ficheros = directorio.listFiles();
                    for (int i = 0; i < ficheros.length; i++) {
                        ficheros[i].delete();
                    }
                    directorio.delete();
                }
                directorio.mkdir();
            } else {
                request.setAttribute("mensaje_login", "El usuario o la contraseña son

```

```

incorrectos.");
        request.getRequestDispatcher("index.jsp").forward(request, response);
    }
} else {
    request.setAttribute("mensaje_login", "Servidor detenido.");
    request.getRequestDispatcher("index.jsp").forward(request, response);
}
}

@Override
protected void doGet(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    processRequest(request, response);
}

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    processRequest(request, response);
}

@Override
public String getServletInfo() {
    return "Short description";
}
}

```

Tabla N° 4.131: Código Fuente Servlet Login.

```

package DAO;

import Beans.BLogin;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

/**
 *
 * @author Nilo
 */
public class DLogin {

    private Connection connection = null;

    public DLogin(Connection connection) {
        this.connection = connection;
    }
}

```

```

public void login(BLogin bLogin) {
    String sql = "SELECT dni, clave FROM cuenta_usuario WHERE usuario = ?";
    PreparedStatement preparedStatement;
    ResultSet resultSet;
    try {
        preparedStatement = connection.prepareStatement(sql);
        preparedStatement.setString(1, bLogin.getUsuario());
        resultSet = preparedStatement.executeQuery();

        while (resultSet.next()) {
            if (resultSet.getString("clave").equals(bLogin.getClave())) {
                bLogin.setDni(resultSet.getString("dni"));
                bLogin.setExiste_cuenta(true);
            } else {
                bLogin.setExiste_cuenta(false);
            }
        }

        resultSet.close();
        preparedStatement.close();
        connection.close();
    } catch (Exception e) {
        System.out.println("[ERROR] DLogin - login" + e);
    }
}
}
}

```

Tabla N° 4.132: Código Fuente DAO Login.

PRUEBAS UNITARIAS

Las figuras 4.68, 4.69, 4.70, 4.71, 4.72 y 4.73 muestran el resultado satisfactorio para las pruebas unitarias "login", "registrar alta", "registrar bovino", "registrar celo", "registrar servicio", "registrar diagnóstico de preñez", "registrar parto" y "registrar aborto" respectivamente.

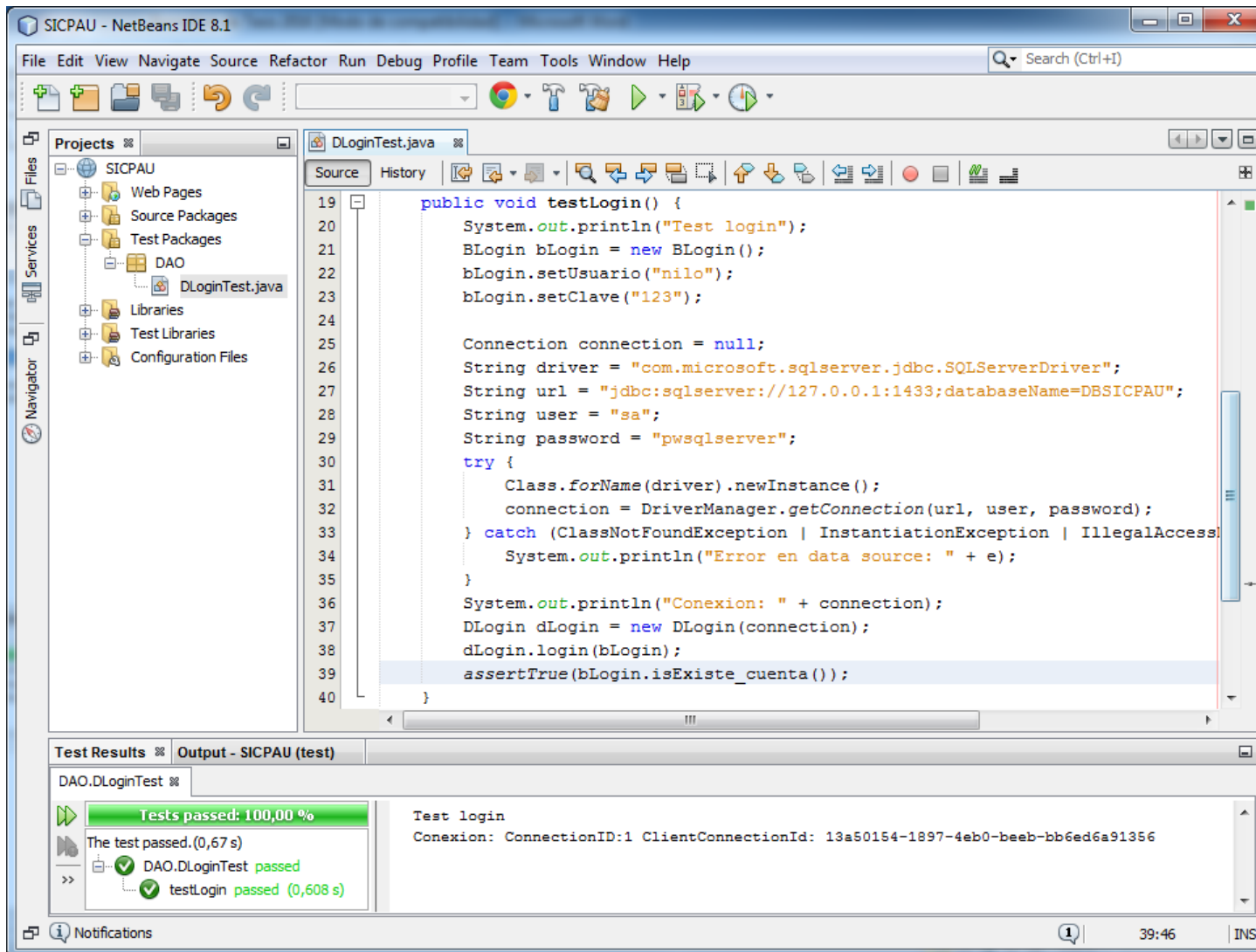


Figura N° 4.68: Prueba Unitaria Login.

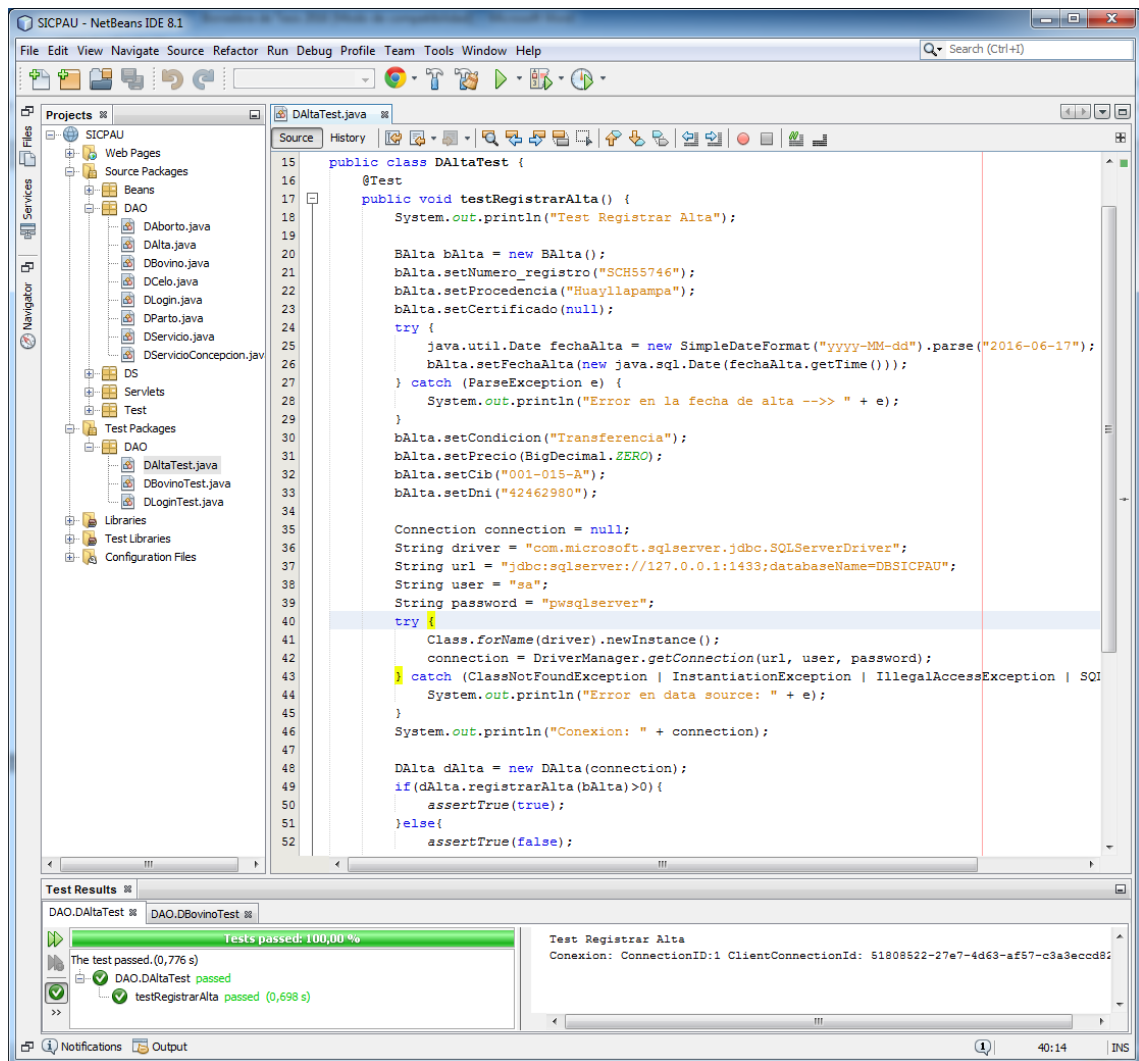


Figura N° 4.69: Prueba Unitaria Registrar Alta.

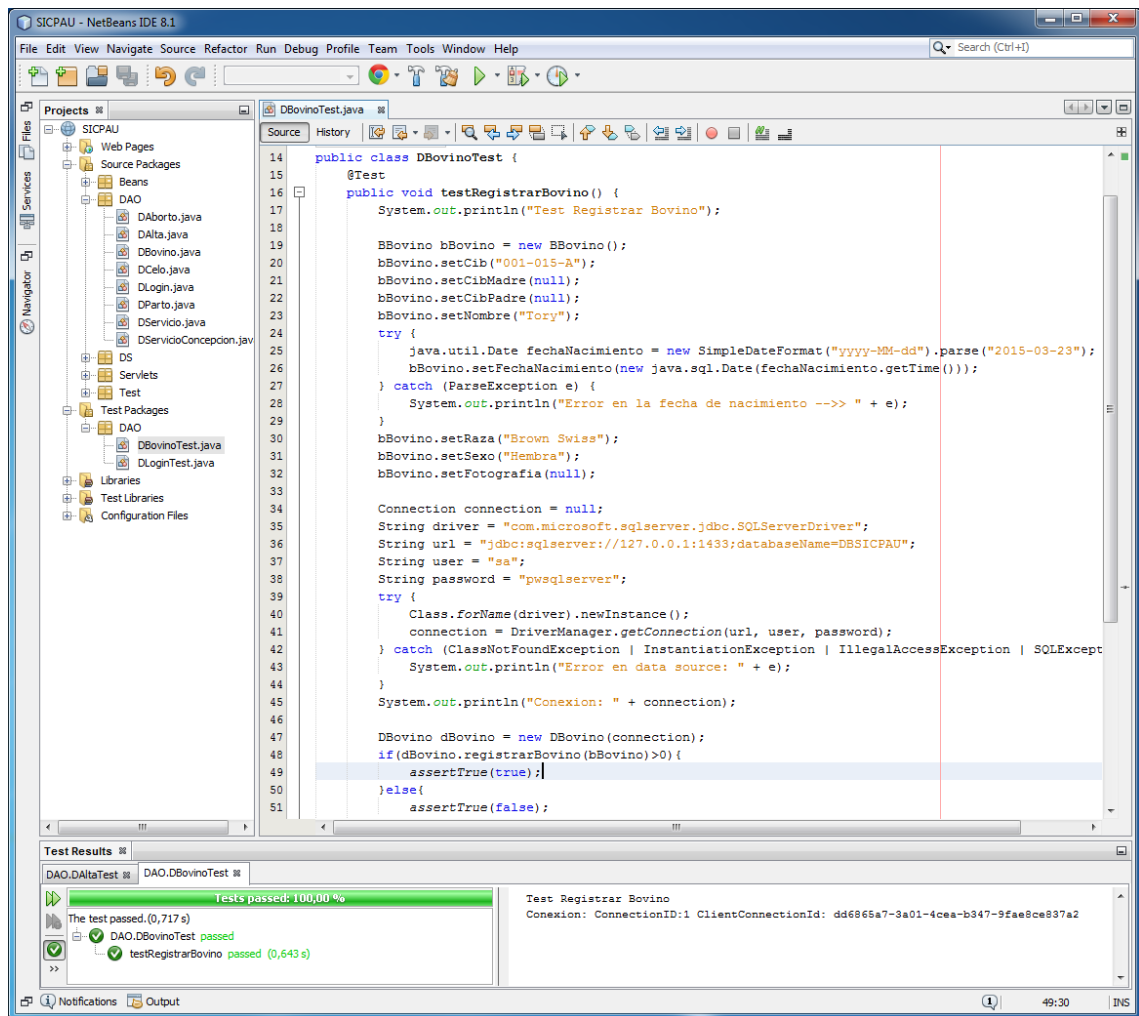


Figura N° 4.70: Prueba Unitaria Registrar Bovino.

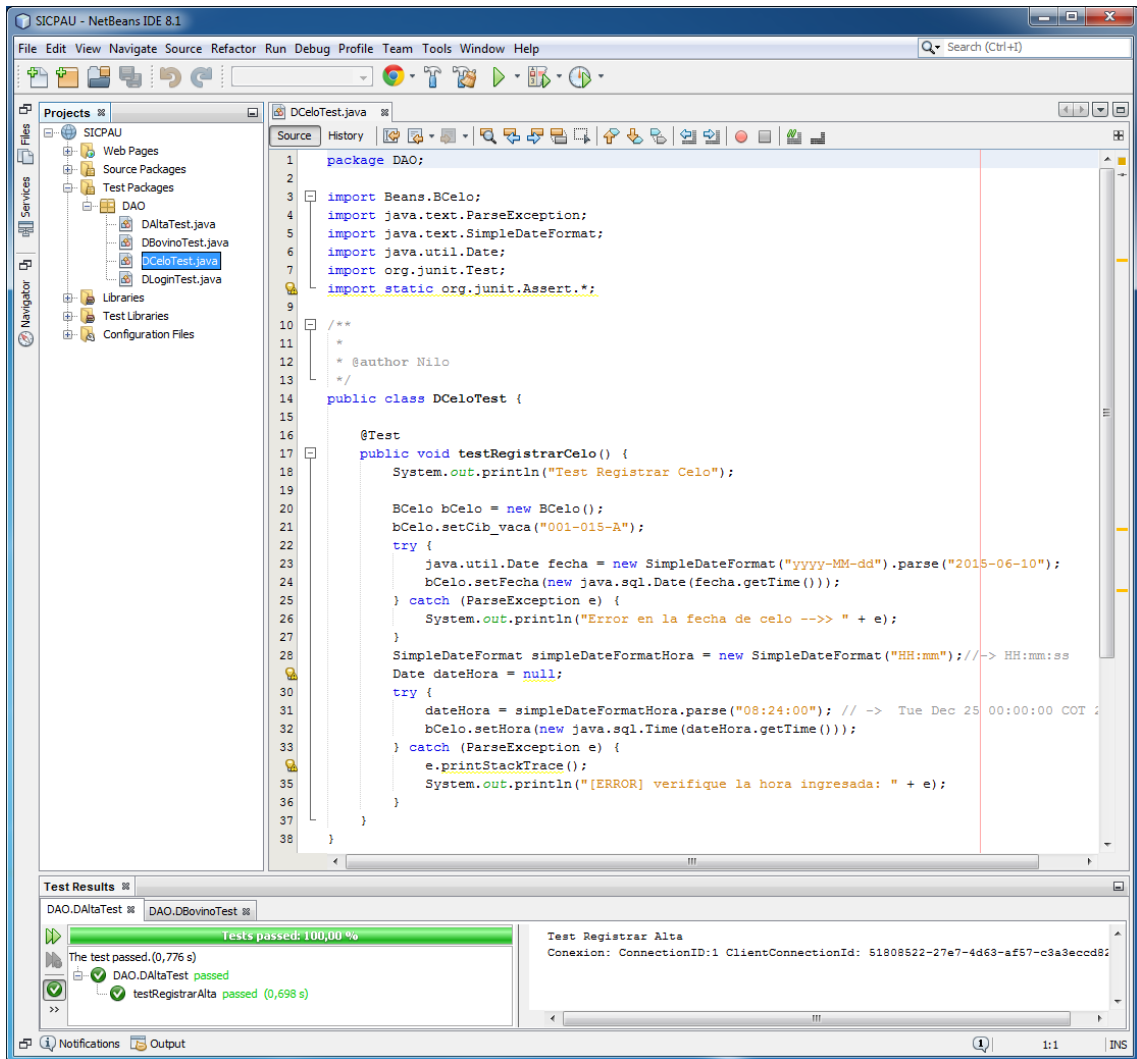


Figura N° 4.71: Prueba Unitaria Registrar Celos.

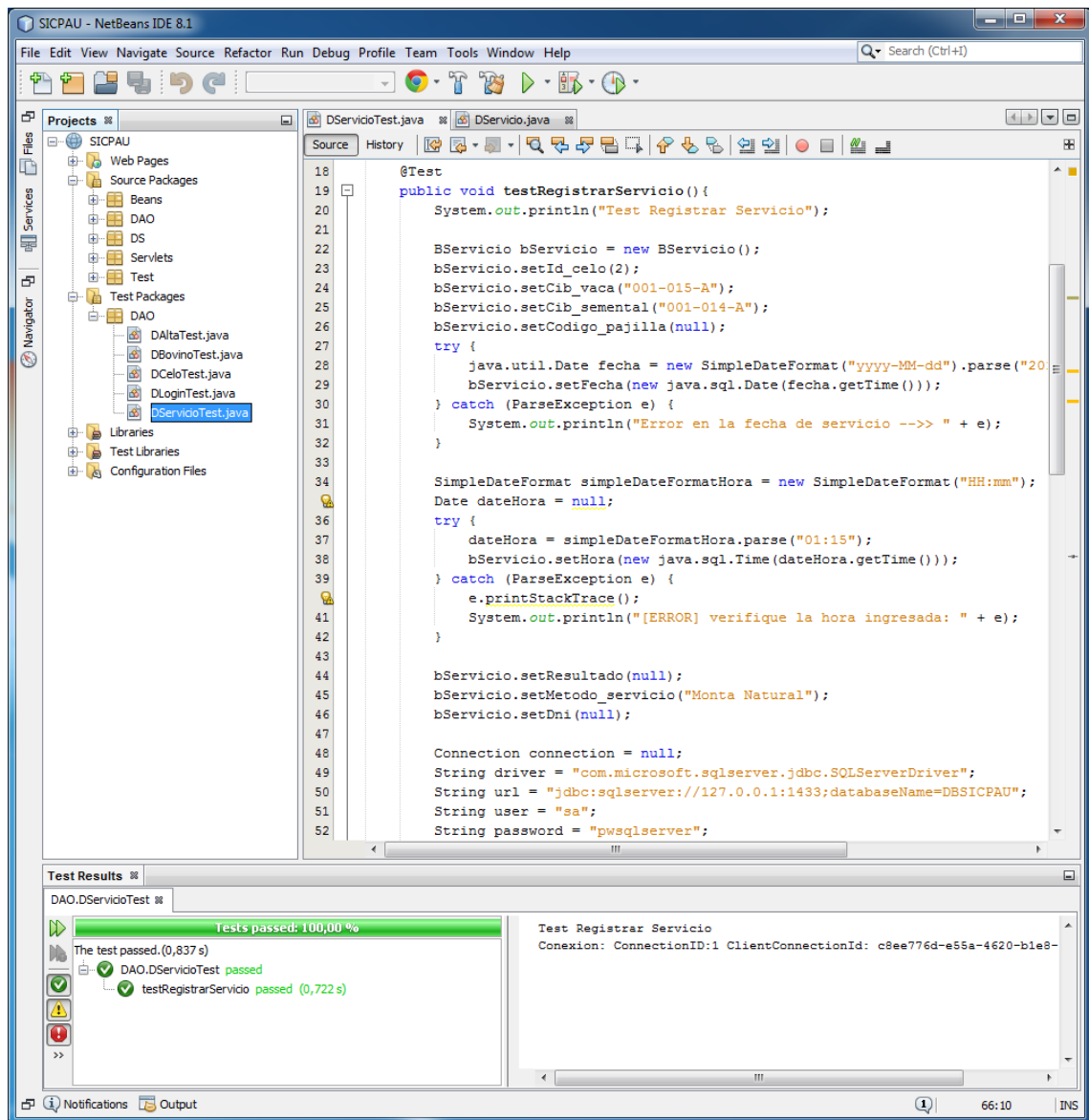


Figura N° 4.72: Prueba Unitaria Registrar Servicio.

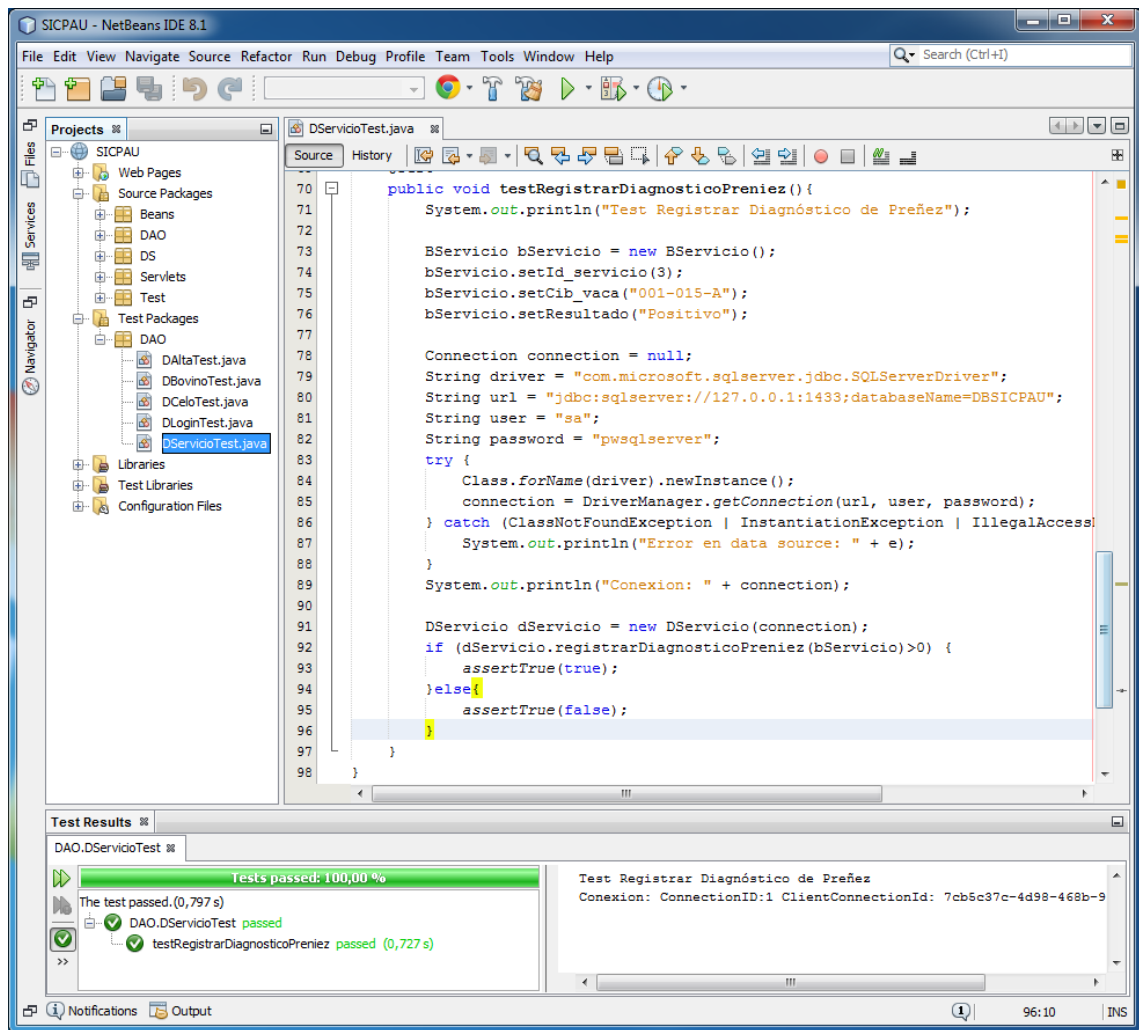


Figura N° 4.73: Prueba Unitaria Registrar Diagnostico de Preñez.

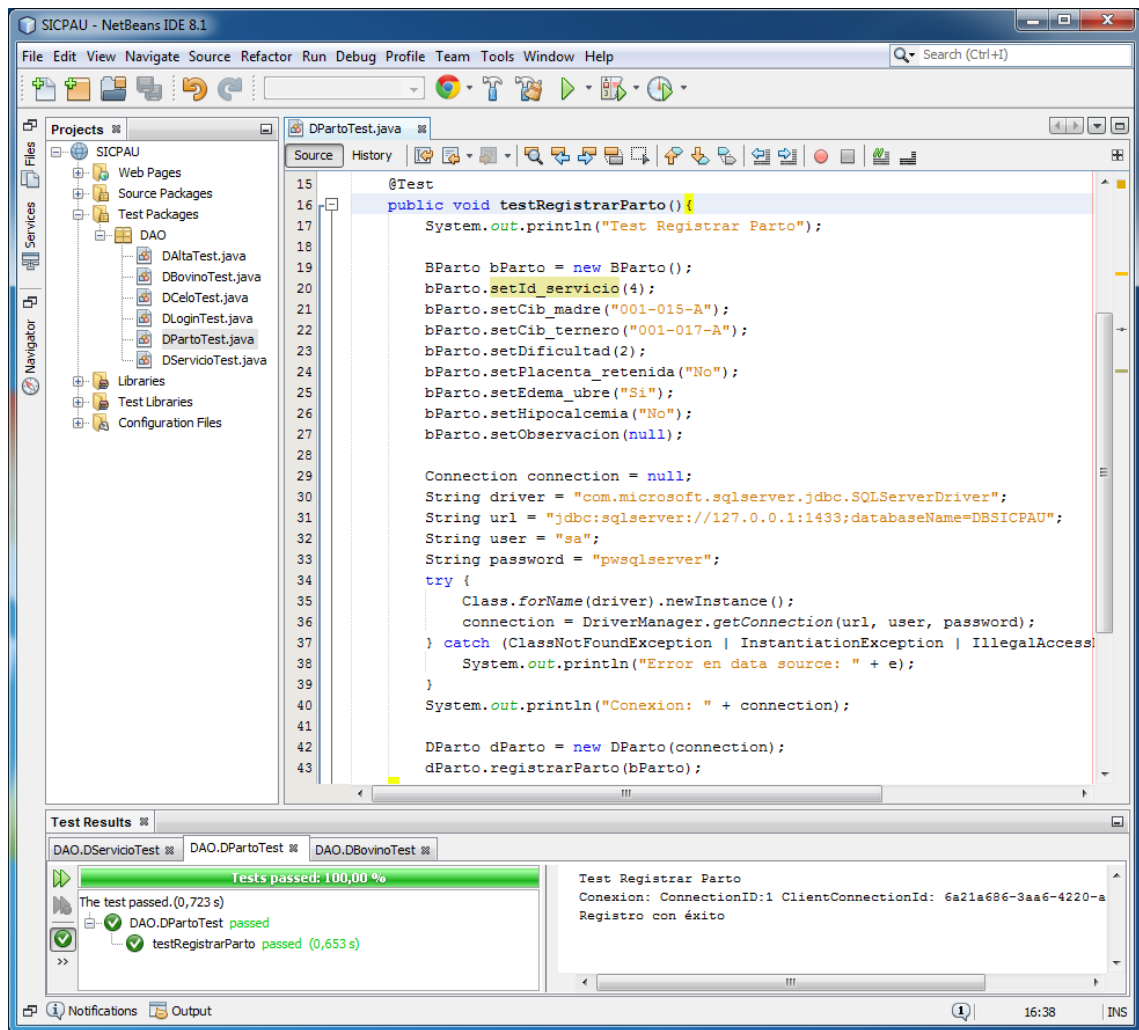


Figura N° 4.74: Prueba Unitaria Registrar Parto.

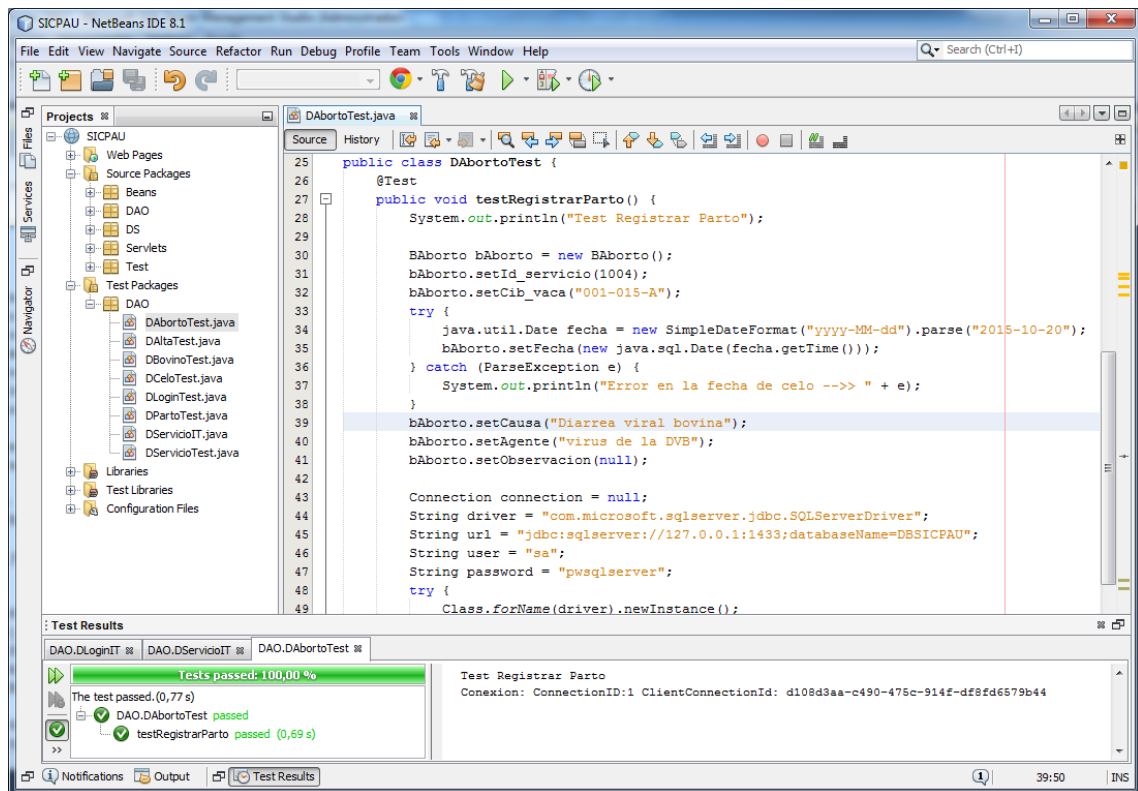


Figura N° 4.75: Prueba Unitaria Registrar Aborto.

PRUEBAS DE INTEGRACIÓN

Las figuras 4.76, 4.77, 4.78 y 4.79 muestran el resultado satisfactorio para las pruebas de integración de los métodos "registrar celo, registrar servicio, y registrar diagnóstico de preñez", "registrar alta y registrar bovino", "registrar bovino y registrar parto" y "registrar servicio, registrar diagnóstico de preñez y registrar aborto" respectivamente.

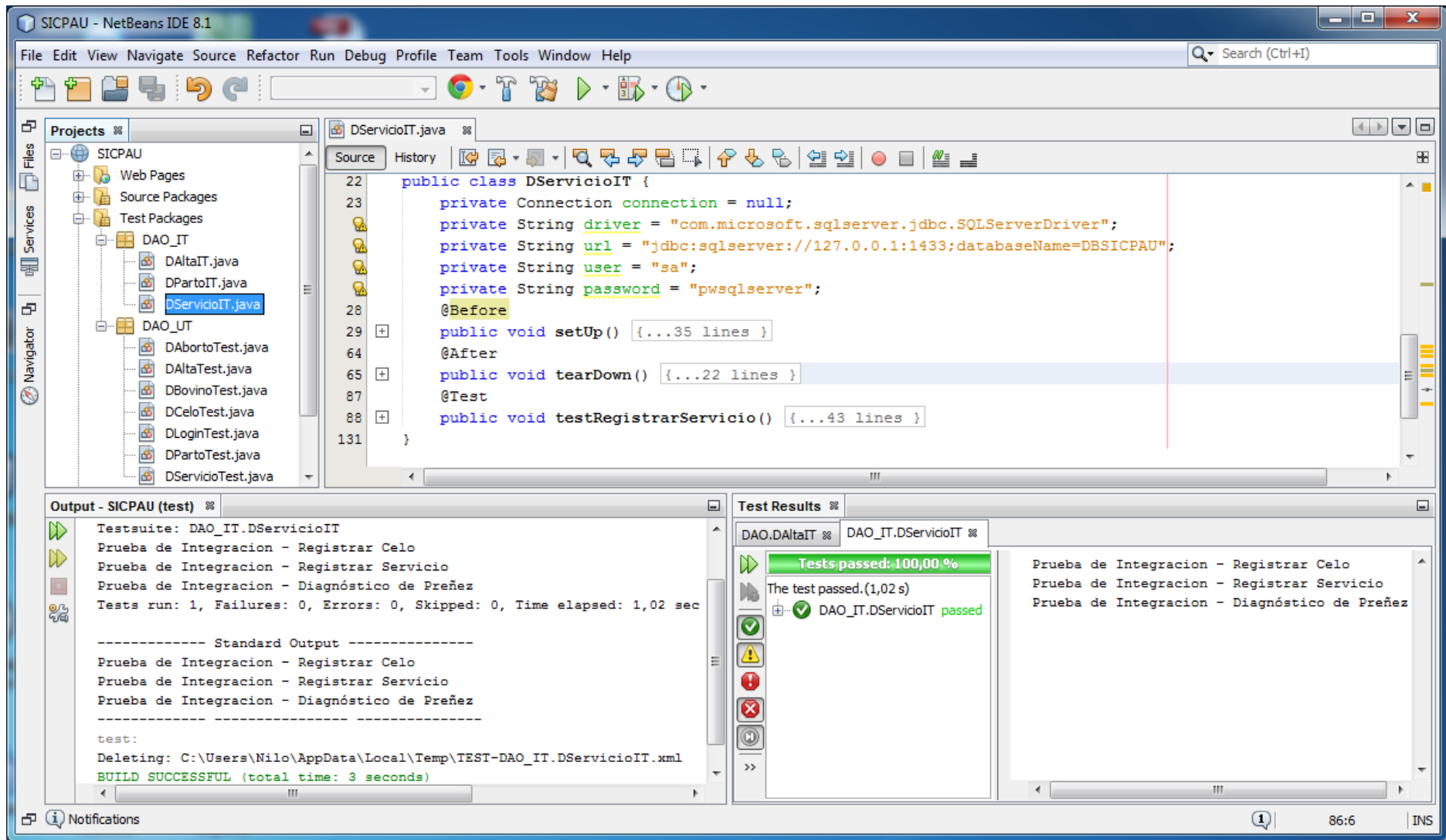


Figura N° 4.76: Prueba de Integración respecto a los métodos Registrar Celo, Registrar Servicio y Registrar Diagnóstico de Preñez.

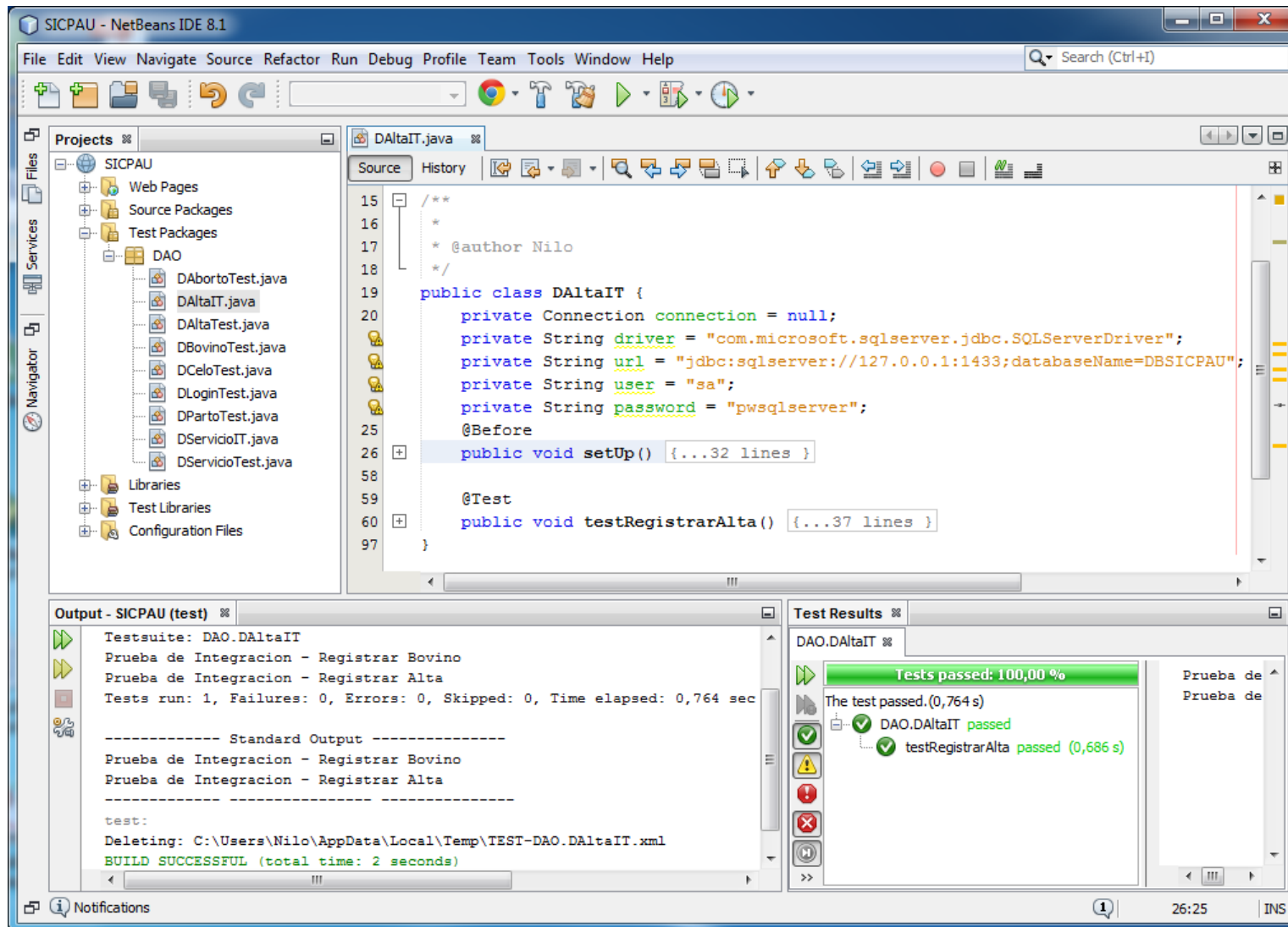


Figura N° 4.77: Prueba de Integración respecto a los métodos Registrar Alta y Registrar Bovino.

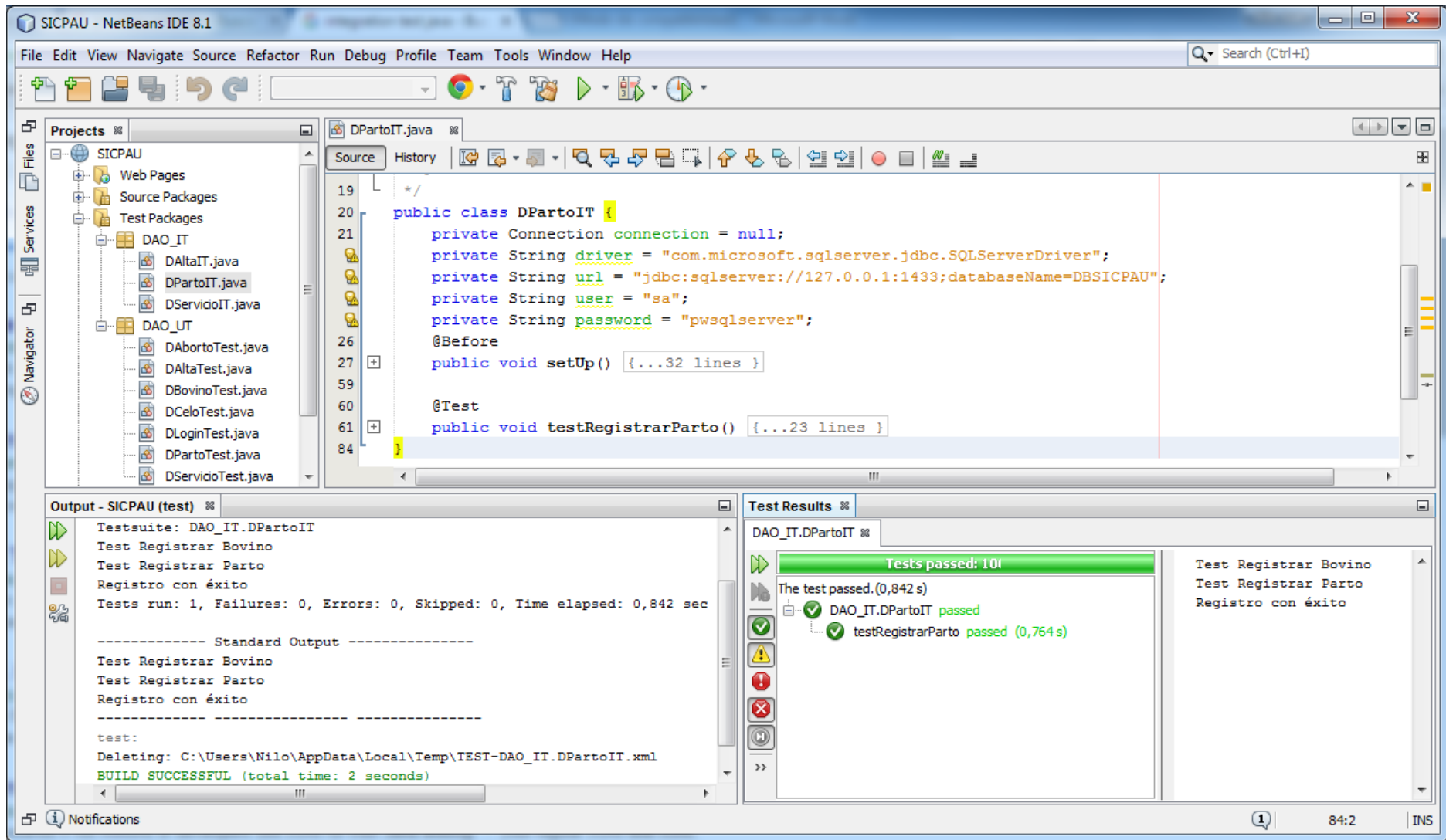


Figura N° 4.78: Prueba de Integración respecto a los métodos Registrar Bovino y Registrar Parto.

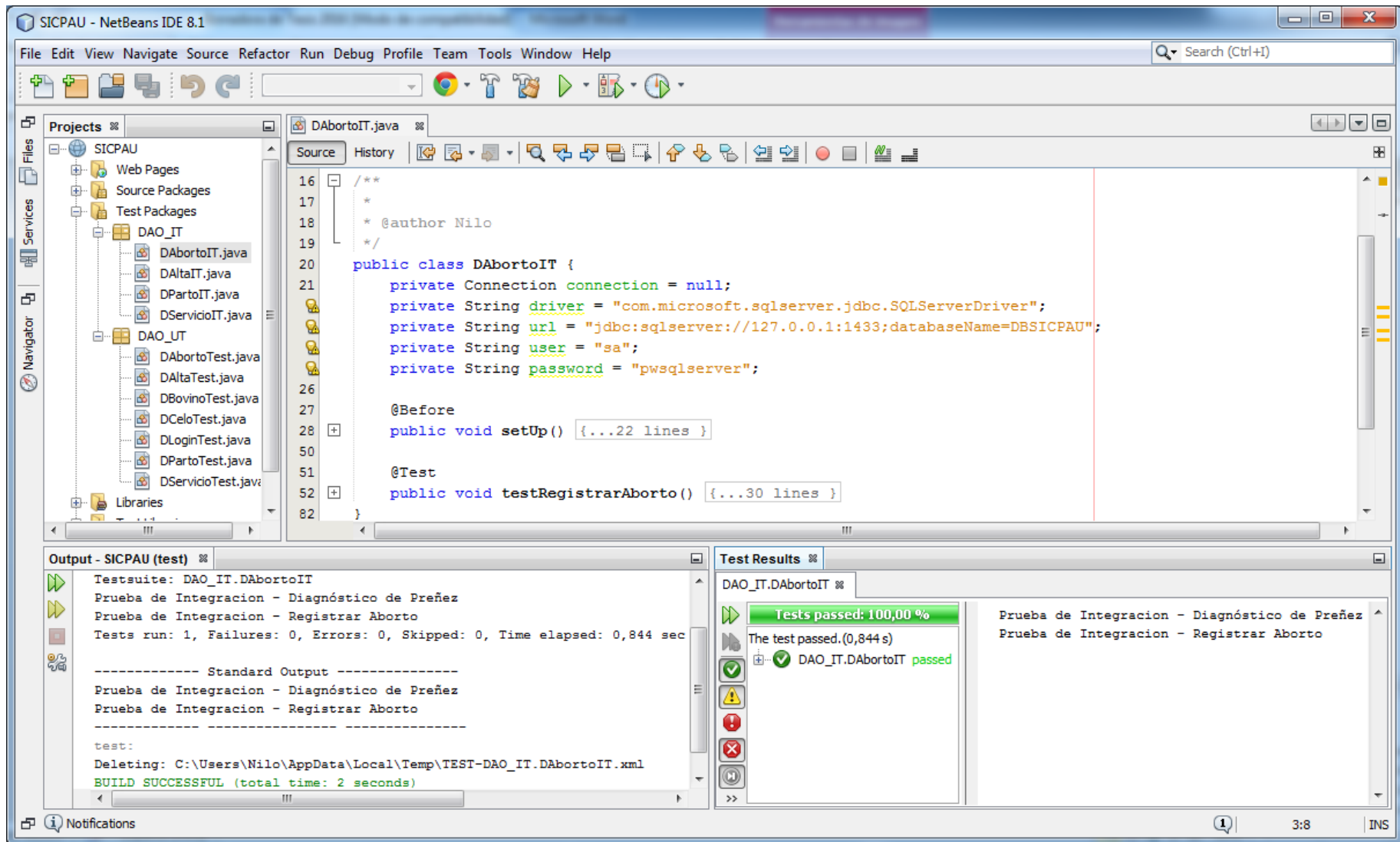


Figura N° 4.79: Prueba de Integración respecto a los métodos Registrar Servicio, Registrar Diagnóstico de Preñez y Registrar Aborto.

PRUEBAS DE ACEPTACIÓN

Las tablas 4.133 a 4.163 muestran los casos de pruebas de aceptación para las historias de usuario “registrar alta” “registrar celo, registrar servicio, y registrar diagnóstico de preñez”, “registrar bovino”, “registrar bovino y registrar parto” y “registrar servicio, registrar diagnóstico de preñez y registrar aborto” respectivamente.

Caso de Prueba	
Número Caso de Prueba: 1	Número Historia de Usuario: 1
Nombre Caso de Prueba: Validar campos del formulario registrar alta.	
Descripción: La aplicación valida los campos del formulario alta.	
Condición de Ejecución: Iniciar sesión para cargar la página “control-movimientos.jsp”.	
Entrada / Pasos de ejecución:	
<ul style="list-style-type: none"> a. Se accederá a la página “control-movimientos.jsp” desde la página principal. b. Se cargará los datos en los campos obligatorios en el formulario “alta”. c. Al hacer clic en el botón “guardar” la aplicación web antes de registrar el alta verificará que los campos obligatorios fueron cargados. d. En caso que falte cargar algún dato obligatorio el sistema muestra un mensaje emergente señalando que campo no está siendo cargado y direcciona el foco a ese campo al darle clic en el botón “aceptar”. 	
Resultado Esperado: La aplicación web no permitirá registrar el alta mientras no se haya cargado todos los campos obligatorios.	
Evaluación de la prueba: Positivo.	

Tabla N° 4.133: Caso de Prueba. Validar campos del formulario registrar alta.

Caso de Prueba	
Número Caso de Prueba: 2	Número Historia de Usuario: 1
Nombre Caso de Prueba: Sugerir código para el bovino.	
Descripción: La aplicación sugiere un código para el bovino.	
Condición de Ejecución: Iniciar sesión para cargar la página “control-movimientos.jsp”.	
Entrada/Pasos de ejecución:	
<ul style="list-style-type: none"> a. Se hará clic en el icono “ayuda” y la aplicación web mostrará un mensaje personalizado con una caja de texto, de modo que el usuario al ingresar un determinado año no menor a 2000 ni mayor a 2999 el sistema le sugiere un código para el bovino. b. En caso que el usuario ingrese un año incorrecto el sistema muestra un mensaje emergente “año incorrecto”. 	
Resultado Esperado: Los códigos registrados están dados en una orden secuencial para un determinado año, es así que el sistema sugiere un código que sucede o continúe al “último” código en ese año.	
Evaluación de la prueba: Positivo.	

Tabla N° 4.134: Caso de Prueba. Sugerir código para el bovino.

Caso de Prueba	
Número Caso de Prueba: 3	Número Historia de Usuario: 1
Nombre Caso de Prueba: Registrar satisfactoriamente el alta de bovino.	
Descripción: La aplicación web registra en la base de datos los campos cargados en el formulario.	
Condición de Ejecución: a. Iniciar sesión para cargar la página "control-movimientos.jsp". b. Cargar los campos en el formulario alta.	
Entrada/Pasos de ejecución: Al hacer clic en el botón "guardar" la aplicación web almacenará los datos del bovino y alta en la base de datos.	
Resultado Esperado: La aplicación mostrará un mensaje emergente "¡Se registró con éxito!".	
Evaluación de la prueba: Positivo.	

Tabla N° 4.135: Caso de Prueba. Registrar satisfactoriamente el alta de bovino.

Caso de Prueba	
Número Caso de Prueba: 1	Número Historia de Usuario: 5
Nombre Caso de Prueba: Validar campos del formulario registrar bovino.	
Descripción: La aplicación valida los campos del formulario bovino	
Condición de Ejecución: Iniciar sesión para cargar la página "reproduccion.jsp".	
Entrada/Pasos de ejecución: a. Se accederá a la página "reproducción.jsp" sección "Parto" desde la página principal. b. Se cargará los campos obligatorios en el formulario "bovino". c. Al hacer clic en el botón "guardar" la aplicación web antes de registrar los datos del bovino, verificará que los campos obligatorios fueron cargados. d. En caso que falte cargar algún campo obligatorio el sistema muestra un mensaje emergente señalando que campo no está siendo cargado y direcciona el foco a ese campo al darle clic en el botón "aceptar".	
Resultado Esperado: La aplicación web no permitirá registrar los datos del bovino, mientras no se haya cargado todos los campos obligatorios.	
Evaluación de la prueba: Positivo.	

Tabla N° 4.136: Caso de Prueba. Validar campos del formulario registrar bovino.

Caso de Prueba	
Número Caso de Prueba: 2	Número Historia de Usuario: 5
Nombre Caso de Prueba: Filtrar búsqueda de código de la vaca de acuerdo a su edad y sexo.	
Descripción: Cada vez que el usuario digite el código del bovino la aplicación sugiere una lista de códigos de bovino de acuerdo al sexo y edad.	
Condición de Ejecución: Iniciar sesión para cargar la página "reproduccion.jsp".	

Debe existir por lo menos un registro de vaca.
Entrada/Pasos de ejecución:
a. Al ir digitando el código del bovino en la caja de texto "Cib _(vaca) " la aplicación web buscará los códigos que coinciden con los caracteres digitados y listará los códigos filtrados.
b. Al dar clic en uno de los códigos sugeridos, la aplicación web carga el campo "Cib _(vaca) " con el código seleccionado.
Resultado Esperado: Listar los códigos de las vaquillonas o vacas en caso de existir algún registro, de lo contrario no listar nada.
Evaluación de la prueba: Positivo.

Tabla N° 4.137: Caso de Prueba. Filtrar búsqueda de código de la vaca de acuerdo a su edad y sexo

Caso de Prueba	
Número Caso de Prueba: 3	Número Historia de Usuario: 5
Nombre Caso de Prueba: Sugerir una lista de códigos de sementales que sirvieron y fecundaron a la vaca.	
Descripción: La aplicación sugiere una lista los códigos de los sementales que sirvieron y fecundaron a la vaca.	
Condición de Ejecución:	
a. Iniciar sesión para cargar la página "reproduccion.jsp".	
b. Debe existir por lo menos un registro de servicio o monta con diagnóstico positivo.	
c. Cargar el campo "Cib _(madre) ".	
Entrada/Pasos de ejecución:	
a. Se hará clic en el icono "sugerencia" y la aplicación web listará los sementales que sirvieron y fecundaron a la vaca mientras el registro de ese servicio no haya sido relacionada con algún registro de parto o aborto.	
b. Al dar clic en uno de los códigos sugeridos, la aplicación web carga el campo "Cib _(padre) " con el código seleccionado.	
Resultado Esperado: Listar los códigos de los sementales en caso de existir algún registro de servicio o monta con diagnóstico positivo y no haya sido relacionada con algún registro de parto o aborto, de lo contrario no listar nada.	
Evaluación de la prueba: Positivo.	

Tabla N° 4.138: Caso de Prueba. Sugerir una lista de códigos de sementales que sirvieron y fecundaron a la vaca.

Caso de Prueba	
Número Caso de Prueba: 4	Número Historia de Usuario: 5
Nombre Caso de Prueba: Sugerir código para el bovino.	
Descripción: La aplicación sugiere al usuario un código de bovino para un determinado año.	
Condición de Ejecución: Iniciar sesión para cargar la página "reproduccion.jsp".	
Entrada/Pasos de ejecución:	
a. Se hará clic en el icono "ayuda" y la aplicación web mostrará un mensaje	

<p>personalizado con una caja de texto, de modo que el usuario al ingresar un determinado año no menor a 2000 ni mayor a 2999 el sistema le sugiere un código para el bovino.</p> <p>b. En caso que el usuario ingrese un año incorrecto el sistema muestra un mensaje emergente "año incorrecto".</p>
<p>Resultado Esperado: Los códigos registrados están dados en una orden secuencial para un determinado año, es así que el sistema sugiere un código que sucede o continúe al "último" código en ese año.</p>
<p>Evaluación de la prueba: Positivo</p>

Tabla N° 4.139: Caso de Prueba. Sugerir código para el bovino.

Caso de Prueba	
Número Caso de Prueba: 5	Número Historia de Usuario: 5
Nombre Caso de Prueba: Registrar satisfactoriamente datos del bovino.	
Descripción: Cada vez que se registre los datos la aplicación muestra un mensaje de registro satisfactorio o exitoso.	
Condición de Ejecución:	
<p>a. Iniciar sesión para cargar la página "reproduccion.jsp".</p> <p>b. Cargar los campos en el formulario bovino.</p>	
Entrada/Pasos de ejecución: Al hacer clic en el botón "guardar" la aplicación web almacenará los datos del bovino en la base de datos.	
Resultado Esperado: La aplicación web mostrará un mensaje emergente "¡Se registró con éxito!".	
Evaluación de la prueba: Positivo.	

Tabla N° 4.140: Caso de Prueba. Registrar satisfactoriamente datos del bovino.

Caso de Prueba	
Número Caso de Prueba: 1	Número Historia de Usuario: 2
Nombre Caso de Prueba: Validar campos del formulario registrar calor o celo.	
Descripción: La aplicación valida los campos del formulario calor o celo.	
Condición de Ejecución: Iniciar sesión para cargar la página "reproduccion.jsp".	
Entrada/Pasos de ejecución:	
<p>a. Se accederá a la página "reproduccion.jsp" sección "Celo" desde la página principal.</p> <p>b. Se cargará los campos en el formulario "celo"</p> <p>c. Al hacer clic en el botón "guardar" la aplicación web antes de registrar el celo, verificará que los campos fueron cargados.</p> <p>d. En caso que falte cargar algún campo el sistema muestra un mensaje emergente señalando que campo no está siendo cargado y direcciona el foco a ese campo al darle clic en el botón "aceptar".</p>	
Resultado Esperado: La aplicación web no permitirá registrar el calor o celo, mientras no se haya cargado todos los campos.	
Evaluación de la prueba: Positivo.	

Tabla N° 4.141: Caso de Prueba. Validar campos del formulario registrar calor o celo.

Caso de Prueba	
Número Caso de Prueba: 2	Número Historia de Usuario: 2
Nombre Caso de Prueba: Filtrar búsqueda de código de la vaca de acuerdo a su edad y sexo.	
Descripción: Cada vez que el usuario digite el código del bovino la aplicación sugiere una lista de códigos de bovino de acuerdo al sexo y edad.	
Condición de Ejecución: a. Iniciar sesión para cargar la página "reproduccion.jsp". b. Debe existir por lo menos un registro de vaca.	
Entrada/Pasos de ejecución: a. Al ir digitando el código del bovino en la caja de texto "Cib _(vaca) " la aplicación web buscará los códigos que coinciden con los caracteres digitados y listará los códigos filtrados. b. Al dar clic en uno de los códigos sugeridos, la aplicación web carga el campo "Cib _(vaca) " con el código seleccionado.	
Resultado Esperado: Listar los códigos de las vaquillonas o vacas en caso de existir algún registro, de lo contrario no listar nada.	
Evaluación de la prueba: Positivo.	

Tabla N° 4.142: Caso de Prueba. Filtrar búsqueda de código de la vaca de acuerdo a su edad y sexo.

Caso de Prueba	
Número Caso de Prueba: 3	Número Historia de Usuario: 2
Nombre Caso de Prueba: Registrar satisfactoriamente el calor o celo.	
Descripción: Cada vez que se registre los datos la aplicación muestra un mensaje de registro satisfactorio o exitoso.	
Condición de Ejecución: a. Iniciar sesión para cargar la página "reproduccion.jsp". b. Cargar los campos del formulario celo.	
Entrada/Pasos de ejecución: Al hacer clic en el botón "guardar" la aplicación web almacenará los datos del bovino en la base de datos.	
Resultado Esperado: La aplicación web mostrará un mensaje emergente "¡Se registró con éxito!".	
Evaluación de la prueba: Positivo.	

Tabla N° 4.143: Caso de Prueba. Registrar satisfactoriamente el calor o celo.

Caso de Prueba	
Número Caso de Prueba: 1	Número Historia de Usuario: 3
Nombre Caso de Prueba: Validar campos del formulario servicio o monta.	
Descripción: La aplicación valida los atributos en el formulario de servicio de monta.	
Condición de Ejecución: Iniciar sesión para cargar la página "reproduccion.jsp".	
Entrada/Pasos de ejecución: a. Se accederá a la página "reproducción.jsp" sección "Servicio" desde la página principal.	

<p>b. Se cargará los campos en el formulario "servicio"</p> <p>c. Al hacer clic en el botón "guardar" la aplicación web antes de registrar el celo, verificará que los campos fueron cargados.</p> <p>d. En caso que falte cargar algún campo el sistema muestra un mensaje emergente señalando que campo no está siendo cargado y direcciona el foco a ese campo al darle clic en el botón "aceptar".</p>
<p>Resultado Esperado: La aplicación web no permitirá registrar el calor o celo, mientras no se haya cargado todos los campos.</p>
<p>Evaluación de la prueba: Positivo.</p>

Tabla N° 4.144: Caso de Prueba. Validar campos del formulario servicio o monta.

Caso de Prueba	
Número Caso de Prueba: 2	Número Historia de Usuario: 3
Nombre Caso de Prueba: Filtrar búsqueda de código de la vaca de acuerdo a su edad y sexo.	
Descripción: Cada vez que el usuario digite el código del bovino la aplicación sugiere una lista de códigos de bovino de acuerdo al sexo y edad.	
Condición de Ejecución:	
<p>a. Iniciar sesión para cargar la página "reproduccion.jsp".</p> <p>b. Debe existir por lo menos un registro de vaca.</p>	
Entrada/Pasos de ejecución:	
<p>a. Al ir digitando el código del bovino en la caja de texto "Cib(vaca)" la aplicación web buscará los códigos que coinciden con los caracteres digitados y listará los códigos filtrados.</p> <p>b. Al dar clic en uno de los códigos sugeridos, la aplicación web carga el campo "Cib(vaca)" con el código seleccionado.</p>	
Resultado Esperado: Listar los códigos de las vaquillonas o vacas en caso de existir algún registro, de lo contrario no listar nada.	
Evaluación de la prueba: Positivo.	

Tabla N° 4.145: Caso de Prueba. Filtrar búsqueda de código de la vaca de acuerdo a su edad y sexo.

Caso de Prueba	
Número Caso de Prueba: 3	Número Historia de Usuario: 3
Nombre Caso de Prueba: Verificar que la fecha y hora de servicio corresponde a la fecha y hora de algún registro de celo.	
Descripción: La aplicación antes de guardar el registro verifica si la fecha de servicio corresponde a la fecha de algún registro de celo.	
Condición de Ejecución: Iniciar sesión para cargar la página "control-movimientos.jsp".	
Entrada/Pasos de ejecución: Después de cargar todos los campos en el formulario servicio pulsar en el botón "guardar".	
Resultado Esperado: Si la correspondencia no es correcta el sistema muestra un mensaje emergente "[ERROR], no existe registro de celo que corresponda a este servicio, verifique la fecha u hora del servicio", de lo contrario muestra "¡Se registró con éxito!".	
Evaluación de la prueba: Positivo.	

Tabla N° 4.146: Caso de Prueba. Verificar que la fecha y hora de servicio corresponde a la fecha y hora de algún registro de celo.

Caso de Prueba	
Número Caso de Prueba: 4	Número Historia de Usuario: 3
Nombre Caso de Prueba: Registrar satisfactoriamente el servicio o monta.	
Descripción: Cada vez que se registre los datos la aplicación muestra un mensaje de registro satisfactorio o exitoso.	
Condición de Ejecución: a. Iniciar sesión para cargar la página "control-movimientos.jsp". b. Debe existir por lo menos un registro de celo. c. Cargar los campos en el formulario servicio.	
Entrada/Pasos de ejecución: Al hacer clic en el botón "guardar" la aplicación web almacenará los datos del servicio en la base de datos.	
Resultado Esperado: La aplicación web mostrará un mensaje emergente "¡Se registró con éxito!".	
Evaluación de la prueba: Positivo.	

Tabla N° 4.147: Caso de Prueba. Registrar satisfactoriamente el servicio o monta.

Caso de Prueba	
Número Caso de Prueba: 1	Número Historia de Usuario: 4
Nombre Caso de Prueba: Validar campos del formulario diagnóstico de preñez.	
Descripción: La aplicación valida los atributos en el formulario de diagnóstico de preñez.	
Condición de Ejecución: Iniciar sesión para cargar la página "reproduccion.jsp".	
Entrada/Pasos de ejecución: a. Se accederá a la página "reproducción.jsp" sección "Diagnóstico de preñez" desde la página principal. b. Se cargará los campos en el formulario "diagnóstico de preñez". c. Al hacer clic en el botón "guardar" la aplicación web antes de registrar el celo, verificará que los campos fueron cargados. d. En caso que falte cargar algún campo el sistema muestra un mensaje emergente señalando que campo no está siendo cargado y direcciona el foco a ese campo al darle clic en el botón "aceptar".	
Resultado Esperado: La aplicación web no permitirá registrar el diagnóstico de preñez, mientras no se haya cargado todos los campos.	
Evaluación de la prueba: Positivo.	

Tabla N° 4.148: Caso de Prueba. Validar campos del formulario diagnóstico de preñez.

Caso de Prueba	
Número Caso de Prueba: 2	Número Historia de Usuario: 4
Nombre Caso de Prueba: Filtrar búsqueda de código de la vaca de acuerdo a	

su edad y sexo.
Descripción: Cada vez que el usuario digite el código del bovino la aplicación sugiere una lista de códigos de bovino de acuerdo al sexo y edad.
Condición de Ejecución: a. Iniciar sesión para cargar la página "reproduccion.jsp". b. Debe existir por lo menos un registro de vaca.
Entrada/Pasos de ejecución: a. Al digitar el código del bovino en la caja de texto "Cib _(vaca) " la aplicación web hará una sugerencia de búsqueda que coinciden con los caracteres digitados. b. Al dar clic en uno de los códigos sugeridos, la aplicación web carga el campo "Cib _(vaca) " con el código seleccionado.
Resultado Esperado: Listar los códigos de las vaquillonas o vacas en caso de existir algún registro, de lo contrario no listar nada.
Evaluación de la prueba: Positivo.

Tabla N° 4.149: Caso de Prueba. Filtrar búsqueda de código de la vaca de acuerdo a su edad y sexo.

Caso de Prueba	
Número Caso de Prueba: 3	Número Historia de Usuario: 4
Nombre Caso de Prueba: Para una determinada vaca listar las fecha de los servicios que se le practico y que aún no fueron diagnosticados.	
Descripción: La aplicación lista la fecha de los servicios que aún no fueron diagnosticados para una determinada vaca.	
Condición de Ejecución: a. Iniciar sesión para cargar la página "reproduccion.jsp". b. Debe existir algún registro de servicio correspondiente al codigo cargado el campo "Cib _(vaca) ".	
Entrada/Pasos de ejecución: Cargar el campo "Cib _(vaca) ".	
Resultado Esperado: La lista de selección "Fecha _(servicio) " cargada con las fechas de los servicios que aún no fueron diagnosticados.	
Evaluación de la prueba: Positivo.	

Tabla N° 4.150: Caso de Prueba. Para una determinada vaca listar las fecha de los servicios que se le practico y que aún no fueron diagnosticados.

Caso de Prueba	
Número Caso de Prueba: 4	Número Historia de Usuario: 4
Nombre Caso de Prueba: Registrar satisfactoriamente el diagnóstico de preñez.	
Descripción: Cada vez que se registre los datos la aplicación muestra un mensaje de registro satisfactorio o exitoso.	
Condición de Ejecución: a. Iniciar sesión para cargar la página "reproduccion.jsp". b. Cargar los campos en el formulario diagnóstico de preñez.	
Entrada/Pasos de ejecución: Hacer clic en el botón "guardar" y la aplicación web actualizará el campo diagnóstico de preñez de la tabla servicio en la base de datos.	
Resultado Esperado: La aplicación web mostrará un mensaje emergente "¡Se registró con éxito!".	

Evaluación de la prueba: Positivo.

Tabla N° 4.151: Caso de Prueba. Registrar satisfactoriamente el diagnóstico de preñez.

Caso de Prueba	
Número Caso de Prueba: 1	Número Historia de Usuario: 6
Nombre Caso de Prueba: Validar campos del formulario parto.	
Descripción: La aplicación valida los atributos en el formulario de parto.	
Condición de Ejecución: Iniciar sesión para cargar la página "reproduccion.jsp".	
Entrada/Pasos de ejecución: <ol style="list-style-type: none">Se accederá a la página "reproducción.jsp" sección "Parto" desde la página principal.Se cargará los campos en el formulario "parto".Al hacer clic en el botón "guardar" la aplicación web antes de registrar el parto, verificará que los campos fueron cargados.En caso que falte cargar algún campo el sistema muestra un mensaje emergente señalando que campo no está siendo cargado y direcciona el foco a ese campo al darle clic en el botón "aceptar".	
Resultado Esperado: La aplicación web no permitirá registrar el diagnóstico de preñez, mientras no se haya cargado todos los campos.	
Evaluación de la prueba: Positivo.	

Tabla N° 4.152: Caso de Prueba. Validar campos del formulario parto.

Caso de Prueba	
Número Caso de Prueba: 2	Número Historia de Usuario: 6
Nombre Caso de Prueba: Filtrar búsqueda de código de la vaca de acuerdo a su edad y sexo.	
Descripción: Cada vez que el usuario digite el código del bovino la aplicación sugiere una lista de códigos de bovino de acuerdo al sexo y edad.	
Condición de Ejecución: <ol style="list-style-type: none">Iniciar sesión para cargar la página "reproduccion.jsp".Debe existir por lo menos un registro de vaca.	
Entrada/Pasos de ejecución: <ol style="list-style-type: none">Al ir digitando el código del bovino en la caja de texto "cib_(vaca)" la aplicación web buscará los códigos que coinciden con los caracteres digitados y listará los códigos filtrados.Al dar clic en uno de los códigos sugeridos, la aplicación web carga el campo "Cib_(vaca)" con el código seleccionado.	
Resultado Esperado: Listar los códigos de las vaquillonas o vacas en caso de existir algún registro, de lo contrario no listar nada.	
Evaluación de la prueba: Positivo.	

Tabla N° 4.153: Caso de Prueba. Filtrar búsqueda de código de la vaca de acuerdo a su edad y sexo.

Caso de Prueba

Número Caso de Prueba: 3	Número Historia de Usuario: 6
Nombre Caso de Prueba: Listar códigos de terneros que aún no fueron relacionados con algún registro de parto o aborto.	
Descripción: La aplicación lista los códigos de los terneros que aún no fueron relacionados con algún registro de parto o aborto.	
Condición de Ejecución: Iniciar sesión para cargar la página "reproduccion.jsp".	
Entrada/Pasos de ejecución: Cargar el campo "Cib _(vacca) ".	
Resultado Esperado: La lista de selección "Cib _(ternero) " cargada con los códigos de los terneros que aún no fueron relacionados con algún registro de parto o aborto.	
Evaluación de la prueba: Positivo.	

Tabla N° 4.154: Caso de Prueba. Listar códigos de terneros que aún no fueron relacionados con algún registro de parto o aborto.

Caso de Prueba	
Número Caso de Prueba: 4	Número Historia de Usuario: 6
Nombre Caso de Prueba: Registrar satisfactoriamente el parto.	
Descripción: Cada vez que se registre los datos la aplicación muestra un mensaje de registro satisfactorio o exitoso.	
Condición de Ejecución: a. Iniciar sesión para cargar la página "reproduccion.jsp". b. Cargar los campos en el formulario parto.	
Entrada/Pasos de ejecución: Al hacer clic en el botón "guardar" la aplicación web almacenará los datos del parto en la base de datos.	
Resultado Esperado: La aplicación web mostrará un mensaje emergente "¡Se registró con éxito!".	
Evaluación de la prueba: Positivo.	

Tabla N° 4.155: Caso de Prueba. Registrar satisfactoriamente el parto.

Caso de Prueba	
Número Caso de Prueba: 1	Número Historia de Usuario: 7
Nombre Caso de Prueba: Validar campos del formulario aborto.	
Descripción: La aplicación valida los atributos en el formulario de aborto.	
Condición de Ejecución: Iniciar sesión para cargar la página "reproduccion.jsp".	
Entrada/Pasos de ejecución: a. Se accederá a la página "reproducción.jsp" sección "Aborto" desde la página principal. b. Se cargará los campos en el formulario "aborto". c. Al hacer clic en el botón "guardar" la aplicación web antes de registrar el aborto, verificará que los campos fueron cargados. d. En caso que falte cargar algún campo el sistema muestra un mensaje emergente señalando que campo no está siendo cargado y direcciona el foco a ese campo al darle clic en el botón "aceptar".	
Resultado Esperado: La aplicación web no permitirá registrar el diagnóstico de preñez, mientras no se haya cargado todos los campos.	

Evaluación de la prueba: Positivo.

Tabla N° 4.156: Caso de Prueba. Validar campos del formulario aborto.

Caso de Prueba	
Número Caso de Prueba: 2	Número Historia de Usuario: 7
Nombre Caso de Prueba: Filtrar búsqueda de código de la vaca de acuerdo a su edad y sexo.	
Descripción: Cada vez que el usuario digite el código del bovino la aplicación sugiere una lista de códigos de bovino de acuerdo al sexo y edad.	
Condición de Ejecución: a. Iniciar sesión para cargar la página "reproduccion.jsp". b. Debe existir por lo menos un registro de vaca.	
Entrada/Pasos de ejecución: a. Al ir digitando el código del bovino en la caja de texto "Cib _(vaca) " la aplicación web buscará los códigos que coinciden con los caracteres digitados y listará los códigos filtrados. b. Al dar clic en uno de los códigos sugeridos, la aplicación web carga el campo "Cib _(vaca) " con el código seleccionado.	
Resultado Esperado: Listar los códigos de las vaquillonas o vacas en caso de existir algún registro, de lo contrario no listar nada.	
Evaluación de la prueba: Positivo.	

Tabla N° 4.157: Caso de Prueba. Filtrar búsqueda de código de la vaca de acuerdo a su edad y sexo.

Caso de Prueba	
Número Caso de Prueba: 3	Número Historia de Usuario: 7
Nombre Caso de Prueba: Listar fechas de servicios con resultado de diagnóstico positivo que aún no fueron relacionados con algún registro de aborto o parto.	
Descripción: La aplicación lista la fecha de los servicios de diagnóstico positivo que aún no fueron relacionados con algún registro de aborto o parto.	
Condición de Ejecución: a. Iniciar sesión para cargar la página "reproduccion.jsp". b. Debe existir algún registro de servicio correspondiente al código cargado el campo "Cib _(vaca) ".	
Entrada/Pasos de ejecución: Cargar el campo "Cib _(vaca) ".	
Resultado Esperado: La lista de selección "Fecha _(servicio) " cargada con las fechas de los servicios que aún no fueron relacionadas con algún registro de parto o aborto.	
Evaluación de la prueba: Positivo.	

Tabla N° 4.158: Caso de Prueba. Listar fechas de servicios con resultado de diagnóstico positivo que aún no fueron relacionados con algún registro de aborto o parto.

Caso de Prueba	
Número Caso de Prueba: 4	Número Historia de Usuario: 7

Nombre Caso de Prueba: Registrar satisfactoriamente el aborto.
Descripción: Cada vez que se registre los datos la aplicación muestra un mensaje de registro satisfactorio o exitoso.
Condición de Ejecución: a. Iniciar sesión para cargar la página "reproduccion.jsp". b. Cargar los campos en el formulario aborto.
Entrada/Pasos de ejecución: Al hacer clic en el botón "guardar" la aplicación web almacenará los datos del aborto en la base de datos.
Resultado Esperado: La aplicación web mostrará un mensaje emergente "¡Se registró con éxito!".
Evaluación de la prueba: Positivo.

Tabla N° 4.159: Caso de Prueba. Registrar satisfactoriamente el aborto.

Caso de Prueba	
Número Caso de Prueba: 1	Número Historia de Usuario: 8
Nombre Caso de Prueba: Filtrar búsqueda de código de la vaca de acuerdo a su edad y sexo.	
Descripción: Cada vez que el usuario digite el código del bovino la aplicación sugiere una lista de códigos de bovino de acuerdo al sexo y edad.	
Condición de Ejecución: a. Iniciar sesión para cargar la página "reproduccion.jsp". b. Debe existir por lo menos un registro de vaca.	
Entrada/Pasos de ejecución: a. Al ir digitando el código del bovino en la caja de texto "Cib(vaca)" la aplicación web buscará los códigos que coinciden con los caracteres digitados y listará los códigos filtrados. b. Al dar clic en uno de los códigos sugeridos, la aplicación web carga el campo "Cib(vaca)" con el código seleccionado.	
Resultado Esperado: Listar los códigos de las vaquillonas o vacas en caso de existir algún registro, de lo contrario no listar nada.	
Evaluación de la prueba: Positivo.	

Tabla N° 4.160: Caso de Prueba. Filtrar búsqueda de código de la vaca de acuerdo a su edad y sexo.

Caso de Prueba	
Número Caso de Prueba: 2	Número Historia de Usuario: 8
Nombre Caso de Prueba: Cálculo de número de días al primer celo, número de días al primer servicio, número de días de vacía, numero de servicios por concepción e intervalo entre partos.	
Descripción: La aplicación realiza los cálculos de número de días al primer celo, número de días al primer servicio, número de días de vacía, numero de servicios por concepción e intervalo entre partos.	
Condición de Ejecución: Iniciar sesión para cargar la página "reproduccion.jsp".	
Entrada/Pasos de ejecución: a. Acceder a la página "reproducción.jsp" sección "Reporte de eventos reproductivos" desde la página principal.	

b. Cargar el campo "Cib _(vaca) " para que la aplicación web muestre la tabla servicio concepción.
Resultado Esperado: Mostrar una tabla con los datos "fecha de parto", "sexo del ternero", "nombre del ternero", "fecha de los celos sin servir", "fecha de servicios", "días al primer celo", "días al primer servicio", "días de vacía", "numero de servicios por concepcion" e "intervalo entre parto" para una determinada vaca.
Evaluación de la prueba: Positivo.

Tabla N° 4.161: Caso de Prueba. Cálculo de número de días al primer celo, número de días al primer servicio, número de días de vacía, numero de servicios por concepción e intervalo entre partos.

Caso de Prueba	
Número Caso de Prueba: 1	Número Historia de Usuario: 24
Nombre Caso de Prueba: Carga la página principal.	
Descripción: La aplicación carga la página principal cuando el usuario se autentica correctamente.	
Condición de Ejecución: debe existir al menos un registro de cuenta de usuario.	
Entrada/Pasos de ejecución:	
a. Ingresar la dirección de la aplicación web en cualquier navegador web y la aplicación web carga la página "index.jsp".	
b. Cargar los campos usuario y clave (autenticarse correctamente).	
Resultado Esperado: Cargar la página principal "principal.jsp" de la aplicación web.	
Evaluación de la prueba: Positivo.	

Tabla N° 4.162: Caso de Prueba. Carga la página principal.

Caso de Prueba	
Número Caso de Prueba: 2	Número Historia de Usuario: 24
Nombre Caso de Prueba: No permitir el acceso a la página principal.	
Descripción: La aplicación muestra un mensaje de error cuando el usuario se autentica incorrectamente.	
Condición de Ejecución: Autenticarse con datos incorrectos.	
Entrada/Pasos de ejecución:	
a. Ingresar la dirección de la aplicación web en cualquier navegador web y la aplicación web carga la página "index.jsp".	
b. Cargar los campos usuario y clave (autenticarse correctamente).	
Resultado Esperado: Un mensaje de color rojo "¡El usuario o contraseña es incorrecto!"	
Evaluación de la prueba: Positivo.	

Tabla N° 4.163: Caso de Prueba. No permitir el acceso a la página principal.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

De acuerdo a los objetivos planteados en relación a los problemas identificados, se recabo la información con los instrumentos que se muestran en los anexos A, B y C, con la información obtenida se realizó las historias de usuario relacionadas a la reproducción de ganado bovino, para monitorear la producción individual y duración de lactancia, y se muestran en las historias de usuario (H.U: N° 1, 2, 3, 4, 5, 6, 7 y 8), interfaces (Figuras N° 4.7, 4.13, 4.15, 4.17, 4.18, 4.21 y 4,24) y base de datos (Figura. N° 4.67), son desarrolladas e implementadas de acuerdo al capítulo III, secciones 3.5.1, 3.5.2, 3.5.3 y 3.5.4, sustentadas en el marco teórico del capítulo II, sección 2.2.3. Asimismo, en el capítulo IV, durante la fase de exploración se obtiene; las historias de usuario, arquitectura técnica inicial y el plan de alto nivel, en la fase de planificación tenemos; las historias de usuario en detalle, el plan de versión (primera iteración), en la fase de iteración tenemos; la arquitectura técnica, tareas de ingeniería, plan de iteración (segunda, tercera, cuarta, quinta y sexta), casos de prueba de aceptación, GUI, tarjetas CRC, base de datos física, código fuente para tareas de ingeniería, pruebas unitarias y pruebas de integración. Por lo expuesto, se puede llegar a las siguientes conclusiones.

- a. Una vez que se instale y ejecute la aplicación web, éste brindará información necesaria (días a primer celo, días a primer servicio, días abiertos, servicio por concepción e intervalo entre partos) de la reproducción bovina, con la finalidad de monitorear la producción láctea en el centro de producción de Allpachaka.

- b. Una vez que se instale y ejecute la aplicación web, éste brindará información necesaria (días a primer celo, días a primer servicio, días abiertos, servicio por concepción e intervalo entre partos) de la reproducción bovina, a fin de monitorear la duración de lactancia en el centro de producción de Allpachaka.

5.2. RECOMENDACIONES

- a. En el Centro de Producción de Allpachaka, la detección de celos de las vacas se realiza mediante métodos visuales, lo cual implica, no identificar el momento exacto cuando la vaca entra en celo, originando problemas tales como; la prolongación del intervalo entre partos y la reducción de la producción láctea. Por tanto, se recomienda implementar una herramienta tecnológica, capaz de detectar los celos en las vacas y anunciar el momento preciso en que se tiene que realizar el servicio respectivo.

- b. De acuerdo a los registros de producción láctea, del Centro de Producción de Allpachaka, se observa que en promedio las vacas en su primera y quinta campaña de producción láctea, produce aproximadamente 7 y 11 litros de leche al día, respectivamente, situación que puede ser mejorada con un buen programa de alimentación, puesto que los bovinos de razas brown swiss, son razas que tienen un mayor potencial de producción láctea. Por tanto, se recomienda desarrollar una aplicación web, para ayudar a balancear la dieta de bovinos, a planificar la alimentación, y a tomar decisiones sobre carga y sistema de pastoreo. Y sea capaz de procesar la información; peso, condición corporal, carga animal, forraje disponible, entre otros, por medio de simuladores de pastoreo y de requerimientos de energía, proteína, fibra, calcio y fósforo. Con el objetivo de ajustar los datos de la calidad de la ingesta, del consumo del animal en pastoreo y del costo energético del pastoreo.

BIBLIOGRAFÍA

- 1) Almeyda, J. y Parreño, J. (2011). *Manejo integrado de ganado vacuno*. Manuscrito no publicado, Universidad Nacional Agraria la Molina, Lima, Perú.
- 2) Alonso, F., Martínez, L. y Segovia, F. (2005). *Introducción a la Ingeniería de Software*. Madrid, España: Delta Publicaciones Universitarias.
- 3) Aumaille, B. (2002). *J2EE Desarrollo de Aplicaciones Web*. Barcelona, España: Ediciones ENI.
- 4) Badenas, J., Llopis, J. y Coltell, O. (2001). *Curso Práctico de Programación en C y C++ (2ª Ed.)*. Castellón de la Plana, España: Universitat Jaume I.
- 5) Balkenende, J. (s.f.). Software para manejo de ganado lechero vacuno. Sistema para bovinos de leche. Córdoba, Argentina: Grupo Sol S.R.L.. Recuperado de: <http://www.infotambo.com.ar/>
- 6) Bazán, O. (2008). *Diagnóstico del comportamiento reproductivo del ganado de carne en fincas de Costa Rica*. Tesis de grado no publicado, Instituto Interamericano de Ciencias Agrícolas de la OEA, Turrialba, Costa Rica.
- 7) Beck, K. y Andres, C. (2004). *Extreme programming explained: embrace change*. E.E.U.U. Oregon: Addison Wesley.
- 8) Bell, D. y Parr, M. (2003). *Java para Estudiantes (3ª ed.)*. México: Pearson Education.
- 9) Bernal, C. (2006). *Metodología de la investigación para administración, economía, humanidades y ciencias sociales (2ª ed.)*. Atlacomulco, México: Pearson Educación.
- 10) Berzal, F., Cubero, j. y Cortijo, F. (2005). *Desarrollo Profesional de Aplicaciones Web con ASP.NET*. Manuscrito no publicado,

Granada, España.

- 11) Borda, M., Tuesca, R. y Navarro, E. (2014). *Métodos Cuantitativos – Herramientas para la Investigación en Salud*. Barranquilla, Colombia: Universidad del Norte.
- 12) Bretschneider, G., Salado, E., Cuatrin, A. y Arias, D. (2015). *Lactancia: Pico y Persistencia*. Recuperado el 18 de Diciembre de 2015, de http://inta.gob.ar/sites/default/files/script-tmp-inta_lactancia_pico_y_persistencia_febrero_2015.pdf
- 13) Caballero, H. y Hervas, T. (2005). *Producción lechera en la sierra ecuatoriana*. Quito, Ecuador: ed. Hernán Caballero y Thelmo Hervas.
- 14) Cardona, S., Jaramillo, S. y Villegas, M. (2008). *Introducción a la Programación en Java*. Colombia: ediciones Elizcom.
- 15) Carrasco, J. (2004). *Creación de programas con Borland C++ 5.0*. Lima, Perú: AG editores.
- 16) Carrillo, A. (2006). *Fundamentos de Programación en C++*. Madrid, España: Delta.
- 17) Castro, A. (2002). *Ganadería de Leche Enfoque Empresarial*. San José, Costa Rica: EUNED.
- 18) Cobo, A. (2008). *Diseño y Programación de Base de Datos*. Madrid, España: Visión Libros.
- 19) Córdova, A. (2008). *Factores ambientales y genéticos que afecta la producción de leche de vacas de la Unidad de Producción Allpachaka (3500 m.s.n.m.) Ayacucho, 2008*. Tesis de grado no publicado, Universidad Nacional de San Cristóbal de Huamanga, Ayacucho, Perú.
- 20) Córdova, S. (2004). *Evaluación de los índices productivos del hato vacuno lechero U.P. Allpachaka de la UNSCH (3550 msnm) años 2000 – 2003, Ayacucho (2004)*. Tesis de grado no publicado, Universidad Nacional de San Cristóbal de Huamanga, Ayacucho, Perú.
- 21) De Alba, J. (2004). *Reproducción y genética animal*. Turrialba,

Costa Rica: SIC.

- 22) Deitel, H. y Deitel, P. (2004). *Como Programar en C/C++ y Java*: Prentice Hall.
- 23) Desongles (2005). *Ayudante técnico de informática*. Sevilla, España: Mad, S.L.
- 24) Díaz, V. (2006). *Metodología de la investigación científica y bioestadística*. Santiago, Chile: RIL.
- 25) De Pablos, C., López, J., Martín, S. y Medina, S. (2004). *Informática y comunicaciones en la empresa*. Madrid, España: ESIC.
- 26) Drozdek, A. (2007). *Estructura de datos y algoritmos en Java*. México, D.F., México: Thomson.
- 27) Duran, F., Gutiérrez, F. y Pimentel, E. (2007). *Programación Orientada a Objetos con Java*. Madrid, España: Thomson.
- 28) Flores, J. (2007). *MÉTODO DE LAS 6'D UML – Pseudocódigo – Java (Enfoque algorítmico)*. Lima, Perú: Fondo Editorial de la USMP.
- 29) García, E. (2007). *Principios Básicos de Informática*. Madrid, España: DYKINSON.
- 30) García, L. (2010). *Programación Orientada a Objetos Java*. Barranquilla, Colombia: Ibáñez.
- 31) Gélvéz, L. (2007). *El proceso reproductivo de vacas y toros*. Recuperado el 13 de diciembre de 2015, de http://mundopecuario.com/tema252/reproduccion_bovinos/
- 32) Grupo Sol, (s.f.). *Aspectos sobre la Producción Lechera*. Recuperado el 13 de diciembre de 2015, de <http://www.gruposol-srl.com.ar/act/curva-lactancia.htm>
- 33) Gutiérrez, G. (2010). *Lenguajes de Consulta SQL: Lenguaje de definición de datos y lenguaje de manipulación de datos*. Recuperado el 21 de enero de 2015, de <http://www.fcca.umich.mx/descargas/apuntes/Academia%20de%20Informatica/Base%20de%20Datos%20%20II%20%20%20G.A.G.C/unidad%201.pdf>.
- 34) Gutiérrez, P. (2009). *Manual práctico de manejo de una*

explotación de vacuno lechero. Recuperado el 21 de enero de 2015, de http://www.agriculturaganaderia.jcyl.es/web/jcyl/AgriculturaGanaderia/es/Plantilla100Detalle/1284243551827/_/1250170548860/Redaccion.

- 35) Heredero, C., Romo, J. y Medina, S. (2004). *Informática y Comunicaciones en la Empresa*. Madrid, España: ESIC.
- 36) Hernández, R., Fernández, C. y Baptista, P. (2006). *Metodología de la investigación* (4ª ed.). México, D.F., México: McGraw Hill Interamericana editores, S.A. de C.V.
- 37) Instituto Interamericano de Cooperación para la Agricultura (2008). *Reproducción animal: métodos de estudio en sistemas*. San José, Costa Rica: ed. Ruiz, M., Rivera, B. y Ruiz, A.
- 38) Jaramillo, S., Cardona, S. y Villa, D. (2008). *Programación Avanzada en Java*. Armenia, Colombia: Elizcom.
- 39) Jamrich, J. y Oja, D. (2008). *Conceptos de computación: nuevas perspectivas* (10ª ed.). México, D. F.: Cengage Learning Editores.
- 40) Joyanes, L. (2003). *Fundamentos de Programación*. Madrid, España: Mc Graw Hill.
- 41) Katrib, M., del Valle, M., Sierra, I. y Hernández, Y. (2009). *Windows Presentation Foundation*. Madrid, España: Luarna.
- 42) Kendall, K. (2005). *Análisis y diseño de sistemas* (6ª Ed.). Naucalpan de Juárez, México: Pearson Educación.
- 43) Kroenke, D. (2003), *Procesamiento de Base de Datos: Fundamentos, Diseño e implementación* (8ª Ed.). México, D.F., México: Pearson Prentice Hall.
- 44) Kurose, J. y Ross, K. (2004). *Redes de Computadores: Un Enfoque descendente basado en Internet* (2ª Ed.). Madrid, España: Pearson Education.
- 45) Letelier, P. y Penadés, C. (15 de enero de 2006), *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Recuperado de: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>
- 46) Marquez, X. (2009). *Software Avanzado para la Gestión*

- Ganadera. Barcelona, España: Agritec Software. Recuperado de:
<http://www.agritecsoft.com>.
- 47) Martel, P. y Díez, J. (1997). *Probabilidad y estadística en medicina*. Madrid, España: Díaz de Santos.
 - 48) Menéndez-Barzanallana, R. (2012), *Bases de datos – introducción*. Recuperado el 27 de enero de 2013, de
<http://www.um.es/docencia/barzana/IAGP/IAGP2-Bases-datos-introduccion.html>
 - 49) Morelli, R. y Walde, R. (2006). *Java, Java, Java, Object-Oriented Problem Solving* (3ª Ed.). United States: Prentice Hall.
 - 50) Nevado, V. (2010). *Introducción a las bases de datos relacionales*, Madrid, España: Visión Libros.
 - 51) Noriega, A. (2007). *Programación en Java 2SE*. Lima, Perú: Megabyte.
 - 52) Pérez, M. (2014). *Programación Extrema: Qué es y principios básicos*. Recuperado el 18 de julio de 2016, de
<https://geekytheory.com/programacion-extrema-que-es-y-principios-basicos/>
 - 53) Porras, E. (2010). *La Metodología Ágil y Formal Iconix para el Desarrollo de Software: Teoría y Práctica*. Ayacucho, Perú: Ed. por Porras Flores, Efrain E.
 - 54) Ramos, A., Ramos, J. (2007). *Operaciones con Bases de Datos Ofimáticas y Corporativas*. Madrid, España: Thomson.
 - 55) Roa, N. (2006). *Ciencia, producción y protección animal*. Maracay, Venezuela: INIA.
 - 56) Rob, P., Coronel, C. (2003) *Sistemas de Base de Datos Diseño Implementación y Administración*: Buenos Aires, Argentina: Thomson.
 - 57) Robles, G. y Ferrer, J. (2002). *Programación eXtrema y Software Libre*. Madrid, España: Congreso Hispalinux. Recuperado de:
<http://es.tldp.org/Presentaciones/200211hispalinux/ferrer/robles-ferrer-ponencia-hispalinux-2002.html>

- 58) Rodríguez, A. (2006). *Iniciación a la Red Internet: Concepto, Funcionamiento, Servicios y Aplicaciones de Internet*. España: Ideaspropias Editorial.
- 59) Rodríguez, R. (26 de noviembre de 2012). *Los 5 valores de la programación extrema (XP)*. Recuperado de:
<http://www.pmoinformatica.com/2012/11/los-5-valores-de-la-programacion.html>
- 60) Rosemberg, M. (2000). *Manejo de ganado bovino de carne y de doble propósito*. Lima, Perú: Proyecto TTA.
- 61) Sánchez, A. (2010). *Parámetros Reproductivos de Bovinos en Regiones Tropicales de México*. Tesis de grado no publicado, Universidad Veracruzana, Veracruz, México.
- 62) Sommerville, I. (2005). *Ingeniería del software (7ª Ed.)*. Madrid, España: Pearson Educación.
- 63) Stallings, W. (2004). *Comunicaciones y Redes de Computadores (7ª Ed.)*. Madrid, España: Pearson Education.
- 64) Taboada J. y Cotos J. (2005). *Sistema de Información Medioambiental*. Madrid, España: Carlos Iglesias.
- 65) Tamayo, M. (2004). *El proceso de la investigación científica*. México D.F., México: Limusa.
- 66) Terrero R., y Morejón R. (2014, Septiembre - Diciembre). Sistema para la gestión y control del ganado vacuno y la inseminación artificial. *Avanzada Científica*, (17), p. 11.
- 67) Universo el periódico de los universitarios (11 de Junio de 2012). *Programación extrema: "Metodología para desarrollo ágil de aplicaciones"*, p. 8. Recuperado de:
<http://www.uv.mx/universo/486/pdf/Universo%20486.pdf>
- 68) Weitzenfeld, A. (2005). *Ingeniería de software Orientada a Objetos con UML, Java e Internet*. México, D.F., México: Cengage Learning.
- 69) Xhafa, F., Vásquez, P., Gómez, J., Molinero, X. y Martín, A. (2006). *Programación en C++ para Ingenieros*, Madrid, España: Thomson.

- 70) Zambon, G y Sekler, M. (2007). *Beginning JSP, JSF and Tomcat Web Development from Novice to Professional*. United States: Apress.

ANEXO A

CUESTIONARIOS PARA LA ENTREVISTA A LOS PRODUCTORES PECUARIOS RESPONSABLES DEL MANEJO DE REPRODUCCIÓN DE GANADO BOVINO Y PRODUCCIÓN LÁCTEA

A. IDENTIFICACIÓN

Nombre del productor: _____

Profesión: _____

Fecha: _____ Número de encuesta: _____

B. REPRODUCCIÓN DE GANADO BOVINO

1. ¿Qué tareas realiza en el Centro de Producción?

.....
.....
.....
.....
.....
.....
.....
.....

2. ¿Cuándo inicia la crianza de ganado bovino?

.....
.....
.....

3. ¿En qué consiste la reproducción de ganado bovino?

.....
.....
.....

4. ¿Cuáles son las prácticas que se tiene en la reproducción de ganado bovino?

.....
.....
.....

5. ¿De qué manera monitorean la reproducción de ganado bovino?
.....
.....
.....
6. ¿Por qué es fundamental llevar los registros y controles de reproducción?
.....
.....
.....
7. ¿Cuáles son los criterios para seleccionar a las vaquillas de reemplazo?
.....
.....
.....
8. ¿Cuáles son los criterios para seleccionar a los toretes de reemplazo?
.....
.....
.....
9. ¿Bajo qué criterios se entoran o sirven a las vaquillas?
.....
.....
.....
10. ¿En qué momento se inicia el empadre de las vacas?
.....
.....
.....
11. ¿El empadre es realizado por monta natural o inseminación artificial?
.....
.....
.....
12. ¿Qué datos se registran en la fase del empadre?
.....
.....
.....
13. ¿Qué datos registran al momento del diagnóstico de preñez?
.....
.....
.....

14. ¿Cuáles son los riesgos que atraviesa la vaca preñada?
.....
.....
15. ¿Cuáles son las causa los abortos?
.....
.....
16. ¿Cuánto dura la preñez?
.....
.....
17. ¿Cuáles son las prácticas de manejo en la fase del parto?
.....
.....
18. ¿Qué datos se registran después del parto?
.....
.....
19. ¿Bajo qué criterios se identifican a los bovinos?
.....
.....
20. ¿En qué momento se desteta al ternero?
.....
.....
21. ¿Cuáles son las prácticas que se deben de tener con el ternero destetado?
.....
.....
22. ¿Qué datos se registran después del destete?
.....
.....
23. ¿Cuáles son los criterios para eliminar vientres improductivos?
.....
.....

24. ¿Cómo está compuesto o categorizado el rebaño de ganado bovino?

.....
.....
.....

25. ¿Cuáles son los criterios de reemplazo de las vacas reproductoras?

.....
.....
.....

26. ¿Cuáles son los criterios de reemplazo de los toros?

.....
.....
.....

27. ¿Cómo determinan el nivel de desempeño reproductivo del hato?

.....
.....
.....

28. ¿Qué nos indica la edad a la pubertad?

.....
.....
.....

29. ¿Qué nos indica la edad al primer servicio?

.....
.....
.....

30. ¿Qué nos indica la edad al primer parto?

.....
.....
.....

31. ¿Qué nos indica los días abiertos?

.....
.....
.....

32. ¿Qué nos indica el servicio por concepción?

.....
.....
.....

33. ¿Qué nos indica el intervalo entre partos?

.....
.....
.....

C. PRODUCCIÓN LÁCTEA

1. ¿En qué consiste la producción láctea de ganado bovino?

.....
.....
.....

2. ¿Cuándo se inicia la producción de leche?

.....
.....
.....

3. ¿Cuáles son las prácticas que se tiene en la producción láctea de ganado bovino?

.....
.....
.....

4. ¿De qué manera monitorean la producción láctea de ganado bovino?

.....
.....
.....

5. ¿Cuántas veces al día se realiza el ordeño?

.....
.....
.....

6. ¿Cuándo es interrumpida la producción láctea?

.....
.....
.....

7. ¿Qué datos se registran en el secado de la vaca?

.....
.....
.....

8. ¿En qué consiste el secado de la vaca?

.....
.....
.....

9. ¿Qué sucede si no se seca a la vaca?

.....
.....
.....

10. ¿Cuándo termina la campaña de producción de leche en una vaca?

.....
.....
.....

11. ¿Cómo es medida la producción láctea?

.....
.....
.....

12. ¿Qué nos indica la producción individual?

.....
.....
.....

13. ¿Qué nos indica las vacas en ordeño?

.....
.....
.....

14. ¿Qué nos indica la duración de lactancia?

.....
.....
.....

D. DATOS ADICIONALES

1. ¿En qué consiste la alta de bovinos?

.....
.....
.....

2. ¿Cuáles son las prácticas que se tiene en el manejo sanitario de ganado bovino?

.....
.....
.....
.....
.....
.....

3.	¿De qué manera monitorean el manejo sanitario de ganado bovino?

4.	¿Cuándo inicia el programa de vacunación?

5.	¿Cuándo inicia el programa de desparasitación?

6.	¿Cuáles son las enfermedades que se presentan con mayor frecuencia?

Tabla N° A.1: Entrevista a los productores pecuarios, sobre la reproducción de ganado bovino y la producción láctea.

ANEXO B

CUESTIONARIOS PARA LA ENCUESTA A LOS PRODUCTORES PECUARIOS RESPONSABLES DEL MANEJO DE REPRODUCCIÓN DE BOVINO Y PRODUCCIÓN LÁCTEA.

A. IDENTIFICACIÓN

Nombre del productor: _____

Cargo: _____

Fecha: _____ Número de encuesta: _____

Marque con una X y conteste el presente cuestionario según su criterio.

B. REPRODUCCIÓN DE GANADO BOVINO

1. ¿Llevan el registro de celo?

Siempre () A veces () Nunca ()

¿Qué datos se registran o deben registrarse?

.....
.....

2. ¿Llevan el registro de empadre?

Siempre () A veces () Nunca ()

¿Qué datos se registran o deben registrarse?

.....
.....
.....

3. ¿Llevan el registro de preñez?

Siempre () A veces () Nunca ()

¿Qué datos se registran o deben registrarse?

.....
.....
.....

4. ¿Llevan el registro de parto?
 Siempre () A veces () Nunca ()
 ¿Qué datos se registran o deben registrarse?

5. ¿Llevan el registro de crecimiento y desarrollo del ganado bovino?
 Siempre () A veces () Nunca ()
 ¿Qué datos se registran o deben registrarse?

6. ¿Evalúan el nivel de desempeño reproductivo del hato?
 Siempre () A veces () Nunca ()
 ¿De qué manera se evalúa o debe de evaluarse?

C. PRODUCCIÓN LÁCTEA

1. ¿Llevan el registro de producción láctea?
 Siempre () A veces () Nunca ()
 ¿Qué datos se registran o deben registrarse?

2. Cómo es el tipo de ordeño:
 Manual () Mecánico () Ambos ()

3. ¿Cuáles son los turnos de ordeño?
 Mañana () Tarde () Noche ()

4. ¿La leche ordeñada es analizada?
 Si () No ()
 ¿Qué datos se registran o deben registrarse?

5. ¿Existe un control en cuanto a la calidad de leche?
 Si () No ()
 De que tipo

Tabla N° B.1: Encuesta a los productores pecuarios, sobre la reproducción de ganado bovino y la producción láctea.

ANEXO C

REGISTRÓ GENERAL DE REPRODUCCIÓN

	NOMBRE	N° ARETE	EDAD DE LA VACA		EDAD AL PRIMER PARTO		N° DE LACTACIÓN
			Meses	Días	Meses	Días	
1							
2							
3							
4							
5							
6							
8							
9							
10							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
29							
30							
31							
32							
33							
34							
35							
36							
37							
38							

ANEXO D

MATRIZ DE OPERACIONALIZACIÓN DE VARIABLE Primera variable: Reproducción de ganado bovino								
DEFINICIÓN	DIMENSIÓN	INDICADOR	ÍTEM	ESCALA DE MEDICIÓN	TIPO	POSIBLE RESULTADO	INTERPRETACIÓN	INSTRUMENTO DE MEDICIÓN
La reproducción de ganado bovino es un conjunto de procesos relacionados a la generación de nuevos terneros, que incluyen; la gametogénesis, fecundación, gestación y parto, y es la actividad esencial para iniciar la producción de leche.	Pubertad	Edad Peso	¿Cuál es la edad y peso promedio de los terneros cuando alcanzan la pubertad?	Escala de intervalo	Cuantitativa continua	< 9 meses y > 255 kg	Valor pésimo	Análisis documental
						9 – 10 meses y 255 – 277 kg	Rango Aceptable	
						> 10 meses y < 277 kg	Valor pésimo	
	Primer servicio	Edad Peso	¿Cuál es la edad y peso promedio de las vaquillas cuando se le practica el servicio?	Escala de intervalo	Cuantitativa continua	< 15 meses y > 370 kg	Valor pésimo	Análisis documental
						15 – 18 meses y 370 – 414 kg	Rango Aceptable	
						> 18 meses y < 414 kg	Valor pésimo	
	Días abiertos	Fecha de parto Fecha de concepción	¿Cuántos días en promedio, transcurren a partir de la fecha de parto a la fecha de concepción?	Escala de intervalo	Cuantitativa discreta	85 a 110 días	Valor óptimo	Análisis documental
						110 – 140 días	Rango Aceptable	
						> 140 días	Valor pésimo	
	Servicios por concepción	Número de servicios Número de vacas preñadas	¿Cuántos servicios en promedio se realizan para preñar a la vaca?	Escala de intervalo	Cuantitativa continua	< 1.7	Valor óptimo	Análisis documental
						1.7 – 2.5	Rango Aceptable	
						> 2.5	Valor pésimo	

	Intervalo entre partos	Fecha de partos	¿Cuál es el promedio de días entre parto y parto?	Escala de intervalo	Cuantitativa discreta	< 390 días	Valor óptimo	Análisis documental
						390 – 420 días	Rango Aceptable	
						> 420 días	Valor pésimo	

Tabla N° D.1: Matriz de operacionalización de la primera variable.

MATRIZ DE OPERACIONALIZACIÓN DE VARIABLE								
Segunda variable: Producción láctea								
DEFINICIÓN	DIMENSIÓN	INDICADOR	ÍTEM	ESCALA DE MEDICIÓN	TIPO	POSIBLE RESULTADO	INTERPRETACIÓN	INSTRUMENTO DE MEDICIÓN
La producción láctea se inicia únicamente cuando nace el ternero, durante las primeras 8 semanas la producción de leche se incrementa y luego disminuye gradualmente hasta el momento del secado, exactamente a los 7 meses de gestación o cuando baja la producción diaria de leche.	Producción individual	Litros de leche por día	¿Cuántos litros de leche en promedio producen las vacas por campaña?	Escala de intervalo	Cuantitativa continua	> 8500 kg	Valor óptimo	Análisis documental
						7500 – 8500 kg	Rango Aceptable	
						< 7500 kg	Valor pésimo	
	Duración de lactancia	Parto reciente Próximo parto	¿En promedio, cuantos días persiste la producción de leche por campaña?	Escala de intervalo	Cuantitativa discreta	< 306 días	Valor óptimo	Análisis documental
						306 – 368 días	Rango Aceptable	
						> 368 días	Valor pésimo	

Tabla N° D.2: Matriz de operacionalización de la segunda variable.