

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

ESCUELA DE INFORMÁTICA E INTELIGENCIA ARTIFICIAL

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

SISTEMA WEB PARA EL SEGUIMIENTO Y CONTROL DE LOS PROCESOS DE CRIANZA DE COBAYAS DE LA GRANJA FAMILIAR QUISHPE.

ELVIS FABIÁN QUISHPE PROAÑO

TUTOR: SANTIAGO DAMIAN QUISHPE MORALES

IBARRA – ECUADOR

JULIO, 2024

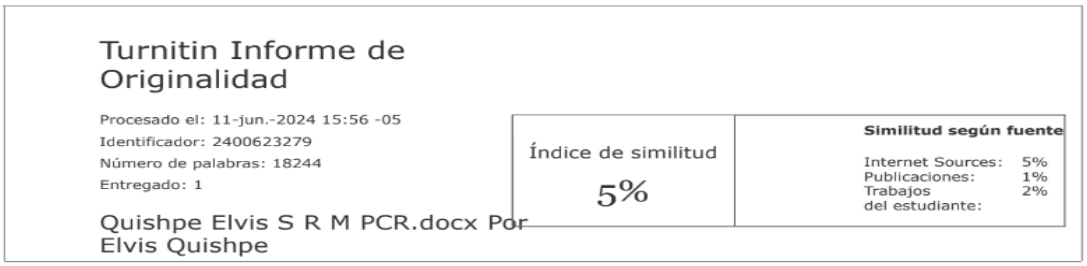
Ibarra, 4 de Julio del 2024

CERTIFICACIÓN TUTOR

En mi calidad de Tutor del Trabajo de Titulación titulado: Sistema web para el seguimiento y control de los procesos de crianza de cobayas de la granja familiar Quishpe**.** presentado por el estudiante Quishpe Proaño Elvis Fabian con cédula de ciudadanía N° 1003113782, para obtener el Título de Ingeniera en Tecnología de la Información.

Certifico que el trabajo cumple con todos los parámetros establecidos, mediante el cual el estudiante demuestra el desarrollo de competencias en el campo de conocimiento de su profesión con un nivel de argumentación coherente, para ser sometido a la evaluación por parte de los lectores.

Adicionalmente, se adjunta el certificado de porcentaje de originalidad de TURNITIN.



Turnitin Informe de Originalidad

Procesado el: 11-jun.-2024 15:56 -05
Identificador: 2400623279
Número de palabras: 18244
Entregado: 1

Quishpe Elvis S R M PCR.docx Por Elvis Quishpe

Índice de similitud	Similitud según fuente	
5%	Internet Sources:	5%
	Publicaciones:	1%
	Trabajos del estudiante:	2%



< 1% match (Internet desde 01-may.-2024)
https://repositorio.unc.edu.pe/handle/20.500.14074/5905?show=full

< 1% match (Internet desde 09-jun.-2020)
https://es.scribd.com/document/410220995/TESIS-ROMERO-PALACIOS-AMAPARO-DE-JESUS-pdf

< 1% match (Internet desde 01-mar.-2024)
http://Repositorio.Upagu.Edu.Pe/bitstream/handle/UPAGU/3179/TESIS.pdf?isAllowed=y&sequence=1

< 1% match (Internet desde 21-jun.-2021)
https://www.scribd.com/document/274226317/Tesis-Kharym-Zapata

< 1% match (Internet desde 31-may.-2023)
https://extranet.wmo.int/edistrib_exped/grp_has/_es/Archives%202011_2019/2016/2016_02_02-26-PR-6890-OBS-WIGOS-PO-OSCAR_es.pdf

(f): _____
Mgs. Santiago Damian Quishpe Morales
TUTOR DE TRABAJO
C.C.: 1002697223

PÁGINA DE APROBACIÓN DEL TRIBUNAL

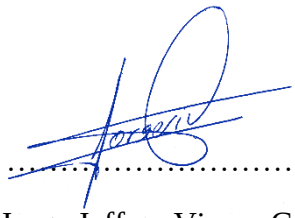
El tribunal examinador, aprueba el presente trabajo en nombre de la Pontificia Universidad Católica del Ecuador Ibarra:



(f): ……

Mgs. Santiago Damian Quishpe Morales

C.C.: 1002697223



(f): ……

Mgs. Jorge Jeffrey Vivero Garcia

C.C.: 1002061420



(f): ……

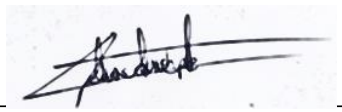
Mgs. Diego Raúl Mafla Rivadeneira

C.C.: 1001698644

ACTA DE CESIÓN DE DERECHOS

Yo, *Elvis Fabian Quishpe Proaño*, declaro conocer y aceptar la disposición del Art. 165 del Código Orgánico de Economía Social de los Conocimientos, Creatividad e Innovación, que manifiesta textualmente: "Se reconoce facultad de los autores y demás titulares de derechos de disponer de sus derechos o autorizar las utilizaciones de sus obras o prestaciones a título gratuito y oneroso, según las condiciones que determinen. Esta facultad podrá ejercerse mediante licencias libres, abiertas y otros modelos alternativos de licenciamiento o la renuncia".

Ibarra, 4 de Julio del 2024

(f): ___  ___

Quishpe Proaño Elvis Fabian.

C.C.: 1003113782

AUTORIA

Yo, *Elvis Fabian Quishpe Proaño*, portador de la cedula de ciudadanía 1003113782, declaro que el presente trabajo de investigación es de total responsabilidad del autor, y eximo expresamente a la Pontificia Universidad Católica del Ecuador Ibarra de posibles reclamos o acciones legales.



(f): …….

Elvis Fabian Quishpe Proaño

C.C.: 1003113782

DEDICATORIA Y AGRADECIMIENTOS

Quiero expresar mi más profunda gratitud a las personas que han sido fundamentales en mi camino académico y en la realización de esta tesis:

A mis padres, por su amor incondicional, su apoyo constante y los sacrificios que han hecho para que yo pueda alcanzar mis metas. Su ejemplo de perseverancia y trabajo duro ha sido mi mayor inspiración.

A mis hermanos, por su compañía, comprensión y ánimo durante todo este proceso. Sus palabras de aliento y su confianza en mí han sido invaluables en los momentos difíciles.

A mis tíos, por su interés en mi desarrollo profesional y por brindarme siempre palabras de sabiduría y consejos valiosos. Su respaldo y cariño han sido una fuente de motivación adicional.

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN TUTOR ..... **ii**
PÁGINA DE APROBACIÓN DEL TRIBUNAL ..... **iii**
ACTA DE CESIÓN DE DERECHOS ..... **iv**
AUTORIA ..... **v**
DEDICATORIA Y AGRADECIMIENTOS ..... **vi**
ÍNDICE DE CONTENIDOS ..... **vii**
ÍNDICE DE TABLAS ..... **x**
ÍNDICE DE FIGURAS ..... **xi**
RESUMEN ..... **xiii**
ABSTRACT ..... **xiv**
INTRODUCCIÓN **11**
CAPÍTULO 1 ..... **13**
ESTADO DEL ARTE **13**
1.1. MARCO TEÓRICO 13
1.1. Crianza de cobayas (cuyes): 13
1.2. Estado actual 13
1.3. Razas y líneas genéticas 13
1.4. Manejo reproductivo 14
1.5. Instalaciones y equipamiento para la crianza 15
1.2.1. Procesos de crianza: 15
1.2.1.1. Ciclo de producción 15
1.2.1.2. Control de inventario y registros 16
1.3.1. Sistemas de información y bases de datos: 17
1.3.1.1. Modelado de datos y diseño de bases de datos relacionales 17
1.3.1.2. Administración de bases de datos (MySQL) 17
1.3.1.3. Seguridad y control de acceso 18
1.4.1. Desarrollo web: 19
1.4.1.1. Lenguajes de programación web (HTML, CSS, JavaScript) 19
1.4.1.2. Protocolos y tecnologías web (HTTP, URLs, APIs) 20
1.4.1.3. Diseño y usabilidad de interfaces web 21
1.5.1. Framework Laravel: 21
1.5.1.1. Arquitectura Model-View-Controller (MVC) 21
1.5.1.2. Gestión de bases de datos con Laravel 22
1.5.1.3. Seguridad y autenticación 23
1.6.1. Gestión de proyectos de software: 24
1.6.1.1. Metodología ágil (Extreme Programming) 24

1.6.1.2. Análisis y gestión de requerimientos 24
1.6.1.3. Pruebas de software y control de calidad 25
1.2. TRABAJOS PREVIOS 26
CAPÍTULO II 28
MATERIALES Y MÉTODOS 28
2.1. Tipo de investigación 28
2.2. Técnicas para la recolección de datos. 28
2.2.1. Observación directa 28
2.2.2. Análisis de la entrevista semiestructurada. 30
2.3. Metodología para el desarrollo del sistema web 32
2.3.1. Metodología Ágil Extreme Programing (XP) 32
2.4. Fases o etapas de Extreme Programming (XP) y su aplicación en el proyecto. 33
2.4.1. Fase 1: Planificación del Proyecto 33
2.4.2. Fase 2: Diseño 39
2.4.2.1. Implementación de la lógica del negocio. 39
2.4.2.2. Arquitectura física de la aplicación: 40
2.4.2.3. Estructura de la Base de datos: 41
2.4.2.4. Casos de uso del sistema web: 42
2.4.2.5. Diagrama de secuencia 43
2.4.2.6. Diseño de la interfaz 45
2.4.3. Fase 3: Codificación 46
2.4.3.1. Herramientas de desarrollo 46
2.4.4. Fase 4: Pruebas 48
2.4.4.1. Casos de pruebas unitarias de las interfaces 48
2.4.4.2. Pruebas de carga con Apache Jmeter 55
CAPÍTULO III 56
RESULTADOS Y DISCUSIÓN 56
3.1. Directorio del código fuente 56
3.2. Interfaces de usuario 56
3.3. Resultados de las pruebas de las unitarias 64
3.4. Resultados de las pruebas de carga y rendimiento. 71
3.5. Seguridad, respaldos y mantenimiento 73
3.5.1. Seguridad 73
3.5.1.1. Seguridad en la contraseña de los usuarios. 73
3.5.1.2. Seguridad en almacenamiento de datos. 74
3.5.1.3. Seguridad en las rutas y controladores. 74
3.5.2. Respaldos 75
3.5.2.1. Respaldo de la Base de datos. 75
3.5.2.2. Respaldo del sistema web creado en Laravel. 75

3.5.3. Mantenimiento....76
3.5.3.1. Actualizaciones de software....76
3.5.3.2. Escalabilidad y rendimiento....77
CONCLUSIONES....78
RECOMENDACIONES....79
REFERENCIAS BIBLIOGRAFÍCAS....80

ÍNDICE DE TABLAS

Tabla 1 Se muestra los registros de Cría o Recrías.....29
Tabla 2 Se muestra el registro para las reproductoras de la granja. 29
Tabla 3 Se muestra el registro de engorde.....30
Tabla 4 Requerimientos Funcionales.....34
Tabla 5 Administración y asignación de roles.....35
Tabla 6 Jaula de reproductores y recrías.....35
Tabla 7 Cobayas reproductores y partos.....36
Tabla 8 Jaula de reproductores y recrías.....36
Tabla 9 Muertes y ventas de reproductores y crías. 37
Tabla 10 Estadísticas y reportes. 37
Tabla 11 Planificación de prioridades 38
Tabla 12 Distribución de sprints.....39
Tabla 13 Plantilla para pruebas de unitarias 48
Tabla 14 Caso de prueba unitaria de la interfaz de inicio de sesión.....49
Tabla 15 Caso de prueba unitaria de la interfaz de jaulas. 49
Tabla 16 Caso de prueba unitaria de la interfaz de cobayas reproductoras. 50
Tabla 17 Caso de prueba unitaria de la interfaz de partos.....51
Tabla 18 Caso de prueba unitaria de la interfaz de cobayas crías. 52
Tabla 19 Caso de prueba unitaria de la interfaz de ventas de cobayas.....53
Tabla 20 Caso de prueba unitaria de la interfaz de muertes de cobayas 54
Tabla 21 Caso de prueba unitaria de la interfaz de recrías.....55
Tabla 22 Resultado del interfaz de inicio de sesión. 64
Tabla 23 Resultado de la interfaz de jaulas. 65
Tabla 24 Resultado de la interfaz de cobayas reproductoras. 66
Tabla 25 Resultado de la interfaz de partos. 67
Tabla 26 Resultado de la interfaz de cobayas crías.....68
Tabla 27 Resultado de la interfaz de ventas de cobayas 69
Tabla 29 Resultado de la interfaz de muertes de cobayas.....70
Tabla 30 Resultado de la interfaz de recrías 70

ÍNDICE DE FIGURAS

Figura 1 Arquitectura del Sistema. 41
Figura 2 Modelado de la Base de Datos. 41
Figura 3 Administración de usuarios. 42
Figura 4 Administración del criadero. 43
Figura 5 Estadísticas y reportes de ventas, muertes y reproducción. 43
Figura 6 Diagrama de interacción del administrador 44
Figura 7 Diagrama de interacción del productor 45
Figura 8 Diseño de la interfaz 46
Figura 9 Aplicación Apache JMeter 55
Figura 10 Directorio del código fuente. 56
Figura 11 Interfaz de inicios de sesión. 57
Figura 12 Interfaz principal. 57
Figura13 Interfaz principal de las jaulas. 58
Figura 14 Interfaz de reproductores 59
Figura 15 Interfaz de recrías. 59
Figura 16 Interfaz de recrías. 60
Figura 17 Interfaz de crías del parto. 60
Figura 18 Interfaz de cobayas muertas. 61
Figura 19 Reporte PDF sobre muertes. 62
Figura 20 Interfaz de cobayas vendidas. 62
Figura 21 Reporte PDF sobre ventas 63
Figura 22 Interfaz de estadísticas de las cobayas. 64
Figura 23 Prueba de carga y rendimiento Login 71
Figura 24 Prueba de carga y rendimiento a la interfaz de usuarios. 72
Figura 25 Prueba de carga y rendimiento a la interfaz de jaulas. 72
Figura 26 Prueba de carga y rendimiento a la interfaz de reproductores. 72
Figura 27 Prueba de carga y rendimiento a la interfaz de partos. 73
Figura 28 Prueba de carga y rendimiento a la interfaz de crías. 73
Figura 29 Seguridad en las contraseñas de los usuarios. 73
Figura 30 Seguridad en almacenamiento de datos. 74
Figura 31 Seguridad en el controlador. 74
Figura 32 Seguridad en las rutas. 75
Figura 33 Respaldo de la base de datos 75
Figura 34 Respaldo del sistema web creado en Laravel. 76
Figura 35 Apache Netbeans IDE 13 76
Figura 36 Composer 77
Figura 37 Apache JMeter. 77

ÍNDICE DE ANEXOS

Anexo 1 Carta de aceptación 83

RESUMEN

La crianza de cuyes constituye una importante fuente de ingresos para las comunidades rurales de la región andina de Latinoamérica. Sin embargo, los productores enfrentan desafíos en el manejo y procesamiento de información clave relacionada con la reproducción, pesos y causas de mortalidad de las cobayas, la cual suele registrarse de forma manual en hojas de cálculo o papel, dificultando su análisis y toma de decisiones.

Reconociendo la necesidad de implementar soluciones tecnológicas en la crianza de animales, la presente investigación se enfocó en el desarrollo de un sistema web utilizando el framework Laravel para el seguimiento y control de los procesos de crianza de cobayas en la granja familiar Quishpe. Este sistema web fue desarrollado siguiendo la metodología ágil de Extreme Programming, lo que permitió un enfoque adaptable y la incorporación constante de retroalimentación del usuario.

Para la codificación se utilizó Visual Studio Code, mientras que MySQL Workbench se empleó para la gestión de la base de datos. Los lenguajes de programación seleccionados incluyeron HTML, JavaScript, PHP y CSS, brindando una solución integral que abarca la creación de interfaces de usuario atractivas y responsivas, la implementación de la lógica de negocios, el manejo de datos y la integración con la base de datos.

El sistema web desarrollado permite controlar y monitorear de manera eficiente la reproducción de las cobayas, administrar los recursos alimenticios y obtener una visión clara sobre las ventas y mortalidad de esta especie. Gracias a esta solución tecnológica, el productor puede visualizar en tiempo real sus ganancias y pérdidas, así como seleccionar los mejores ejemplares en base a la información proporcionada, asegurando la continuidad del negocio y el cumplimiento de los estándares de peso y características deseadas. Además, se llevaron a cabo pruebas de unitarias exhaustivas para validar el correcto funcionamiento del sistema web, garantizando que cumpliera con los requisitos y necesidades específicas de la granja.

Palabras clave: crianza de cuyes, sistema web, Extreme Programming, Laravel, MySQL Workbench, HTML, JavaScript, PHP, CSS, gestión de procesos, granja familiar, producción de cobayas, pesos, causas de mortalidad de las cobayas, responsivas.

ABSTRACT

The raising of guinea pigs constitutes an important source of income for rural communities in the Andean region of Latin America. However, producers face challenges in managing and processing key information related to the reproduction, weights, and causes of mortality of guinea pigs, which is usually recorded manually in spreadsheets or on paper, making analysis and decision-making difficult.

Recognizing the need to implement technological solutions in animal husbandry, the present research focused on the development of a web system using the Laravel framework for tracking and controlling the guinea pig breeding processes at the Quishpe family farm. This web system was developed following the agile Extreme Programming methodology, which allowed for an adaptable approach and constant incorporation of user feedback.

Visual Studio Code was used for coding, while MySQL Workbench was employed for database management. The selected programming languages included HTML, JavaScript, PHP, and CSS, providing a comprehensive solution that encompasses the creation of attractive and responsive user interfaces, the implementation of business logic, data management, and integration with the database.

The developed web system allows for efficient control and monitoring of guinea pig reproduction, management of food resources, and a clear overview of sales and mortality of this species. Thanks to this technological solution, the producer can visualize profits and losses in real-time, as well as select the best specimens based on the provided information, ensuring business continuity and compliance with desired weight standards and characteristics. Furthermore, comprehensive unit tests were carried out to validate the correct functioning of the web system, ensuring that it met the specific requirements and needs of the farm.

Keywords: guinea pig breeding, web system, Extreme Programming, Laravel, MySQL Workbench, HTML, JavaScript, PHP, CSS, process management, family farm, guinea pig production, weights, causes of guinea pig mortality, responsive.

INTRODUCCIÓN

El software juega un papel vital al proporcionar nuevas soluciones para mejorar la eficiencia, el rendimiento y la excelencia de los productos y servicios. Ha sido un impulsor clave de la innovación y la mejora continua de diversos procesos en diversos sectores e industrias. Desde la automatización de tareas repetitivas hasta la optimización de flujos de trabajos complejos, el software ha demostrado ser una herramienta indispensable para aumentar y mejorar la calidad en todo tipo de operaciones.

La producción animal destinada al consumo humano desempeña un papel fundamental en la alimentación global, proporcionando una fuente vital de proteínas, nutrientes y otros productos derivados. Desde tiempos ancestrales, la domesticación y cría de animales ha sido una actividad esencial para el sustento humano, y en la actualidad sigue siendo una parte integral de la economía y la sociedad en todo el mundo. (Caravaca, y otros, 2024)

La tecnología ha transformado radicalmente la producción animal en las últimas décadas, revolucionando la forma en la que se crían, manejan y gestionan los animales en granjas y exportaciones alrededor del mundo. (Herrera & Ortiz, 2022) Este cambio ha sido impulsado en gran parte por avances significativos en campos como la electrónica, la informática y la genética, que han permitido el desarrollo e implementación de soluciones innovadoras para mejorar la eficiencia, la productividad y el bienestar animal.

La crianza de cobayas es una actividad habitual en varias regiones del Ecuador y la introducción de una aplicación web puede mejorar la eficiencia y rentabilidad del negocio. La gestión manual de los procesos relacionados con la crianza de cobayas es complicada, susceptible a errores y difícil de adaptar a medida que la granja aumenta en tamaño y producción.

En una granja de crianza de cobayas, la identificación y selección de los mejores ejemplares puede complicarse debido al elevado número de animales que se generan en un lapso determinado. Esta tarea, realizada de forma manual, implica un considerable consumo de tiempo y esfuerzo por parte de los productores. Sin embargo, la introducción de un sistema tecnológico podría brindar una solución eficiente a esta problemática.

La granja familiar Quishpe, dedicada a la crianza de cobayas, se enfrenta a desafíos diarios en la gestión eficiente de sus procesos de producción. La falta de herramientas

tecnológicas adecuadas para el seguimiento y control de la crianza de estos animales dificulta la optimización de recursos y la garantía de bienestar de los mismos, Esta investigación consiste en el desarrollo de una plataforma tecnológica que proporciono una solución integral para la gestión y monitoreo de los procesos de crianza de cobayas en la granja familiar Quishpe.

Para tal fin se plantearon los siguientes objetivos:

Objetivo General

- Desarrollar un sistema web para el seguimiento y control de los procesos de crianza de cobayas de la granja familiar Quishpe utilizando las herramientas de Laravel

A partir de esta premisa, se desglosan los siguientes objetivos secundarios que se irán alcanzando gradualmente a medida que se lleve a cabo la investigación:

- Analizar la literatura actual sobre uso de la tecnología en criaderos de animales domésticos mediante la revisión de material científico para fundamentar la investigación.
- Identificar los procesos claves en la crianza de cuyes que deberán automatizarse a través del sistema web.
- Diseñar un sistema web para control la reproductividad, mortalidad y ventas que implican en la crianza de cobayas.
- Validar la eficiencia del software mediante pruebas de rendimiento y carga de software para evaluar el correcto funcionamiento y almacenamiento de información.

El documento consta de tres capítulos. El primero cubre la investigación bibliográfica y los conceptos clave, estableciendo el marco teórico. El segundo se centra en las herramientas tecnológicas y el proceso de desarrollo mediante la metodología XP. El tercero presenta los resultados finales, incluyendo una descripción detallada de la aplicación desarrollada.

CAPÍTULO 1
ESTADO DEL ARTE

La crianza de cobayas, conocidas también como cuyes, ha sido una actividad tradicional y relevante en el contexto de la agricultura familiar en diversas regiones. Estas especies representan una fuente importante de proteína y sustento económico para numerosas familias dedicadas a su producción. Sin embargo, la gestión eficiente de los recursos y el control adecuado de los procesos de crianza constituyen desafíos críticos que pueden impactar directamente en la productividad y la rentabilidad de esta actividad.

En este escenario, el desarrollo de una aplicación web para el seguimiento y control de los procesos de crianza de cobayas se ha convertido en una necesidad apremiante. Esta herramienta tecnológica tiene el potencial de optimizar la gestión de los recursos, mejorar el monitoreo de las etapas clave de la producción y facilitar la toma de decisiones informadas por parte de los productores.

1.1. MARCO TEÓRICO

1.1. Crianza de cobayas (cuyes):

1.2. Estado actual

Las cobayas son animales originarios de América del Sur y se encuentran en muchos países, incluidos Ecuador, Perú, Bolivia y Colombia. Este roedor es un alimento muy nutritivo y es consumido principalmente por población rural de escasos recursos. Además de su valor nutricional, la cría de cuyes puede proporcionar ingresos adicionales y oportunidades de empleo para mujeres y niños en las comunidades agrícolas. (Antonio, 2014)

1.3. Razas y líneas genéticas

Las cobayas o cuyes pueden ser clasificadas en función de diversos criterios, como su conformación física, la forma y longitud de su pelaje, así como las tonalidades del mismo. A lo

largo del tiempo, los criadores han seleccionado ejemplares que cumplen con determinadas condiciones reproductivas, mejorando gradualmente las líneas genéticas hasta obtener diferentes razas o especies adaptadas a distintos propósitos. (Antonio, 2014)

Algunas de estas razas se caracterizan por poseer una excelente masa muscular y niveles de grasa óptimos, lo que las hace ideales para la producción de carne. Otras, por su parte, destacan por su alta capacidad reproductora, lo que las convierte en opciones valiosas para la cría y reproducción. Además, existen razas que se han adaptado de manera excepcional a diversos climas y condiciones ambientales, mostrando una mayor resistencia a enfermedades. (Antonio, 2014)

Conocer sobre las diferentes razas y líneas de cobayas es fundamental, ya que esta información permite determinar diversos factores relacionados con su producción, ganancia de peso y fecundidad. Tener este conocimiento posibilita mejorar la genética de las crías, seleccionando a las mejores especies con mayor resistencia a enfermedades. Esto contribuye a optimizar la calidad y productividad del criadero.

1.4. Manejo reproductivo

El proceso de reproducción es una función biológica fundamental en todos los seres vivos, que les permite generar descendencia. En el ámbito agropecuario, esta reproducción puede tener diferentes fines productivos, como el mejoramiento genético, el aumento de peso en un determinado tiempo o el incremento de la capacidad reproductora, con el objetivo final de mejorar la productividad alimentaria de la especie. (Gabriela, 2020)

En el caso de la crianza de cobayas, existen varios factores clave a considerar al momento de realizar el emparejamiento para la reproducción. Es necesario conocer el peso adecuado de los machos y hembras, así como los tiempos óptimos para el destete, el crecimiento y el engorde. Además, se debe tener en cuenta el método de crianza y apareamiento más apropiado, y la nutrición que se proporcionará a los animales durante el período de reproducción. (Gabriela, 2020)

Otro aspecto importante es el conocimiento del comportamiento de las camadas y las posibles enfermedades que puedan afectar a los galpones y jaulas individuales. Esto permite tomar las medidas preventivas necesarias y garantizar un ambiente saludable para la cría de cobayas. (Gabriela, 2020)

Conocer los datos importantes de las cobayas permite mejorar factores y automatizar los procesos de recolección y procesamiento de información. Al criar esta especie, se puede seleccionar a los mejores individuos considerando datos como ganancia de peso, tiempo de engorde y tiempo de venta o saca. Estos datos posibilitan determinar el momento óptimo para vender las cobayas y recuperar los ingresos, optimizando así la productividad y rentabilidad del criadero.

1.5. Instalaciones y equipamiento para la crianza

Una adecuada infraestructura y equipamiento en el criadero de cobayas es fundamental, ya que contribuye a mejorar la producción, la salud y el bienestar de estos animales. La implementación de gazaperas, bebederos y una alimentación balanceada permite que las cobayas se desarrollen de manera óptima y tengan una crianza más eficiente, optimizando el tiempo de crecimiento. (Mauricio & Rueda, 2021)

Además, es crucial monitorear la temperatura del galpón y garantizar su seguridad para evitar la presencia de roedores externos. Estos roedores pueden ocasionar pérdidas significativas y representar un riesgo de transmisión de enfermedades al criadero. Mantener un entorno adecuado y condiciones sanitarias óptimas es clave para el éxito de la crianza de cobayas. (Mauricio & Rueda, 2021).

Mejorar las instalaciones y equipamientos en la crianza de cobayas es fundamental, ya que garantiza que las crías y reproductores mantengan un buen bienestar, lo que se traduce en una mayor producción de gazapos y, por ende, mayores ingresos para la granja. Conocer sobre la seguridad de las instalaciones es crucial, pues evita la entrada de enfermedades y plagas exteriores. Además, contar con equipamientos adecuados, como gazaperas, alimentadores y bebederos, permite distribuir de manera óptima los nutrientes a las cobayas, potenciando su desarrollo y productividad.

1.2.1. Procesos de crianza:

1.2.1.1. Ciclo de producción

En la crianza de cobayas, es fundamental conocer los datos clave relacionados con su proceso reproductivo. Estas especies tienen la capacidad de adaptarse a temperaturas que oscilan entre los 38 y 39 grados centígrados. Su expectativa de vida se extiende desde los 6 hasta los 8 años, y su

etapa reproductiva se prolonga aproximadamente hasta los 2 años, con la posibilidad de tener entre 4 y 5 partos por reproductora. (Xavier et al., 2024)

En cuanto a la gestación, los gazapos (crías) tienen un período de gestación que puede variar entre 2 y 3 meses, dependiendo del tamaño de la camada. Una vez nacidos, alcanzan su madurez reproductiva en un lapso de 30 a 35 días, momento en el cual deben cumplir con un peso específico para poder ser apareados. (Xavier et al., 2024)

Otro aspecto relevante es el celo post-parto que experimentan las reproductoras, lo que permite que el macho pueda preñarlas nuevamente poco después del alumbramiento. Esta característica contribuye a maximizar la productividad reproductiva de las cobayas. (Xavier et al., 2024)

Es importante tener en cuenta estos datos reproductivos, como la temperatura de adaptación, la expectativa de vida, la etapa reproductiva, el período de gestación, el tiempo de madurez de los gazapos, el peso requerido para el apareamiento y el celo post-parto, ya que son factores clave que influyen directamente en el éxito y la rentabilidad de la crianza de estas especies. (Xavier et al., 2024)

1.2.1.2. Control de inventario y registros

En la investigación, se destaca la importancia de implementar un sistema de producción que permita llevar registros detallados, los cuales brindarán información valiosa sobre los aspectos más relevantes de la crianza de cobayas. Uno de los datos clave a registrar es la fecha de empadre y de los partos, lo que permitirá identificar a las hembras que presenten dificultades o retrasos en cuanto a fertilidad o alumbramiento. (Ortiz, & Armando, 2024)

Además, es fundamental llevar un control del número de partos y el tamaño de cada camada, ya que esto facilitará la selección de los ejemplares que se mantendrán en la granja y aquellos que serán destinados a la venta. En cuanto a las crías, se deben registrar datos como la fecha de destete, el sexo y el peso al nacer y al ser destetadas, lo que contribuirá a identificar a las mejores crías de cada camada. (Ortiz, & Armando, 2024)

1.3.1. Sistemas de información y bases de datos:

1.3.1.1. Modelado de datos y diseño de bases de datos relacionales

Una base de datos se define como una colección o repositorio de datos que refleja las interrelaciones existentes en el mundo real. Estos datos pueden ser utilizados por diferentes usuarios y aplicaciones. Una base de datos debe cumplir con ciertos parámetros de seguridad, donde la información debe ser íntegra, estar disponible y ser confidencial. Además, los datos no deben ser redundantes o repetitivos, a menos que sean imprescindibles (Piñeiro, 2024)

En cuanto a las relaciones, existen tres tipos principales: uno a uno, uno a muchos y muchos a muchos. El tipo de relación se establece en función del proceso de datos que se desea agregar. La relación uno a uno se utiliza cuando un registro pertenece al registro de otra tabla. La relación uno a muchos se emplea cuando un registro puede estar relacionado con uno o más registros de otra tabla. Por último, la relación muchos a muchos implica la creación de una tabla intermedia, donde un registro puede tener múltiples registros relacionados en otra tabla, y viceversa. (Piñeiro, 2024)

La creación de una base de datos es una tarea fundamental que permite optimizar el tiempo y la eficiencia en la recolección y gestión de datos. Este proceso implica el desarrollo de un modelo, el cual puede definirse como una representación abstracta de la realidad que se desea capturar en la base de datos. El modelo consta de un conjunto de símbolos, conceptos y reglas que permiten estructurar y organizar la información de manera lógica y coherente. (Piñeiro, 2024)

En el diseño de bases de datos, se distinguen tres tipos principales de modelos: conceptuales, lógicos y físicos. El modelo conceptual representa una visión general de los datos y sus interrelaciones, sin entrar en detalles técnicos. Por otro lado, el modelo lógico traduce el modelo conceptual a una estructura más detallada y específica, definiendo las entidades, atributos y relaciones de manera precisa. Finalmente, el modelo físico se enfoca en la implementación física de la base de datos en un sistema de gestión de bases de datos específico, considerando aspectos como el almacenamiento, el rendimiento y la optimización. (Piñeiro, 2024)

1.3.1.2. Administración de bases de datos (MySQL)

MySQL es un sistema de gestión de bases de datos relacionales ampliamente utilizado en el desarrollo de aplicaciones web y otros entornos de software. Este sistema de código abierto proporciona una interfaz gráfica de usuario (GUI) y un lenguaje de consulta estructurado (SQL) para la administración y manipulación de datos de manera eficiente. (Huillcen, Palomino, & Solís, 2022)

Una de las ventajas destacadas de MySQL es su capacidad para manejar modelos de bases de datos con relaciones complejas entre tablas. Además, ofrece un conjunto de sentencias SQL estándar, como INSERT, UPDATE, DELETE y SELECT, que permiten realizar operaciones de inserción, actualización, eliminación y consulta de datos de manera precisa y flexible (Huillcen, Palomino, & Solís, 2022).

MySQL también cuenta con características avanzadas, como la capacidad de combinar y encadenar múltiples sentencias SQL para obtener resultados más complejos y personalizados. Esta flexibilidad y potencia han contribuido a su amplia adopción en proyectos de diferentes escalas y sectores, convirtiéndolo en una herramienta esencial para el desarrollo de sistemas de información robustos y escalables (Huillcen, Palomino, & Solís, 2022)

1.3.1.3. Seguridad y control de acceso

En un sistema de gestión de bases de datos (SGBD), el servidor desempeña un rol fundamental en la administración de los privilegios de acceso y la gestión de usuarios. Este componente es capaz de otorgar permisos específicos a los usuarios, lo que les permite realizar diversas operaciones y peticiones sobre la base de datos. (Ibáñez, 2016)

El proceso de autenticación y autorización se lleva a cabo en dos etapas. La primera implica la comprobación de la identidad del usuario mediante el uso de un nombre de usuario y una contraseña. Una vez autenticado, el usuario puede acceder a la segunda etapa, donde se determinan los privilegios y operaciones que puede realizar sobre el SGBD. (Ibáñez, 2016)

El servidor también ofrece la posibilidad de asignar roles y niveles de acceso a las tablas de la base de datos. Esto permite controlar quién puede realizar cambios, como inserciones, actualizaciones o eliminaciones, y quién solo tiene acceso de lectura a los datos. (Ibáñez, 2016)

El control de acceso es esencial para garantizar la integridad y seguridad de la información almacenada en la base de datos. Este mecanismo verifica los privilegios de cada usuario y valida las transacciones que puede realizar en las tablas correspondientes. (Ibáñez, 2016)

Además, el SGBD implementa medidas de seguridad, como la creación de contraseñas y usuarios, para controlar y restringir el acceso a la base de datos. Estas medidas son gestionadas por el

administrador del sistema, quien determina los permisos y niveles de acceso adecuados para cada usuario o rol. (Ibáñez, 2016)

Al implementar una base de datos, es fundamental considerar la seguridad de la información y definir los usuarios con acceso a modificar estos datos. La creación de roles y la asignación de privilegios adecuados es crucial, ya que permite que los usuarios tengan acceso a la información necesaria y, de ser requerido, puedan manipularla de acuerdo con sus responsabilidades. Esto garantiza la continuidad del negocio y la protección de los datos, asegurando que solo las personas autorizadas puedan acceder y realizar cambios en el sistema.

1.4.1. Desarrollo web:

1.4.1.1. Lenguajes de programación web (HTML, CSS, JavaScript)

CSS (Cascading Style Sheets) es un lenguaje de estilos utilizado en el desarrollo web que permite aplicar estilos visuales y definir la presentación de los elementos HTML de una página web. La principal ventaja de CSS es que separa la estructura y el contenido (HTML) de la apariencia visual, lo que facilita la personalización y el mantenimiento del diseño. Mediante la selección de elementos por su identificador (id), clase o etiquetas, se pueden aplicar estilos específicos, como colores, fuentes, tamaños, posicionamiento y efectos visuales, mejorando la experiencia del usuario y la apariencia general del sitio web. (Condor & Solís, 2014)

HTML (HyperText Markup Language) es el lenguaje de marcado estándar utilizado para estructurar y dar forma al contenido de una página web. Una página HTML consta de tres partes principales: la cabecera (head), el cuerpo (body) y el pie (footer). En el cuerpo se pueden agregar elementos como títulos, párrafos, listas, imágenes y tablas, utilizando etiquetas específicas. Además, se pueden asignar identificadores únicos (id) y clases (class) a los elementos, lo que permite aplicar estilos CSS y manipularlos con JavaScript. HTML también ofrece la posibilidad de crear formularios para la recolección de datos del usuario, los cuales pueden ser procesados en el servidor. (Condor & Solís, 2014)

JavaScript es un lenguaje de programación interpretado que se ejecuta principalmente en el navegador web (cliente), pero también puede ser utilizado en el servidor. Su principal función es mejorar la interactividad y el dinamismo de las páginas web. Con JavaScript, se pueden seleccionar y manipular elementos HTML, responder a eventos (como clics o movimientos del ratón), validar formularios, realizar operaciones matemáticas y lógicas, y realizar solicitudes

asíncronas (AJAX) al servidor. Además, JavaScript cuenta con una amplia gama de bibliotecas y frameworks que facilitan el desarrollo de aplicaciones web modernas y complejas. (Condor & Solís, 2014)

Estos tres lenguajes (CSS, HTML y JavaScript) son fundamentales en el desarrollo web moderno y trabajan de manera complementaria para crear experiencias interactivas, atractivas y funcionales para los usuarios.

1.4.1.2. Protocolos y tecnologías web (HTTP, URLs, APIs)

HTTP (Hypertext Transfer Protocol) es el protocolo utilizado para la comunicación entre clientes (como navegadores web) y servidores en la World Wide Web. Se basa en un modelo de solicitud-respuesta, donde el cliente envía una solicitud al servidor y este responde con los datos solicitados. HTTP define varios métodos de solicitud, como GET (obtener recursos), POST (enviar datos al servidor), PUT (actualizar recursos) y DELETE (eliminar recursos), que indican la acción que se desea realizar. (Castillo, 2022)

Las URLs (Uniform Resource Locators) son cadenas de caracteres que proporcionan direcciones únicas para recursos en la web, como páginas, imágenes, archivos y servicios. Una URL consta de varios componentes, incluyendo el nombre de dominio, la ruta del recurso y, en algunos casos, parámetros de consulta. Las URLs permiten que los clientes localicen y accedan a los recursos de manera eficiente y estandarizada en Internet. (Castillo, 2022)

Las APIs (Interfaces de Programación de Aplicaciones) son conjuntos de reglas, protocolos y herramientas que facilitan la comunicación e interacción entre diferentes aplicaciones o sistemas. Las APIs definen métodos y formatos estandarizados para acceder y utilizar los servicios y funcionalidades ofrecidos por una aplicación. En el contexto web, las APIs se basan comúnmente en el protocolo HTTP, proporcionando acceso a los recursos y servicios a través de solicitudes y respuestas bien definidas. (Castillo, 2022)

Estos conceptos son fundamentales en el desarrollo de aplicaciones web modernas y en la integración de diferentes sistemas y servicios. HTTP proporciona el mecanismo de comunicación, las URLs permiten la localización y acceso a recursos, y las APIs facilitan la interoperabilidad y el intercambio de datos entre aplicaciones, promoviendo así el desarrollo de sistemas distribuidos y escalables. (Castillo, 2022)

Al desarrollar una página web, es importante conocer los diferentes protocolos que permiten transmitir información a través de las URLs o generar vistas. Métodos como GET, POST, DELETE y UPDATE facilitan la realización de cambios, creaciones o visualizaciones en el sistema. Además, es fundamental comprender el funcionamiento de las APIs, ya que esto garantiza un buen desempeño y escalabilidad de la aplicación. Contar con este conocimiento técnico permite crear una solución web robusta y adaptable a las necesidades del negocio.

1.4.1.3. Diseño y usabilidad de interfaces web

Las interfaces de usuario son un componente crucial en el desarrollo de aplicaciones y sitios web, ya que representan el punto de interacción directa con los usuarios finales. Para garantizar una experiencia satisfactoria, las interfaces deben cumplir con ciertos aspectos clave, como la organización y presentación clara de la información, lo que facilita la navegación y el acceso a los contenidos y funcionalidades. (Erika et al., 2020)

Uno de los aspectos fundamentales es el diseño responsivo, que implica que la interfaz se adapte de manera óptima a diferentes dispositivos y tamaños de pantalla, desde computadoras de escritorio hasta tabletas y dispositivos móviles. Esta capacidad de adaptación asegura que los usuarios puedan acceder y utilizar la aplicación o sitio web de manera fluida, independientemente del dispositivo que estén utilizando. (Erika et al., 2020)

Además, la usabilidad desempeña un papel primordial en el éxito de una interfaz de usuario. Una interfaz usable debe permitir un fácil aprendizaje de las funcionalidades y una navegación intuitiva, brindando al usuario una experiencia eficiente al realizar tareas o buscar contenido específico. Asimismo, debe ser agradable y satisfactoria, lo que implica una interacción sencilla y fluida, evitando frustraciones o confusiones innecesarias. (Erika et al., 2020)

1.5.1. Framework Laravel:

1.5.1.1. Arquitectura Model-View-Controller (MVC)

El patrón de diseño Modelo-Vista-Controlador (MVC) es una arquitectura de software ampliamente utilizada en el desarrollo de aplicaciones web y otros sistemas de software. Esta estructura organiza el código de manera modular y separada en tres componentes principales, lo que facilita la gestión, el mantenimiento y la escalabilidad del sistema. (Cíceri, 2019)

El Modelo representa la capa de acceso a datos y la lógica de negocio de la aplicación. En esta capa se definen las entidades o estructuras de datos que representan los objetos del dominio del problema, así como las reglas y operaciones asociadas a estas entidades. Comúnmente, los modelos interactúan con la base de datos para realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre los datos. (Cíceri, 2019)

La Vista es responsable de la presentación de la información al usuario final. En esta capa se define la interfaz de usuario mediante tecnologías como HTML, CSS y plantillas de renderizado, que permiten mostrar los datos obtenidos desde el Modelo de manera atractiva y comprensible para el usuario. (Cíceri, 2019)

Por último, el Controlador actúa como intermediario entre el Modelo y la Vista. Esta capa maneja la lógica de control de la aplicación, procesando las solicitudes entrantes, interactuando con los modelos para obtener o manipular datos, y enviando los datos necesarios a las vistas correspondientes para su renderizado y presentación al usuario. (Cíceri, 2019)

Al desarrollar un sistema web utilizando el framework Laravel, es fundamental comprender su funcionamiento, ya que esto permite optimizar tiempos y mantener un código ordenado y protegido. Laravel estructura los sistemas bajo el patrón Modelo-Vista-Controlador (MVC), donde el modelo representa la lógica y conexión con la base de datos, encargándose de la interacción con las diferentes tablas; el controlador implementa la lógica del negocio y consume los datos del modelo; y la vista presenta la información a los usuarios de manera visual y manejable, permitiéndoles manipular dichos datos.

1.5.1.2. Gestión de bases de datos con Laravel

En los frameworks de PHP modernos, se utiliza una capa de abstracción para facilitar la conexión y la interacción con bases de datos. En el caso de Laravel, esta capa de abstracción se conoce como Eloquent ORM (Object-Relational Mapping). Para establecer la conexión con la base de datos, es necesario configurar los parámetros correspondientes en el archivo config/database.php. Esta configuración permite conectarse no solo a un motor de base de datos específico, sino también a diferentes motores, ya que algunos pueden requerir parámetros adicionales para establecer la conexión de manera adecuada. (Cíceri, 2019)

Otra característica importante en Laravel son las migraciones de bases de datos. Las migraciones permiten versionar y controlar los cambios en la estructura de la base de datos mediante archivos PHP, los cuales se incorporan como parte del código fuente del proyecto. De esta manera, todos los desarrolladores involucrados tienen acceso a la información actualizada de la base de datos y pueden realizar modificaciones de manera controlada. (Cíceri, 2019)

Laravel proporciona herramientas que facilitan la conexión y manipulación de la base de datos. A través de las migraciones, se puede crear el modelo de datos y, mediante un simple comando, cargar toda la estructura en la base de datos. Adicionalmente, el ORM (Mapeo Objeto-Relacional) de Laravel permite manejar los datos de manera eficiente una vez que se han establecido las relaciones entre los diferentes modelos del sistema web. Estas funcionalidades simplifican y aceleran el desarrollo, al tiempo que mantienen la integridad y coherencia de la base de datos.

1.5.1.3. Seguridad y autenticación

La seguridad es un aspecto fundamental en el desarrollo de aplicaciones web, ya que permite protegerlas contra vulnerabilidades y garantizar que solo los usuarios autorizados tengan acceso a los recursos. En Laravel, se ofrecen diversas soluciones para abordar la autenticación y la seguridad. (Cíceri, 2019)

En cuanto a la autenticación, Laravel proporciona métodos basados en sesiones, tokens de acceso y autenticación OAuth, los cuales pueden configurarse en las rutas, controladores y vistas de la aplicación. Estas opciones permiten implementar mecanismos robustos de autenticación y control de acceso. (Cíceri, 2019)

Por otro lado, Laravel cuenta con medidas de seguridad para mitigar ataques como inyecciones SQL y scripting entre sitios (XSS). Una de estas medidas es el uso de consultas preparadas, las cuales brindan protección contra la inyección de código malicioso en las consultas a la base de datos. Además, Laravel genera automáticamente tokens CSRF (Cross-Site Request Forgery) para proteger contra ataques de este tipo. (Cíceri, 2019)

1.6.1. Gestión de proyectos de software:

1.6.1.1. Metodología ágil (Extreme Programming)

La metodología ágil es un enfoque utilizado en el desarrollo de software que se centra en la entrega iterativa e incremental de soluciones de alta calidad. Esta metodología no se aplica a todo tipo de proyectos, sino que es más adecuada para proyectos de pequeña o mediana escala, así como para aquellos en los que los requisitos pueden cambiar con frecuencia. (Lainez, 2015)

Un aspecto clave de las metodologías ágiles es la importancia que se le da a la comunicación efectiva entre el equipo de desarrollo y el cliente. Esta comunicación constante permite abordar los detalles del proyecto de manera más precisa y facilita la incorporación de modificaciones cuando sean necesarias. Esto permite que el software evolucione de manera segura y ágil, adaptándose a los cambios en los requisitos o las necesidades del cliente. (Lainez, 2015)

Las metodologías ágiles promueven la entrega incremental de software funcional, lo que permite una retroalimentación temprana y una mayor flexibilidad para realizar ajustes o incorporar nuevas características. Además, fomentan la colaboración estrecha entre los miembros del equipo de desarrollo, lo que facilita la resolución de problemas y la toma de decisiones informadas. (Lainez, 2015)

La implementación de una metodología adecuada permite desarrollar un software de calidad, siguiendo un proceso estructurado. Una buena comunicación con el usuario es fundamental, ya que permite dividir el proyecto en fases, modificar e incorporar nuevas funcionalidades a medida que se van resolviendo los problemas. Esto garantiza que el sistema final se ajuste a las necesidades reales del usuario y se adapte a los cambios que puedan surgir durante el desarrollo. La aplicación de una metodología ágil, como Scrum o Extreme Programming, facilita la colaboración, la entrega iterativa de funcionalidades y la retroalimentación constante, asegurando la calidad y la satisfacción del cliente.

1.6.1.2. Análisis y gestión de requerimientos

El análisis de requerimientos es un proceso fundamental en el desarrollo de software, ya que sienta las bases para la creación exitosa de un proyecto. Durante esta etapa, se busca definir y comprender de manera clara y precisa las necesidades y expectativas del cliente o usuario final.

Esto permite establecer una ruta de desarrollo adecuada y evitar costosos errores o trabajos innecesarios durante la implementación del sistema. (Calero Muñoz, C., Piattini Velthuis, M. G., 2010).

Además, el análisis de requerimientos fomenta la creación de una relación sólida entre el equipo de desarrollo y los usuarios finales, lo que facilita una comprensión mutua y una comunicación efectiva. Esta interacción cercana entre todas las partes interesadas es clave para asegurar que el producto final cumpla con las expectativas y satisfaga las necesidades reales del cliente. (Calero Muñoz, C., Piattini Velthuis, M. G., 2010).

1.6.1.3. Pruebas de software y control de calidad

Las pruebas de software son un proceso fundamental que permite identificar y corregir potenciales dificultades o errores antes de que el sistema sea implementado en el entorno de producción. Su objetivo principal es garantizar que el software se ejecute de manera correcta y estable, previniendo inconvenientes que puedan afectar su funcionamiento. (Julián Mera Paz, 2016)

Existen diversos métodos y enfoques para llevar a cabo las pruebas de software. Entre ellos se encuentran la verificación, que consiste en determinar si cada fase del desarrollo cumple con las condiciones y requisitos establecidos inicialmente. Otro método es la validación, que se realiza al final del proceso de desarrollo para asegurar que el producto final satisface los requerimientos especificados por el cliente o usuario final. (Julián Mera Paz, 2016)

Es fundamental incorporar las pruebas de software a lo largo de todo el ciclo de desarrollo del sistema. Algunas de las pruebas comúnmente realizadas incluyen la identificación de defectos, las pruebas funcionales y no funcionales, y la generación de informes que reflejen el nivel de calidad y confianza del software. (Julián Mera Paz, 2016)

Las pruebas de software son fundamentales para conocer el estado y la calidad del sistema web desarrollado. Mediante la aplicación de diversas pruebas, se puede verificar la funcionalidad y seguridad del sistema, identificando posibles vulnerabilidades. Es crucial abordar estas vulnerabilidades y aplicar las mejores soluciones antes de que el proyecto esté en un entorno de producción y pueda ser manipulado por los usuarios finales.

1.2. TRABAJOS PREVIOS

En el proyecto "Sistema de identificación orientado a la web para el control y seguimiento pecuario de una finca con el uso de tecnología RFID" desarrollado por Remberto Ballesta Ávila, se aborda la optimización de los procesos pecuarios mediante la implementación de tecnología. El objetivo principal fue facilitar las tareas administrativas y operativas, permitiendo que los involucrados puedan realizar transacciones de manera eficiente y mejorar los tiempos de respuesta. Un aspecto clave del proyecto es el mejoramiento del entorno de identificación animal, lo que implica una organización más efectiva de los ejemplares. Además, se busca incorporar recursos tecnológicos que contribuyan a optimizar la economía y reducir los costos asociados a la actividad pecuaria. (Ballestas Ávila, Remberto, 2020)

Tal como lo señala Karen Natalia en su tesis "Sistema web para el control sanitario y seguimiento médico en la producción tecnificada de cuyes", garantizar la salud y prevención de enfermedades en los animales es fundamental para asegurar una producción óptima. En este sentido, la implementación de un sistema web tuvo como objetivo reducir los gastos en medicamentos y, por consiguiente, disminuir la mortalidad de las cobayas (Galarza Pincay, 2022). El sistema pretende brindar un seguimiento exhaustivo de los procesos de crianza, abarcando el tratamiento de las instalaciones, el monitoreo de enfermedades, la vacunación y la desparasitación en todas las etapas, desde la cría hasta la recría y reproducción. Uno de sus beneficios clave es la optimización del tiempo en la toma de decisiones relacionadas con los registros de sanidad, los cuales recopilan información esencial como fechas de nacimiento, períodos de vacunación y desparasitación, con el objetivo de garantizar un crecimiento saludable de las cobayas en el menor tiempo posible, maximizando así las ganancias y la rentabilidad. (Karen Natalia, 2022)

En el proyecto "Sistema web para el control de costos de producción y comercialización de cuyes" se destaca la importancia de la tecnología en la producción de animales y la reducción de tiempo al momento de tomar registros sobre su crecimiento, criaderos y tareas. Se pretendía mejorar los procesos de costos y ventas de cobayas, y de esta manera, tener una mejor visión de los ingresos y egresos que se generan. Mediante la implementación de un sistema web, se obtuvieron buenos resultados al realizar transacciones de venta (Choez Choez, 2022). En el desarrollo de este proyecto, se tomaron en consideración varios factores clave, como el registro de la compra de cuyes, el registro de proveedores de alimentos, la adquisición de alimento y el registro de clientes que compran con frecuencia este producto. Además, se creó una lista donde se puede visualizar el stock de cuyes disponibles. (Choez Choez, 2022)

En el proyecto de Orlando Bautista" las tecnologías en la crianza de cuyes y su incidencia en la generación de ingresos económicos para los productores del distrito Antonio Raymondi, Áncash 2018", se hace énfasis en la falta de tecnología en los distintos procesos que implica la producción de cuyes y las dificultades que han enfrentado los pequeños productores por no optimizar sus procesos y tareas al realizar esta actividad. Se enfatiza que la tecnología puede ayudar a mejorar los procesos de alimentación, producción, sanidad e instalaciones. Este proyecto busca identificar las tecnologías que han implementado los productores para mejorar y optimizar su producción de cobayas, logrando así obtener una mayor cantidad de crías y reproductores, y percibir mejores ingresos al momento de la venta de los mismos. (Orlando, 2018)

De manera adicional, cabe destacar el proyecto "Sistema de gestión de información para el manejo de inventario en el proceso de crianza de cuyes de la granja de la federación colombiana de productores de papa "fedepapa" DE OBONUCO—NARIÑO" optimizo el proceso de crianza de cobayas mediante la implementación de un sistema computacional. Se han automatizado tareas como la recolección de datos, identificación de cuyes y el registro del peso de cada animal. Este sistema contribuye al manejo eficiente de la información durante el proceso de crianza de cobayas. El presente proyecto está dirigido a ayudar a pequeños y medianos productores que poseen granjas tecnificadas y necesitan un control óptimo del inventario. (Patichoy Calpa & Armando Paredes, 2022)

En la investigación" Sistema web para el control administrativo de la granja de cuy coordinadora PROCUY Mantaro 2021", Jean Carlos Vásquez Soto analiza la implementación de tecnología en los procesos de crianza de cobayas. Se enfoca en la toma de registros que pueden ayudar a tomar mejores decisiones, el aumento de la producción de cobayas en los últimos tiempos y el consumo de las mismas. Existen varios problemas en las áreas administrativa y operativa debido a la falta de un sistema web que permita mejorar los procesos de recolección de datos de los galpones con los que cuenta la granja. Los administradores enfrentan dificultades al no tener acceso a la información de manera instantánea para la toma de decisiones. (Carlos, 2021)

Después de revisar las investigaciones mencionadas, se concluye que la implementación de tecnología contribuye a optimizar los procesos de crianza de cobayas. En estos estudios, se resalta la importancia de la recolección de información para mejorar los procesos y aumentar la producción. La información proporcionada por los sistemas es vital para que los productores puedan tomar decisiones oportunas y prevenir gastos innecesarios. Cabe recalcar que se han identificado deficiencias en la gestión y recolección de datos relacionados con la crianza de cobayas por falta de tecnología.

CAPÍTULO II
MATERIALES Y MÉTODOS

En este apartado se detallan los aspectos metodológicos que orientaron el proceso de investigación, junto con los procedimientos y herramientas utilizados para alcanzar los objetivos propuestos.

2.1. Tipo de investigación

El presente proyecto se enmarca una investigación de tipo aplicada, ya que busca desarrollar una solución tecnológica para abordar una problemática específica en el ámbito de la crianza de cobayas.

Se empleó un enfoque cualitativo, dado que fue necesario identificar los procesos de recolección de datos y las respuestas que el usuario requiere del sistema web. Para la recopilación de información, se utilizó la técnica de la entrevista al dueño de la granja Quishpe, lo que permitió conocer los aspectos que debe abordar el sistema y los resultados esperados con su funcionamiento. Además, se revisaron los registros en los que el dueño realiza el seguimiento a la producción y crecimiento de las cobayas.

2.2. Técnicas para la recolección de datos.

Las técnicas empleadas para la recopilación de datos y la identificación de las necesidades de la granja familiar Quishpe fueron la observación directa y entrevista. Luego de aplicar las técnicas de recopilación de datos y entrevista, así como la observación directa, y analizar la información obtenida, se concluyó que era fundamental conocer los requisitos para desarrollar un sistema web para la granja familiar Quishpe.

2.2.1. Observación directa

Mediante la observación directa, se lograron identificar los procesos que el productor realizaba al momento de recolectar información del criadero. Asimismo, se pudo identificar la información primordial y las necesidades clave para optimizar los procesos de selección y mejorar el tiempo de respuesta ante la problemática.

Uno de los casos observados, fue sobre la forma en la que se almacenaban los datos (formato en Word o manuscrito) en estos documentos se llenaba con información de las recrías, los registros

debían llenarse mensualmente con el fin de determinar el crecimiento y la cantidad de animales que existían por jaula. (Ver Tabla 1).

Tabla 1

Se muestra los registros de Cría o Recrías

FICHA DE OBSERVACIÓN		
Fecha:	***10/04/2024***	*N°1*
Lugar:	*Criadero Quishpe*	
Descripción	*Registro de cría y recría*	

Fecha de destete_____ fecha final de recría_____
Número de animales_____ sexo_____

Fecha	N° Jaula	Cantidad	Sexo	Total	Observación (venta, muerte)

***Descripción**: El productor llenaba estos formatos con información como número de jaula, la cantidad de crías que existían en esa jaula, el sexo de las crías y una observación sobre todas las crías vendidas o muertas en el transcurso de su crecimiento*.

Otro caso observado fue los registros de las cobayas reproductoras, de igual manera estos registros se almacenaban de las siguientes maneras (formato en Word o manuscrito) en donde el productor debía ingresar todos los datos sobre las Hembras y Macho para comenzar con su reproducción. (Ver Tabla 2)

Tabla 2

Se muestra el registro para las reproductoras de la granja.

FICHA DE OBSERVACIÓN		
Fecha:	***10/04/2024***	*N°2*
Lugar:	*Criadero Quishpe*	
Descripción	*Registro de reproductores*	

N° de jaula_____ N° de macho_____

N° Hembra	Fecha de parto	N° Parto	N° Crías	Observación

Descripción: *El productor debía ingresar el número de la Jaula y el código único de las cobayas reproductoras, debe existir un macho por jaula y hasta 8 hembras máximo, una vez que las hembras parían se registraba el número del parto, numero de crías nacidas y una observación sobre el parto.*

Otro caso observado fue el registro de las crías de las cobayas en donde la información se almacenaba de las siguientes maneras (formato en Word o manuscrito), en estos registros se realizaba un seguimiento de las crías individualmente, se tomaban datos sobre sus pesos en un determinado tiempo. (Ver Tabla 3).

Tabla 3

Se muestra el registro de engorde

FICHA DE OBSERVACIÓN		
Fecha:	***10/04/2024***	*N°2*
Lugar:	*Criadero Quishpe*	
Descripción	*Registro de engorde*	

N° de jaula_____ Total de macho_____

N° animal	Peso nacimiento	Peso destete	Peso de venta	Peso en canal	Observación

Descripción: *El productor debía ingresar información sobre el peso de nacimiento y destete para de esta manera poder determinar la ganancia de peso de la cría, de igual forma al ser vendida la cobaya se ingresaba el peso de venta y el precio.*

2.2.2. Análisis de la entrevista semiestructurada.

Esta entrevista es de tipo semiestructurada, con preguntas abiertas dirigidas al propietario de la granja. El objetivo es conocer las necesidades del productor en el proceso de crianza de cobayas y obtener su opinión sobre los posibles cambios que la implementación de tecnología podría traer a la granja.

ENTREVISTA SEMIESTRUCTURADA.

Fecha realización: 5 de abril del 2024

Lugar: Granja familiar Quishpe

Duración entrevista: 30 minutos

Entrevista dirigida a: Lic. Vinicio Quishpe, propietario de la granja familiar Quishpe.

Tipo de preguntas: Abiertas

1. ¿Qué tipo de registros y controles maneja actualmente la granja para el seguimiento de la crianza de cobayas? ¿Considera que estos métodos son eficientes?

   Los registros que se manejan son de recría, reproductores, nacimientos y ganancia de peso de las recrías, de igual manera estos registros se toman cada vez que una cobaya tiene un parto y cuando se desteta el animal y después de 2 meses para determinar el peso de los animales para la saca o venta.
2. ¿Cuáles son los principales desafíos que enfrenta el personal de la granja en la recopilación y procesamiento de la información relacionada con la producción de cobayas?

   Al momento de tomar decisiones es muy tedioso el realizar para bastantes registros, y estar calculando individualmente la ganancia de peso y el índice productivo de cada jaula, también al momento de buscar registros de un animal en específico se toma mucho tiempo o los datos no se encuentra en las hojas o en el digital donde se almacena la información.
3. ¿Considera que la implementación de un sistema web podría mejorar la toma de decisiones y optimizar los procesos de crianza en la granja familiar Quishpe? ¿Por qué?

   Todos los sistemas en donde mejoren la productividad y la toma de decisiones son indispensables para las empresas o micro empresas, en la granja este sistema web nos ayudaría a mejorar reduciendo el tiempo en la búsqueda de información y filtrado de animales con mejores características.
4. ¿Qué tipo de funcionalidades o reportes le gustaría que el sistema web proporcionara para facilitar el seguimiento y control de la producción de cobayas?

   En la funcionalidad queríamos que se puedan agregar, actualizar y eliminar registros sobre las jaulas, reproductores, partos y crías. Mientras que en los reportes nos gustaría conocer sobre las ventas y muertes de las cobayas en un tiempo determinado.
5. ¿De qué manera la integración de tecnología, como registros electrónicos y generación automática de informes, podría beneficiar la gestión y eficiencia de la granja?

   Se podría conocer sobre las ganancias y pérdidas en tiempo real, también se podría optimizar procesos de búsqueda de mejores especies, y tener una rápida respuesta mediante la tecnología aplicada.
6. ¿Qué tan importante sería para usted que el sistema web tenga diferentes perfiles de usuario con accesos y privilegios específicos para la administración y el personal de la granja?

Como administrador de la jaula quisiera que el productor pueda acceder a distintas funciones y que no pueda realizar ciertos cambios, mientras que el productor solo podría acceder a la información que es primordial para realizar sus funcionalidades.

Análisis. _ Luego de recopilar toda la información, se pudo concluir que la implementación de un sistema web en la granja familiar Quishpe era indispensable. Esto debido a que el sistema ayudaría a optimizar los procesos de recolección de datos requeridos para un adecuado manejo de la crianza de cobayas. Asimismo, la generación de reportes permitiría al productor y al dueño tener un conocimiento real de las ganancias y pérdidas de ingresos. El sistema web también respondería a las necesidades de conocer información crucial, como la ganancia de peso, la saca de animales que cumplían con su ciclo productivo y la identificación de las mejores especies para la continuidad del negocio.

METODOLOGÍA

2.3. Metodología para el desarrollo del sistema web

Las metodologías empleadas en el desarrollo de software permiten realizar cambios durante el proceso de creación, acogiendo las solicitudes de los usuarios. Esto posibilita contar con un software funcional, donde exista una interacción constante entre el desarrollador y el cliente, lo que garantiza que el producto final se ajuste a las necesidades reales. (Laínez Fuentes, 2015)

2.3.1. Metodología Ágil Extreme Programing (XP)

Se ha seleccionado la metodología ágil Extreme Programming (XP) para el desarrollo del sistema web de seguimiento y control de procesos de crianza de cobayas de la granja familiar Quishpe debido a las siguientes razones:

La metodología seleccionada para este proyecto está diseñada específicamente para pequeños y medianos proyectos, donde se mantiene un diálogo constante con el cliente. Esto permite realizar mejoras y cambios durante el transcurso del desarrollo y codificación del sistema de software. (Lainez, 2015)

La metodología brinda una estructura bien definida, que inicia con una fase de planificación. En esta etapa, se crean las historias de usuario, se establecen los roles y responsabilidades de cada miembro del equipo. Luego, se procede con el diseño, realizando reuniones para identificar problemas y crear diagramas, así como modelos que representen la estructura.

Posteriormente, se da paso a la codificación, donde se lleva a cabo todo el proceso de desarrollo del software, complementado con la ejecución de pruebas unitarias para garantizar la calidad. Más adelante, llega la fase de pruebas, en la que se identifican y resuelven posibles errores. Finalmente, se realiza el lanzamiento y entrega del producto al cliente.

2.4. Fases o etapas de Extreme Programming (XP) y su aplicación en el proyecto.

2.4.1. Fase 1: Planificación del Proyecto

En esta etapa, se lleva a cabo la recopilación exhaustiva de todas las necesidades y expectativas relacionadas con el proyecto. Se fomenta un diálogo activo con los usuarios finales para comprender a profundidad sus requerimientos. Durante este proceso, se definen claramente las metas y objetivos que guiarán el desarrollo exitoso del sistema.

a) Levantamiento de requerimientos.

En el proceso de recopilación de información, se emplean entrevistas como herramienta fundamental para obtener los detalles necesarios. Durante estas sesiones, los usuarios finales especifican las funcionalidades principales que debe cumplir la aplicación.

b) **Requerimientos no funcionales.**

a) Usabilidad.

- ✓ Los diseños de las diferentes interfaces deben mantener una consistencia en cuanto a sus botones, tablas y componentes, los cuales deben compartir estilos uniformes.
- ✓ En las diferentes interfaces, se deben mostrar mensajes de alerta cuando el usuario realice acciones de crear, editar y eliminar registros. Esto permitirá que el usuario tenga clara la interacción que está llevando a cabo.

b) Disponibilidad.

- ✓ El sistema web debe estar disponible al menos el 99% del tiempo, excluyendo los periodos de mantenimiento o cuando no exista acceso a la base de datos, asegurando la continuidad de las operaciones de crianza de cobayas.

c) **Rendimiento.**

- ✓ Las consultas a la base de datos y la visualización de registros deben estar optimizadas para minimizar el tiempo de respuesta del sistema web, especialmente cuando se manejan grandes volúmenes de información. Esto evita que el sistema se sobrecargue y colapse, garantizando un rendimiento eficiente y una experiencia fluida para los usuarios.

c) Requerimientos funcionales.

A continuación, se presenta la tabla con los requisitos funcionales que el sistema web debe cumplir. Estos requisitos fueron planteados por el cliente encargado de los procesos de crianza de cobayas de la granja Quishpe, los cuales se detallarán a continuación. (ver tabla 4)

Tabla 4
Requerimientos Funcionales

No	Requerimiento
1	El sistema web debe permitir el ingreso de usuarios mediante la autenticación.
2	El usuario administrador podrá asignar roles a los nuevos usuarios para la gestión de la sistema web.
3	El usuario administrador ejecutara funciones como crear, editar y eliminar los usuarios.
4	El sistema web permitirá al productor y administrador registrar datos sobre la producción de cobayas.
5	El sistema web permitirá modificar, eliminar y crear información sobre el proceso de producción de cobayas.
6	El productor y administrador podrán registrar datos sobre jaulas, productores, recrías, ventas y muertes de cobayas.
7	El sistema web permitirá la visualización de toda la información proporcionada.
8	El sistema web permitirá la exportación de reportes sobre ventas y muertes de las cobayas.
9	El sistema web permitirá la visualización de graficas sobre información de ventas, muertes y producción de crías.
10	El sistema web proporcionara información sobre el estado de las cobayas y su reproducción.

d) Usuarios Y roles que interactúan en el sistema web.

Hay dos roles fundamentales encargados de gestionar el sistema web. Estos roles tendrán la capacidad de ingresar información que será procesada, lo que permitirá la toma de decisiones. Los roles son los siguientes:

Administrador: El rol de administrador estará a cargo de la gestión administrativa del sistema. Este rol tendrá la capacidad de otorgar acceso al sistema a los productores, así como crear, editar y eliminar cualquier tipo de registro relacionado con la gestión del criadero.

Productor: El rol de productor tendrá acceso para crear información sobre jaulas, reproductores, partos, crías, muertes y ventas de crías y reproductores. Los productores podrán eliminar, crear y editar registros en estas vistas, los cuales serán procesados y mostrarán información relevante para la reproducción y crianza de cobayas. Además, tendrán la capacidad de crear y visualizar registros sobre las ventas y muertes de las cobayas.

e) Historias de Usuario

Historia de Usuario 1. Administración y asignación de roles

Tabla 5
Administración y asignación de roles

	Historia de Usuario
N°: 1	**Usuario:** Administrador
Nombre de la historia: Creación y asignación de roles y usuarios	
Interacción Asignada: 1	**Prioridad en el negocio:** Alta
Descripción: Como administrador del sistema, necesito la capacidad de crear, asignar y modificar los roles de los usuarios registrados. Además, los usuarios deben proporcionar información esencial al registrarse, incluyendo su nombre, apellido, dirección de correo electrónico (Gmail), fotografía y contraseña. Es esencial que la contraseña proporcionada sea verificada en el momento de la creación de la cuenta del usuario para garantizar la seguridad de acceso al sistema.	
Observaciones: Se deben establecer criterios claros para la verificación de contraseñas, garantizando su fortaleza y seguridad, almacenándolas en forma de hash. La funcionalidad debe incluir una interfaz intuitiva y segura para la gestión de roles de usuario.	

Historia de Usuario 2. Gestión de las jaulas de reproductores y recrías.

Tabla 6
Jaula de reproductores y recrías.

	Historia de Usuario
N°: 2	**Usuario:** Administrador, Productor
Nombre de la historia: Gestión de reproductores y recrías.	
Interacción Asignada: 2	**Prioridad en el negocio:** Media
Descripción: El administrador como el productor pueden agregar, editar y eliminar los registros de la interfaz de jaulas las cual se dividen entre reproductores y recría aquí se encuentran las crías destetadas de las cobayas reproductoras.	
Observaciones: Validar la información antes de ser subida a la base de datos. Proporcionar alertas antes de eliminar cualquier registro. El número de jaula debe ser único.	

Historia de Usuario 3. Gestión de reproductores y partos.

Tabla 7
Cobayas reproductores y partos

	Historia de Usuario
N°: 3	**Usuario:** Administrador, Productor
Nombre de la historia: Gestión de reproductores y partos	
Interacción Asignada: 2	**Prioridad en el negocio:** Media
Descripción: El administrador como el productor pueden agregar, editar y eliminar los registros de la interfaz de reproductores y partos, cada jaula debe consistir en 8 hembras máximo y un macho. Y cada hembra dispone de una interfaz con sus partos.	
Observaciones: Validar la información antes de ser subida a la base de datos. Proporcionar alertas antes de eliminar cualquier registro. Validar existencia de un solo macho y código único. Notificar cuando la cobaya cumpla 5 partos. El total de hijos vivos y muertos debe ser igual al total de hijos nacidos.	

Historia de Usuario 4. Gestión de crías.

Tabla 8
Jaula de reproductores y recrías.

	Historia de Usuario
N°: 4	**Usuario:** Administrador, Productor
Nombre de la historia: Gestión de crías.	

Interacción Asignada: 2	**Prioridad en el negocio:** Media
Descripción: El administrador como el productor pueden agregar, editar y eliminar los registros de la interfaz de crías, una vez que se produce un parto debe ingresar los registros de las crías vivas y muertas.	
Observaciones: Validar la información antes de ser subida a la base de datos. Proporcionar alertas antes de eliminar cualquier registro. El código de la cría debe ser único. Se debe ingresar todos los registros de vivos y muertos para proceder con el destete.	

Historia de Usuario 5. Gestión de muertes y ventas de cobayas.

Tabla 9
Muertes y ventas de reproductores y crías.

	Historia de Usuario
N°: 5	**Usuario:** Administrador, Productor
Nombre de la historia: Gestión de muertes y ventas de cobayas.	
Interacción Asignada: 2	**Prioridad en el negocio:** Media
Descripción: El administrador como el productor pueden agregar, editar y eliminar los registros de la interfaz de ventas y muertes de cobayas, al momento de ingresar registros deben seleccionar la jaula e ingresar datos en los formularios.	
Observaciones: Validar la información antes de ser subida a la base de datos. Proporcionar alertas antes de eliminar cualquier registro.	

Historia de Usuario 6. Gestión de estadísticas y Reportes.

Tabla 10
Estadísticas y reportes.

	Historia de Usuario
N°: 6	**Usuario:** Administrador, Productor
Nombre de la historia: gestión de visualización y exportación de reportes	
Interacción Asignada: 4	**Prioridad en el negocio:** Media
Descripción: El administrador como el productor deben poder exportar los registros de ventas, muertes y producción de cobayas. Asimismo, necesitan poder visualizar los datos de ventas, muertes y producción de cobayas durante un tiempo determinado.	

Observaciones: El filtrado de las estadísticas debe ser por total de cobayas, fechas y precios.

Los reportes deben ser en tiempo real.

Tabla 11
Planificación de prioridades

ID	Nombre	Estimación (horas)	Importancia	Tareas	Tiempo
1	**Autenticación y login**	**10**	**Alta**	**Modelar la interfaz.**	**3h**
				Validación de usuarios.	**3h**
				Testeo de los campos.	**4h**
2	**Usuarios y roles**	**20**	**Alta**	**Modelar la interfaz.**	**4h**
				Asignar Rol al usuario.	**4h**
				Construir el CRUD.	**5h**
				Comprobar el CRUD.	**5h**
				Testeo de los campos.	**2h**
3	**Registrar Jaula**	**10**	**Media**	**Modelar la interfaz.**	**3h**
				Construir el CRUD.	**3h**
				Comprobar el CRUD.	**2h**
				Testeo de los campos.	**2h**
4	**Registrar Reproductor**	**10**	**Media**	**Modelar la interfaz.**	**3h**
				Construir el CRUD.	**3h**
				Comprobar el CRUD.	**2h**
				Testeo de los campos.	**2h**
5	**Registrar Parto**	**10**	**Alta**	**Modelar la interfaz.**	**3h**
				Construir el CRUD.	**3h**
				Comprobar el CRUD.	**2h**
				Testeo de los campos.	**2h**
6	**Registrar Cría**	**10**	**Media**	**Modelar la interfaz.**	**3h**
				Construir el CRUD.	**3h**
				Comprobar el CRUD.	**2h**
				Testeo de los campos.	**2h**
7	**Registrar Ventas**	**10**	**Alta**	**Modelar la interfaz.**	**3h**
				Construir el CRUD.	**3h**
				Comprobar el CRUD.	**2h**

				Testeo de los campos.	**2h**
8	**Registrar Muertes**	**10**	**Alta**	**Modelar la interfaz.**	**3h**
				Construir el CRUD.	**3h**
				Comprobar el CRUD.	**2h**
				Testeo de los campos.	**2h**
9	**Registrar Recrías**	**10**	**Media**	**Modelar la interfaz.**	**3h**
				Construir el CRUD.	**3h**
				Comprobar el CRUD.	**2h**
				Testeo de los campos.	**2h**
10	**Estadísticas y Reportes**	**15**	**Alta**	**Modelar la interfaz.**	**3h**
				Validar gráficas.	**5h**
				Generar Reportes.	**5h**
				Testeo de los campos.	**2h**

Tabla 12
Distribución de sprints

Id	Fecha de inicio	Fecha de Finalización	Historias de usuario a desarrollar
1	01/04/2024	15/04/2024	Historia de Usuario 1
2	16/04/2024	29/04/2024	Historia de Usuario 2
3	30/04/2024	10/05/2024	Historia de Usuario 3
4	11/05/2024	17/05/2024	Historia de Usuario 4
5	18/05/2024	30/05/2024	Historia de Usuario 5
6	30/05/2024	31/05/2024	Historia de Usuario 6

2.4.2. Fase 2: Diseño

Durante esta fase, se prioriza la simplicidad en el diseño, tanto para los usuarios finales como para los desarrolladores. El objetivo es minimizar la complejidad, facilitar la comprensión y permitir una implementación eficiente del sistema. Se enfoca en obtener un Modelo Relacional claro y una arquitectura de aplicación simple y eficiente que promueva la eficacia en todas las etapas del desarrollo de software, desde la concepción hasta la implementación.

2.4.2.1. Implementación de la lógica del negocio.

A continuación, se presentan los módulos que conforman el sistema web, detallando la funcionalidad específica de cada uno:

- Módulo de Usuarios.

- Módulo del Criadero.
- Módulo de Producción
- Módulo de visualización de datos.

Módulo de Usuarios: El módulo de gestión de usuarios permite crear y asignar roles para acceder al sistema web. Existen dos tipos de roles principales con diferentes niveles de acceso:

- ✓ Administrador. – El rol de administrador faculta la creación, modificación y eliminación de usuarios, así como la asignación de roles para regular el acceso a la información del sistema. Además, el administrador tiene la capacidad de realizar cambios en las interfaces, obtener reportes y acceder a todas las vistas del sistema.
- ✓ Productor. – El rol de productor permite crear, actualizar y eliminar registros relacionados con la producción de cobayas. Además, los productores pueden visualizar la información y exportar reportes sobre el estado de la producción.

Modulo del Criadero: El módulo de gestión del criadero facilita la creación, modificación, eliminación y visualización de información relativa a jaulas, reproductores, crías de las cobayas. En este módulo, se integran interfaces que permiten a los productores interactuar de manera directa con el sistema web.

Módulo de Producción: El módulo de gestión de producción facilita la creación, modificación, eliminación y visualización de información relativa a número de hijos por parto, tiempo de gestación, capacidad reproductora, registros de ventas y muertes.

Módulo de visualización de datos: El módulo de visualización de datos brinda al productor información sobre nacimientos de crías, muertes y ventas en un período de tiempo específico. Además, permite exportar informes detallados sobre el estado de ventas, muertes de crías y reproductores.

2.4.2.2. Arquitectura física de la aplicación:

Para el desarrollo del sistema web se planteó una secuencia de capas, las cuales permiten manipular y almacenar los datos que serán proporcionados por el criadero de cobayas.

La primera capa consistió en el etiquetado y recolección de datos de las cobayas, ya fueran de los reproductores como de las crías. Se recopilaron datos sobre el sexo, código, peso inicial, número de partos, número de hijos por parto y el comienzo del empadre, la segunda capa implicó enviar los datos a la aplicación web a través de la vista de nuestro framework, La tercera capa consistió en realizar solicitudes de select, create, update y delete a la base de datos. Una vez procesada la

información, la base de datos envía los resultados a la aplicación web, para que el productor pueda visualizarlos. (Ver figura 2)

Figura 1

Arquitectura del Sistema.

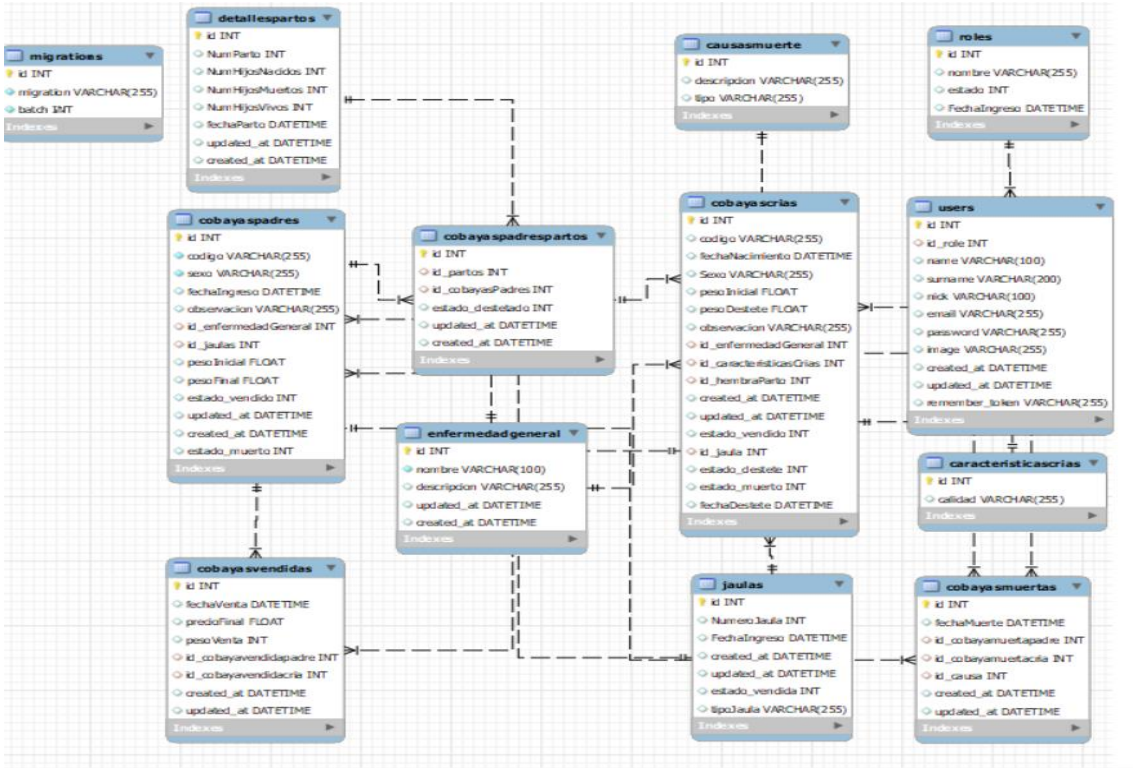


2.4.2.3. Estructura de la Base de datos:

Para el proyecto web se implementó una solución de base de datos que incluyó la creación de tablas relacionadas de uno a uno, uno a muchos, y para las relaciones de muchos a muchos, se establecieron tablas intermedias. Esto permitió unir los datos de manera eficiente y mantener relaciones sólidas entre las diferentes entidades, optimizando el almacenamiento y gestión de la información necesaria para el proyecto web. (Ver figura 3)

Figura 2

Modelado de la Base de Datos.



2.4.2.4.Casos de uso del sistema web:

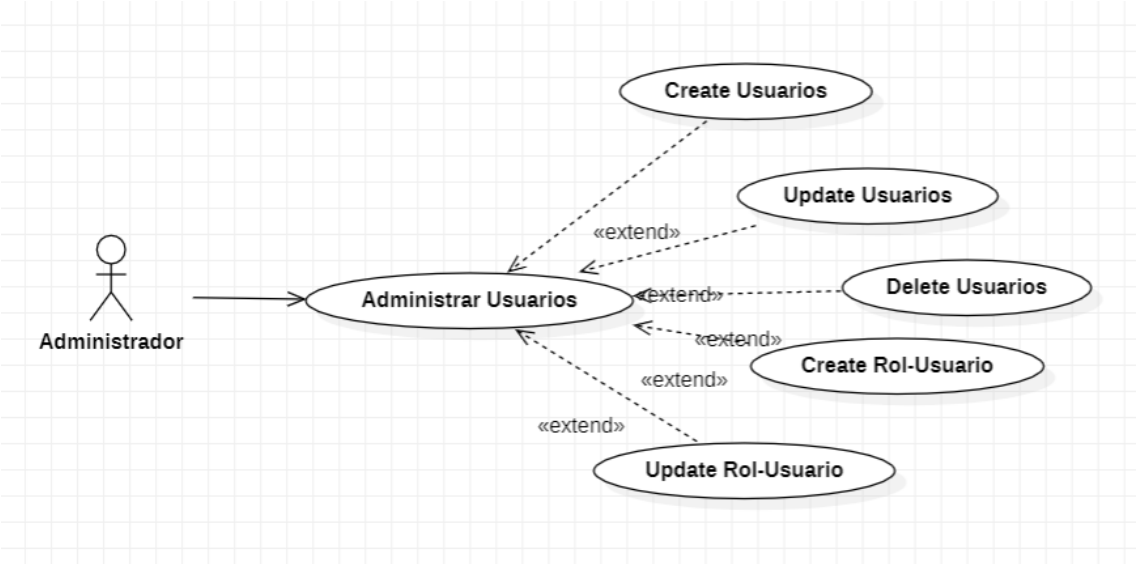
Los siguientes casos de uso ilustran la interacción de los usuarios con el sistema web. En ellos se detallan las funcionalidades a las que pueden acceder los diferentes tipos de usuarios, así como las acciones que pueden realizar dentro del sistema.

Administrador: Proceso de administración de usuarios.

El usuario administrador tiene la capacidad de crear, actualizar y eliminar cuentas de usuario en el sistema web. Además, puede asignar o remover roles a los diferentes usuarios, otorgándoles o quitándoles permisos dentro del sistema. (Ver figura 4).

Figura 3

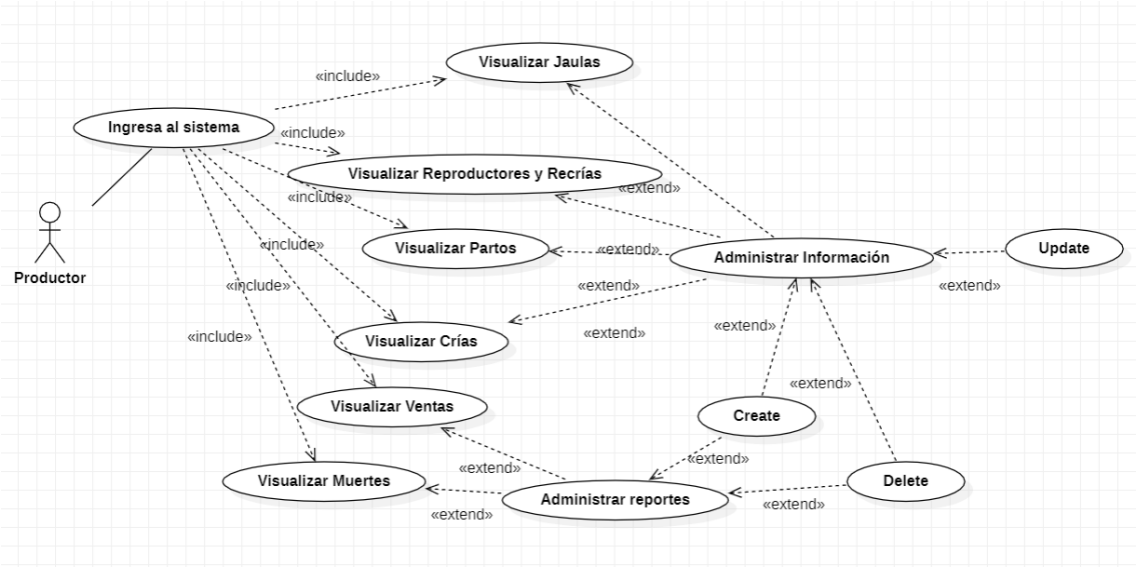
Administración de usuarios.



Productor: Proceso de administración del criadero.

Tanto el usuario productor como el administrador pueden participar en la administración del criadero. Ambos pueden acceder a las diferentes interfaces del sistema, donde tienen la capacidad de crear, actualizar y eliminar registros en las distintas secciones de la aplicación web. (Ver Figura 5)

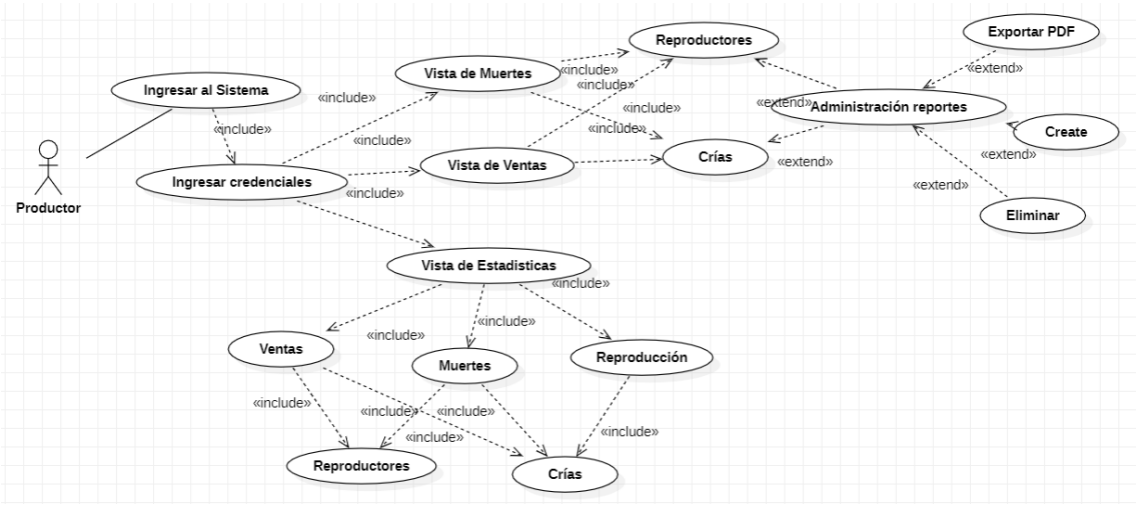
Figura 4
Administración del criadero.



Productor: Estadísticas y reportes de ventas, muertes y reproducción.

Después de iniciar sesión, el usuario productor puede acceder a interfaces que le permiten visualizar gráficas y exportar reportes sobre el estado de la producción de cobayas. Estas interfaces muestran información relevante, como las cobayas vendidas, las cobayas muertas y las crías nacidas, brindando al productor una visión general del desempeño de la granja. (Ver figura 6).

Figura 5
Estadísticas y reportes de ventas, muertes y reproducción.



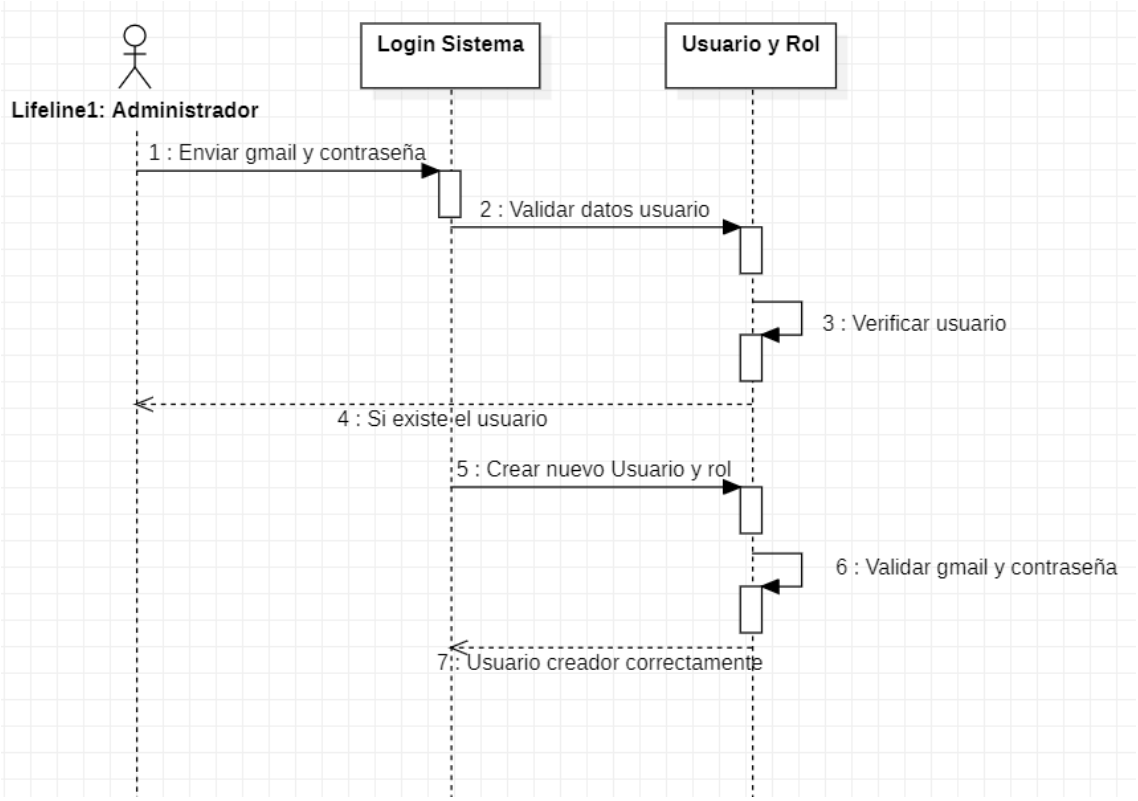
2.4.2.5. Diagrama de secuencia

Administrador

El usuario administrador puede ingresar al sistema una vez que sus credenciales han sido verificadas. Además, el administrador tiene la capacidad de crear nuevos usuarios y asignarles roles, ya sea como productores o como administradores adicionales. (Ver figura 7)

Figura 6

Diagrama de interacción del administrador

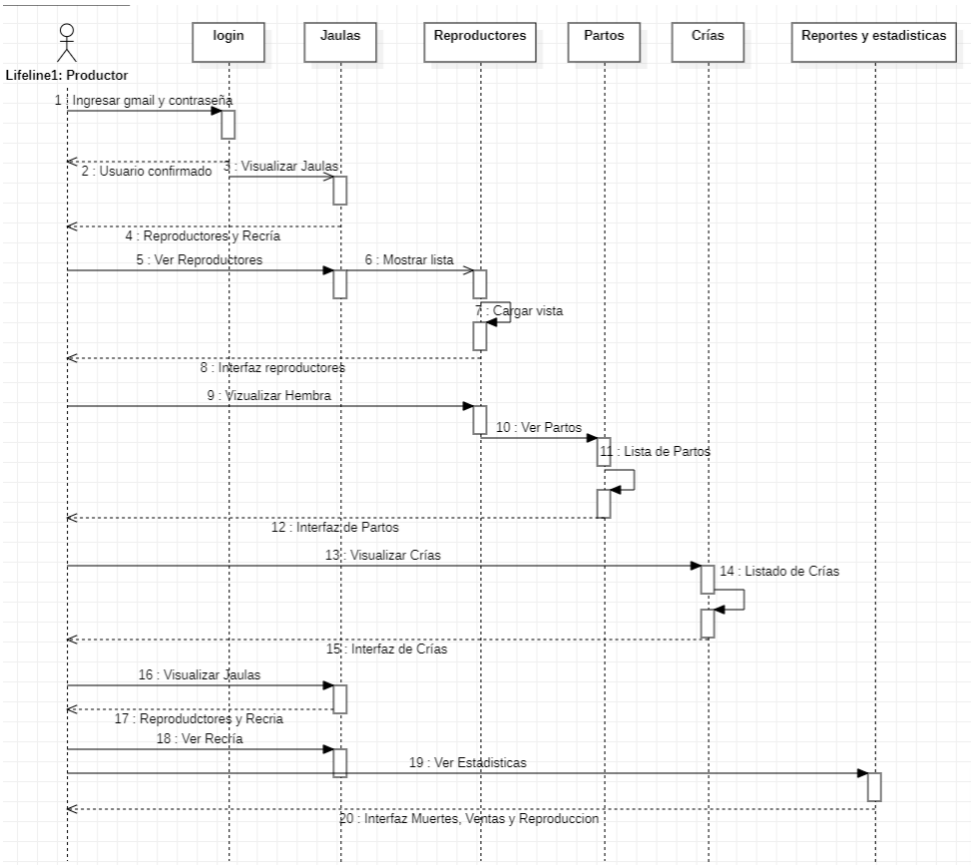


Productor

El usuario productor, tras autenticarse con su cuenta de Gmail y contraseña, puede acceder a varias interfaces del sistema. Estas le permiten crear, editar y eliminar registros de jaulas, reproductores, partos y crías. Además, puede visualizar las interfaces de muertes, ventas y reproducción de las crías. Asimismo, el productor puede generar reportes sobre las muertes y ventas de las cobayas, ya sean reproductoras o crías. (Ver figura 8)

Figura 7

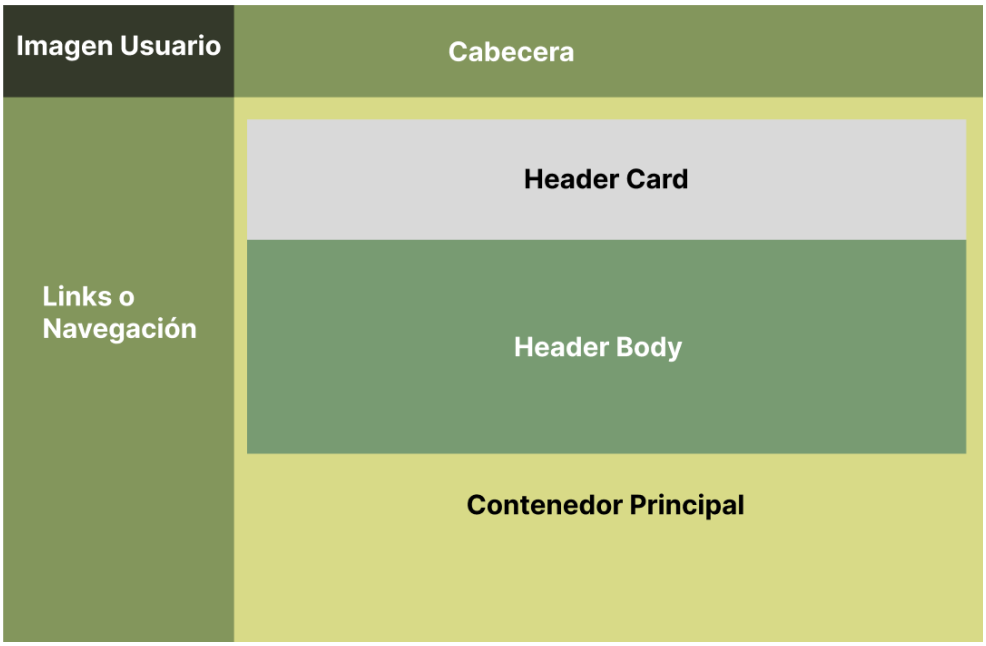
Diagrama de interacción del productor



2.4.2.6. Diseño de la interfaz

Para la construcción del sistema web, se realizó un prototipo de la interfaz principal, el cual sirvió de base para el desarrollo de las demás interfaces. En la siguiente imagen, se muestra un ejemplo de este diseño inicial, el cual fue creado utilizando la herramienta Figma. (Ver figura 8)

Figura 8
Diseño de la interfaz



2.4.3. Fase 3: Codificación

En la etapa de implementación del software, se llevó a cabo un diálogo constante con el cliente para identificar mejoras y corregir fallos del sistema. Adicionalmente, a medida que se desarrollaban las diferentes vistas, el programador revisaba el funcionamiento del sistema, asegurándose de que se acoplaran adecuadamente a los casos de uso previamente establecidos. Este enfoque colaborativo y de revisión continua permitió refinar y mejorar la solución implementada, en línea con los requerimientos del cliente.

2.4.3.1. Herramientas de desarrollo

Para la creación del sistema web, se utilizaron diversas herramientas y aplicaciones que permitieron codificar, implementar funcionalidades y agregar librerías que mejoraron la interacción del sistema. A continuación, se detallarán las herramientas y aplicaciones empleadas.

Lenguajes de programación.

Para el desarrollo de la aplicación, se utilizaron los siguientes lenguajes de programación:

- PHP
- JavaScript
- CSS
- HTML

La implementación de estas herramientas permitió crear las interfaces, facilitar la interacción de los usuarios con el sistema, así como establecer la conexión con la base de datos y manipular los datos.

Editor de texto

Para codificar todo el sistema, se utilizó la herramienta de desarrollo Apache NetBeans. Esta herramienta proporcionó un entorno integrado que permitió crear, editar y eliminar líneas de código. Además, ayudó a depurar el código y monitorizar la interacción de los datos, lo que facilitó la optimización y mejora del código desarrollado. NetBeans también cuenta con extensiones que ayudan a formatear y visualizar mejor cualquier tipo de lenguaje de programación utilizado en el proyecto.

Administrador de dependencias.

Para la creación de este proyecto, se utilizó la herramienta Composer. Esta herramienta permitió instalar dependencias, ejecutar el proyecto y agilizar la generación de modelos, controladores y vistas del sistema web. Además, Composer proporciona otras funcionalidades útiles, como visualizar las rutas creadas, realizar migraciones a la base de datos, crear nuevos proyectos, desarrollar middlewares para validar variables y generar helpers que ayuden a mejorar y formatear el código de la aplicación. En resumen, Composer se convirtió en una herramienta fundamental para optimizar el desarrollo del sistema web.

Framework

El sistema web fue desarrollado utilizando el framework Laravel, el cual proporciona una estructura y herramientas que permiten tener un proyecto organizado y seguro. Laravel funciona bajo el patrón Modelo-Vista-Controlador, lo que permite crear un sistema ordenado. En el modelo se maneja la conexión a la base de datos, el controlador consume los datos, mientras que la vista se encarga de visualizar la información. Además, Laravel brinda seguridad en las rutas y controladores, lo cual es fundamental al desarrollar el sistema web. Esto permite otorgar a los usuarios accesos diferenciados a las distintas áreas y la capacidad de modificar únicamente la información que les corresponde.

Administrador de Base de datos.

Para administrar la información y visualizar las relaciones entre las distintas tablas de la base de datos del sistema web, se utilizaron herramientas como Workbench y TablePlus. Estas herramientas permitieron navegar, consultar y administrar la base de datos de manera eficiente. Además, facilitaron la edición de datos, la búsqueda y el filtrado de información mediante

sentencias SQL. Otra funcionalidad clave es la posibilidad de exportar datos sobre las tablas y el modelo de la base de datos.

2.4.4. Fase 4: Pruebas

Esta es una etapa crucial, ya que asegura la calidad y confiabilidad del sistema de software. Durante esta fase, se llevan a cabo pruebas exhaustivas para verificar que todas las funcionalidades se ejecuten correctamente. Si se identifican errores, estos se corrigen antes de que el producto se entregue al cliente y se implemente en un entorno real. De esta manera, se garantiza que el software funcione de manera óptima cuando llegue a manos del usuario final.

En la Tabla 13 que se muestra a continuación, se presenta la plantilla que se empleó para documentar y registrar de manera organizada los diversos casos de prueba implementados durante el desarrollo del sistema.

Tabla 13
Plantilla para pruebas de unitarias

Prueba unitaria N°		
Nombre:		
Descripción:		
Escenarios		
Campo de entrada	**Campo de entrada**	**Resultado esperado**
Observaciones:		

En las tablas que se presentan a continuación, se muestra un conjunto de casos de prueba utilizados para evaluar el correcto funcionamiento de cada una de las interfaces del sistema. Dicha evaluación permitió asegurar el adecuado desempeño del sistema en su conjunto.

2.4.4.1.Casos de pruebas unitarias de las interfaces

Caso de prueba unitarias de la interfaz de inicio de sesión.

El proceso de inicio de sesión (login) es un componente fundamental del sistema. Debe permitir a los usuarios acceder a la aplicación web de manera segura y confiable. Asimismo, es importante que el login proporcione a los usuarios retroalimentación clara sobre las acciones que están realizando, como una confirmación de acceso exitoso o notificaciones sobre posibles errores. (Ver tabla 14)

Tabla 14
Caso de prueba unitaria de la interfaz de inicio de sesión.

Prueba de unitaria Numero 1		
Nombre: Inicio de sesión.		
Descripción: El sistema web debe contar con las validaciones necesarias para permitir que los usuarios accedan al sistema de manera segura. Para ello, se requiere que el usuario ingrese su correo electrónico y contraseña válidos a fin de poder iniciar sesión en el sistema.		
Escenarios		
Correo electrónico	**Contraseña**	**Resultado esperado**
correo@correo12.com (Gmail invalida)	Correo12 (contraseña valida)	El Gmail fue ingresado incorrectamente por el usuario no tiene la estructura adecuada.
correo@correo.com (Gmail valido)	Juan123 (contraseña invalida)	La contraseña proporcionada por el usuario es incorrecta.
Campo vacío	Correo12 (contraseña valida)	El sistema muestra al usuario que el campo de Gmail es obligatorio.
correo@correo.com (Gmail valido)	Campo vacío	El sistema muestra al usuario que el campo de contraseña es obligatorio.
correo@correo.com (Gmail valido)	Correo12 (contraseña valida)	Los campos e información es correcta, el usuario es redirigido a la página principal del sistema.
Observaciones:		

Caso de prueba unitaria de la interfaz jaulas.

La interfaz de creación de jaulas debe verificar la información de las jaulas existentes, a fin de asegurar que el número de jaula a registrar no esté ocupado y se encuentre disponible. Asimismo, es importante que todos los campos requeridos para la creación de una nueva jaula sean debidamente completados. (Ver tabla 15)

Tabla 15
Caso de prueba unitaria de la interfaz de jaulas.

Prueba unitaria N° 2

Nombre: Jaulas			
Descripción: El módulo de jaulas tiene que tener validación al momento de ingresar la información, todos los campos debe estar llenos y algunos registros deben ser de valor único.			
Escenarios			
Número de Jaula	**Tipo de Jaula**	**Fecha de Ingreso**	**Resultado esperado**
Campo vacío	Campo vacío	Campo vacío	Todos los campos son obligatorios.
Uno (invalido)	Reproductores12 (invalido)	18/4/2030 (invalido)	Los datos ingresados no son válidos.
1 (invalido)	Reproductores (valido)	18/4/2024 (valido)	El número de jaula debe ser único para los reproductores y recría.
1 (valido)	Reproductores (valido)	18/4/2024 (valido)	Todos los campos son correctos, se puede crear o actualizar el registro
Observaciones:			

Caso de prueba unitaria de la interfaz de cobaya reproductora.

La interfaz de cobayas reproductoras permite al usuario realizar las siguientes acciones: crear, actualizar y eliminar registros. Es fundamental que la información proporcionada en esta interfaz sea debidamente validada para garantizar su integridad y coherencia. (Ver Tabla 16)

Tabla 16
Caso de prueba unitaria de la interfaz de cobayas reproductoras.

Prueba unitaria N 3						
Nombre: Cobayas Reproductoras.						
Descripción: La interfaz de cobayas reproductoras debe validar la información proporcionada por el usuario.						
Escenarios						
Código	**Sexo**	**Fecha de Ingreso**	**Observación**	**Enfermedad General**	**Peso Inicial**	**Resultado Esperado**
Campo Vacío	Campo Vacío	Campo Vacío	Campo Vacío	Campo Vacío	Campo Vacío	Todos los campos son obligatorios

12@ (invalido)	Machos (invalido)	18/04/39 (invalido)	12131311 (invalido)	Parasitos12 (invalido)	1kg (invalido)	Los datos ingresados no son válidos.
12 (invalido)	Macho (invalido)	18/04/24 (valido)	Ninguna (valido)	Parásitos (valido)	900g (valido)	Solo debe existir un macho por jaula y el código.
12 (valido)	Macho (valido)	18/04/24 (valido)	Ninguna (valido)	Parásitos (valido)	900g (valido)	Todos los campos son correctos, se puede crear o actualizar el registro
Observaciones:						

Caso de prueba unitaria de la interfaz de partos.

En la interfaz de partos, se debe asegurar que el usuario ingrese información válida y que todos los campos requeridos se encuentren debidamente completados. Esto es crucial para mantener la integridad y precisión de los datos relacionados a los partos de las cobayas. (Ver tabla 17)

Tabla 17

Caso de prueba unitaria de la interfaz de partos.

Prueba unitaria N°4					
Nombre: Partos					
Descripción: La siguiente interfaz debe validar todos los campos, también que la información sea la adecuada y solicitada por el sistema.					
Escenarios					
Número Parto	**Número de hijos**	**Número de hijos vivos**	**Número de hijos muertos**	**Fecha de Parto**	**Resultado esperado**

Campo Vacío	Campo Vacío	Campo Vacío	Campo Vacío	Campo Vacío	Todos los campos son obligatorios
12a (invalido)	13a (invalido)	8a (invalido)	8a (invalido)	18/04/2030 (invalido)	Los datos ingresados no son válidos.
1 (valido)	5 (valido)	3 (invalido)	3 (invalido)	18/04/24 (valido)	El número de hijos vivos y muertos deben sumar y ser igual al total.
1 (valido)	5 (valido)	2 (valido)	3 (valido)	18/04/24 (valido)	Todos los campos son correctos se puede crear o actualizar el registro.
Observaciones:					

Caso de prueba unitaria de cobayas crías.

La interfaz de crías permite al usuario interactuar con los registros de las mismas. En esta, el usuario puede crear nuevos registros, los cuales deben cumplir con ciertas validaciones para asegurar que la información proporcionada sea la adecuada y esté completa. (Ver Tabla 18)

Tabla 18

Caso de prueba unitaria de la interfaz de cobayas crías.

Prueba unitaria N° 5							
Nombre: Cobayas crías.							
Descripción: El sistema web debe verificar que la información de las cobayas crías debe ser la correcta y que todos los campos estén llenos.							
Escenarios							
Código	**Sexo**	**Estado**	**Peso Inicial**	**Observación**	**Enfermedad General**	**Aspecto**	**Resultado esperado**

Campo Vacío	Campo Vacío	Campo Vacío	Campo Vacío	Campo Vacío	Campo Vacío	Campo Vacío	Todos los campos son obligatorios
12b1 (invalido)	Hembras (invalido)	Vivos (invalido)	1kg (invalido)	1012121 (invalido)	Parasitos12 (invalido)	Normal 12 (invalido)	Los datos no son validos
12 (valido)	Hembra (valido)	Vivo (valido)	900g (valido)	Ninguna (valido)	Parásitos (valido)	Normal (valido)	Todos los datos son correcto se puede crear o actualizar.
Observaciones:							

Caso de prueba unitaria de la interfaz de ventas de cobayas

En la interfaz de ventas, se registran todas las transacciones relacionadas a las cobayas reproductoras y a las crías. Las crías se venden una vez que han pasado por el proceso de destete, mientras que las cobayas reproductoras se venden una vez que han cumplido su ciclo reproductivo. Esta interfaz permite llevar un control detallado de todas las ventas realizadas. (Ver tabla 19)

Tabla 19

Caso de prueba unitaria de la interfaz de ventas de cobayas

Prueba unitaria N° 6				
Nombre: Ventas de cobayas reproductoras y recrías.				
Descripción: El sistema web permite al usuario registrar datos sobre las ventas de las cobayas en esta interfaz debe ingresar datos sobre el precio y peso final de la cobaya vendida.				
Escenarios				
Jaula	**Código**	**Precio**	**Peso**	**Resultado esperado**
Campo vacío	Campo vacío	Campo vacío	Campo vacío	Todos los campos son obligatorios

1as – reproductores (invalido)	120a1 (invalido)	90 (invalido)	1kg (invalido)	Los datos ingresados son inválidos
1 – reproductores (valido)	120a (valido)	9 (valido)	900g (valido)	Todos los campos son correctos crear cobaya vendida.
Observaciones:				

Caso de prueba de la interfaz de muertes de cobayas

La interfaz de muertes de cobayas permite al usuario llevar un registro detallado de todas las cobayas que han fallecido, ya sea que se trate de animales reproductores, crías destetadas o ejemplares en la etapa de recría. Este registro centralizado facilita el seguimiento y control de la mortalidad en la granja. (Ver tabla 20)

Tabla 20

Caso de prueba unitaria de la interfaz de muertes de cobayas

Prueba unitaria N° 7			
Nombre: Mueres de cobayas reproductoras y recrías.			
Descripción: La siguiente interfaz necesita información solida sobre la causa de muerte y la jaula de procedencia de la cobaya, para de esta forma llenar un conteo de perdidas tanto en reproductores y crías.			
Escenarios			
Jaula	**Código**	**Causa de Muerte**	**Resultado esperado**
Campo vacío	Campo vacío	Campo vacío	Todos los campos son obligatorios
12a - Cría (invalido)	120a1 (invalido)	Aborto12 (invalido)	Los datos ingresados son inválidos
12 – Cría (valido)	120a (valido)	Aborto (valido)	Todos los campos son correctos registrar datos.
Observaciones:			

Caso de prueba unitaria de la interfaz de recrías

La interfaz de crías destetadas muestra los registros de las crías que han pasado por el destete, clasificadas en jaulas de hembras y machos, listas para la venta. Es crucial validar la información de esta interfaz, ya que se utilizará para analizar la ganancia de peso de las crías. (Ver Tabla 21)

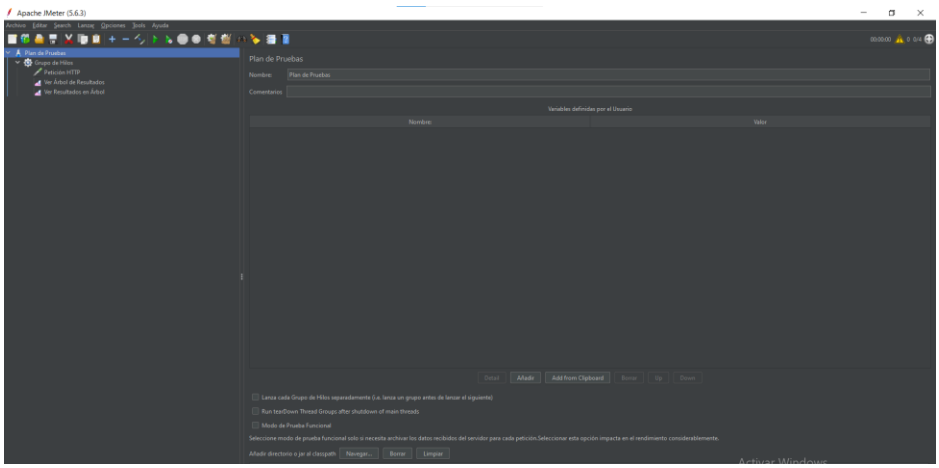
Tabla 21
Caso de prueba unitaria de la interfaz de recrías

Prueba unitaria N° 8		
Nombre: Recrías		
Descripción: La siguiente interfaz permite al usuario ingresar datos sobre el peso de destete de las crías después de un tiempo de 15 días para luego evaluar su crecimiento		
Escenarios		
Código	**Peso destete**	**Resultado esperado**
Campo vacío	Campo vacío	Todos los campos son obligatorios
120a1 (invalido)	2kg (invalido)	Los datos ingresados son inválidos
120 (valido)	900g (valido)	Todos los campos son correctos se agrega la cría a una jaula de recría.
Observaciones:		

2.4.4.2. Pruebas de carga con Apache Jmeter

JMeter es una herramienta que permite realizar diferentes tipos de pruebas a aplicaciones software. Algunas de sus capacidades principales incluyen simular cargas de múltiples usuarios concurrentes para medir el rendimiento bajo distintos niveles de tráfico. También puede someter al sistema a condiciones extremas para determinar sus límites. Admite diversos protocolos y genera informes detallados con métricas de rendimiento como tiempos de respuesta y tasas de transferencia. Estas características la convierten en una opción versátil para evaluar la capacidad y el desempeño de aplicaciones. (ver Figura 9)

Figura 9
Aplicación Apache JMeter



CAPÍTULO III
RESULTADOS Y DISCUSIÓN

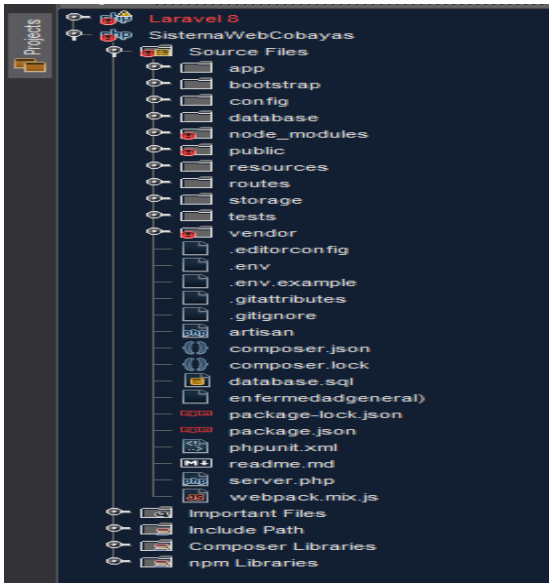
En este capítulo, se presentan los resultados obtenidos tras el desarrollo e implementación del sistema web para el criadero de cobayas. Se detallan las pruebas realizadas, los hallazgos clave y se discuten las implicaciones de estos resultados.

3.1. Directorio del código fuente

Para el desarrollo de este proyecto, se utilizó el framework de Laravel, permitiendo iniciar el proyecto desde cualquier parte del ordenador, y recurriendo a Composer para instalar otras dependencias y librerías. El proyecto se creó utilizando la versión 5.7 de Laravel, y para implementar todo el sistema de autenticación de usuarios, se instaló una extensión de Laravel mediante el comando php artisan make:auth, lo que facilitó la creación de la funcionalidad de inicio de sesión de los usuarios. (Ver figura 10)

La estructura de directorios de la aplicación Laravel se organiza de la siguiente manera:

Figura 10
Directorio del código fuente.



3.2.Interfaces de usuario

Para el desarrollo del sistema web, se implementaron vistas que cubrían funcionalidades clave, como inicio de sesión, administración de jaulas, manejo de reproductores, recría, partos, crías, ventas, muertes y estadísticas. Esto permitió crear un sistema web completo y funcional, capaz de satisfacer los diversos requerimientos del negocio y facilitar la interacción del usuario con la aplicación.

Interfaz de inicio de sesión.

Para acceder al sistema web de control de reproducción de cobayas, el usuario debe ingresar sus credenciales, es decir, su correo electrónico y contraseña, las cuales son creadas por el administrador del sistema. (Ver figura 11)

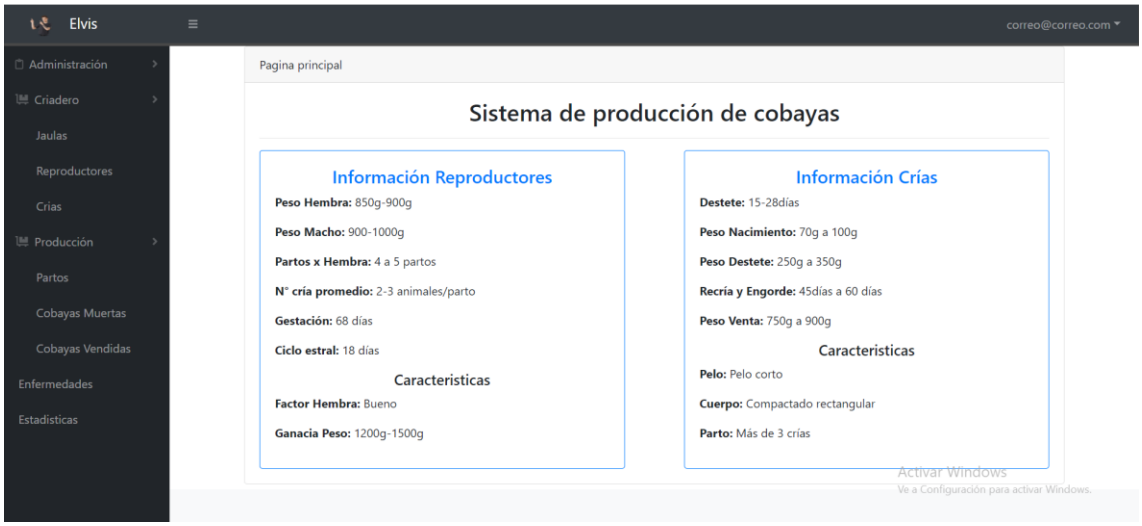
Figura 11
Interfaz de inicios de sesión.



Interfaz principal con el rol de productor

En esta interfaz se presenta el contenido general sobre la cría de cobayas, incluyendo información detallada sobre reproductores, crías y características clave a considerar por el productor. (Ver figura 12)

Figura 12
Interfaz principal.

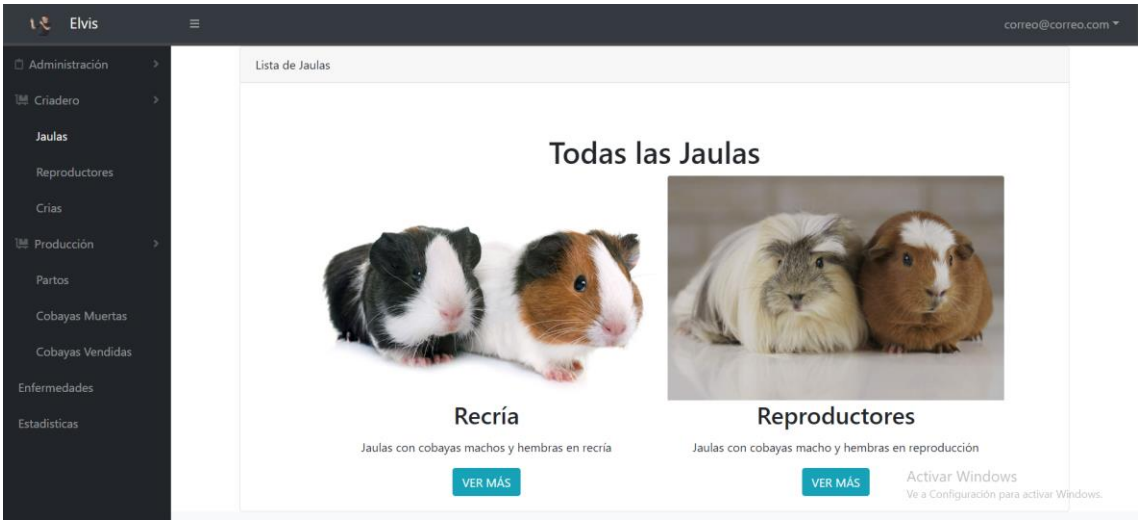


Interfaz de Jaulas

Existe una interfaz principal de jaulas que se divide en recría y reproductores. El productor, al acceder a esta interfaz, puede encontrar información sobre las jaulas existentes. Además, en esta misma interfaz, el productor puede crear nuevos registros, pero solo podrá eliminar y actualizar aquellos que no tengan datos asociados. (Ver figura 13)

Figura13

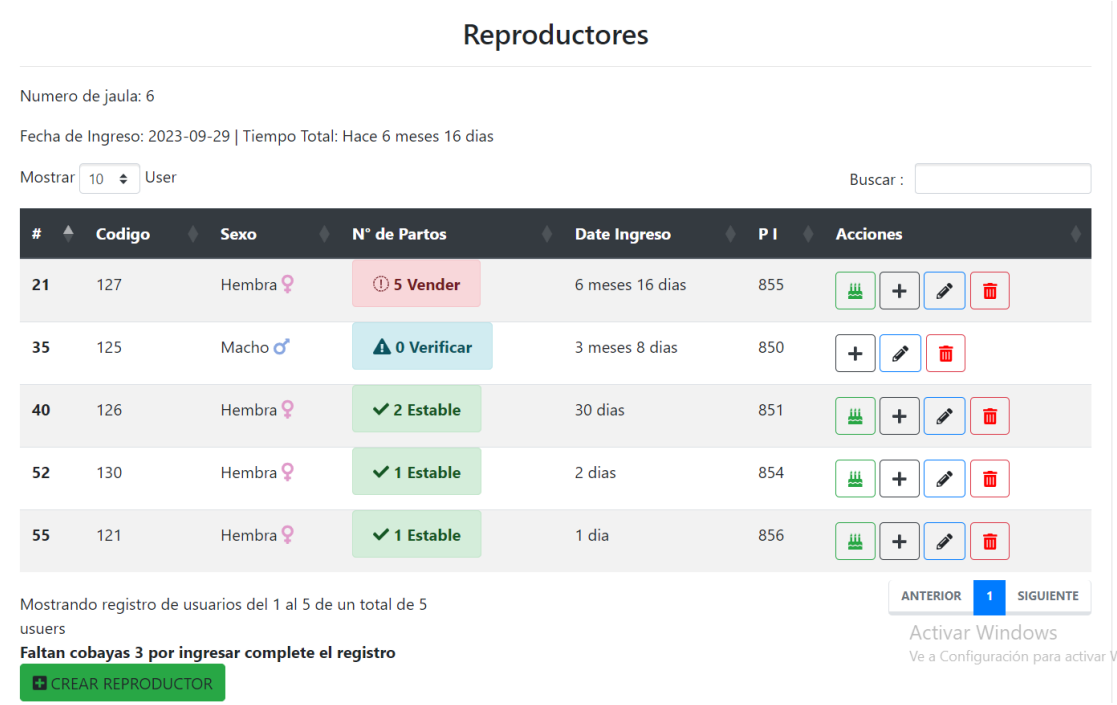
Interfaz principal de las jaulas.



Interfaz de Reproductores.

La interfaz muestra los datos de las cobayas reproductoras por jaula. Cada jaula de reproducción debe tener un registro que incluye un macho y entre 7 a 8 hembras para iniciar el ciclo reproductivo. Los datos registrados incluyen el comienzo del empadre, el total de partos por cobaya y el peso inicial de los animales. (Ver figura 14)

Figura 14
Interfaz de reproductores



Interfaz de Recrías

En esta interfaz se listan todas las crías que han sido destetadas y se encuentran en proceso de crecimiento, ya sea para su posterior venta o, en caso de fallecimiento, para el registro de dicha circunstancia. (Ver figura 15)

Figura 15
Interfaz de recrías.

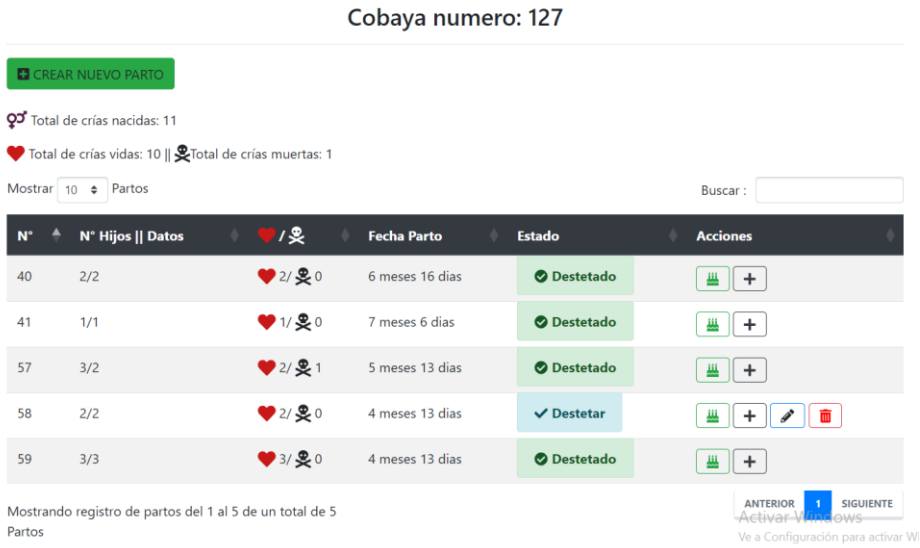


Interfaz de partos de hembras.

En esta interfaz, el productor puede visualizar información sobre los partos de cada cobaya. Por cada parto, se registra el total de crías nacidas, las que nacieron vivas, las que nacieron muertas,

la fecha del parto, así como el estado actual del parto, que puede ser: destetado, en proceso de destete, o aún en curso porque falta tiempo para el destete. (Ver figura 16)

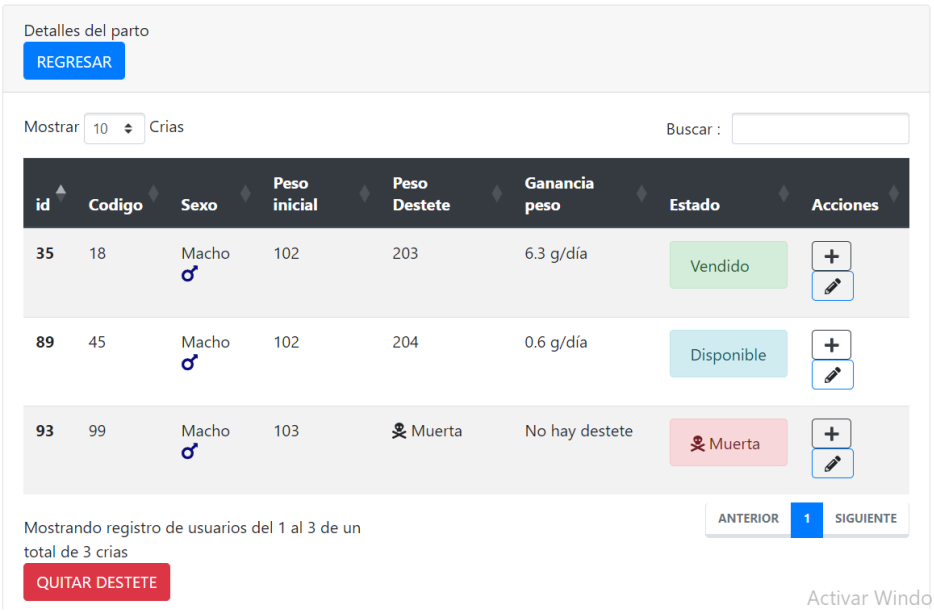
Figura 16
Interfaz de recrías.



Interfaz de crías del parto.

En esta interfaz, el productor puede acceder a los datos sobre las crías nacidas. Para cada cría, se muestra si nació viva o muerta, su peso de nacimiento, la ganancia de peso, el sexo y un código único de identificación. (Ver figura 17)

Figura 17
Interfaz de crías del parto.



Interfaz de cobayas muertas.

En esta interfaz, el productor puede registrar a las cobayas que han fallecido durante el proceso de reproducción o crecimiento, en el caso de las crías. Aquí, el productor puede visualizar los datos de las cobayas muertas, como el sexo, la causa de fallecimiento y la fecha. Además, puede generar un reporte consolidado de todas las muertes ocurridas. (Ver figura 18)

Figura 18
Interfaz de cobayas muertas.



Reporte de cobayas muertas.

Para obtener una visión general de las cobayas muertas, tanto reproductores como crías, se debe descargar un archivo en formato PDF. Este documento contiene información detallada sobre las muertes registradas en ambos grupos de cobayas, lo que permite al usuario tener un panorama completo de la mortalidad en la granja. (Ver figura 19)

Figura 19
Reporte PDF sobre muertes.



GRANJA FAMILIAR QUISHPE

Listado de registros de muertes de Recria del año 2024 del mes de Abril

Fecha de la creación de registro:
01 de Mayo de 2024

Productor: Elvis

Total de cobayas Muertas 19

Total de hembras: 10

Total de machos: 9

Codigo	Sexo	Descripción	Causa	Fecha Muerte
111	Macho	Ninguna Observacion	Nacimiento	01 de Abril de 2024
112	Hembra	Ninguna Observacion	Nacimiento	01 de Abril de 2024

Interfaz de cobayas vendidas.

En esta interfaz, el productor puede acceder a la información detallada de todas las cobayas que habían sido vendidas. Aquí se registran el peso y el precio al que fueron vendidas, la fecha de la venta y la jaula a la que pertenecen. Además, el productor puede generar un reporte consolidado de todas las ventas realizadas en un periodo determinado. (Ver figura 20)

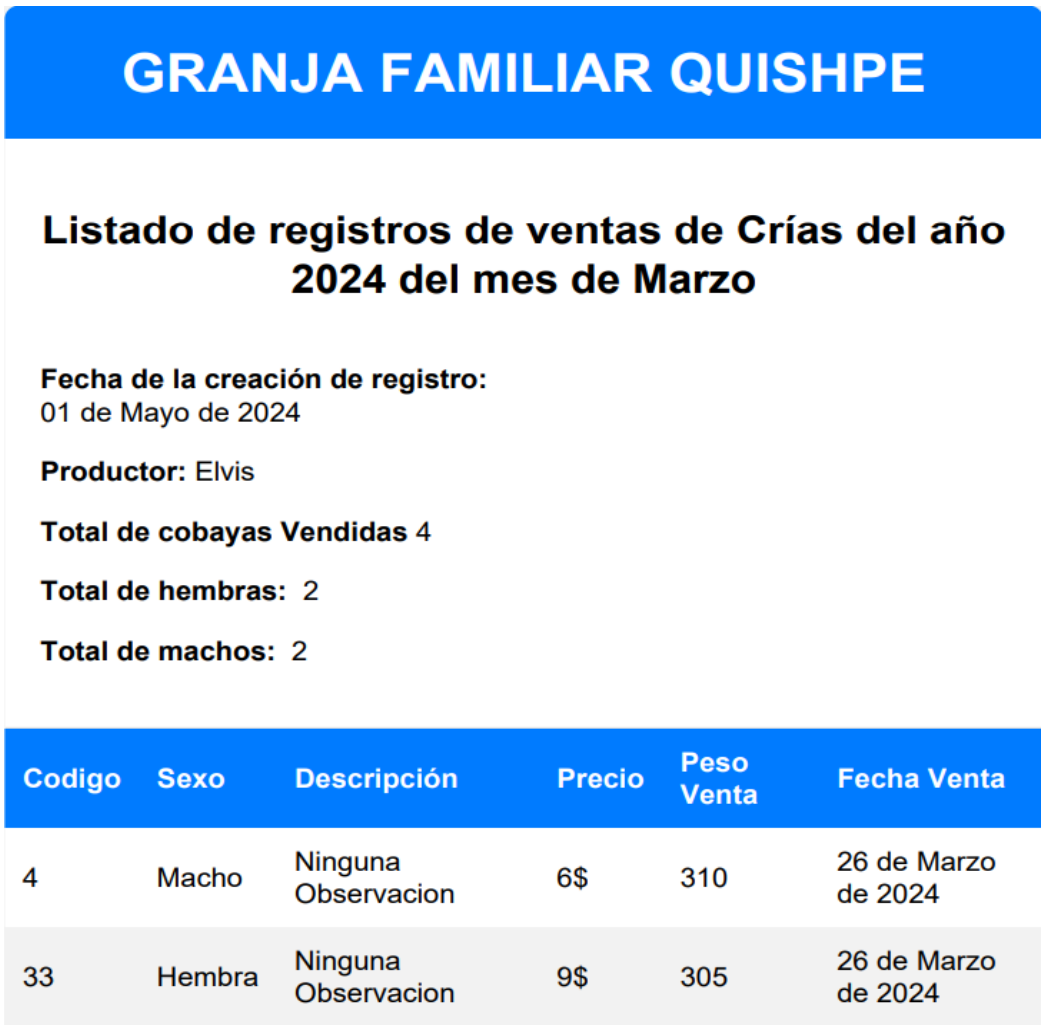
Figura 20
Interfaz de cobayas vendidas.



Reporte de cobayas vendidas

Para obtener una visión general de las cobayas vendidas, tanto reproductores como crías, se debe descargar un archivo en formato PDF. Este documento contiene información detallada sobre las ventas registradas en ambos grupos de cobayas, lo que permite al usuario tener un panorama completo de las transacciones realizadas en la granja. (Ver figura 21)

Figura 21
Reporte PDF sobre ventas



GRANJA FAMILIAR QUISHPE

Listado de registros de ventas de Crías del año 2024 del mes de Marzo

Fecha de la creación de registro:
01 de Mayo de 2024

Productor: Elvis

Total de cobayas Vendidas 4

Total de hembras: 2

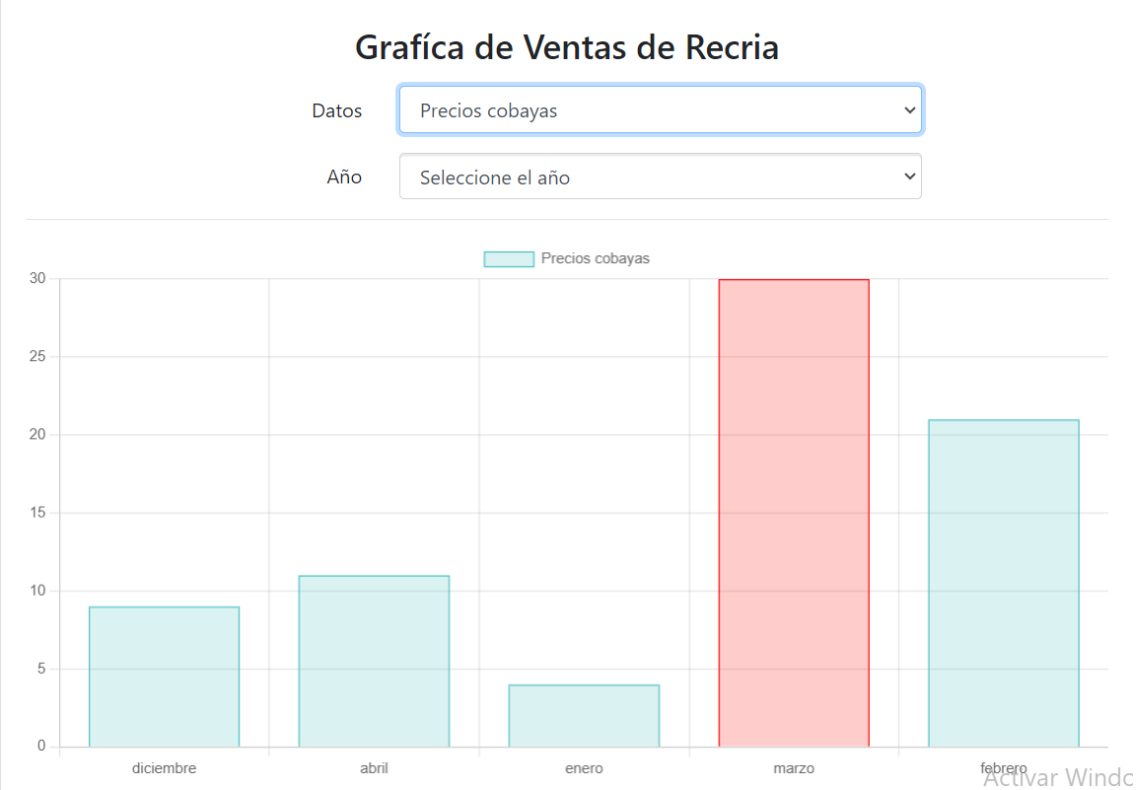
Total de machos: 2

Codigo	Sexo	Descripción	Precio	Peso Venta	Fecha Venta
4	Macho	Ninguna Observacion	6$	310	26 de Marzo de 2024
33	Hembra	Ninguna Observacion	9$	305	26 de Marzo de 2024

Interfaz de estadísticas de las cobayas.

En esta interfaz el productor puede visualizar los datos de las cobayas muertas, vendidas y nacimientos de las crías, puede obtener datos sobre el precio, causa de muerte, el total de cobayas por cada mes, los datos se pueden filtrar por año o pueden ser todos los datos existentes. (Ver figura 22)

Figura 22
Interfaz de estadísticas de las cobayas.



3.3. Resultados de las pruebas de las unitarias

Tras diseñar los casos de prueba y realizar las pruebas de unitarias, los resultados obtenidos se detallan en las Tablas 22 a 29 que se presentan a continuación.

Tabla 22
Resultado del interfaz de inicio de sesión.

Prueba unitaria Numero 1			
Nombre: Inicio de sesión.			
Descripción: El sistema web debe contar con las validaciones necesarias para permitir que los usuarios accedan al sistema de manera segura. Para ello, se requiere que el usuario ingrese su correo electrónico y contraseña válidos a fin de poder iniciar sesión en el sistema.			
Escenarios			
Correo electrónico	**Contraseña**	**Resultado esperado**	**Resultado Obtenido.**
correo@correo12.com (Gmail invalida)	Correo12 (contraseña valida)	El Gmail fue ingresado incorrectamente por el usuario no tiene la estructura adecuada.	Mensaje de error en el campo del Gmail.

correo@correo.com (Gmail valido)	Juan123 (contraseña invalida)	La contraseña proporcionada por el usuario es incorrecta.	Mensaje de error en el campo de contraseña.
Campo vacío	Correo12 (contraseña valida)	El sistema muestra al usuario que el campo de Gmail es obligatorio.	Mensaje de validación en el campo de Gmail.
correo@correo.com (Gmail valido)	Campo vacío	El sistema muestra al usuario que el campo de contraseña es obligatorio.	Mensaje de validación en el campo de la contraseña.
correo@correo.com (Gmail valido)	Correo12 (contraseña valida)	Los campos e información es correcta, el usuario es redirigido a la página principal del sistema.	Redirigir al usuario a la página principal del sistema.
Observaciones:			

Tabla 23
Resultado de la interfaz de jaulas.

Prueba unitaria N° 2				
Nombre: Jaulas				
Descripción: El módulo de jaulas tiene que tener validación al momento de ingresar la información, todos los campos debe estar llenos y algunos registros deben ser de valor único.				
Escenarios				
Número de Jaula	**Tipo de Jaula**	**Fecha de Ingreso**	**Resultado esperado**	**Resultado Obtenido.**
Campo vacío	Campo vacío	Campo vacío	Todos los campos son obligatorios.	Mensaje de error en cada campo vacío
Uno (invalido)	Reproductores12 (invalido)	18/4/2030 (invalido)	Los datos ingresados no son válidos.	Mensaje de error que los datos son incorrectos
1 (invalido)	Reproductores (valido)	18/4/2024 (valido)	El número de jaula debe ser único para	

			los reproductores y recría.	
1 (valido)	Reproductores (valido)	18/4/2024 (valido)	Todos los campos son correctos, se puede crear o actualizar el registro	Mensaje de éxito al final del formulario y redirección a la misma página.
Observaciones:				

Tabla 24

Resultado de la interfaz de cobayas reproductoras.

Prueba unitaria N 3							
Nombre: Cobayas Reproductoras.							
Descripción: La interfaz de cobayas reproductoras debe validar la información proporcionada por el usuario.							
Escenarios							
Código	**Sexo**	**Fecha de Ingreso**	**Observación**	**Enfermedad General**	**Peso Inicial**	**Resultado Esperado**	**Resultado Obtenido.**
Campo Vacío	Campo Vacío	Campo Vacío	Campo Vacío	Campo Vacío	Campo Vacío	Todos los campos son obligatorios	Mensaje de error en cada campo vacío
12@ (invalido)	Machos (invalido)	18/04/39 (invalido)	12131311 (invalido)	Parasitos12 (invalido)	1kg (invalido)	Los datos ingresados no son válidos.	Mensaje de error que los datos son incorrectos
12 (invalido)	Macho (invalido)	18/04/24 (valido)	Ninguna (valido)	Parásitos (valido)	900g (valido)	Solo debe existir un macho por jaula	Mensaje de error en código y macho

						y el código.	por que ya existen esos registros
12 (valido)	Macho (valido)	18/04/24 (valido)	Ninguna (valido)	Parásitos (valido)	900g (valido)	Todos los campos son correctos, se puede crear o actualizar el registro	Mensaje de éxito al final del formulario y redirección a la misma página.
Observaciones:							

Tabla 25
Resultado de la interfaz de partos.

Prueba unitaria N°4						
Nombre: Partos						
Descripción: La siguiente interfaz debe validar todos los campos, también que la información sea la adecuada y solicitada por el sistema.						
Escenarios						
Número Parto	**Número de hijos**	**Número de hijos vivos**	**Número de hijos muertos**	**Fecha de Parto**	**Resultado esperado**	**Resultado Obtenido.**
Campo Vacío	Campo Vacío	Campo Vacío	Campo Vacío	Campo Vacío	Todos los campos son obligatorios	Mensaje de error en cada campo vacío
12a (invalido)	13a (invalido)	8a (invalido)	8a (invalido)	18/04/2030 (invalido)	Los datos ingresados no son válidos.	Mensaje de error que los datos son incorrectos
1 (valido)	5 (valido)	3 (invalido)	3 (invalido)	18/04/24 (valido)	El número de hijos	Mensaje de error al final

					vivos y muertos deben sumar y ser igual al total.	del formulario que la suma de hijos es incorrecta.
1 (valido)	5 (valido)	2 (valido)	3 (valido)	18/04/24 (valido)	Todos los campos son correctos se puede crear o actualizar el registro.	Mensaje de éxito al final del formulario y redirección a la misma página.
Observaciones:						

Tabla 26
Resultado de la interfaz de cobayas crías.

Prueba unitaria N° 5								
Nombre: Cobayas crías.								
Descripción: El sistema web debe verificar que la información de las cobayas crías debe ser la correcta y que todos los campos estén llenos.								
Escenarios								
Código	**Sexo**	**Estado**	**Peso Inicial**	**Observación**	**Enfermedad General**	**Aspecto**	**Resultado esperado**	**Resultado Obtenido.**
Campo Vacío	Campo Vacío	Campo Vacío	Campo Vacío	Campo Vacío	Campo Vacío	Campo Vacío	Todos los campos son obligatorios	Mensaje de error en cada campo vacío
12b1 (invalido)	Hembras (invalido)	Vivos (invalido)	1kg (invalido)	1012121 (invalido)	Parasitos 12 (invalido)	Normal12 (invalido)	Los datos no son validos	Mensaje de error que los

								datos son incorrectos
12 (valido)	Hembra (valido)	Vivo (valido)	900g (valido)	Ninguna (valido)	Parásitos (valido)	Normal (valido)	Todos los datos son correcto se puede crear o actualizar.	Mensaje de éxito al final del formulario y redirección a la misma página.
Observaciones:								

Tabla 27
Resultado de la interfaz de ventas de cobayas

Prueba unitaria N° 6					
Nombre: Ventas de cobayas reproductoras y recrías.					
Descripción: El sistema web permite al usuario registrar datos sobre las ventas de las cobayas en esta interfaz debe ingresar datos sobre el precio y peso final de la cobaya vendida.					
Escenarios					
Jaula	**Código**	**Precio**	**Peso**	**Resultado esperado**	**Resultado Obtenido.**
Campo vacío	Campo vacío	Campo vacío	Campo vacío	Todos los campos son obligatorios	Mensaje de error en cada campo vacío
1as – reproductores (invalido)	120a1 (invalido)	90 (invalido)	1kg (invalido)	Los datos ingresados son inválidos	Mensaje de error que los datos son incorrectos
1 – reproductores (valido)	120a (valido)	9 (valido)	900g (valido)	Todos los campos son correctos crear cobaya vendida.	Mensaje de éxito al final del formulario y

					redirección a la misma página.
Observaciones:					

Tabla 28

Resultado de la interfaz de muertes de cobayas

Prueba unitaria N° 7				
Nombre: Mueres de cobayas reproductoras y recrías.				
Descripción: La siguiente interfaz necesita información solida sobre la causa de muerte y la jaula de procedencia de la cobaya, para de esta forma llenar un conteo de perdidas tanto en reproductores y crías.				
Escenarios				
Jaula	**Código**	**Causa de Muerte**	**Resultado esperado**	**Resultado Obtenido.**
Campo vacío	Campo vacío	Campo vacío	Todos los campos son obligatorios	Mensaje de error en cada campo vacío
12a - Cría (invalido)	120a1 (invalido)	Aborto12 (invalido)	Los datos ingresados son inválidos	Mensaje de error que los datos son incorrectos
12 – Cría (valido)	120a (valido)	Aborto (valido)	Todos los campos son correctos registrar datos.	Mensaje de éxito al final del formulario y redirección a la misma página.
Observaciones:				

Tabla 29

Resultado de la interfaz de recrías

Prueba unitaria N° 8
Nombre: Recrías
Descripción: La siguiente interfaz permite al usuario ingresar datos sobre el peso de destete de las crías después de un tiempo de 15 días para luego evaluar su crecimiento

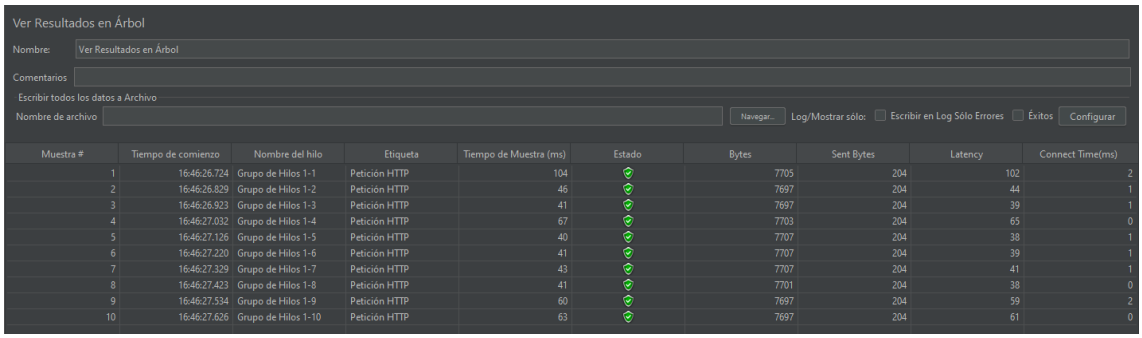
Escenarios			
Código	**Peso destete**	**Resultado esperado**	**Resultado Obtenido.**
Campo vacío	Campo vacío	Todos los campos son obligatorios	Mensaje de error en cada campo vacío
120a1 (invalido)	2kg (invalido)	Los datos ingresados son inválidos	Mensaje de error que los datos son incorrectos
120 (valido)	900g (valido)	Todos los campos son correctos se agrega la cría a una jaula de recría.	Mensaje de éxito al final del formulario y redirección a la misma página.
Observaciones:			

3.4. Resultados de las pruebas de carga y rendimiento.

Tras configurar JMeter con los escenarios de prueba, se realizaron pruebas de carga y rendimiento incrementando gradualmente el número de usuarios concurrentes. Los resultados, que muestran métricas clave como tiempos de respuesta, uso de recursos y rendimiento, se presentan en las siguientes figuras para evaluar el desempeño del sistema bajo diferentes cargas.

Prueba de carga y rendimiento a la interfaz de login

Figura 23
Prueba de carga y rendimiento Login



Prueba de carga y rendimiento a la interfaz de usuarios.

Figura 24

Prueba de carga y rendimiento a la interfaz de usuarios.

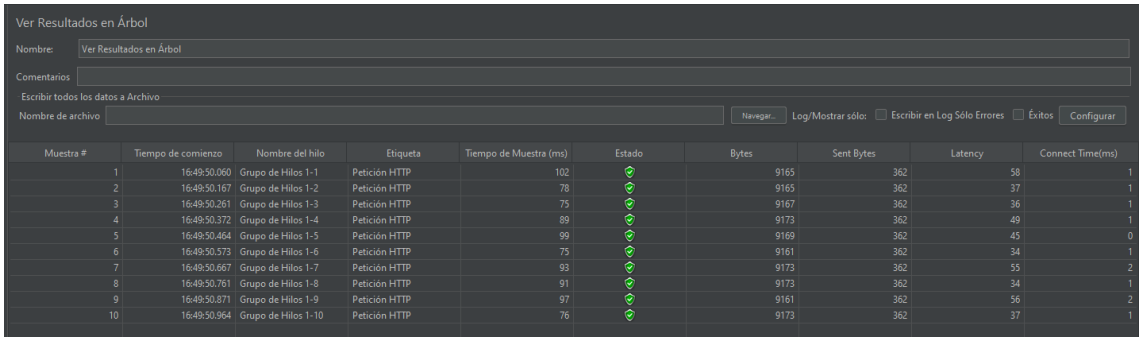


Muestra #	Tiempo de comienzo	Nombre del hilo	Etiqueta	Tiempo de Muestra (ms)	Estado	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	16:48:49.522	Grupo de Hilos 1-1	Petición HTTP	144		9171	351	76	1
2	16:48:49.614	Grupo de Hilos 1-2	Petición HTTP	112		9163	351	57	1
3	16:48:49.714	Grupo de Hilos 1-3	Petición HTTP	51		21017	207	47	0
4	16:48:49.814	Grupo de Hilos 1-4	Petición HTTP	87		9163	351	41	1
5	16:48:49.914	Grupo de Hilos 1-5	Petición HTTP	96		9159	351	42	1
6	16:48:50.015	Grupo de Hilos 1-6	Petición HTTP	94		9161	351	45	1
7	16:48:50.115	Grupo de Hilos 1-7	Petición HTTP	77		9179	351	35	1
8	16:48:50.215	Grupo de Hilos 1-8	Petición HTTP	89		9175	351	42	1
9	16:48:50.315	Grupo de Hilos 1-9	Petición HTTP	86		9163	351	39	1
10	16:48:50.415	Grupo de Hilos 1-10	Petición HTTP	76		9175	351	35	1

Prueba de carga y rendimiento a la interfaz de jaulas.

Figura 25

Prueba de carga y rendimiento a la interfaz de jaulas.



Muestra #	Tiempo de comienzo	Nombre del hilo	Etiqueta	Tiempo de Muestra (ms)	Estado	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	16:49:50.060	Grupo de Hilos 1-1	Petición HTTP	102		9165	362	58	1
2	16:49:50.167	Grupo de Hilos 1-2	Petición HTTP	78		9165	362	37	1
3	16:49:50.261	Grupo de Hilos 1-3	Petición HTTP	75		9167	362	36	1
4	16:49:50.372	Grupo de Hilos 1-4	Petición HTTP	89		9173	362	49	1
5	16:49:50.464	Grupo de Hilos 1-5	Petición HTTP	99		9169	362	45	0
6	16:49:50.573	Grupo de Hilos 1-6	Petición HTTP	75		9161	362	34	1
7	16:49:50.667	Grupo de Hilos 1-7	Petición HTTP	93		9173	362	55	2
8	16:49:50.761	Grupo de Hilos 1-8	Petición HTTP	91		9173	362	34	1
9	16:49:50.871	Grupo de Hilos 1-9	Petición HTTP	97		9161	362	56	2
10	16:49:50.964	Grupo de Hilos 1-10	Petición HTTP	76		9173	362	37	1

Prueba de carga y rendimiento a la interfaz de reproductores.

Figura 26

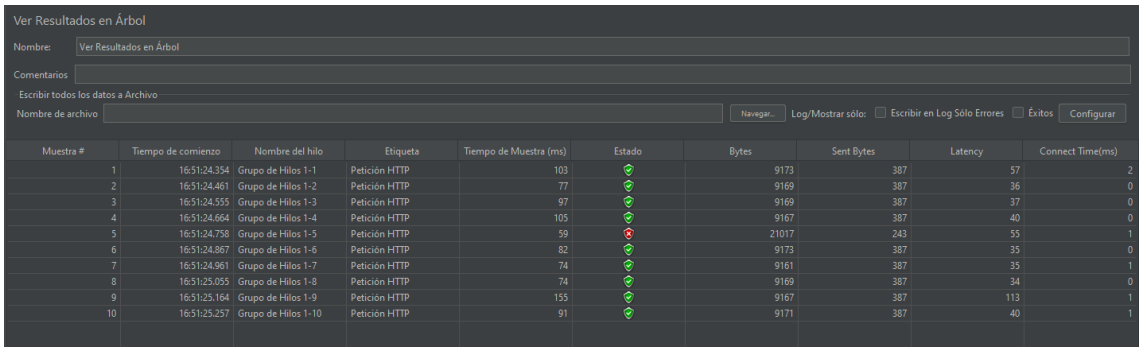
Prueba de carga y rendimiento a la interfaz de reproductores.



Muestra #	Tiempo de comienzo	Nombre del hilo	Etiqueta	Tiempo de Muestra (ms)	Estado	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	16:50:36.309	Grupo de Hilos 1-1	Petición HTTP	95		9165	380	50	1
2	16:50:36.416	Grupo de Hilos 1-2	Petición HTTP	76		9177	380	37	1
3	16:50:36.510	Grupo de Hilos 1-3	Petición HTTP	73		9165	380	33	1
4	16:50:36.621	Grupo de Hilos 1-4	Petición HTTP	101		9175	380	60	1
5	16:50:36.713	Grupo de Hilos 1-5	Petición HTTP	45		21017	236	42	0
6	16:50:36.823	Grupo de Hilos 1-6	Petición HTTP	111		9173	380	71	2
7	16:50:36.916	Grupo de Hilos 1-7	Petición HTTP	75		9165	380	37	0
8	16:50:37.010	Grupo de Hilos 1-8	Petición HTTP	139		9173	380	84	1
9	16:50:37.119	Grupo de Hilos 1-9	Petición HTTP	74		9169	380	34	1
10	16:50:37.213	Grupo de Hilos 1-10	Petición HTTP	74		9167	380	37	1

Prueba de carga y rendimiento a la interfaz de partos.

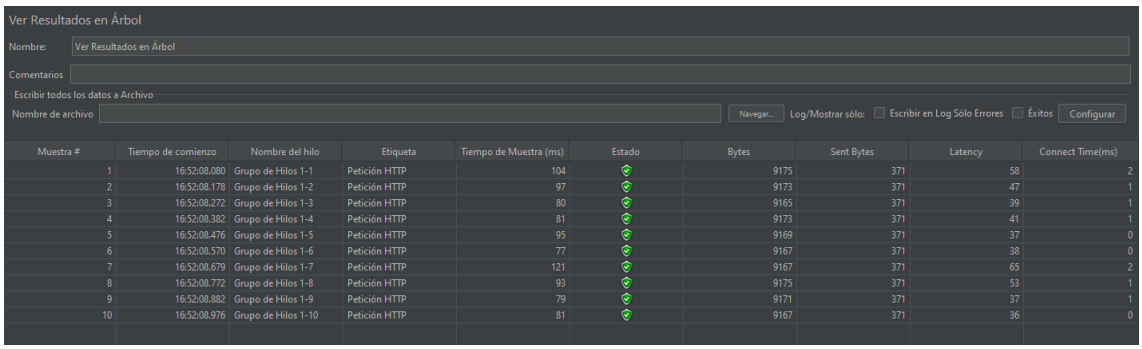
Figura 27
Prueba de carga y rendimiento a la interfaz de partos.



Muestra #	Tiempo de comienzo	Nombre del hilo	Etiqueta	Tiempo de Muestra (ms)	Estado	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	16:51:24.354	Grupo de Hilos 1-1	Petición HTTP	103		9173	387	57	2
2	16:51:24.461	Grupo de Hilos 1-2	Petición HTTP	77		9169	387	36	0
3	16:51:24.555	Grupo de Hilos 1-3	Petición HTTP	97		9169	387	37	0
4	16:51:24.664	Grupo de Hilos 1-4	Petición HTTP	105		9167	387	40	0
5	16:51:24.758	Grupo de Hilos 1-5	Petición HTTP	59		21017	243	55	1
6	16:51:24.867	Grupo de Hilos 1-6	Petición HTTP	82		9173	387	35	0
7	16:51:24.961	Grupo de Hilos 1-7	Petición HTTP	74		9161	387	35	1
8	16:51:25.055	Grupo de Hilos 1-8	Petición HTTP	74		9169	387	34	0
9	16:51:25.164	Grupo de Hilos 1-9	Petición HTTP	155		9167	387	113	1
10	16:51:25.257	Grupo de Hilos 1-10	Petición HTTP	91		9171	387	40	1

Prueba de carga y rendimiento a la interfaz de crías.

Figura 28
Prueba de carga y rendimiento a la interfaz de crías.



Muestra #	Tiempo de comienzo	Nombre del hilo	Etiqueta	Tiempo de Muestra (ms)	Estado	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	16:52:08.080	Grupo de Hilos 1-1	Petición HTTP	104		9175	371	58	2
2	16:52:08.178	Grupo de Hilos 1-2	Petición HTTP	97		9173	371	47	1
3	16:52:08.272	Grupo de Hilos 1-3	Petición HTTP	80		9165	371	39	1
4	16:52:08.382	Grupo de Hilos 1-4	Petición HTTP	81		9173	371	41	1
5	16:52:08.476	Grupo de Hilos 1-5	Petición HTTP	95		9169	371	37	0
6	16:52:08.570	Grupo de Hilos 1-6	Petición HTTP	77		9167	371	38	0
7	16:52:08.679	Grupo de Hilos 1-7	Petición HTTP	121		9167	371	65	2
8	16:52:08.772	Grupo de Hilos 1-8	Petición HTTP	93		9175	371	53	1
9	16:52:08.882	Grupo de Hilos 1-9	Petición HTTP	79		9171	371	37	1
10	16:52:08.976	Grupo de Hilos 1-10	Petición HTTP	81		9167	371	36	0

3.5. Seguridad, respaldos y mantenimiento

3.5.1. Seguridad

3.5.1.1. Seguridad en la contraseña de los usuarios.

Se utilizó Laravel para hashear las contraseñas de usuarios. Al registrarse, la contraseña se cifra y solo el hash se almacena en la base de datos, sin poder visualizar la contraseña original. Para acceder, se valida si la contraseña ingresada coincide con el hash almacenado. Al cambiar la contraseña, se crea un nuevo hash, sin revertir el anterior. Esto preserva la confidencialidad de las contraseñas.

Figura 29
Seguridad en las contraseñas de los usuarios.

id	id_role	name	surname	nick	email	password	image
2	1 →	Mishell	Anrango	mljuan	juan12@gmail.oom	$2y$10$Zv4okaspnSENjeAoI6uevOjp8.rcplaoxGbuCfgzSZvyS33EjMRdq	1716939704speaker_15.…
3	3 →	Pedro	Torres	rppadro	pedro12@gmail.oom	$2y$10$HFYSYio7jaxJp3wmt2RHTuxjrB0w18438qDR/.9ZyxmParMuSegvG	1711481734gato.png
5	1 →	Elvis	Quishpe	Efquishpe	correo@correo.com	$2y$10$m0dcJb881UL1vCXR12XIg.eBMOWcec1q8ua79hh2Z4h6TgsdP8D92	1711432558speaker_18.…
6	3 →	Juan	Torres	juTorres	correo5@correo.com	$2y$10$hMAWeUVsGC/LLPTcWrsLv.HkVzZb/8HCFefhmCwXry0ws/vXG2vQW	1716939574speaker_17.…
7	3 →	Alex	Garcia	ManoloGa…	correo2@correo.com	$2y$10$Mtxt4FrJqgkiWVC/NKEmyu759vgDuss1gfxBAo8.HQzt6Q2foYGYu	1716937994speaker_6.p…

3.5.1.2. Seguridad en almacenamiento de datos.

Se aprovecharon las herramientas de validación y protección de datos que ofrece Laravel. Se implementaron validadores que filtran la información ingresada por el usuario, asegurando que cumpla con los tipos de datos requeridos antes de almacenarse en la base de datos. Además, Laravel facilita la gestión de errores de validación, proporcionándolos en forma de array. Esto permite retornar estos errores a las vistas de manera sencilla, brindando así una retroalimentación clara al usuario sobre los campos que deben ser corregidos o completados antes de procesar la información.

Figura 30
Seguridad en almacenamiento de datos.

```
protected function validator(array $data) {

    $id = $data['iduser'];
    $validator = Validator::make($data, [
                'name' => ['required', 'string', 'max:255', 'alpha'],
                'surname' => ['required', 'string', 'max:255', 'alpha'],
                'nick' => ['required', 'string', 'max:255'],
                'email' => ['required', 'string', 'email', 'max:255', Rule::unique('users')->ignore($id, 'id')],
                'role' => ['required', 'integer', 'max:255'],
                'password' => ['required', 'string', 'min:6', 'confirmed'],
                'password_confirmation' => ['required', 'string', 'min:6',],
    ]);
    if ($validator->fails()) {
        return response()->json(['errors' => $validator->errors()], 422); // Código de respuesta 422 (Unprocessable Enti
    } else {
        return response()->json(['message' => 'Datos válidos'], 200);
    }

    // El código aquí se ejecutará si la validación pasa
}
```

3.5.1.3. Seguridad en las rutas y controladores.

Se crearon funciones en Laravel para controlar el acceso a rutas y funcionalidades según los roles de usuario. Así, el rol de administrador tiene acceso completo, incluyendo modificar usuarios y todas las interfaces. Mientras que el rol de productor sólo accede a interfaces relacionadas con la producción. Esto permite gestionar correctamente los privilegios y reforzar la seguridad del sistema.

Figura 31
Seguridad en el controlador.

```
class UserController extends Controller {

    public function __construct() {
        $this->middleware('checkRole:admin');
    }
```

Figura 32
Seguridad en las rutas.

```
//Rutas de Criadero Jaula
Route::middleware(['checkRole:admin,productor,criador'])->group(function () {
Route::get('/inicio/jaula', 'JaulaController@showInicio')->name('jaula.show');
//saca las jaulas
Route::get('/fildradoJaula/{tipo?}', 'JaulaController@filtradodeJaula');
//saca una Jaula
Route::get('/jaula/one/{id?}/{tipoJaula?}', 'JaulaController@sacarOneJaula');

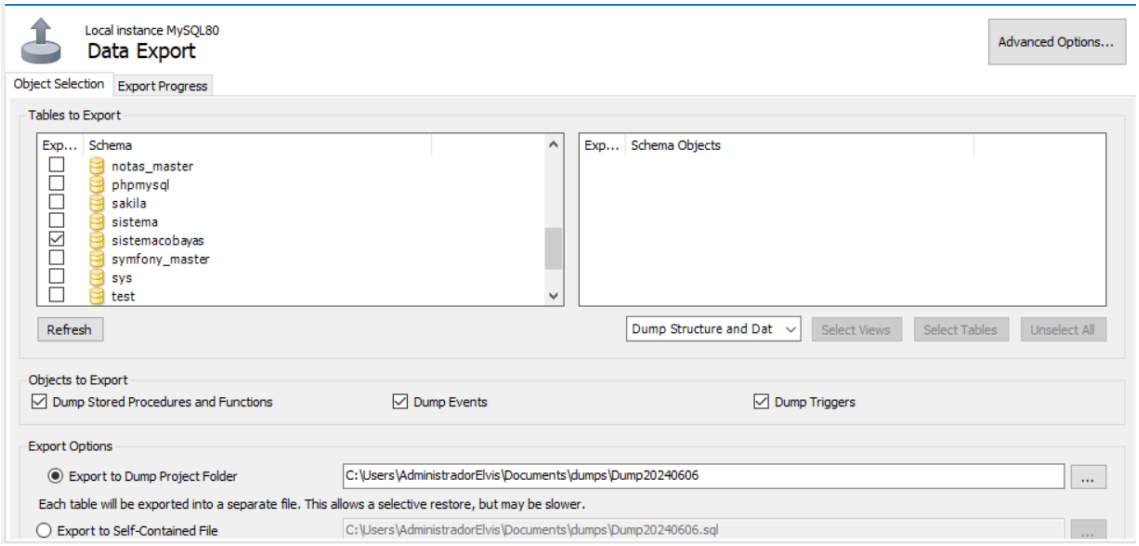
Route::get('/jaula/detalle/{id}/{tipoJaula?}/{tipoSexo?}/{page?}', 'JaulaController@jaulaDetalle')->name('jaula.detalle');
```

3.5.2. Respaldos

3.5.2.1. Respaldo de la Base de datos.

Se utilizó WorkBench para generar respaldos completos de la base de datos, incluyendo la estructura y los datos ingresados por productores y administradores. Esto permite preservar la información crítica del sistema y restaurarla en caso necesario, brindando una capa adicional de protección a los datos.

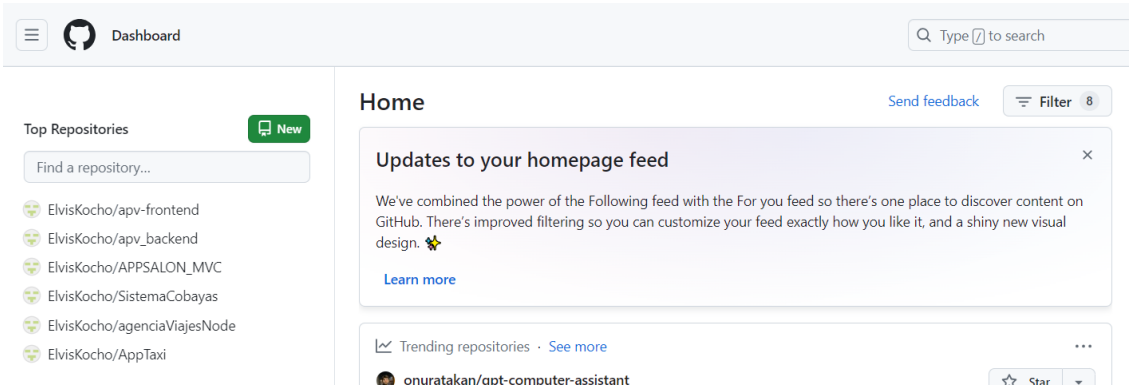
Figura 33
Respaldo de la base de datos



3.5.2.2. Respaldo del sistema web creado en Laravel.

Se creó un repositorio en GitHub para almacenar el código fuente del proyecto. Esta plataforma permite documentar librerías y extensiones, actualizar el código mediante comandos, y descargar el sistema en otros dispositivos. De esta manera, se aprovechan las ventajas de un sistema de control de versiones distribuido para un mejor desarrollo y mantenimiento del software.

Figura 34
Respaldo del sistema web creado en Laravel.



3.5.3. Mantenimiento

3.5.3.1. Actualizaciones de software

Para actualizar el software de la aplicación web, se utiliza Apache NetBeans para modificar el código fuente y Composer para instalar nuevas librerías y actualizar las versiones de los componentes utilizados. Esto permite mantener el sistema actualizado, incorporar nuevas funcionalidades y aplicar parches de manera ágil.

Figura 35
Apache Netbeans IDE 13

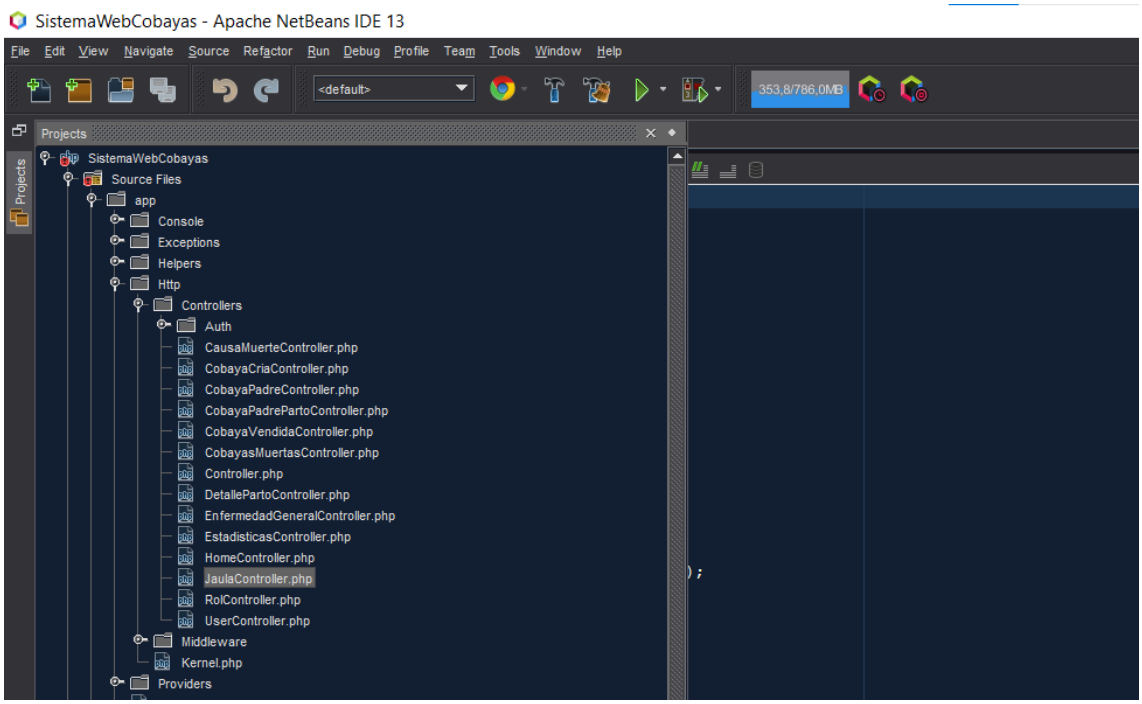
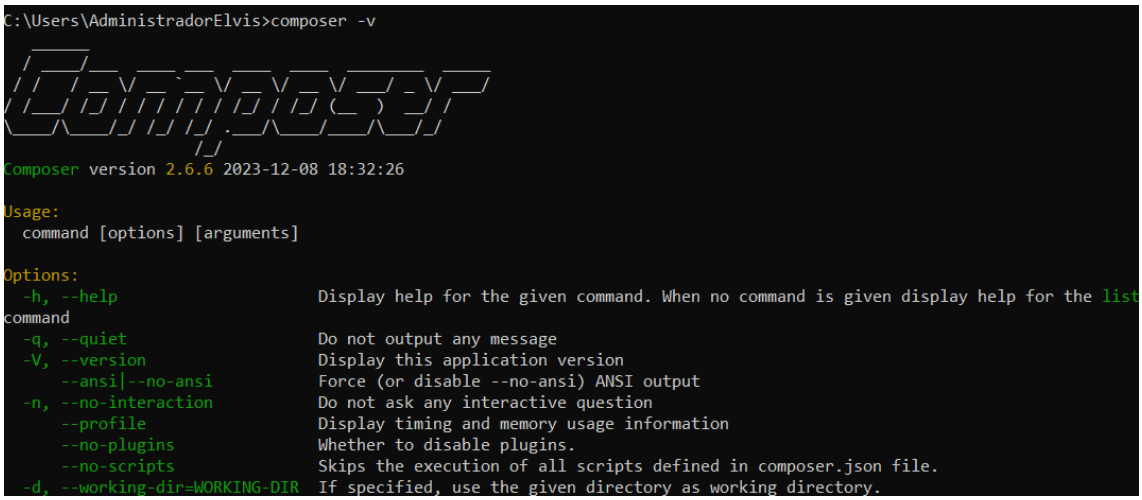


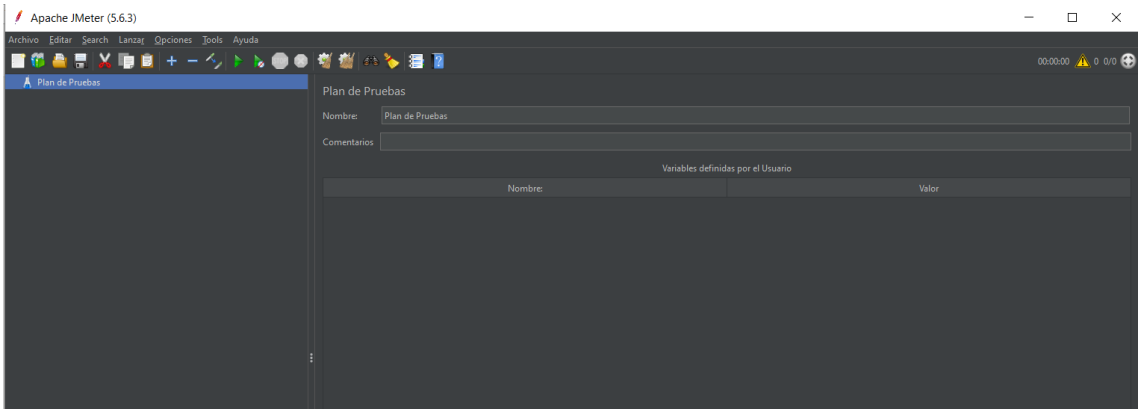
Figura 36
Composer



3.5.3.2. Escalabilidad y rendimiento

Se utiliza Apache JMeter para monitorear el rendimiento bajo diferentes cargas de tráfico. Esto permite identificar cuellos de botella y realizar optimizaciones o escalamiento de recursos para mantener un rendimiento adecuado.

Figura 37
Apache JMeter.



CONCLUSIONES

Después de culminar este proyecto, es posible establecer una serie de conclusiones significativas, las cuales se detallan a continuación:

- Laravel ofrece una amplia gama de herramientas y funcionalidades integradas, como autenticación, enrutamiento, gestión de bases de datos, pruebas unitarias, entre otras, lo que agilizó el desarrollo del sistema web. Además, al utilizar Laravel, se pudo aprovechar su capacidad de gestión de bases de datos relacionales y no relacionales, brindando flexibilidad en el almacenamiento y manipulación de los datos de crianza de cobayas.

- La identificación de los procesos clave sentó las bases para el diseño y desarrollo de módulos específicos dentro del sistema web, adaptados a las necesidades particulares de la crianza de cobayas.

- Se implementaron funcionalidades avanzadas, como la generación de informes, análisis de datos y visualizaciones, para facilitar la toma de decisiones y el monitoreo de los procesos clave.

- La aplicación web facilitó el registro organizado de información clave, como datos de reproductores, partos, crías, ventas y muertes, brindando una visión general del desempeño de la granja en tiempo real.

- Se validó la eficiencia del almacenamiento y recuperación de información, garantizando la integridad y consistencia de los datos relacionados con la crianza de cobayas.

- El proyecto sentó las bases para la adopción de tecnologías digitales en la gestión de la granja, abriendo la posibilidad de explorar soluciones adicionales que optimicen aún más los procesos productivos.

RECOMENDACIONES

Después de completar el desarrollo del sistema web, se presentan las siguientes recomendaciones con el objetivo de mejorar su eficiencia operativa y ampliar sus capacidades funcionales:

- Realizar capacitaciones periódicas al personal de la granja para asegurar el uso óptimo de todas las funcionalidades del sistema web y mantener actualizado su conocimiento sobre el mismo.
- Mantener una comunicación constante con los desarrolladores del sistema web para identificar oportunidades de mejora, reportar posibles errores o incidencias, y solicitar actualizaciones o nuevas funcionalidades según las necesidades cambiantes de la granja.
- Considerar la implementación de una versión móvil del sistema web, o el desarrollo de una aplicación móvil complementaria, para facilitar el acceso y registro de información desde dispositivos móviles en tiempo real.
- Evaluar la posibilidad de ampliar el alcance del sistema web para incluir módulos adicionales, como la gestión de recursos humanos, control de inventarios o análisis de costos de producción, según las necesidades específicas de la granja.
- Desarrollar una funcionalidad de alertas y notificaciones que informe al personal sobre eventos importantes, como próximos partos, registros vencidos, o niveles críticos de existencias, para una gestión más proactiva.
- Evaluar la integración con soluciones de Internet de las Cosas (IoT) para monitorear automáticamente variables ambientales como temperatura, humedad y calidad del aire en las jaulas, optimizando así las condiciones de crianza.

REFERENCIAS BIBLIOGRAFÍCAS

Antonio, J. (2014). Especies alternativas: Manual de crianza de cobayos (Cavia porcellus) - Repositorio Institucional de la Universidad Nacional Agraria. https://repositorio.una.edu.ni/2472/1/renl01v856.pdf

Ballestas Ávila, R. (2020, 11 de junio). Implementación de un sistema de identificación orientado a la web para el control y seguimiento pecuario de una finca con el uso de tecnología RFID [Trabajo de grado, Universidad de Córdoba]. Repositorio Institucional UniCórdoba. https://repositorio.unicordoba.edu.co/entities/publication/93092005-2237-40c9-9146-337a68ec54ff

Caravaca, F., Castel, J., Guzmán, J., Delgado, M., Mena, Y., Alcalde, M., & González, P. (2024). Google Books. https://books.google.es/books?id=yqxte3v1gqkc&pg=PA17

Carlos, J. (2021). Sistema web para el control administrativo de la granja de cuy Coordinadora Procuy Mantaro 2021 [Tesis de grado, Universidad César Vallejo]. Repositorio Institucional UCV. https://hdl.handle.net/20.500.12692/74912

Castillo, C. (2022). Google Books. https://books.google.es/books?id=7taveaaaqbaj&pg=PA10

Ceballos, E., & Maricela, D. (2020). Interacción humano - tecnología, interfaces y usabilidad. RECIAMUC, 4(2), 21-28. https://doi.org/10.26820/reciamuc/4.(2).abril.2020.21-28

Choez Choez, B. (2022). Sistema web para el control de costos de producción y comercialización de cuyes [Trabajo de titulación, Universidad Agraria del Ecuador]. Repositorio Institucional. https://cia.uagraria.edu.ec/archivos/choez%20choez%20bianca%20adriana.pdf

Cíceri, M. (2019). Google Books. https://books.google.es/books?id=spyldwaaqbaj&pg=PA4

Condor, E., & Solís, I. (2014). Google Books. https://books.google.es/books?id=qrg-cqaaqbaj&pg=PA1

Gabriela. (2020). Elaboración de un etograma de gazapos de cobayos (Cavia porcellus) en un sistema de producción en jaula, mediante el uso de un registro focal continuo

[Trabajo de titulación, Universidad Politécnica Salesiana]. Repositorio Institucional UPS. https://dspace.ups.edu.ec/handle/123456789/18907

Galarza Pincay, K. N. (2022). Sistema web para el control sanitario y seguimiento médico en la producción tecnificada de cuyes [Trabajo de titulación, Universidad Agraria del Ecuador]. Repositorio Institucional. https://cia.uagraria.edu.ec/archivos/galarza%20pincay%20karen%20natalia.pdf

Huillcen, H., Palomino, F., & Solís, I. (2022). Google Books. https://books.google.es/books?id=xq5weaaaqbaj&pg=PT3

Ibáñez, L. (2016). Google Books. https://books.google.es/books?id=4aa6eaaaqbaj&pg=PP1

Laínez Fuentes, J. R. (2015). Google Books. https://books.google.es/books?id=m4fjcgaaqbaj&pg=PA6

Lainez, J. (2015). Google Books. https://books.google.es/books?id=m4fjcgaaqbaj&pg=PA6

Mauricio, I., & Rueda, X. (2021). Universidad Politécnica Salesiana sede Cuenca Carrera de Medicina Veterinaria y Zootecnia [Trabajo de titulación, Universidad Politécnica Salesiana]. Repositorio Institucional UPS. https://dspace.ups.edu.ec/bitstream/123456789/21486/1/ups-ct009456.pdf

Mera Paz, J. (2016). Análisis del proceso de pruebas de calidad de software. Revista Ingeniería Solidaria, 12(20), 163-176. https://doi.org/10.16925/in.v12i20.1482

Orlando. (2018). Las tecnologías en la crianza de cuyes y su incidencia en la generación de ingresos económicos para los productores del Distrito Antonio Raymondi, Ancash 2018 [Tesis de maestría, Universidad Nacional del Callao]. Repositorio Institucional UNC. http://hdl.handle.net/20.500.14074/5905

Ortiz, G., & Armando, E. (2024). Establecimiento de la colonia de cobayos linea Hartley en el bioterio de la Universidad Autónoma Metropolitana, Unidad Xochimilco [Trabajo de investigación, Universidad Autónoma Metropolitana]. Repositorio Institucional UAM-X. https://repositorio.xoc.uam.mx/jspui/handle/123456789/41157

Patichoy Calpa, A., & Armando Paredes, D. (2022). Sistema de gestión de información para el manejo de inventario en el proceso de crianza de cuyes de la granja de la

Federación Colombiana de Productores de Papa "Fedepapa" de Obonuco--Nariño [Trabajo de grado, Universidad Mariana]. Repositorio Institucional. https://repositorio.umariana.edu.co/handle/20.500.14112/19397

Piñeiro, J. (2024). Google Books. https://books.google.es/books?id=udfecqaaqbaj&pg=PA1

Xavier, H., Fernando, & Soto, I. dir. (2024). Implementación de un centro de producción y mejoramiento genético de cobayos (cuyes) comerciales [Trabajo de titulación, Universidad Estatal Península de Santa Elena]. Repositorio Institucional UPSE. https://doi.org/ne17362

CAPÍTULO IX

ANEXOS

Anexo 1 *Carta de aceptación*

San pablo, 8 de julio de 2024

Magister

Stalin Marcelo Arciniegas Aguirre

DIRECTOR DE LA ESCUELA DE INGENIERIA DE LA PUCE-SI.

De mi consideración:

La presente tiene como objetivo, hacerle conocer que yo Vinicio Fabian Quishpe Yaselga con CI: 1003113782, **PROPIETARIA** del espacio de terreno donde se implanto el sistema que se encuentra ubicado en el cantón Otavalo Provincia de Imbabura.

Por medio de la presente la **GRANJA FAMILIAR QUISHPE**, tiene el agrado de informar que el Sr. **Elvis Fabián Quishpe Proaño,** portador de la cedula de ciudadanía N.° 1003133782 realizo la entrega del proyecto "**SISTEMA WEB PARA EL SEGUIMIENTO Y CONTROL DE LOS PROCESOS DE CRIANZA DE COBAYAS DE LA GRANJA FAMILIAR QUISHPE.**" Que se desarrolló en beneficio de nuestra granja.

Es grato informar que el proyecto culminado tubo un alto agrado de satisfacción dentro de la granja, cumpliendo con todos los requisitos establecidos. Demostrando compromiso, capacidad y profesionalismo en el desarrollo del proyecto.

Atentamente.



C.C. 100240530-4