



“Sistema de gestión de ganado basado en la nube: una herramienta para la optimización de la trazabilidad financiera.”

TESIS PARA TITULACIÓN INTEGRAL

INGENIERÍA SISTEMAS COMPUTACIONALES

EGRESADO:

LUIS EDUARDO LOPEZ GARCIA

DIRECTOR Y CO-DIRECTOR:

DRA. MARLENE MÉNDEZ MORENO

DR. MARIO BENHUR CHUC ARMENDÁRIZ

Calkiní, Campeche, México 2025





INSTITUTO TECNOLÓGICO SUPERIOR DE CALKINÍ

Sistema de gestión de ganado basado en la nube: una herramienta para la optimización de la trazabilidad financiera.

Que presenta:

Luis Eduardo López García

Ingeniero en Sistemas Computacionales

Asesor:

DRA. MARLENE MENDEZ MORENO

DR. MARIO BENHUR CHUC ARMENDÁRIZ

Calkiní, Campeche, México

FEBRERO, 2025



Educación
Secretaría de Educación Pública



INSTITUTO TECNOLÓGICO
SUPERIOR DE CALKINÍ



Instituto Tecnológico Superior de Calkiní
Subdirección de Planeación y Desarrollo

Instituto Tecnológico Superior de Calkiní
Departamento de Administración y
Apoyo al Estudiante

Calkiní, Campeche 07/03/2025
P/03/25/636

INGENIERÍA EN SISTEMAS COMPUTACIONALES

CONSTANCIA DE NO INCONVENIENCIA

MODALIDAD: TESIS

C. LUIS EDUARDO LOPEZ GARCIA 7257
PASANTE DE LA CARRERA DE
INGENIERÍA EN SISTEMAS COMPUTACIONALES.
PRESENTE

Con relación a lo establecido en el Reglamento de Estudios de Licenciatura, capítulo VII, fracción I, vigente en este Instituto, y una vez emitido el fallo favorable por el director DRA. MARLENE MÉNDEZ MORENO, y el co-director DR. MARIO BENHUR CHUC ARMENDÁRIZ, respectivamente, este Instituto consta de no tener inconveniencia para la realización del acto protocolario de titulación integral de su trabajo profesional titulado: **"SISTEMA DE GESTIÓN DE GANADO BASADO EN LA NUBE: UNA HERRAMIENTA PARA LA OPTIMIZACION DE LA TRAZABILIDAD FINANCIERA"**.

ATENTAMENTE

Excelencia en Educación Tecnológica®
Sabiduría, Fortaleza de Nuestra Cultura®

Dr. Jorge Alfonso Hau Puc.
Departamento de Administración y
Apoyo al Estudiante.

C.c.p. Archivo



INSTITUTO TECNOLÓGICO SUPERIOR
DE CALKINÍ
"DEPARTAMENTO DE ADMINISTRACIÓN
ESCOLAR Y APOYO A
ESTUDIANTES"
CLAVE: 04MSU0013P
CALKINÍ, CAMPECHE, MÉXICO.



2025
Año de
La Mujer
Indígena

Av. Ah Canul 8/N por Carretera Federal, Calkiní, Campeche,
C.P. 24900 Tel. 996-8134870,
e-mail: se_calkini@tecnm.mx | itescam.edu.mx



Formato de Evaluación de asesores interno y externo



Dedicatoria.

A ti, mamá, porque has sido mi mayor inspiración y mi fuerza en cada paso que he dado. Gracias por estar siempre ahí, por creer en mí incluso cuando yo dudaba, por motivarme a seguir adelante y por enseñarme que con esfuerzo y dedicación todo es posible. Cada logro que alcanzo es también tuyo, porque sin tu amor, tus desvelos y tus sacrificios, este camino habría sido mucho más difícil. Sé que estás orgullosa de cada uno de tus hijos, y eso es lo que más nos motiva a seguir adelante. Tu felicidad y tu orgullo son nuestro mayor motor, y no hay nada más valioso para nosotros que verte sonreír.

A mi familia, mi refugio y mi mayor apoyo. Gracias por ser mi impulso en los momentos difíciles, por su cariño incondicional y por acompañarme en cada etapa de este proceso. Su confianza en mí me ha dado la fuerza para seguir adelante y superar cada obstáculo.

Y a todas las personas que han creído en mí, que me han alentado, aconsejado y apoyado en este camino, les dedico este esfuerzo. Sus palabras, su confianza y su compañía han sido fundamentales para llegar hasta aquí.

Este logro es tan mío como de ustedes. Gracias por estar conmigo en este viaje, por hacer que cada paso valiera la pena y por ser parte de este momento tan especial.



Agradecimientos.

En primer lugar, agradezco a Dios, quien me ha dado la fortaleza, sabiduría y perseverancia para enfrentar cada reto que se presentó a lo largo de este proyecto y de mi formación académica. Sin su guía, este logro no habría sido posible.

A mi madre, quien siempre me motivó, confió y creyó en mí incluso en los momentos más difíciles. Gracias por ser mi mayor inspiración, por darme ánimos cuando más los necesitaba y por enseñarme que con esfuerzo y dedicación se pueden alcanzar todas las metas. Este logro no es solo mío, sino también tuyo, porque detrás de cada paso que di, estuviste tú, guiándome con amor, paciencia y sacrificio. Me enseñaste todo lo que sé, me apoyaste en cada decisión que tomé y te preocupaste por mí en cada etapa de mi vida. En las largas noches de desvelo, mientras yo aprendía, tú estabas ahí, acompañándome, asegurándote de que nunca me rindiera. Gracias por inculcarme el valor de la responsabilidad, la importancia del esfuerzo y el poder de la perseverancia. Todo lo que soy y todo lo que logre será siempre un reflejo de tu amor incondicional.

A mis maestros, quienes con su paciencia, experiencia y guía contribuyeron de manera invaluable al desarrollo de este proyecto. Sus consejos y retroalimentación fueron fundamentales para darle forma y profundidad a mi trabajo.

A mis compañeros y amigos, por compartir conmigo este viaje lleno de aprendizajes, retos y momentos inolvidables. Su compañía, motivación y apoyo hicieron de este camino una experiencia enriquecedora y gratificante.

A mi institución, el Instituto Tecnológico Superior de Calkiní, por brindarme la oportunidad de formarme como profesional y por facilitarme las herramientas y conocimientos necesarios para superar cada desafío académico y personal.

A Aldeas Infantiles por todo el apoyo brindado a lo largo de mi camino. Su ayuda ha sido fundamental en mi desarrollo personal y académico, ofreciéndome oportunidades y acompañándome en cada paso. Gracias por abrirme puertas, por estar ahí cuando más lo necesité y por darme las herramientas para seguir adelante. Su labor ha sido una luz en mi vida, y siempre llevaré en mi corazón el impacto positivo que han tenido en mi formación.

Finalmente, agradezco a todas las personas que, de una forma u otra, contribuyeron al desarrollo de este proyecto. Cada palabra de aliento, cada gesto de apoyo y cada enseñanza dejaron una huella importante en este camino. Este logro es el resultado de un esfuerzo colectivo, y siempre estaré profundamente agradecido por ello.



Contenido

Dedicatoria.....	5
Agradecimientos.	6
Índice Figuras.....	9
Índice Tablas.....	10
Resumen.....	11
Abstract	11
Capítulo 1.....	12
Introducción.....	12
1.2 ANTECEDENTES. Y/O MARCO TEÓRICO	13
PLANTEAMIENTO DEL PROBLEMA.....	14
OBJETIVOS GENERALES Y ESPECIFICOS.	15
JUSTIFICACIÓN.....	16
ALCANCES Y LIMITACIONES.	16
Capítulo 2 Marco Teórico	18
CAPITULO 3	22
METODOLOGIA	22
Casos de Uso.....	23
Desarrollo de la Base de Datos.....	23
Capítulo 4 Resultados y discusión	26
Interfaz de Usuario	26
Pruebas.....	43
Capítulo 5 Conclusiones y recomendaciones.....	45
Conclusiones.....	45
Recomendaciones.....	46
Referencias Bibliográficas.....	47
Anexos	48
BusquedaAnimal3.java	48
HomeBusqueda.java	60
BuscarAnimal2	65
HomeBusquedaLotes.java.....	75

BuscarAnimal1	79
HomeGanado.java	87
Gasto.java	91
Gastos.java	94
Editar.java	104
Precios.java	108
PreciosV.java	113
VerAnimal.java.....	117
VerMas.java.....	124
VerMas1.java.....	128
VerMasAnimal.java.....	132
VerAnimalClient.java.....	136
VerMasClient.java.....	143
Animal.java	147
BuscarAnimal.java	149
EditarAnimal.java	174
Ganado.java.....	188
RegistrarAnimal.java.....	193
RegistrarAnimales.java.....	206
Registro.java.....	210
Conexión.java.....	211
Inicio.java.....	212
Inicio1.java.....	224
Login.java.....	232
Movimientos.java	240
RegistrarUsuario.java	244

Índice Figuras.

Ilustración 1 diagrama de uso	23
Ilustración 2 Tabla base de datos.....	24
Ilustración 3 Estructura de la tabla Animales.....	24
Ilustración 4 Estructura de la tabla Gastos	25
Ilustración 5 Estructura de la tabla Historial	25
Ilustración 6 Estructura de la tabla Movimientos	25
Ilustración 7 Estructura de la tabla Precios	26
Ilustración 8 Estructura de la tabla Presupuesto	26
Ilustración 9 Estructura de la tabla Registro de usuarios	26
Ilustración 10 Pantalla inicio de sesión.....	27
Ilustración 11 Pantalla Registro de Usuarios.....	28
Ilustración 12 Vista Cliente	29
Ilustración 13 Vista administrador	30
Ilustración 14 búsqueda por orden de registro	31
Ilustración 15 Editar Registro de ganado	32
Ilustración 16 Información del ganado ya registrado	34
Ilustración 17 Trazabilidad del ganado	35
Ilustración 18 Trazabilidad del ganado por filtros	36
Ilustración 19 Registro de los movimientos en el sistema	37
Ilustración 20 Administrador de los precios por kilo	38
Ilustración 21 Administrador de presupuestos.....	39
Ilustración 22 Registro del ganado	40
Ilustración 23 Trazabilidad financiera	42

Índice Tablas.



Resumen.

El presente trabajo de investigación tiene como objetivo el desarrollo de un sistema de gestión de ganado basado en la nube enfocado en la optimización de la trazabilidad financiera de las operaciones ganaderas. Este sistema permite a los ganaderos gestionar de manera eficiente los aspectos financieros relacionados con la adquisición, manejo y comercialización de los animales. A través de una plataforma accesible desde cualquier dispositivo, los usuarios pueden llevar un control detallado de los gastos y los ingresos asociados a cada animal, lo que facilita la toma de decisiones y mejora la rentabilidad de la explotación ganadera. Además, se implementan herramientas que permiten la integración de los movimientos financieros con los registros del ganado, proporcionando una visión clara de la situación económica de la granja. La solución desarrollada no solo mejora la eficiencia operativa, sino que también optimiza el control y seguimiento de los recursos financieros a lo largo de todo el ciclo de vida del ganado.

Abstract

This research aims to develop a cloud-based livestock management system focused on optimizing the financial traceability of livestock operations. This system enables farmers to efficiently manage the financial aspects related to the acquisition, management, and commercialization of animals. Through an accessible platform from any device, users can maintain a detailed record of expenses and revenues associated with each animal, facilitating decision-making and improving the profitability of the livestock enterprise. Additionally, tools are implemented to integrate financial movements with livestock records, providing a clear view of the farm's economic situation. The developed solution not only improves operational efficiency but also optimizes the control and monitoring of financial resources throughout the entire lifecycle of the livestock.

Capítulo 1

Introducción.

La ganadería es una de las actividades económicas más relevantes dentro del sector agropecuario, desempeñando un papel crucial en la producción de carne, leche y otros derivados. Sin embargo, la administración financiera de las unidades productivas ganaderas sigue siendo un desafío debido a la falta de herramientas digitales especializadas que permitan un control eficiente de los ingresos y egresos. La ausencia de trazabilidad financiera impide a los productores realizar una gestión óptima de sus recursos, dificultando la planificación presupuestaria y la toma de decisiones estratégicas.

El avance tecnológico ha permitido la implementación de sistemas de gestión basados en la nube, los cuales han demostrado ser una solución efectiva para mejorar la organización y el acceso a la información en tiempo real. Estos sistemas no solo han optimizado la trazabilidad de los animales en términos de sanidad y movilidad, sino que también ofrecen la posibilidad de llevar un registro estructurado de los aspectos financieros del negocio. A través de la recopilación y análisis de datos económicos, los ganaderos pueden evaluar con mayor precisión su rentabilidad, identificar patrones de gasto y optimizar sus inversiones.

Este estudio tiene como objetivo analizar cómo un sistema de gestión de ganado basado en la nube contribuye a la trazabilidad financiera del negocio ganadero. Para ello, se explorará cómo la digitalización de los registros económicos permite mejorar la eficiencia en la administración de recursos, asegurando un mejor control de los ingresos provenientes de la venta de ganado y de los egresos asociados a su mantenimiento.

La investigación se estructura en diferentes secciones. Primero, se presentan los antecedentes y fundamentos teóricos que sustentan la importancia de la trazabilidad financiera y el uso de la tecnología en la ganadería. Luego, se detalla la metodología utilizada para el desarrollo e implementación del sistema de gestión. Posteriormente, se analizan los resultados obtenidos a partir del uso del sistema en un entorno ganadero real. Finalmente, se exponen las conclusiones y recomendaciones que buscan impulsar la modernización financiera del sector mediante el uso de herramientas digitales avanzadas.

Con esta investigación, se espera demostrar que un sistema de gestión basado en la nube no solo optimiza el control del ganado, sino que también proporciona una trazabilidad financiera más precisa y accesible, permitiendo a los productores tomar decisiones informadas y mejorar la rentabilidad de sus negocios.

1.2 ANTECEDENTES. Y/O MARCO TEÓRICO

El sector ganadero ha evolucionado significativamente con la adopción de tecnologías digitales, especialmente en lo que respecta a la trazabilidad y gestión financiera. La digitalización ha permitido optimizar la administración de los recursos, facilitando el seguimiento de ingresos y egresos en la producción pecuaria.

Trazabilidad y digitalización en la ganadería

La trazabilidad es un proceso clave en la ganadería moderna, ya que permite registrar la información relevante de los animales, desde su nacimiento hasta su comercialización. En muchos países, los sistemas de trazabilidad animal son obligatorios y están digitalizados, mejorando el control sanitario y comercial del ganado (González-García, 2020). En México, aunque la adopción de estas tecnologías aún es limitada, existe un creciente interés en su implementación (INEGI, 2021).

El uso de tecnologías en la nube ha facilitado la digitalización del sector agropecuario. Estas herramientas permiten la recopilación y almacenamiento de grandes volúmenes de datos, mejorando la toma de decisiones estratégicas (Wolfert, Verdouw & Bogaardt, 2017). Sin embargo, uno de los desafíos en México es la baja infraestructura tecnológica en zonas rurales, lo que dificulta la adopción de estos sistemas (Klerkx, Jakku & Labarthe, 2019).

Trazabilidad financiera en la ganadería

Si bien la trazabilidad animal ha sido ampliamente estudiada, la trazabilidad financiera en la ganadería es un área en desarrollo. En la actualidad, los productores necesitan herramientas que les permitan registrar sus ingresos y egresos de manera eficiente, a fin de mejorar la rentabilidad de sus explotaciones.

Un sistema financiero digital permite llevar un control detallado de los costos asociados a la producción ganadera, como la compra de insumos, alimentación, servicios veterinarios y mano de obra (Sørensen et al., 2019). A su vez, estos sistemas pueden ayudar a optimizar el flujo de efectivo y mejorar la rentabilidad mediante la identificación de patrones de gasto innecesarios (FAO, 2020).

En este contexto, la implementación de sistemas de gestión financiera digital en la ganadería mexicana representa una oportunidad para mejorar la sostenibilidad del sector. Según el INEGI (2021), solo el 18% de las fincas del país han adoptado herramientas digitales para la gestión económica, lo que resalta la necesidad de modernización.

La digitalización de la trazabilidad financiera permite a los productores:

- Identificar los costos de producción con mayor precisión.
- Controlar los ingresos derivados de la comercialización del ganado.

- Evaluar la rentabilidad de la explotación en tiempo real.

Importancia del sistema propuesto

El sistema de gestión financiera basado en la nube desarrollado en este estudio proporciona un mecanismo para la trazabilidad financiera en la ganadería. A diferencia de otros sistemas que se enfocan solo en la trazabilidad animal, esta herramienta permite:

- Registrar los ingresos provenientes de la venta de ganado.
- Administrar presupuestos mensuales mediante la categorización de gastos.
- Calcular automáticamente las ganancias, proporcionando a los ganaderos una visión clara de su situación financiera.

La implementación de este tipo de soluciones puede contribuir significativamente a la eficiencia operativa del sector, ayudando a los productores a optimizar sus recursos y mejorar su competitividad en el mercado.

PLANTEAMIENTO DEL PROBLEMA

La ganadería es una de las actividades más relevantes para la economía agrícola, pero enfrenta importantes desafíos debido a la escasa adopción de tecnologías digitales que optimicen su gestión. Este rezago tecnológico impacta negativamente en áreas clave como la trazabilidad, la eficiencia operativa y la gestión de datos críticos, lo que limita la capacidad de los productores para adaptarse a las exigencias del mercado global y garantizar estándares de calidad. Además, la falta de herramientas adecuadas repercute directamente en la rentabilidad, ya que incrementa los costos operativos, aumenta el riesgo de errores humanos y genera pérdidas económicas, afectando la competitividad del sector.

En México, aunque existen soluciones tecnológicas emergentes para la ganadería, la implementación efectiva de estas herramientas en las fincas sigue siendo limitada. Esto ha revelado una brecha significativa entre las tecnologías disponibles y su adopción real en la práctica. Los sistemas actuales no permiten una integración eficiente de los datos ni un control adecuado de los costos, lo que compromete la calidad de la información y eleva los costos operativos y las pérdidas financieras.

El sistema de gestión de ganado basado en la nube con un enfoque en la trazabilidad financiera ha sido desarrollado como una solución para superar estas barreras, pero aún no ha sido implementado en un entorno real de trabajo. Este sistema permite la integración de funciones de organización, almacenamiento y acceso remoto a los datos, brindando a los ganaderos herramientas para tomar decisiones más informadas y mejorar la eficiencia operativa. Además, el sistema facilita la trazabilidad tanto de los animales como de los flujos financieros, lo que contribuirá a una mayor transparencia y control de los costos operativos, ayudando a los productores a mejorar su rentabilidad y adaptarse a los estándares exigidos

por el mercado global. Su efectividad será evaluada a través de este proyecto, lo que permitirá comprobar su impacto y viabilidad para su futura implementación.

Se plantea la siguiente pregunta de investigación: ¿Cómo puede un sistema de gestión de ganado basado en la nube optimizar la trazabilidad financiera en la gestión ganadera? Para abordar este tema, se enfoca en el aspecto financiero, explorando cómo esta tecnología puede mejorar el control y la transparencia de los flujos económicos relacionados con la ganadería, como los costos operativos y los presupuestos anuales. A partir de ello, se formula la hipótesis de que la implementación de un sistema basado en la nube permite la asignación y consulta de presupuestos anuales, además del seguimiento del precio del ganado, lo que facilita un mejor control de los costos operativos y contribuye a la rentabilidad a lo largo del tiempo.

OBJETIVOS GENERALES Y ESPECIFICOS.

1) Objetivo General:

Evaluar la efectividad de un sistema de gestión de ganado basado en la nube para mejorar la trazabilidad financiera, facilitando la asignación y consulta de presupuestos anuales, el seguimiento de los costos operativos y el precio del ganado, y contribuyendo a la rentabilidad y eficiencia en la gestión ganadera.

2) Objetivos específicos:

- Evaluar la interfaz de usuario del sistema para determinar su eficacia en la experiencia del usuario, asegurando que sea intuitiva y fácil de usar para los ganaderos y administradores.
- Verificar la seguridad del sistema de autenticación, comprobando si ofrece un acceso adecuado y seguro para los diferentes tipos de usuarios, con roles y permisos específicos.
- Comprobar la eficacia del sistema para facilitar la consulta de los animales registrados, proporcionando acceso rápido y preciso a la información general y detallada sobre el ganado.
- Evaluar la funcionalidad de la búsqueda avanzada por número de identificación, analizando si permite una consulta rápida y precisa de datos específicos sobre los animales.
- Verificar la eficiencia de la gestión de lotes de animales, comprobando si el sistema permite clasificar y filtrar animales según su estado (ej. preñados, recién nacidos, etc.).
- Comprobar que la base de datos en la nube esté sincronizada y accesible en tiempo real, garantizando un acceso remoto adecuado para todos los usuarios autorizados.
- Evaluar la efectividad del sistema en el registro y consulta de movimientos de los animales, comprobando si proporciona un historial detallado y preciso de las actividades realizadas sobre cada uno de ellos.

- Analizar la gestión de los presupuestos anuales en el sistema, verificando su capacidad para asignar, consultar y actualizar presupuestos relacionados con la operación de la finca.
- Evaluar la trazabilidad financiera en el sistema, comprobando si facilita el seguimiento detallado de los flujos financieros vinculados a la gestión del ganado y contribuye al control de los costos operativos.

JUSTIFICACIÓN

El desarrollo de un sistema de gestión de ganado basado en la nube con un enfoque específico en la trazabilidad financiera ofrece una solución potencial para superar estos retos. Aunque el sistema ya ha sido desarrollado, aún no ha sido implementado, lo que hace que este proyecto sea crucial para comprobar su viabilidad y eficacia. El sistema permite la centralización de la información sobre el ganado y los flujos financieros de la finca, lo que facilita la asignación y consulta de los presupuestos anuales, el seguimiento del precio del ganado y un control más preciso de los costos operativos.

Al integrar las funciones operativas y financieras en un solo sistema accesible de forma remota y segura, el sistema tiene el potencial de mejorar la toma de decisiones, optimizar la rentabilidad y promover una gestión más eficiente de los recursos. Además, facilita la trazabilidad de los animales y ofrece un análisis detallado de los flujos financieros, lo que contribuirá a una mayor transparencia y control de los costos operativos.

Los principales beneficiarios de este sistema serían los ganaderos, quienes experimentarán un incremento en la eficiencia operativa, mayor control sobre sus recursos y una mejor capacidad para tomar decisiones informadas. Asimismo, una sede en Mérida, que podrá acceder a la información de las fincas de manera remota, contribuiría a una supervisión más transparente y efectiva sin comprometer la seguridad de los datos.

Este proyecto no solo busca solucionar los problemas actuales en la gestión ganadera, sino que también tiene el potencial de ser un referente para la implementación de tecnologías digitales en el sector agrícola. La validación y análisis de este sistema proporcionará información crucial sobre su impacto, abriendo la puerta a una mayor adopción de soluciones tecnológicas en la ganadería, mejorando la competitividad y sostenibilidad del sector tanto a nivel nacional como internacional.

ALCANCES Y LIMITACIONES.

La presente investigación tiene como propósito evaluar la implementación de un sistema de gestión de ganado basado en la nube y su impacto en la trazabilidad financiera. Para ello, se analizarán las siguientes características del sistema:

- Interfaz amigable: Se verificará la facilidad de uso de la interfaz para los ganaderos y administradores, asegurando su accesibilidad desde computadoras con Windows.
- Registro de cambios en el ganado: Se analizará cómo el sistema gestiona y almacena cambios en el estado reproductivo, alimenticio, sanitario y corporal de los animales.
- Consultas personalizadas: Se evaluará la eficacia del sistema para permitir la consulta y filtrado de los datos del ganado según diferentes criterios (estado reproductivo, condición corporal, salud, etc.).
- Historial de modificaciones: Se examinará la funcionalidad de registro automático de cambios en los estados del ganado para evaluar su utilidad en el seguimiento de los animales.
- Almacenamiento en la nube: Se analizará la accesibilidad y seguridad de los datos almacenados en la nube.
- Base de datos MySQL: Se verificará el correcto almacenamiento y consulta de la información en la base de datos MySQL utilizada por el sistema.
- Gestión financiera: Se evaluará la funcionalidad del módulo de presupuestos anuales para registrar y consultar gastos e ingresos de la finca.
- Trazabilidad financiera: Se analizará cómo el sistema permite el seguimiento detallado de ingresos y egresos para mejorar la toma de decisiones y la rentabilidad ganadera.

Si bien el sistema ofrece herramientas para mejorar la gestión del ganado y la trazabilidad financiera, su implementación presenta las siguientes limitaciones:

- Enfoque en gestión, no en producción: El sistema no mide indicadores productivos como la cantidad de leche o carne generada por los animales.
- Ausencia de estadísticas en tiempo real: No se generan reportes en tiempo real sobre la productividad o eficiencia del ganado.
- Dependencia de conexión a internet: La funcionalidad del sistema está limitada en entornos sin acceso estable a internet.
- Compatibilidad con Windows únicamente: Actualmente, el sistema solo puede ser utilizado en computadoras con sistema operativo Windows, excluyendo dispositivos móviles.
- Dependencia de MySQL: La administración de la base de datos MySQL requiere ciertos conocimientos técnicos.
- Restricción en la edición del presupuesto: Actualmente, los montos asignados a cada concepto dentro del presupuesto no pueden ser modificados ni eliminados una vez registrados. Esto garantiza la integridad de los datos financieros, pero podría representar una limitación para ajustes o correcciones. En futuras actualizaciones, se podría evaluar la implementación de mecanismos de edición controlada para mejorar la flexibilidad en la gestión financiera.

Capítulo 2 Marco Teórico

2.1 Fundamentación teórica

En este capítulo se presentan los fundamentos teóricos que sustentan el desarrollo e implementación del sistema de gestión de ganado basado en la nube. Se abordan conceptos clave relacionados con la trazabilidad financiera, la gestión de datos en la nube y las tecnologías utilizadas para garantizar la eficiencia operativa en el sector ganadero.

2.1.1 Fundamentos de la gestión ganadera

La ganadería es una actividad fundamental para la economía agrícola, ya que proporciona productos esenciales como carne, leche y derivados. Sin embargo, su eficiencia depende en gran medida de la capacidad de los productores para gestionar adecuadamente sus recursos, optimizar costos y mejorar la trazabilidad de sus operaciones.

El control y registro de información sobre el ganado, su estado reproductivo, sanitario y alimenticio, así como los costos operativos asociados, son aspectos clave para garantizar una administración eficiente. La digitalización de estos procesos a través de sistemas de gestión permite mejorar la toma de decisiones y reducir la probabilidad de errores humanos, incrementando la competitividad del sector.

2.1.2 Trazabilidad financiera en la ganadería

La trazabilidad financiera se refiere al seguimiento detallado de los movimientos económicos dentro de una explotación ganadera. Esto incluye el registro de costos operativos, ingresos generados por la venta de animales o productos derivados y la administración de presupuestos anuales.

El acceso a datos financieros actualizados y organizados permite a los ganaderos tomar decisiones estratégicas fundamentadas en información precisa, lo que contribuye a una mayor rentabilidad. Implementar un sistema que permita registrar y analizar estos datos no solo mejora la planificación y el control presupuestario, sino que también contribuye a la rentabilidad del negocio ganadero. Una adecuada trazabilidad financiera reduce la incertidumbre, optimiza la asignación de recursos y ayuda a prever escenarios económicos adversos.

2.1.3 Uso de tecnologías en la nube para la gestión ganadera



Los sistemas de información en la nube han revolucionado múltiples sectores, incluida la ganadería, al permitir el almacenamiento, procesamiento y consulta remota de datos sin depender de infraestructuras físicas costosas.

El sistema de gestión de ganado basado en la nube proporciona a los usuarios herramientas para registrar, consultar y analizar datos sobre sus animales y finanzas en tiempo real. Aunque su funcionamiento requiere una conexión a internet estable, no demanda velocidades de conexión particularmente altas, lo que lo hace viable en entornos rurales con acceso limitado a redes de alta velocidad.

Entre los beneficios del uso de la nube en la gestión ganadera destacan:

- **Accesibilidad remota:** Permite la consulta y actualización de información desde cualquier ubicación.
- **Seguridad de datos:** Reduce el riesgo de pérdida de información al almacenarla en servidores protegidos.
- **Optimización de costos:** Minimiza la necesidad de infraestructura local para el almacenamiento de datos.

2.1.4 Bases de datos para la gestión ganadera

El uso de bases de datos en la gestión ganadera es fundamental para garantizar la integridad y disponibilidad de la información. En este sistema se utiliza MySQL como gestor de bases de datos, lo que permite un manejo estructurado y eficiente de los registros.

La organización de los datos en tablas facilita la consulta y actualización de la información, permitiendo el acceso rápido a detalles sobre cada animal, su historial y los costos asociados. Además, el uso de consultas SQL permite extraer datos específicos de manera rápida y eficiente, optimizando el análisis financiero y operativo.

2.2 Antecedentes tecnológicos

La transformación digital ha impactado diversos sectores, incluyendo la ganadería, donde la gestión de datos y la trazabilidad han evolucionado con la implementación de sistemas informáticos especializados. Existen diversas soluciones tecnológicas diseñadas para mejorar la administración del ganado, optimizar los procesos de producción y facilitar el seguimiento financiero de las explotaciones ganaderas.

2.2.1 Soluciones tecnológicas en la gestión ganadera

En los últimos años, han surgido múltiples plataformas y aplicaciones enfocadas en la gestión de ganado, cada una con características particulares que buscan atender las necesidades específicas de los productores. Algunas de las soluciones más relevantes incluyen:



- Agroganadero: Plataforma que permite llevar un registro detallado del inventario de ganado, control de vacunas, alimentación y reproducción. Está diseñada para pequeños y medianos productores y su almacenamiento es local, lo que limita el acceso remoto.
- BovControl: Aplicación móvil utilizada para registrar información sobre el ganado, incluyendo datos de reproducción, peso y salud. Funciona con almacenamiento en la nube, pero depende de la conectividad móvil para sincronizar datos.
- CattleMax: Sistema que permite a los ganaderos registrar información sobre sus animales y administrar la producción. Sin embargo, su enfoque está más orientado a la producción lechera y cárnica, no tanto a la trazabilidad financiera.

Si bien estas herramientas ofrecen funcionalidades útiles, presentan ciertas limitaciones como la dependencia de una conexión a internet constante, costos elevados de suscripción o una curva de aprendizaje pronunciada para los productores con poca experiencia en tecnologías digitales.

2.2.2 Estado actual de la tecnología en la gestión ganadera

La implementación de tecnologías en la gestión ganadera ha evolucionado significativamente en los últimos años. Actualmente, las soluciones más avanzadas incluyen:

- Sistemas en la nube: Facilitan el acceso remoto a la información desde cualquier dispositivo con conexión a internet. Esto mejora la trazabilidad y permite un monitoreo constante de los datos de los animales y la administración financiera.
- Internet de las Cosas (IoT): Dispositivos como collares con GPS y sensores biométricos permiten monitorear en tiempo real el estado de salud y el comportamiento del ganado. Estos datos pueden integrarse con sistemas de gestión para mejorar la toma de decisiones.
- Inteligencia Artificial y Big Data: Algoritmos de análisis predictivo ayudan a identificar patrones en la salud, reproducción y alimentación del ganado, lo que permite anticiparse a problemas y optimizar recursos.
- Aplicaciones móviles: Cada vez más soluciones permiten el registro y consulta de datos directamente desde un smartphone, facilitando la gestión sobre el terreno sin necesidad de estar frente a una computadora.

Si bien estas tecnologías han mejorado la eficiencia en la gestión ganadera, muchos sistemas existentes aún presentan barreras como costos elevados, dificultades de integración con infraestructuras locales y dependencia de conexiones a internet estables.

2.2.3 Justificación de la implementación del sistema



Ante el panorama actual, la implementación del sistema de gestión de ganado basado en la nube responde a una necesidad del sector al proporcionar:

- Almacenamiento en la nube con menor dependencia de conexiones de alta velocidad, lo que facilita su uso en zonas rurales con conectividad limitada.
- Un enfoque en la trazabilidad financiera y el control de estados del ganado, diferenciándose de otras soluciones que priorizan la productividad cárnica o lechera.
- Una interfaz sencilla y accesible desde Windows, adaptada a las necesidades de los ganaderos que requieren herramientas intuitivas y funcionales.

La implementación de este sistema representa un avance en la digitalización del sector ganadero, permitiendo una administración más eficiente de los recursos y facilitando la toma de decisiones estratégicas basadas en datos confiables.

2.3.1 Fases del desarrollo

Planificación y diseño del sistema: En esta fase se definieron la arquitectura del sistema, los requisitos funcionales y no funcionales, así como la base de datos y los módulos del sistema. Los módulos que componen el sistema fueron claramente identificados y estructurados para facilitar el desarrollo.

Desarrollo modular: El sistema fue desarrollado en módulos, lo que permitió una implementación escalonada. Cada módulo fue construido, probado e implementado de manera independiente, lo que facilitó la identificación y corrección de errores de forma aislada, sin que estos afectaran al sistema en su conjunto.

Integración: Una vez que los módulos fueron desarrollados y probados de forma independiente, se procedió a la integración de los módulos, asegurando que la comunicación entre ellos fuera fluida y que el sistema operara correctamente como un todo.

Despliegue e implementación: El sistema fue implementado en el entorno de producción, garantizando que todos los módulos estén funcionando de manera continua y sin interrupciones.

2.3.2 Tipos de pruebas

Una vez completado el desarrollo e implementación del sistema, se procederá a realizar las siguientes pruebas para garantizar la calidad, funcionalidad y facilidad de uso del sistema. Los detalles de cada tipo de prueba son los siguientes:

Pruebas de sistema: Estas pruebas se centrarán en evaluar la funcionalidad general del sistema ya integrado. Se verificará que todos los módulos trabajen conjuntamente, asegurando que el sistema sea capaz de registrar, consultar y gestionar la información sobre los animales y la trazabilidad financiera sin errores.

Pruebas de usabilidad: Se llevarán a cabo pruebas con usuarios finales (ganaderos) para evaluar la facilidad de uso del sistema. El enfoque estará en la interfaz de usuario, la navegación y la eficiencia de las tareas más comunes dentro del sistema. El objetivo es garantizar que el sistema sea intuitivo y accesible para los ganaderos, incluso si no están familiarizados con tecnologías complejas.

Pruebas de aceptación: Las pruebas de aceptación validarán si el sistema cumple con los requisitos y expectativas establecidos por los usuarios y las partes interesadas. Los ganaderos participarán en estas pruebas para confirmar que el sistema satisface sus necesidades y proporciona las funcionalidades que esperan.

Pruebas de compatibilidad: Se realizarán pruebas de compatibilidad para garantizar que el sistema funcione correctamente en computadoras con diferentes características. En particular, se probará en máquinas que tengan Windows 10 y Windows 11, debido a las diferencias que existen entre estas versiones de sistema operativo. Esto asegurará que la interfaz y las funcionalidades del sistema sean consistentes y operen sin problemas en ambas plataformas.

2.3.3 Capacitación y soporte

Si bien no se ofrecerá una capacitación formal, el sistema está diseñado para ser intuitivo y fácil de usar. Se proporcionará un manual de usuario y ayudas contextuales dentro de la interfaz del sistema, que permitirán a los ganaderos aprender a utilizar las herramientas sin dificultad.

CAPITULO 3

METODOLOGIA

El desarrollo e implementación del sistema de gestión de ganado basado en la nube se ha llevado a cabo bajo una metodología modular. Este enfoque modular permitió dividir el proyecto en componentes independientes, facilitando el manejo de tareas, y garantizando flexibilidad para realizar cambios o mejoras sin afectar la totalidad del sistema. Además, este enfoque favorece la realización de pruebas específicas para cada módulo, garantizando un enfoque más detallado y enfocado en cada parte del sistema.

Arquitectura del Sistema

El sistema se diseñó con una arquitectura modular, donde cada módulo cumple una función específica. La estructura del sistema está compuesta por:

Cliente (Interfaz Gráfica de Usuario - GUI): Aplicación de escritorio desarrollada en NetBeans.



Servidor: Base de datos MySQL alojada en un servidor remoto.

Conectividad: Interacción entre la GUI y la base de datos mediante consultas SQL.

Casos de Uso

Los casos de uso definen las acciones principales que los usuarios registrados pueden realizar en el sistema.

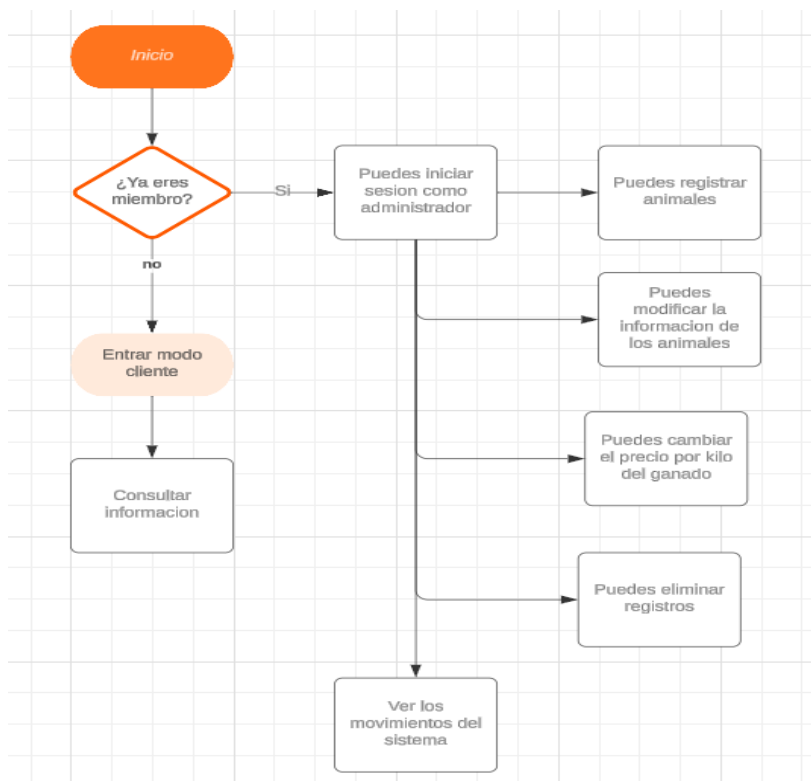


Ilustración 1 diagrama de uso

Desarrollo de la Base de Datos

La base de datos fue diseñada en MySQL y contiene las siguientes tablas principales:

registros-us: Almacena los datos de los usuarios.

ganado: Contiene la información del ganado registrado.

movimientos: Registra los eventos y cambios en el ganado.

presupuesto: Permite la asignación de presupuesto anual y el seguimiento de la trazabilidad financiera.

La estructura de la base de datos permite realizar consultas eficientes y garantizar la integridad de los datos mediante claves primarias y relaciones bien definidas.

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
☐ animales	Examinar Estructura Buscar Insertar Vaciar Eliminar	7	InnoDB	utf8mb4_general_ci	112.0 KB	-
☐ gastos	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8mb4_general_ci	16.0 KB	-
☐ historial	Examinar Estructura Buscar Insertar Vaciar Eliminar	19	InnoDB	utf8mb4_general_ci	16.0 KB	-
☐ movimientos	Examinar Estructura Buscar Insertar Vaciar Eliminar	68	InnoDB	utf8mb4_general_ci	16.0 KB	-
☐ precios	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8mb4_general_ci	16.0 KB	-
☐ presupuesto	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8mb4_general_ci	16.0 KB	-
☐ registros-us	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8mb4_general_ci	16.0 KB	-
7 tablas	Número de filas	102	InnoDB	utf8mb4_general_ci	208.0 KB	0 B

Ilustración 2 Tabla base de datos

Estructura de tabla		Vista de relaciones							
#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
☐ 1	Id	int(25)			No	Ninguna			Cambiar Eliminar Más
☐ 2	Raza	varchar(25)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 3	Peso	int(11)			No	Ninguna			Cambiar Eliminar Más
☐ 4	Sexo	varchar(25)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 5	Numero_economico	varchar(25)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 6	Salud	varchar(25)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 7	Reproductivo	varchar(25)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 8	Etapas	varchar(25)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 9	Corporal	varchar(25)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 10	Alimenticio	varchar(25)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 11	Sanitario	varchar(25)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 12	Modo_adquisicion	varchar(25)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 13	Peso_adquirido	int(11)			No	Ninguna			Cambiar Eliminar Más
☐ 14	Peso_destete	int(11)			No	Ninguna			Cambiar Eliminar Más
☐ 15	Raza_padre	varchar(25)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 16	Raza_madre	varchar(25)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 17	Fecha_adquisicion	varchar(11)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 18	Fecha_destete	varchar(11)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 19	Imagen	blob			No	Ninguna			Cambiar Eliminar Más
☐ 20	precio	varchar(25)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más

Ilustración 3 Estructura de la tabla Animales

Servidor: 127.0.0.1 » Base de datos: mi-granja » Tabla: gastos

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaci

Estructura de tabla Vista de relaciones

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
☐ 1	Anio	varchar(100)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 2	Mes	varchar(100)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 3	Concepto	varchar(100)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 4	Proveedor	varchar(100)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 5	Monto	varchar(100)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más

Ilustración 4 Estructura de la tabla Gastos

Servidor: 127.0.0.1 » Base de datos: mi-granja » Tabla: historial

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaci

Estructura de tabla Vista de relaciones

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
☐ 1	Fecha	date			No	Ninguna			Cambiar Eliminar Más
☐ 2	Movimiento	varchar(100)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 3	Id	varchar(100)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 4	Dato	varchar(100)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 5	ValorAnterior	varchar(100)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 6	ValorNuevo	varchar(100)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más

Ilustración 5 Estructura de la tabla Historial

Servidor: 127.0.0.1 » Base de datos: mi-granja » Tabla: movimientos

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Opera

Estructura de tabla Vista de relaciones

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
☐ 1	Fecha	date			No	Ninguna			Cambiar Eliminar Más
☐ 2	Hora	time			No	Ninguna			Cambiar Eliminar Más
☐ 3	Usuario	varchar(25)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 4	privilegio	varchar(25)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 5	Movimiento	varchar(25)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 6	id	varchar(25)	utf8mb4_general_ci		Sí	No aplica			Cambiar Eliminar Más

Ilustración 6 Estructura de la tabla Movimientos

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
☐ 1	Ternero	varchar(25)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 2	Joven	varchar(25)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 3	Adulto	varchar(25)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 4	Anciano	varchar(25)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más

Ilustración 7 Estructura de la tabla Precios

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
☐ 1	Presupuesto	varchar(60)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 2	Anio	varchar(60)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más

☐ Seleccionar todo
 Para los elementos que están marcados:
 Examinar
 Cambiar
 Eliminar
 Primaria
 Usuario

Ilustración 8 Estructura de la tabla Presupuesto

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
☐ 1	Usuario	varchar(25)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 2	Password	varchar(25)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 3	PasswordMaster	varchar(25)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
☐ 4	Privilegio	varchar(25)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más

Ilustración 9 Estructura de la tabla Registro de usuarios

Capítulo 4 Resultados y discusión

Interfaz de Usuario

La interfaz de usuario se desarrolló en Java con NetBeans, utilizando un diseño absoluto para mejorar la organización de los componentes. Se incluyen funcionalidades como:

- Registro y autenticación de usuarios.
- Ingreso y consulta de ganado.

- Registro de movimientos y trazabilidad.
- Asignación de presupuesto y control financiero.

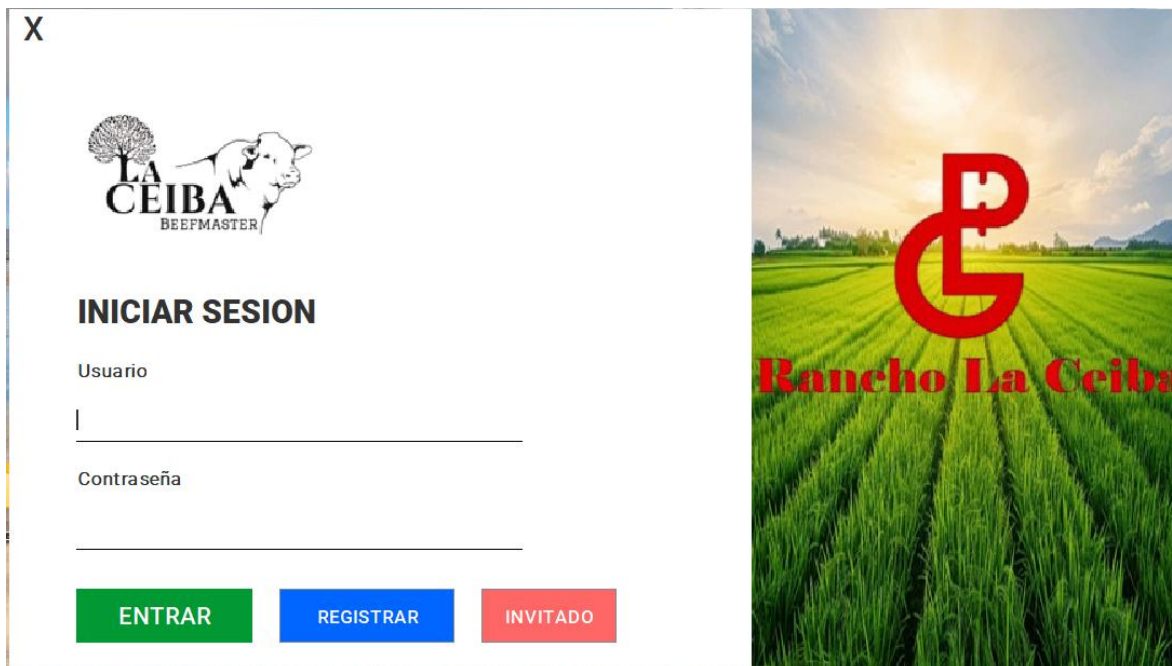


Ilustración 10 Pantalla inicio de sesión.

La pantalla de inicio de sesión es el punto de acceso al sistema y permite a los usuarios autenticarse para ingresar a la aplicación. En ella, se presentan dos campos de entrada: uno para el nombre de usuario y otro para la contraseña.

El botón "Entrar" solo entra al sistema cuando los datos ingresados coinciden con un usuario registrado en la base de datos.

Además, la pantalla ofrece dos opciones adicionales:

"Registrar": Permite a los nuevos usuarios crear una cuenta en el sistema.

"Modo Invitado": Brinda acceso limitado a la aplicación sin necesidad de autenticación.

Esta pantalla garantiza un acceso seguro y flexible para distintos tipos de usuarios.

X



Registrar Usuario

Usuario

Contraseña

Contraseña Maestra

Administrador

ENTRAR



Ilustración 11 Pantalla Registro de Usuarios

La pantalla de registro de usuario permite la creación de nuevas cuentas en el sistema. Para completar el registro, el usuario debe ingresar la siguiente información:

Usuario: Nombre con el que se identificará en el sistema.

Contraseña: Clave de acceso para la cuenta.

Contraseña Maestra: Código especial requerido para validar la creación de nuevos usuarios.

Además, existe una opción para indicar si el nuevo usuario será un administrador. Si esta opción no se selecciona, el sistema lo registrará por defecto como usuario cliente, con permisos restringidos.

X

BIENVENIDO AL SISTEMA

PORFAVOR ELIJA UNA OPCION



Ilustración 12 Vista Cliente

Esta pantalla permite a los usuarios clientes consultar la información del ganado registrado en el sistema. Para facilitar la búsqueda y visualización de los datos, se presentan tres opciones principales:

Ver ganado por orden de ID: Muestra los registros en orden secuencial según su identificador único.

Búsqueda específica: Permite localizar un animal ingresando directamente su ID.

Búsqueda por lote: Facilita la consulta de varios animales agrupados bajo un mismo lote, optimizando la gestión de múltiples registros al mismo tiempo.

X

BIENVENIDO AL SISTEMA

PORFAVOR ELIJA UNA OPCION



VER GANADO



REGISTRAR USUARIO



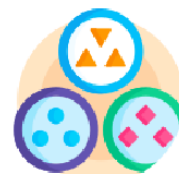
MOVIMIENTOS



PRECIOS



ADMINISTRACION



REGISTRAR GANADO

Ilustración 13 Vista administrador

Esta pantalla proporciona acceso a las funciones avanzadas del sistema, permitiendo a los administradores gestionar la información del ganado y otros aspectos clave. Se presentan seis opciones principales:

Ver ganado: Permite consultar todos los registros de animales en el sistema.

Registrar usuario: Opción para crear nuevas cuentas de usuario con los permisos correspondientes.

Movimientos en el sistema: Muestra un historial de actividades realizadas por los usuarios, como registros, modificaciones o eliminaciones de datos.

Precios: Gestión de valores relacionados con la venta del ganado.

Administración financiera: Control y seguimiento del presupuesto de la finca ganadera.

Registro de ganado: Permite añadir nuevos animales al sistema con su información correspondiente.

X

BUSQUEDA EN EL REGISTRO

Precio \$\$\$: 11000.00

Mostrar Registros por Orden



Numero de ID: 1

No. Economico Propiedad: 256

Fecha de Adquisición: 2024-11-01

Tipo o Raza: Angus

Peso: 55.0

Sexo: Hembra

Estados o Categorías

Salud: Saludable

Reproductivo: En celo

Etapas: Joven

Corporal: Óptima

Sanitario: En cuarentena

Alimenticio: En engorda

ANTERIOR

SIGUIENTE

VER MÁS

EDITAR

ELIMINAR

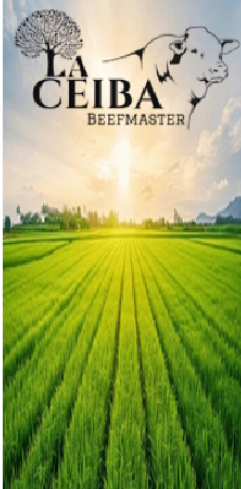



Ilustración 14 búsqueda por orden de registro

Esta pantalla permite visualizar la información del ganado de manera ordenada, mostrando los datos de un animal a la vez. La información presentada incluye:

Precio: Calculado según el precio por kilo del animal.

Foto: Imagen del animal registrado.

Número de ID: Identificador único del animal en el sistema.

Número Económico de Propiedad: Código interno asignado por el propietario.

Fecha de Adquisición: Día en que el animal fue incorporado al sistema.

Tipo o Raza: Clasificación del animal según su especie.

Peso: Peso actual del animal en kilogramos.

Sexo: Masculino o femenino.

Estado de Salud: Condición médica del animal.

Estado Reproductivo: Situación del animal respecto a la reproducción (Gestante, En celo, Vacío, Parida).

Etapas del Ganado: Fase de desarrollo del animal (cría, adulto, etc.).

Estado Corporal: Condición física del ganado.

Estado Sanitario: Situación respecto a vacunas y tratamientos.

Estado Alimenticio: Evaluación de la nutrición del animal.

ITESCAM

Además, la pantalla cuenta con cinco botones de acción:

Anterior: Muestra el registro anterior.

Siguiente: Muestra el siguiente registro.

Ver más información: Permite acceder a detalles adicionales del animal.

Editar: Opción para modificar los datos del ganado.

Eliminar: Permite eliminar el registro del animal del sistema.

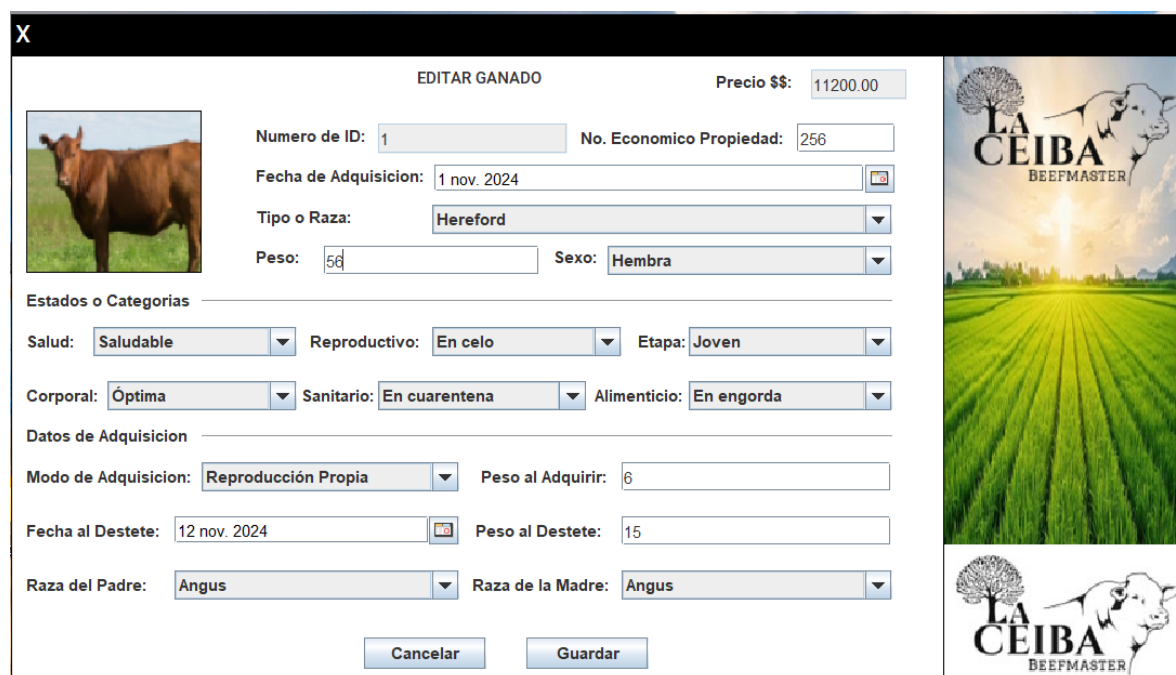


Ilustración 15 Editar Registro de ganado

Esta pantalla permite modificar la información de un animal previamente registrado en el sistema, con excepción del ID, que es un identificador único y no puede ser editado.

El precio del ganado se calcula automáticamente en función del precio por kilo según la etapa del ganado y su peso, por lo que no es un campo editable.

Los datos que se pueden actualizar incluyen:

Foto: Imagen del animal.

Número Económico: Código interno del animal dentro de la propiedad.

Fecha de Adquisición: Día en que el animal fue incorporado al sistema.

Tipo o Raza: Clasificación del animal según su especie.

Peso: Peso actual del animal en kilogramos.

Sexo: Masculino o femenino.

Estado de Salud: Condición médica general.

Estado Reproductivo: Situación reproductiva del animal (Gestante, En celo, Vacío, Parida).

Etapas: Fase de desarrollo del ganado.

Estado Corporal: Condición física general.

Estado Sanitario: Estado respecto a vacunas y tratamientos médicos.

Estado Alimenticio: Evaluación de la nutrición del animal.

Modo de Adquisición: Forma en la que el animal fue adquirido (compra, cría interna, etc.).

Peso al Adquirir: Peso del animal en el momento de su compra o incorporación.

Fecha al Destete: Día en que el animal fue separado de su madre.

Peso al Destete: Peso registrado al momento del destete.

Raza del Padre: Información sobre la genética paterna.

Raza de la Madre: Información sobre la genética materna.

La pantalla cuenta con dos botones de acción:

Cancelar: Descartar los cambios y cerrar la ventana sin guardar.

Guardar: Confirmar y actualizar la información del animal en el sistema.



Peso al Adquirir

Fecha al Destete

Peso al Destete

Raza del Padre

Raza de la Madre

La única acción disponible en esta pantalla es el botón de Siguiente, que permite al usuario avanzar a la información detallada de Trazabilidad del Ganado, donde se podrán ver los eventos y registros históricos asociados con ese animal.

X

BUSQUEDA EN EL REGISTRO

Id Animal

10

Fecha	Movimiento	Dato	Valor Anterior	Valor Nuevo
2024-12-05	Creacion	raza	----	Limousin
2024-12-05	Creacion	peso	----	60
2024-12-05	Creacion	sexo	----	Hembra
2024-12-05	Creacion	numeroEconomico	----	595-54
2024-12-05	Creacion	salud	----	Saludable
2024-12-05	Creacion	reproductivo	----	Gestante
2024-12-05	Creacion	etapa	----	Adulto
2024-12-05	Creacion	corporal	----	Óptima
2024-12-05	Creacion	alimenticio	----	Bien alimentado
2024-12-05	Creacion	sanitario	----	Vacunado
2024-12-05	Creacion	modoAdquisicion	----	Reproducción Propia
2024-12-05	Creacion	pesoAdquirido	----	10
2024-12-05	Creacion	pesoDestete	----	26
2024-12-05	Creacion	razaPadre	----	Limousin
2024-12-05	Creacion	razaMadre	----	Charolais
2024-12-05	Creacion	fechaAdquisicion	----	2024-12-01
2024-12-05	Creacion	fechaDestete	----	2024-12-04
2024-12-05	Modificacion	peso	60	62



Ilustración 17 Trazabilidad del ganado

Esta pantalla muestra la información de trazabilidad del ganado seleccionado, comenzando con su ID y permitiendo al usuario elegir el dato específico que desea filtrar.

La pantalla incluye:

ID del animal: Identificador único del ganado.

Campo de elección: Un filtro que permite seleccionar el tipo de dato que se desea consultar.

A continuación, se muestra una tabla con los siguientes campos:

Fecha: La fecha en la que se registró el movimiento o modificación del ganado.

Movimiento: El tipo de acción realizada (creación, modificación, etc.).

Dato del Ganado: El dato específico que fue modificado (por ejemplo, peso, estado de salud, etc.).

Valor Anterior: El valor antes de que se realizara la modificación.

Valor Nuevo: El valor posterior a la modificación.

Esta pantalla permite a los usuarios visualizar el historial de cambios y movimientos de un animal, lo que facilita el seguimiento de su trazabilidad a lo largo del tiempo.

X

BUSQUEDA EN EL REGISTRO

Id Animal

10

peso

Fecha	Movimiento	Dato	Valor Anterior	Valor Nuevo
2024-12-05	Creacion	peso	----	60
2024-12-05	Modificacion	peso	60	62



Ilustración 18 Trazabilidad del ganado por filtros

Aquí se muestra un ejemplo de los datos filtrados de ese animal.

 MIS MOVIMIENTOS 					
FECHA	HORA	USUARIO	PRIVILEGIO	MOVIMIENTO	ID DEL GANADO
2024-12-10	07:46:42	WCHO	Administrador	Actualizacion	10
2024-12-10	07:46:15	WCHO	Administrador	Actualizacion	3
2024-12-10	07:45:46	WCHO	Administrador	Actualizacion	1
2024-12-10	07:45:23	WCHO	Administrador	Inicio de sesión	No aplica

Ilustración 19 Registro de los movimientos en el sistema

Esta pantalla muestra el historial de movimientos en el sistema relacionados con el ganado, brindando información sobre las acciones realizadas en un periodo determinado.

La pantalla incluye una tabla con los siguientes campos:

Fecha: La fecha en que se registró el movimiento.

Hora: El momento exacto en que se llevó a cabo el movimiento.

Usuario: Nombre del usuario que realizó el movimiento.

Privilegio: El nivel de acceso o rol del usuario que realizó el movimiento (por ejemplo, administrador, cliente, etc.).

Movimiento Realizado: Descripción del tipo de acción realizada (por ejemplo, creación, modificación, eliminación).

ID del Ganado: El identificador único del ganado involucrado en el movimiento (si aplica).

Esta pantalla permite a los usuarios seguir de cerca las acciones realizadas dentro del sistema, proporcionando transparencia y trazabilidad de las actividades relacionadas con el ganado.

X

Precios segun la etapa

Ternero o becerro	65.0
Joven o recria	200.0
Adulto	300.0
Anciano	250.0

Guardar



Ilustración 20 Administrador de los precios por kilo

Esta pantalla permite al usuario ingresar los precios por kilo para cada una de las 4 etapas del ganado.

Cada campo corresponde a una etapa específica del ganado, permitiendo al usuario asignar un precio por kilo para cada fase del desarrollo del animal.

La pantalla cuenta con un botón de Guardar que permite guardar los precios ingresados para las distintas etapas.

Administración de Gastos

Presupuesto Asignado: 50000.0 Restante: 40000.00 Año: 2025

MES Concepto Proveedor Monto Añadir

AÑO	MES	CONCEPTO	PROVEEDOR	MONTO
2025	ENERO	alimentacion	GBC	10000.0

Gasto Total: 10000.00 Asignar Presupuesto Terminar Presupuesto

LA CEIBA BEEFMASTER

Ilustración 21 Administrador de presupuestos

Esta pantalla permite al administrador gestionar el presupuesto asignado para el ganado, registrando gastos y controlando el presupuesto restante. La pantalla incluye los siguientes elementos:

Presupuesto Asignado: Campo donde se ingresa el presupuesto inicial asignado para el año.

Año: Campo para seleccionar o ingresar el año correspondiente al presupuesto.

Una vez que el Presupuesto Asignado y el Año son ingresados, se habilita el botón "Asignar Presupuesto", el cual registra estos datos en el sistema.

El campo "Restante" es no modificable y muestra el resultado de restar el total de los gastos del presupuesto asignado.

Mes: Un campo desplegable que permite seleccionar el mes correspondiente en el que se está registrando un gasto.

Concepto: Campo donde se describe el concepto del gasto (por ejemplo, alimentación, medicamentos, etc.).

Proveedor: Campo donde se ingresa el nombre del proveedor del servicio o producto relacionado con el gasto.

Monto: Campo para ingresar el monto del gasto.

Al pulsar el botón "Añadir", el gasto se registra en el sistema y se muestra en una tabla debajo con la siguiente información:

Año: El año en que se registró el gasto.

Mes: El mes correspondiente al gasto.

Concepto: Descripción del gasto.

Proveedor: Nombre del proveedor del gasto.

Monto: Monto registrado para ese gasto.

En la parte inferior, se muestra el total acumulado de todos los montos de los gastos registrados, permitiendo al usuario tener un control visual de los gastos totales.

Botón "Terminar Presupuesto": Este botón permite limpiar el presupuesto asignado y los gastos registrados, para poder comenzar a registrar un nuevo presupuesto.

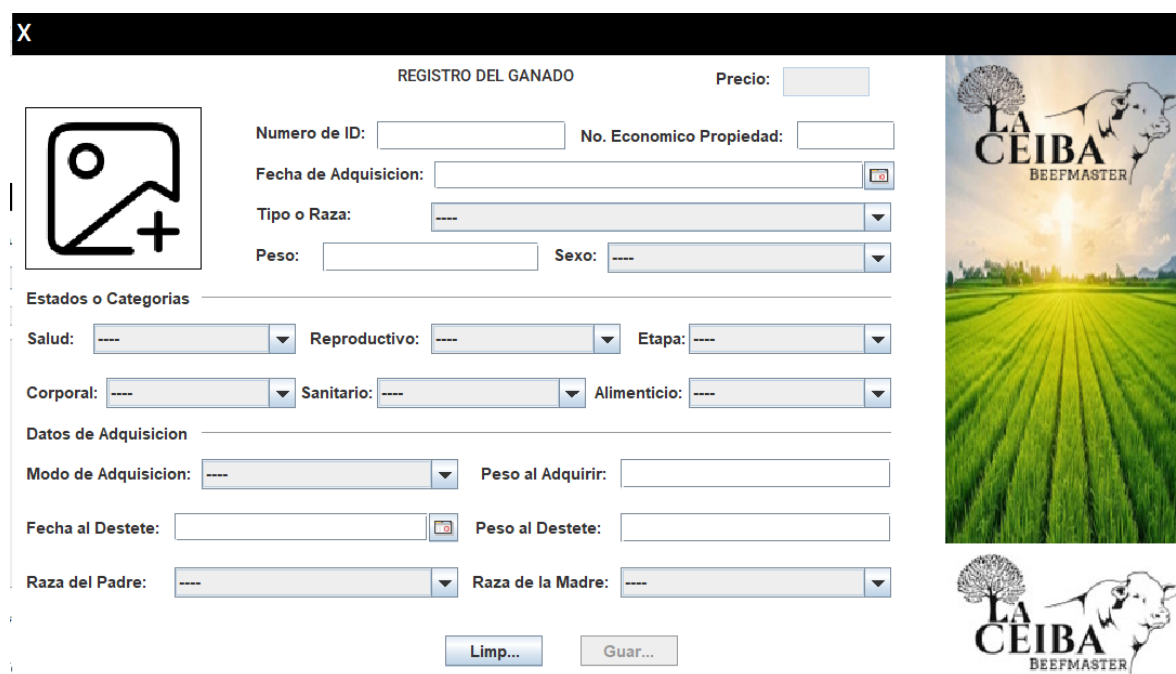


Ilustración 22 Registro del ganado

Esta pantalla permite al usuario registrar un nuevo animal en el sistema. Los campos solicitados son los siguientes:

ID: Identificador único del ganado, asignado automáticamente por el sistema.

Precio: El precio del ganado no se ingresa, se calcula automáticamente según el precio por kilo según la etapa y peso del ganado.

Foto: Cargar la imagen del ganado.

Número Económico: Código interno del animal dentro de la propiedad.

Fecha de Adquisición: Fecha en que el animal fue adquirido.

Tipo o Raza: Clasificación del animal según su especie.

Peso: Peso del animal en kilogramos.

Sexo: Masculino o femenino.

Estado de Salud: Condición médica del animal.

Estado Reproductivo: Situación reproductiva del animal (Gestante, En celo, Vacío, Parida).

Etapas: Fase de desarrollo del ganado.

Estado Corporal: Condición física general del animal.

Estado Sanitario: Estado respecto a vacunas y tratamientos médicos.

Estado Alimenticio: Evaluación de la nutrición del animal.

Modo de Adquisición: Forma en que el animal fue adquirido (compra, cría interna, etc.).

Peso al Adquirir: Peso del animal al momento de su adquisición.

Fecha al Destete: Fecha en que el animal fue separado de su madre.

Peso al Destete: Peso registrado al momento del destete.

Raza del Padre: Información sobre la genética paterna.

Raza de la Madre: Información sobre la genética materna.

La pantalla incluye los siguientes botones:

Limpiar: Borra los datos ingresados en los campos de la pantalla, dejando todo en blanco para un nuevo registro.

Guardar: Guarda el registro del ganado en el sistema. Este botón solo se habilita cuando todos los campos están completos, asegurando que la información esté completa antes de ser guardada.



X

Trazabilidad Financiera de la finca

Precio ganado:

Gastos:

Ganancias:

Año	Gastos	Ganancias
2026	1000.0	0.0
2025	2000.0	27610.0





Ilustración 23 Trazabilidad financiera

Esta pantalla proporciona al administrador una visión general del estado financiero de la finca, mostrando los costos, los gastos y las ganancias relacionadas con el ganado. Los elementos que componen esta pantalla son los siguientes:

Costo Total de los Ganados: Un campo que muestra la suma total del costo de todos los ganados registrados en la finca. Este valor es no editable y se calcula automáticamente según los datos del sistema.

Gastos Totales: Un campo que muestra la suma total de los gastos registrados para la finca, como alimentación, medicamentos, etc. Este valor es no editable y se calcula automáticamente a partir de los registros de gastos.

Ganancias: Un campo que muestra las ganancias netas de la finca, calculadas automáticamente como la diferencia entre el costo total de los ganados y los gastos totales. Este valor es no editable.

Debajo de estos campos, se muestra una tabla que presenta los siguientes datos:

Año: El año correspondiente a los registros financieros.

Gastos Totales de ese Año: Los gastos acumulados de ese año, calculados a partir de los registros de gastos.

Ganancias de ese Año: Las ganancias de ese año, calculadas como la diferencia entre los ingresos generados por los ganados y los gastos totales de ese año.

Pruebas

PRUEBAS DE SISTEMA

Objetivo: Comprobar que el sistema pueda guardar correctamente la información del ganado en la base de datos.

Caso de prueba: Intentar registrar un nuevo animal en el sistema y verificar si la información se guarda correctamente.

Resultado: Ocurrió un error porque el identificador del animal (ID) incluía letras y números, pero la base de datos solo permitía números. Como consecuencia, el sistema no pudo guardar el registro.

Conclusión: La prueba no fue exitosa. Para solucionarlo, se modificó la base de datos para que acepte identificadores con letras y números, permitiendo así el registro sin problemas; En un segundo intento ya con la corrección la prueba fue exitosa.

Objetivo: Evaluar la funcionalidad del módulo de búsqueda en el sistema.

Caso de prueba: Realizar búsquedas por ID, raza y estado reproductivo para comprobar que los resultados sean correctos.

Resultado: Los datos se muestran de manera precisa y sin retrasos.

Conclusión: La prueba fue exitosa.

Objetivo: Comprobar que los movimientos realizados en la granja se registran correctamente en la base de datos.

Caso de prueba: Registrar diferentes movimientos de animales (peso, estado, reproductivo, etc.) y verificar si se guardan en la tabla correspondiente.

Resultado: Cada movimiento quedó registrado correctamente con la fecha, hora, usuario y tipo de movimiento.

Conclusión: La prueba fue exitosa.

PRUEBAS DE USABILIDAD

Objetivo: Evaluar la facilidad de uso de la interfaz para un usuario sin experiencia.



Caso de prueba: Pedir a un usuario nuevo que registre un animal sin instrucciones previas.

Resultado: El usuario logró completar el proceso sin dificultades en menos de 4 minutos.

Conclusión: La interfaz es intuitiva y fácil de usar.

Objetivo: Verificar la accesibilidad y visibilidad de los botones de acción en la tabla de registros.

Caso de prueba: Pedir a un usuario que edite y elimine un registro usando los botones de acción.

Resultado: El usuario encontró y utilizó los botones sin problemas.

Conclusión: Los botones son accesibles y cumplen con su función.

Objetivo: Medir la velocidad con la que un usuario puede encontrar un registro específico.

Caso de prueba: Pedir a un usuario que busque un animal por su número de identificación.

Resultado: El usuario encontró el registro en menos de 10 segundos.

Conclusión: La búsqueda es eficiente y rápida.

PRUEBAS DE ACEPTACIÓN

Objetivo: Confirmar que el sistema satisface las necesidades de los usuarios ganaderos.

Caso de prueba: Pedir a un ganadero que utilice la plataforma y registre su opinión sobre su utilidad.

Resultado: El usuario encontró el sistema útil y adecuado para su actividad.

Conclusión: La prueba fue exitosa y el sistema es aceptado.

Objetivo: Verificar que el usuario cliente solo pueda visualizar los datos sin modificarlos.

Caso de prueba: Intentar modificar un registro desde la cuenta de un usuario cliente.



Resultado: El usuario cliente no pudo modificar los datos, solo visualizar la información.

Conclusión: La restricción de permisos funciona correctamente.

Objetivo: Confirmar que los usuarios pueden administrar el ganado sin errores técnicos.

Caso de prueba: Registrar, editar y eliminar un animal en la plataforma.

Resultado: Todas las operaciones se realizaron correctamente sin errores.

Conclusión: La prueba fue exitosa.

PRUEBAS DE COMPATIBILIDAD

Objetivo: Comprobar la ejecución del sistema en diferentes versiones de Windows.

Caso de prueba: Instalar y ejecutar la aplicación en Windows 10 y Windows 11.

Resultado: El sistema funcionó sin errores en ambas versiones.

Conclusión: La prueba fue exitosa.

Objetivo: Probar la conexión del sistema con la base de datos en distintas redes de internet.

Caso de prueba: Conectar el sistema a través de Wi-Fi, datos móviles y red cableada.

Resultado: La conexión se mantuvo estable en todos los casos.

Conclusión: La prueba fue exitosa.

Capítulo 5 Conclusiones y recomendaciones.

Conclusiones

La implementación del Sistema de gestión de ganado basado en la nube ha demostrado ser un enfoque eficaz para mejorar la trazabilidad financiera en la gestión ganadera. Durante las pruebas, se verificó que el sistema es capaz de almacenar correctamente la información de los animales, realizar búsquedas precisas y registrar los movimientos de la granja de manera confiable.

La hipótesis planteada, que sugiere que la implementación de este sistema mejora la trazabilidad financiera mediante un control preciso de los presupuestos anuales y el seguimiento de los precios del ganado, se cumplió. Los resultados obtenidos mostraron que el sistema permite una gestión financiera más eficiente, contribuyendo a la optimización de los costos operativos y mejorando la rentabilidad a largo plazo.

En relación con la pregunta de investigación, el sistema responde adecuadamente, ya que facilita la gestión de los flujos financieros relacionados con la ganadería, permitiendo una mayor transparencia y un control más eficaz sobre las decisiones económicas.

En resumen, el sistema no solo cumple con los objetivos propuestos, sino que también sienta las bases para una evolución continua que podría enriquecer aún más la gestión ganadera en el futuro.

Recomendaciones

Mejoras continuas en el sistema: Este proyecto tiene un gran potencial para evolucionar. Se recomienda seguir desarrollando el sistema con nuevas funcionalidades y mejoras, asegurando que siempre esté alineado con las necesidades cambiantes del sector ganadero.

Desarrollo de una aplicación móvil complementaria: Una futura ampliación del sistema podría ser la creación de una aplicación móvil que permita a los usuarios gestionar el ganado y controlar las finanzas desde cualquier lugar, proporcionando acceso rápido y eficiente a la plataforma en dispositivos móviles.

Capacitación y soporte al usuario: Para maximizar el impacto del sistema, es crucial proporcionar capacitación constante a los usuarios, así como un servicio de soporte técnico para resolver posibles inconvenientes.

Implementación de nuevas funciones financieras: Se recomienda agregar herramientas más avanzadas de análisis y reporte financiero, como proyecciones de ingresos, análisis de rentabilidad y seguimiento de costos en tiempo real.

Escalabilidad y adaptación a diferentes necesidades: A medida que crezca el número de usuarios y registros, es necesario realizar un análisis de la escalabilidad del sistema, asegurando que pueda manejar un mayor volumen de datos sin comprometer el rendimiento.

Referencias Bibliográficas.

Club Ganadero. (2024, febrero 14). *Tips para mejorar la gestión financiera de la actividad ganadera*. Club ganadero. <https://www.clubganadero.com/actividad-ganadera/>

Codegalac, X. (s/f). *COMISION DE DESARROLLO GANADERO PARA AMÉRICA LATINA Y EL CARIBE*. Fao.org. Recuperado el 1 de marzo de 2025, de <https://openknowledge.fao.org/server/api/core/bitstreams/ea1d222b-cd83-4012-8298-128bb7e1bde6/content>

de Castilla y León, J. (s/f). *Trazabilidad e Higiene Ganadera*. Jcyl.es. Recuperado el 1 de marzo de 2025, de <https://agriculturaganaderia.jcyl.es/web/es/ganaderia/trazabilidad-higiene-ganadera.html>

La tecnología en la nube facilita el pesaje del ganado. (s/f). Gallagher.com. Recuperado el 1 de marzo de 2025, de <https://am.gallagher.com/es-AR/Solutions/Case-Study-Listings/cloud-technology-for-easier-livestock-weighing>

Sutton, N. (2023, junio 13). *Traceability in finance: Why is it important?* OneAdvanced; Advanced. <https://www.oneadvanced.com/news-and-opinion/traceability-in-finance-why-is-it-important/>

Trazabilidad en la Ganadería de América Latina. (s/f). Intagri.com. Recuperado el 1 de marzo de 2025, de <https://www.intagri.com/articulos/ganaderia/trazabilidad-en-la-ganaderia-de-america-latina>

Trazabilidad financiera. (s/f). Rojjer Consultores. Recuperado el 1 de marzo de 2025, de <https://www.rojjerconsultores.com.mx/trazabilidad-financiera>

Vaquitec. (s/f). Agritecsoft.com. Recuperado el 1 de marzo de 2025, de <https://www.agritecsoft.com/vaquitec/es/>

Klerkx, L., Jakku, E., & Labarthe, P. (2019). Digital agriculture and the agriculture innovation system. *Agricultural Systems*, 171, 1-6.

Sørensen, C. G., Fountas, S., Nash, E., Pesonen, L., Bochtis, D., Pedersen, S. M., & Blackmore, B. S. (2019). Concepts, perspectives and the value chain in precision agriculture. *European Journal of Agronomy*, 92, 1-16.

Wolfert, S., Verdouw, C., & Bogaardt, M. J. (2017). Big data in smart farming – A review. *Agricultural Systems*, 153, 69-80.

Anexos

A continuación, se presenta el código completo del sistema, el cual ha sido desarrollado para gestionar de manera eficiente las operaciones y registros relacionados con el ganado. Este código incluye funcionalidades clave como el registro y edición de datos, control de movimientos, administración de gastos y visualización de información detallada.

El sistema ha sido diseñado con un enfoque modular, asegurando que cada componente cumpla con su función específica y permita una interacción fluida entre las diferentes secciones. Además, se han implementado controles de validación y seguridad para garantizar la integridad de los datos.

BusquedaAnimal3.java

```
package Busqueda;

import com.VerMasClient.VerMasClient;
import com.admin.*;
import com.conexion.Conexion;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.ImageIcon;

/**@author wicho */
public class BuscarAnimal3 extends javax.swing.JPanel {

    String id = null;

    public BuscarAnimal3() {
```



```

initComponents();
ocultarComponentes(); // Asegura que todos los componentes empiecen ocultos
mostrarBusquedaPorID();
jTextField14.setText("");
jTextField14.setEditable(false);

}

private void dispose() {
    throw new UnsupportedOperationException("Not supported yet.");
}

private void mostrarBusquedaPorID() {
    jButton1.setVisible(true);
    jTextField1.setVisible(true);

    // Obtener el texto del campo de ID y asegurarse de que no esté vacío
    String idText = jTextField1.getText().trim();
    if (idText.isEmpty()) {
        System.out.println("Por favor, ingresa un ID.");
        return;
    }

    int id;
    try {
        // Intentar convertir el texto a un número entero
        id = Integer.parseInt(idText);
    } catch (NumberFormatException e) {
        System.out.println("ID no válido. Asegúrate de ingresar un número.");
        return;
    }

    // Conectar a la base de datos y realizar la consulta
    try (Connection conn = Conexion.conectar()) {
        String query = "SELECT id, Numero_economico, Fecha_adquisicion, raza, peso, sexo, salud,
reproductivo, etapa, corporal, sanitario, alimenticio, imagen, precio FROM animales WHERE id = ?";

        try (PreparedStatement stmt = conn.prepareStatement(query)) {

```



```

stmt.setInt(1, id);
try (ResultSet rs = stmt.executeQuery()) {
    if (rs.next()) {
        // Crear un objeto Animal con los datos del registro encontrado
        Animal animal = new Animal(
            rs.getInt("id"),
            rs.getString("Numero_economico"),
            rs.getDate("Fecha_adquisicion"),
            rs.getString("raza"),
            rs.getFloat("peso"),
            rs.getString("sexo"),
            rs.getString("salud"),
            rs.getString("reproductivo"),
            rs.getString("etapa"),
            rs.getString("corporal"),
            rs.getString("sanitario"),
            rs.getString("alimenticio"),
            rs.getBytes("imagen"), // Obtener la imagen como byte[]
            rs.getString("precio")
        );
        mostrarAnimal(animal); // Mostrar los datos en la interfaz

        // Bloquear todos los campos de entrada donde se muestran los datos
        jTextField2.setEditable(false);
        jTextField3.setEditable(false);
        jTextField4.setEditable(false);
        jTextField5.setEditable(false);
        jTextField6.setEditable(false);
        jTextField7.setEditable(false);
        jTextField8.setEditable(false);
        jTextField9.setEditable(false);
        jTextField10.setEditable(false);
        jTextField11.setEditable(false);
        jTextField12.setEditable(false);
        jTextField13.setEditable(false);
        jButton3.setVisible(true);

    } else {

```



```

        System.out.println("No se encontró ningún registro con el ID especificado.");
        // Valores predeterminados
        jTextField1.setText("");
        jComboBox1.setSelectedItem("----");
        jTextField3.setText("");
        jTextField7.setText("");
        jTextField8.setText("");
        jTextField9.setText("");
        jTextField10.setText("----");
        jTextField11.setText("");
        jTextField13.setText("");
        jTextField12.setText("");
        jTextField6.setText("");
        jTextField4.setText("");
        jTextField2.setText("");
        jTextField5.setText("");
        imagen.setIcon(null);

        // Bloquear los campos
        jTextField2.setEditable(false);
        jTextField3.setEditable(false);
        jTextField4.setEditable(false);
        jTextField5.setEditable(false);

    }
}
} catch (SQLException e) {
    e.printStackTrace();
}
}

private void mostrarAnimal(Animal animal) {
    imagen.setVisible(true);
    // Convertir byte[] a Image y establecer en JLabel
    if (animal.getImagen() != null) {
        ImageIcon imagenIcon = new ImageIcon(animal.getImagen());
        imagen.setIcon(imagenIcon);
    }
}

```



```
} else {  
    imagen.setIcon(null);  
}  
  
Idtxt.setVisible(true);  
jTextField2.setVisible(true);  
jTextField2.setText(String.valueOf(animal.getId()));  
id = jTextField2.getText();  
  
NoTxt.setVisible(true);  
jTextField4.setVisible(true);  
jTextField4.setText(animal.getNumeroEconomico());  
  
FechaAdqTxt.setVisible(true);  
jTextField5.setVisible(true);  
jTextField5.setText(animal.getFechaAdquisicion().toString());  
  
RazaTxt.setVisible(true);  
jTextField6.setVisible(true);  
jTextField6.setText(animal.getRaza());  
  
PesoTxt.setVisible(true);  
jTextField3.setVisible(true);  
jTextField3.setText(String.valueOf(animal.getPeso()));  
  
SexoTxt.setVisible(true);  
jTextField7.setVisible(true);  
jTextField7.setText(animal.getSexo());  
  
jLabel8.setVisible(true);  
jSeparator1.setVisible(true);  
  
SaludTxt.setVisible(true);  
jTextField8.setVisible(true);  
jTextField8.setText(animal.getSalud());  
  
ReproductivoTxt.setVisible(true);  
jTextField9.setVisible(true);
```



```
    jTextField9.setText(animal.getReproductivo());

    EtapaTxt.setVisible(true);
    jTextField10.setVisible(true);
    jTextField10.setText(animal.getEtapa());

    CorporalTxt.setVisible(true);
    jTextField11.setVisible(true);
    jTextField11.setText(animal.getCorporal());

    SanitarioTxt.setVisible(true);
    jTextField12.setVisible(true);
    jTextField12.setText(animal.getSanitario());

    AlimenticioTxt.setVisible(true);
    jTextField13.setVisible(true);
    jTextField13.setText(animal.getAlimenticio());

    jLabel8.setVisible(true);
    jSeparator1.setVisible(true);
    jTextField14.setText(animal.getPrecio());

}
```

```
private void ocultarComponentesExclusivos() {
    imagen.setVisible(false);
    ldtxt.setVisible(false);
    jTextField2.setVisible(false);
    NoTxt.setVisible(false);
    jTextField4.setVisible(false);
    FechaAdqTxt.setVisible(false);
    jTextField5.setVisible(false);
    RazaTxt.setVisible(false);
    jTextField6.setVisible(false);
    PesoTxt.setVisible(false);
    jTextField3.setVisible(false);
    SexoTxt.setVisible(false);
    jTextField7.setVisible(false);
```



```

jLabel8.setVisible(false);
jSeparator1.setVisible(false);
SaludTxt.setVisible(false);
jTextField8.setVisible(false);
ReproductivoTxt.setVisible(false);
jTextField9.setVisible(false);
EtapasTxt.setVisible(false);
jTextField10.setVisible(false);
CorporalTxt.setVisible(false);
jTextField11.setVisible(false);
SanitarioTxt.setVisible(false);
jTextField12.setVisible(false);
AlimenticioTxt.setVisible(false);
jTextField13.setVisible(false);
jButton1.setVisible(false);
jButton3.setVisible(false);
}

```

```

private void ocultarComponentes() {
    ocultarComponentesExclusivos();
    jTextField1.setVisible(true);
}

```

```

/**

```

```

 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */

```

```

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

```

```

    jLabel2 = new javax.swing.JLabel();
    jComboBox1 = new javax.swing.JComboBox<>();
    jTextField1 = new javax.swing.JTextField();
    imagen = new javax.swing.JLabel();
    Idtxt = new javax.swing.JLabel();
    FechaAdqTxt = new javax.swing.JLabel();

```



```

RazaTxt = new javax.swing.JLabel();
PesoTxt = new javax.swing.JLabel();
jTextField2 = new javax.swing.JTextField();
jTextField3 = new javax.swing.JTextField();
SexoTxt = new javax.swing.JLabel();
NoTxt = new javax.swing.JLabel();
jTextField4 = new javax.swing.JTextField();
jSeparator1 = new javax.swing.JSeparator();
jLabel8 = new javax.swing.JLabel();
SaludTxt = new javax.swing.JLabel();
ReproductivoTxt = new javax.swing.JLabel();
EtapaTxt = new javax.swing.JLabel();
CorporalTxt = new javax.swing.JLabel();
AlimenticioTxt = new javax.swing.JLabel();
SanitarioTxt = new javax.swing.JLabel();
jTextField6 = new javax.swing.JTextField();
jTextField5 = new javax.swing.JTextField();
jTextField7 = new javax.swing.JTextField();
jTextField8 = new javax.swing.JTextField();
jTextField9 = new javax.swing.JTextField();
jTextField10 = new javax.swing.JTextField();
jTextField11 = new javax.swing.JTextField();
jTextField12 = new javax.swing.JTextField();
jTextField13 = new javax.swing.JTextField();
jButton3 = new javax.swing.JButton();
jButton1 = new javax.swing.JButton();
jTextField14 = new javax.swing.JTextField();
jLabel1 = new javax.swing.JLabel();

setBackground(new java.awt.Color(255, 255, 255));
setLayout(null);

jLabel2.setFont(new java.awt.Font("Roboto Medium", 0, 12)); // NOI18N
jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel2.setText("BUSQUEDA EN EL REGISTRO POR ID");
add(jLabel2);
jLabel2.setBounds(0, 0, 690, 30);

```



```

jComboBox1.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Busqueda
por ID" }));
add(jComboBox1);
jComboBox1.setBounds(20, 50, 200, 22);
add(jTextField1);
jTextField1.setBounds(240, 50, 220, 22);

```

```

imagen.setIcon(new javax.swing.ImageIcon(getClass().getResource("/iconos/añadir.png"))); //
NOI18N

```

```

imagen.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
imagen.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        imagenMouseClicked(evt);
    }
});
add(imagen);
imagen.setBounds(20, 120, 130, 120);

```

```

Idtxt.setText("Numero de ID:");
add(Idtxt);
Idtxt.setBounds(190, 110, 90, 16);

```

```

FechaAdqTxt.setText("Fecha de Adquisicion:");
add(FechaAdqTxt);
FechaAdqTxt.setBounds(190, 150, 130, 16);

```

```

RazaTxt.setText("Tipo o Raza:");
add(RazaTxt);
RazaTxt.setBounds(190, 190, 80, 16);

```

```

PesoTxt.setText("Peso:");
add(PesoTxt);
PesoTxt.setBounds(190, 230, 50, 16);
add(jTextField2);
jTextField2.setBounds(280, 110, 140, 22);
add(jTextField3);
jTextField3.setBounds(240, 230, 160, 22);

```



```
SexoTxt.setText("Sexo:");  
add(SexoTxt);  
SexoTxt.setBounds(410, 230, 50, 16);
```

```
NoTxt.setText("No. Economico Propiedad:");  
add(NoTxt);  
NoTxt.setBounds(430, 110, 160, 20);  
add(jTextField4);  
jTextField4.setBounds(590, 110, 73, 22);
```

```
jSeparator1.setForeground(new java.awt.Color(153, 153, 153));  
add(jSeparator1);  
jSeparator1.setBounds(150, 270, 510, 10);
```

```
jLabel8.setText("Estados o Categorías");  
add(jLabel8);  
jLabel8.setBounds(20, 260, 130, 20);
```

```
SaludTxt.setText("Salud:");  
add(SaludTxt);  
SaludTxt.setBounds(20, 300, 60, 20);
```

```
ReproductivoTxt.setText("Reproductivo:");  
add(ReproductivoTxt);  
ReproductivoTxt.setBounds(230, 300, 90, 20);
```

```
EtapasTxt.setText("Etapas:");  
add(EtapasTxt);  
EtapasTxt.setBounds(470, 300, 50, 20);
```

```
CorporalTxt.setText("Corporal:");  
add(CorporalTxt);  
CorporalTxt.setBounds(20, 350, 70, 20);
```

```
AlimenticioTxt.setText("Alimenticio:");  
add(AlimenticioTxt);  
AlimenticioTxt.setBounds(440, 350, 80, 20);
```



```

SanitarioTxt.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
SanitarioTxt.setText("Sanitario:");
add(SanitarioTxt);
SanitarioTxt.setBounds(220, 350, 60, 20);
add(jTextField6);
jTextField6.setBounds(320, 190, 340, 22);
add(jTextField5);
jTextField5.setBounds(320, 150, 340, 22);
add(jTextField7);
jTextField7.setBounds(450, 230, 210, 22);
add(jTextField8);
jTextField8.setBounds(70, 300, 150, 22);
add(jTextField9);
jTextField9.setBounds(320, 300, 140, 22);
add(jTextField10);
jTextField10.setBounds(510, 300, 150, 22);
add(jTextField11);
jTextField11.setBounds(80, 350, 140, 22);
add(jTextField12);
jTextField12.setBounds(280, 350, 160, 22);
add(jTextField13);
jTextField13.setBounds(510, 350, 150, 22);


jButton3.setText("VER MÁS");
jButton3.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jButton3MouseClicked(evt);
    }
});
add(jButton3);
jButton3.setBounds(270, 420, 110, 23);


jButton1.setText("BUSCAR");
jButton1.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jButton1MouseClicked(evt);
    }
});

```



```

add(jButton1);
jButton1.setBounds(480, 50, 90, 23);
add(jTextField14);
jTextField14.setBounds(590, 20, 64, 22);

jLabel1.setText("Precio $$$:");
add(jLabel1);
jLabel1.setBounds(510, 20, 70, 20);
} // </editor-fold>

private void imagenMouseClicked(java.awt.event.MouseEvent evt) {

}

private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
    mostrarBusquedaPorID();
}

private void jButton3MouseClicked(java.awt.event.MouseEvent evt) {
    VerMasClient verWindow = new VerMasClient(id);
    verWindow.setVisible(true);
}

// Variables declaration - do not modify
private javax.swing.JLabel AlimenticioTxt;
private javax.swing.JLabel CorporalTxt;
private javax.swing.JLabel EtapaTxt;
private javax.swing.JLabel FechaAdqTxt;
private javax.swing.JLabel Idtxt;
private javax.swing.JLabel NoTxt;
private javax.swing.JLabel PesoTxt;
private javax.swing.JLabel RazaTxt;
private javax.swing.JLabel ReproductivoTxt;
private javax.swing.JLabel SaludTxt;
private javax.swing.JLabel SanitarioTxt;
private javax.swing.JLabel SexoTxt;
private javax.swing.JLabel imagen;

```



```

private javax.swing.JButton jButton1;
private javax.swing.JButton jButton3;
private javax.swing.JComboBox<String> jComboBox1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel8;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField10;
private javax.swing.JTextField jTextField11;
private javax.swing.JTextField jTextField12;
private javax.swing.JTextField jTextField13;
private javax.swing.JTextField jTextField14;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField4;
private javax.swing.JTextField jTextField5;
private javax.swing.JTextField jTextField6;
private javax.swing.JTextField jTextField7;
private javax.swing.JTextField jTextField8;
private javax.swing.JTextField jTextField9;
// End of variables declaration
}

```

HomeBusqueda.java

```

package Busqueda;

import com.frame.Inicio1;
import java.awt.BorderLayout;
import java.awt.Color;
import javax.swing.border.Border;

/**@author wicho*/
public class HomeBusqueda extends javax.swing.JFrame {

    int xMouse, yMouse;
    private String username;

```



```

public HomeBusqueda(String username) {
    initComponents();
    this.username = username;

    BuscarAnimal3 buscarAnimal3 = new BuscarAnimal3();
    ventanas.removeAll();
    ventanas.setLayout(new BorderLayout());
    buscarAnimal3.setSize(ventanas.getWidth(), ventanas.getHeight());
    ventanas.add(buscarAnimal3);
    ventanas.revalidate();
    ventanas.repaint();

    buscarAnimal3.setVisible(true);
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jPanel2 = new javax.swing.JPanel();
    ventanas = new javax.swing.JPanel();
    menu = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel1 = new javax.swing.JLabel();
    cierre = new javax.swing.JPanel();
    cierre = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setUndecorated(true);
    getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    jPanel2.setBackground(new java.awt.Color(0, 0, 0));

```



```

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 840, Short.MAX_VALUE)
);
jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 31, Short.MAX_VALUE)
);

getContentPane().add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 0, 840,
31));

ventanas.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));

javax.swing.GroupLayout ventanasLayout = new javax.swing.GroupLayout(ventanas);
ventanas.setLayout(ventanasLayout);
ventanasLayout.setHorizontalGroup(
    ventanasLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 688, Short.MAX_VALUE)
);
ventanasLayout.setVerticalGroup(
    ventanasLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 468, Short.MAX_VALUE)
);

getContentPane().add(ventanas, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 30,
690, 470));

menu.setBackground(new java.awt.Color(255, 153, 0));
menu.setPreferredSize(new java.awt.Dimension(870, 500));
menu.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO.jpg"))); //
NOI18N
menu.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 360, -1, 110));

```



```

jLabel3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO-
removebg-preview.png"))); // NOI18N
menu.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, -1));

jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/campo (1).jpg"))); //
NOI18N
menu.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, 470));

getContentPane().add(menu, new org.netbeans.lib.awtextra.AbsoluteConstraints(690, 30, 180,
470));

cierree.setBackground(new java.awt.Color(0, 0, 0));
cierree.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        cierreeMouseClicked(evt);
    }
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        cierreeMouseEntered(evt);
    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        cierreeMouseExited(evt);
    }
});

cierre.setFont(new java.awt.Font("Roboto Medium", 0, 18)); // NOI18N
cierre.setForeground(new java.awt.Color(255, 255, 255));
cierre.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
cierre.setText("X");

javax.swing.GroupLayout cierreeLayout = new javax.swing.GroupLayout(cierree);
cierree.setLayout(cierreeLayout);
cierreeLayout.setHorizontalGroup(
    cierreeLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            cierreeLayout.createSequentialGroup()
                .addComponent(cierre, javax.swing.GroupLayout.DEFAULT_SIZE, 24,
Short.MAX_VALUE)

```

```

        .addContainerGap()
    );
    cierreLayout.setVerticalGroup(
        cierreLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
cierreLayout.createSequentialGroup()
        .addGap(0, 0, Short.MAX_VALUE)
        .addComponent(cierre, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE))
    );

    getContentPane().add(cierre, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 30, -
1));

    pack();
    setLocationRelativeTo(null);
} // </editor-fold>

private void cierreMouseClicked(java.awt.event.MouseEvent evt) {
    Inicio1 inicioWindow = new Inicio1();
    inicioWindow.setVisible(true);
    this.dispose();
}

private void cierreMouseEntered(java.awt.event.MouseEvent evt) {
    cierre.setBackground(Color.red);
}

private void cierreMouseExited(java.awt.event.MouseEvent evt) {
    cierre.setBackground(Color.black);
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(() -> {
        new HomeBusqueda("Usuario").setVisible(true);
    });
}

```



```

    });

}

// Variables declaration - do not modify
private javax.swing.JLabel cierre;
private javax.swing.JPanel cierree;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel menu;
private javax.swing.JPanel ventanas;
// End of variables declaration

private void setBorder(Border createEmptyBorder) {
    throw new UnsupportedOperationException("Not supported yet.");
}
}

```

BuscarAnimal2

```

package BusquedaLotes;

import com.VerMasClient.VerMasClient;
import com.admin.*;
import com.conexion.Conexion;
import java.awt.Component;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.DefaultCellEditor;
import javax.swing.JButton;
import javax.swing.JCheckBox;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableCellRenderer;
import java.util.List;
import java.util.ArrayList;
import javax.swing.table.TableColumn;

```



```

/**@author wicho*/
public class BuscarAnimal2 extends javax.swing.JPanel {
    private List<Animal> listaAnimales = new ArrayList<>();
    String id = null;
    private int registroActual = 0; // Índice del registro actualmente mostrado

    public BuscarAnimal2() {
        initComponents();
        ocultarComponentes(); // Asegura que todos los componentes empiecen ocultos
        configurarEventos();
        jTextField1.setEditable(false);
    }

    public void buscarDatos(String query) {
        try {
            Connection con = Conexion.conectar();
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery(query);
            limpiarTabla();
            double totalPrecio = 0;
            // Obtener datos de la base de datos y agregarlos a la tabla
            while (rs.next()) {
                String[] fila = new String[jTable1.getColumnCount() - 1];
                for (int i = 0; i < fila.length; i++) {
                    fila[i] = rs.getString(i + 1);
                }
                // Añadir la fila sin el botón
                ((DefaultTableModel) jTable1.getModel()).addRow(fila);

                // Agregar el botón en la columna "Acciones"
                JButton actionButton = createActionButton();
                ((DefaultTableModel) jTable1.getModel()).setValueAt(actionButton, jTable1.getRowCount() -
1, jTable1.getColumnCount() - 1);

                // Acumular el precio de cada animal
                double precio = rs.getDouble("precio");
                totalPrecio += precio;
            }

            // Actualizar el JTextField1 con el precio total
            jTextField1.setText(String.valueOf(totalPrecio));

            // Configurar el comportamiento de los botones en la tabla
            TableColumn column = jTable1.getColumnModel().getColumn(jTable1.getColumnCount() - 1);
            column.setCellRenderer(new ButtonRenderer());
            column.setCellEditor(new ButtonEditor(new JCheckBox()));
            rs.close();
            stmt.close();
            con.close();
        } catch (SQLException e) {

```



```

        e.printStackTrace();
        JOptionPane.showMessageDialog(this, "Error al obtener datos de la base de datos.");
    }
}

private JButton createActionButton() {
    JButton button = new JButton("Ver Mas");
    button.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            System.out.println("pulso ver más del siguiente id"+id);
            VerMasClient verWindow = new VerMasClient(id);
            verWindow.setVisible(true);

        }
    });
    return button;
}

class ButtonRenderer extends JButton implements TableCellRenderer {
    public ButtonRenderer() {
        setOpaque(true);
    }

    @Override
    public Component getTableCellRendererComponent(JTable table, Object value, boolean
isSelected, boolean hasFocus, int row, int column) {
        if (value instanceof JButton) {
            return (JButton) value;
        }
        return this;
    }
}

class ButtonEditor extends DefaultCellEditor {
    protected JButton button;
    private String label;
    private boolean isPushed;

    public ButtonEditor(JCheckBox checkBox) {
        super(checkBox);
        button = new JButton();
        button.setOpaque(true);
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                fireEditingStopped();
            }
        });
    }

    @Override
    public Component getTableCellEditorComponent(JTable table, Object value, boolean isSelected,
int row, int column) {

```

```

        if (value instanceof JButton) {
            button = (JButton) value;
        }
        id = (String) table.getValueAt(row, 1);
        return button;
    }

    @Override
    public Object getCellEditorValue() {
        return label;
    }
}

class ModeloTablaConBotones extends DefaultTableModel {
    public ModeloTablaConBotones(Object[][] data, Object[] columnNames) {
        super(data, columnNames);
    }

    @Override
    public Class<?> getColumnClass(int columnIndex) {
        if (columnIndex == getColumnCount() - 1) { // La columna "Acciones"
            return JButton.class;
        }
        return super.getColumnClass(columnIndex);
    }
}

private void configurarEventos() {
    jComboBox1.setSelectedIndex(0);

    jComboBox1.addItemListener(e -> {
        if (e.getStateChange() == ItemEvent.SELECTED) {
            String selectedOption = (String) jComboBox1.getSelectedItem();
            ocultarComponentes();

            switch (selectedOption) {
                case "Busqueda por Categoria":
                    mostrarBusquedaPorCategoria();
                    break;
                default:
                    ocultarComponentes();
                    break;
            }
        }
    });

    jComboBox2.addItemListener(e -> {
        if (e.getStateChange() == ItemEvent.SELECTED) {
            String selectedOption = (String) jComboBox2.getSelectedItem();
            String query = "";
            ocultarComponentesExclusivos();

```

```

switch (selectedOption) {
    case "Busqueda por Raza":
        configurarComboBoxRaza();
        configurarModeloTabla(new String[]{"Raza", "Id", "No.Economico", "FechaAdq",
"Peso", "precio"});
        query = "SELECT Raza, Id, Numero_economico, Fecha_adquisicion, Peso, precio
FROM animales";
        break;
    case "Busqueda por Sexo":
        configurarComboBoxSexo();
        configurarModeloTabla(new String[]{"Sexo", "Id", "No.Economico", "FechaAdq",
"Peso", "precio"});
        query = "SELECT Sexo, Id, Numero_economico, Fecha_adquisicion, Peso, precio
FROM animales";
        break;
    case "Busqueda por Salud":
        configurarComboBoxSalud();
        configurarModeloTabla(new String[]{"Salud", "Id", "No.Economico","Raza", "Peso",
"precio"});
        query = "SELECT Salud, Id, Numero_economico, Raza, Peso, precio FROM
animales";
        break;
    case "Busqueda Reproductivo":
        configurarComboBoxReproductivo();
        configurarModeloTabla(new String[]{"Reproductivo", "Id", "Raza","No.Economico",
"Raza", "Peso", "precio"});
        query = "SELECT Reproductivo, Id, Raza, Numero_economico, Peso, precio FROM
animales";
        break;
    case "Busqueda por Etapa":
        configurarComboBoxEtapa();
        configurarModeloTabla(new String[]{"Etapa", "Id", "No.Economico", "Raza", "Peso",
"precio"});
        query = "SELECT Etapa, Id, Numero_economico, Raza, Peso, precio FROM
animales";
        break;
    case "Busqueda Corporal":
        configurarComboBoxCorporal();
        configurarModeloTabla(new String[]{"Corporal", "Id", "No.Economico", "Raza",
"Peso", "precio"});
        query = "SELECT Corporal, Id, Numero_economico, Raza, Peso, precio FROM
animales";
        break;
    case "Busqueda Sanitario":
        configurarComboBoxSanitario();
        configurarModeloTabla(new String[]{"Sanitario", "Id", "No.Economico", "Raza",
"Peso", "precio"});
        query = "SELECT Sanitario, Id, Numero_economico, Raza, Peso, precio FROM
animales";
        break;
    case "Busqueda Alimenticio":
        configurarComboBoxAlimenticio();

```

```

        configurarModeloTabla(new String[]{"Alimenticio", "Id", "No.Economico", "Raza",
"Peso", "precio"});
        query = "SELECT Alimenticio, Id, Numero_economico, Raza, Peso, precio FROM
animales";
        break;
        case "Modo de Adquisicion":
            configurarComboBoxModoAdquisicion();
            configurarModeloTabla(new String[]{"ModoAdqui", "Id", "No.Economico",
"FechaAdqui", "Peso", "precio"});
            query = "SELECT Modo_adquisicion, Id, Numero_economico, Fecha_adquisicion,
Peso, precio FROM animales";
            break;
        default:
            jTable1.setVisible(false);
            limpiarTabla();
            return;
    }
    if (!query.isEmpty()) {
        buscarDatos(query);
    }
}
});

```

```

jComboBox11.addItemListener(e -> {
    if (e.getStateChange() == ItemEvent.SELECTED && jComboBox2.getSelectedIndex() > 0) {
        String valorSeleccionado = (String) jComboBox11.getSelectedItem();
        String selectedOption = (String) jComboBox2.getSelectedItem();

        String query = "";
        switch (selectedOption) {
            case "Busqueda por Raza":
                query = "SELECT Raza, Id, Numero_economico, Fecha_adquisicion, Peso, precio
FROM animales WHERE Raza = '" + valorSeleccionado + "'";
                break;
            case "Busqueda por Sexo":
                query = "SELECT Sexo, Id, Numero_economico, Fecha_adquisicion, Peso, precio
FROM animales WHERE Sexo = '" + valorSeleccionado + "'";
                break;
            case "Busqueda por Salud":
                query = "SELECT Salud, Id, Numero_economico, Raza, Peso, precio FROM animales
WHERE Salud = '" + valorSeleccionado + "'";
                break;
            case "Busqueda Reproductivo":
                query = "SELECT Reproductivo, Id, Numero_economico, Raza, Peso, precio FROM
animales WHERE Reproductivo = '" + valorSeleccionado + "'";
                break;
            case "Busqueda por Etapa":
                query = "SELECT Etapa, Id, Numero_economico, Raza, Peso, precio FROM animales
WHERE Etapa = '" + valorSeleccionado + "'";
                break;
            case "Busqueda Corporal":

```

```

        query = "SELECT Corporal, Id, Numero_economico, Raza, Peso, precio FROM
animales WHERE Corporal = '" + valorSeleccionado + "'";
        break;
        case "Busqueda Sanitario":
            query = "SELECT Sanitario, Id, Numero_economico, Raza, Peso, precio FROM
animales WHERE Sanitario = '" + valorSeleccionado + "'";
            break;
        case "Busqueda Alimenticio":
            query = "SELECT Alimenticio, Id, Numero_economico, Raza, Peso, precio FROM
animales WHERE Alimenticio = '" + valorSeleccionado + "'";
            break;
        case "Modo de Adquisicion":
            query = "SELECT Modo_adquisicion, Id, Numero_economico, Fecha_adquisicion,
Peso, precio FROM animales WHERE Modo_adquisicion = '" + valorSeleccionado + "'";
            break;
        default:
            limpiarTabla();
            return;
    }

    if (!query.isEmpty()) {
        buscarDatos(query);
    }
}
});
}

private void mostrarBusquedaPorCategoria() {
    jComboBox2.setVisible(true);
}

private void ocultarComponentesExclusivos() {

    jScrollPane1.setVisible(false);
    jTable1.setVisible(false);
    jComboBox11.setVisible(false);

}

private void ocultarComponentes() {
    ocultarComponentesExclusivos();
}

// Métodos para configurar los ComboBox específicos
private void configurarComboBoxRaza() {
    jComboBox11.setVisible(true);
    jScrollPane1.setVisible(true);
    jTable1.setVisible(true);
    jComboBox11.removeAllItems();
    String[] razas = {"----", "Angus", "Hereford", "Charolais", "Limousin", "Brahman", "Simmental",
"Holstein", "Jersey", "Guernsey", "Pardo Suizo", "Ayrshire", "Shorthorn", "Normando", "Fleckvieh",
"Nellore", "Criollo"};

```

```

        for (String raza : razas) {
            jComboBox11.addItem(raza);
        }
    }

    private void configurarComboBoxSexo() {
        jComboBox11.setVisible(true);
        jScrollPane1.setVisible(true);
        jTable1.setVisible(true);
        jComboBox11.removeAllItems();
        jComboBox11.addItem("----");
        jComboBox11.addItem("Macho");
        jComboBox11.addItem("Hembra");
    }

    private void configurarComboBoxSalud() {
        jComboBox11.setVisible(true);
        jScrollPane1.setVisible(true);
        jTable1.setVisible(true);
        jComboBox11.removeAllItems();
        jComboBox11.addItem("----");
        jComboBox11.addItem("Saludable");
        jComboBox11.addItem("Enfermo");
        jComboBox11.addItem("En recuperación");
        jComboBox11.addItem("Lesionado");
    }

    private void configurarComboBoxReproductivo() {
        jComboBox11.setVisible(true);
        jScrollPane1.setVisible(true);
        jTable1.setVisible(true);
        jComboBox11.removeAllItems();
        jComboBox11.addItem("----");
        jComboBox11.addItem("Gestante");
        jComboBox11.addItem("En celo");
        jComboBox11.addItem("Vacío");
        jComboBox11.addItem("Parida");
        jComboBox11.addItem("Activo");
        jComboBox11.addItem("Inactivo");
    }

    private void configurarComboBoxEtapa() {
        jComboBox11.setVisible(true);
        jScrollPane1.setVisible(true);
        jTable1.setVisible(true);
        jComboBox11.removeAllItems();
        jComboBox11.addItem("----");
        jComboBox11.addItem("Ternero o becerro");
        jComboBox11.addItem("Joven o recr  a");
        jComboBox11.addItem("Adulto");
        jComboBox11.addItem("Anciano");
    }
}

```

```

private void configurarComboBoxCorporal() {
    jComboBox11.setVisible(true);
    jScrollPane1.setVisible(true);
    jTable1.setVisible(true);
    jComboBox11.removeAllItems();
    jComboBox11.addItem("----");
    jComboBox11.addItem("Óptima");
    jComboBox11.addItem("Baja condición");
    jComboBox11.addItem("Sobrepeso");
}

private void configurarComboBoxSanitario() {
    jComboBox11.setVisible(true);
    jScrollPane1.setVisible(true);
    jTable1.setVisible(true);
    jComboBox11.removeAllItems();
    jComboBox11.addItem("----");
    jComboBox11.addItem("Vacunado");
    jComboBox11.addItem("En cuarentena");
    jComboBox11.addItem("En tratamiento");
    jComboBox11.addItem("Sin tratamiento");
}

private void configurarComboBoxAlimenticio() {
    jComboBox11.setVisible(true);
    jScrollPane1.setVisible(true);
    jTable1.setVisible(true);
    jComboBox11.removeAllItems();
    jComboBox11.addItem("----");
    jComboBox11.addItem("Sano");
    jComboBox11.addItem("Desnutrido");
    jComboBox11.addItem("Obeso");
}

private void configurarComboBoxModoAdquisicion() {
    jComboBox11.setVisible(true);
    jTable1.setVisible(true);
    jScrollPane1.setVisible(true);
    jComboBox11.removeAllItems();
    jComboBox11.addItem("----");
    jComboBox11.addItem("Compra");
    jComboBox11.addItem("Donación");
    jComboBox11.addItem("Intercambio");
}

private void limpiarTabla() {
    DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
    model.setRowCount(0);
}

```



```

private void configurarModeloTabla(String[] columnas) {
    // Usar el modelo personalizado con botones
    DefaultTableModel model = new ModeloTablaConBotones(new Object[0][0], columnas);
    model.addColumn("Acciones");

    // Configurar la tabla con el nuevo modelo
    jTable1.setModel(model);
}

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel2 = new javax.swing.JLabel();
    jComboBox1 = new javax.swing.JComboBox<>();
    jComboBox2 = new javax.swing.JComboBox<>();
    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    jComboBox11 = new javax.swing.JComboBox<>();
    jLabel3 = new javax.swing.JLabel();
    jTextField1 = new javax.swing.JTextField();

    setBackground(new java.awt.Color(255, 255, 255));
    setLayout(null);

    jLabel2.setFont(new java.awt.Font("Roboto Medium", 0, 12)); // NOI18N
    jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    jLabel2.setText("BUSQUEDA EN EL REGISTRO");
    add(jLabel2);
    jLabel2.setBounds(0, 0, 470, 30);

    jComboBox1.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Busqueda
por Categoria" }));
    add(jComboBox1);
    jComboBox1.setBounds(20, 50, 200, 22);

    jComboBox2.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Categoria",
"Busqueda por Raza", "Busqueda por Sexo", "Busqueda por Salud", "Busqueda Reproductivo",
"Busqueda por Etapa", "Busqueda Corporal", "Busqueda Sanitario", "Busqueda Alimenticio", "Modo
de Adquisicion" }));
    add(jComboBox2);
    jComboBox2.setBounds(240, 50, 220, 22);

    jTable1.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null}
        },
        new String [] {
            "Title 1", "Title 2", "Title 3", "Title 4"
        }
    ));

```

```

    }
    ));
    jScrollPane1.setViewportViewView(jTable1);

    add(jScrollPane1);
    jScrollPane1.setBounds(10, 90, 670, 360);

    jComboBox11.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "----" }));
    add(jComboBox11);
    jComboBox11.setBounds(480, 50, 200, 22);

    jLabel3.setText("Precio $$$.");
    add(jLabel3);
    jLabel3.setBounds(490, 10, 80, 20);
    add(jTextField1);
    jTextField1.setBounds(560, 10, 64, 22);
} // </editor-fold>

// Variables declaration - do not modify
private javax.swing.JComboBox<String> jComboBox1;
private javax.swing.JComboBox<String> jComboBox11;
private javax.swing.JComboBox<String> jComboBox2;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
private javax.swing.JTextField jTextField1;
// End of variables declaration
}

```

HomeBusquedaLotes.java

```

package BusquedaLotes;

import com.frame.Inicio1;
import java.awt.BorderLayout;
import java.awt.Color;
import javax.swing.border.Border;

/** @author wicho */
public class HomeBusquedaLotes extends javax.swing.JFrame {

    int xMouse, yMouse;
    private String username;

    /**
     * Creates new form Home
     */
    public HomeBusquedaLotes(String username) {
        initComponents();
        this.username = username;
        BuscarAnimal2 buscarAnimal2 = new BuscarAnimal2();
    }

```



```

        ventanas.removeAll();
        ventanas.setLayout(new BorderLayout());
        buscarAnimal2.setSize(ventanas.getWidth(), ventanas.getHeight());
        ventanas.add(buscarAnimal2);
        ventanas.revalidate();
        ventanas.repaint();
        buscarAnimal2.setVisible(true);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        cierre = new javax.swing.JPanel();
        cierre = new javax.swing.JLabel();
        jPanel2 = new javax.swing.JPanel();
        ventanas = new javax.swing.JPanel();
        menu = new javax.swing.JPanel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel1 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setUndecorated(true);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        cierre.setBackground(new java.awt.Color(0, 0, 0));
        cierre.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                cierreMouseClicked(evt);
            }
            public void mouseEntered(java.awt.event.MouseEvent evt) {
                cierreMouseEntered(evt);
            }
            public void mouseExited(java.awt.event.MouseEvent evt) {
                cierreMouseExited(evt);
            }
        });

        cierre.setFont(new java.awt.Font("Roboto Medium", 0, 18)); // NOI18N
        cierre.setForeground(new java.awt.Color(255, 255, 255));
        cierre.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        cierre.setText("X");

        javax.swing.GroupLayout cierreLayout = new javax.swing.GroupLayout(cierre);
        cierre.setLayout(cierreLayout);
        cierreLayout.setHorizontalGroup(

```

```

        cierreLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
cierreLayout.createSequentialGroup()
        .addComponent(cierre, javax.swing.GroupLayout.DEFAULT_SIZE, 24,
Short.MAX_VALUE)
        .addContainerGap())
    );
    cierreLayout.setVerticalGroup(
        cierreLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
cierreLayout.createSequentialGroup()
        .addGap(0, 0, Short.MAX_VALUE)
        .addComponent(cierre, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE))
    );

    getContentPane().add(cierre, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 30, -
1));

    jPanel2.setBackground(new java.awt.Color(0, 0, 0));

    javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
    jPanel2.setLayout(jPanel2Layout);
    jPanel2Layout.setHorizontalGroup(
        jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 840, Short.MAX_VALUE)
    );
    jPanel2Layout.setVerticalGroup(
        jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 31, Short.MAX_VALUE)
    );

    getContentPane().add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 0, 840,
31));

    ventanas.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));

    javax.swing.GroupLayout ventanasLayout = new javax.swing.GroupLayout(ventanas);
    ventanas.setLayout(ventanasLayout);
    ventanasLayout.setHorizontalGroup(
        ventanasLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 688, Short.MAX_VALUE)
    );
    ventanasLayout.setVerticalGroup(
        ventanasLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 468, Short.MAX_VALUE)
    );

    getContentPane().add(ventanas, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 30,
690, 470));

    menu.setBackground(new java.awt.Color(255, 153, 0));

```



```

        menu.setPreferredSize(new java.awt.Dimension(870, 500));
        menu.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabel2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO.jpg"))); //
NOI18N
        menu.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 360, -1, 110));

        jLabel3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO-
removebg-preview.png"))); // NOI18N
        menu.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, -1));

        jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/campo (1).jpg"))); //
NOI18N
        menu.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, 470));

        getContentPane().add(menu, new org.netbeans.lib.awtextra.AbsoluteConstraints(690, 30, 180,
470));

        pack();
        setLocationRelativeTo(null);
    } // </editor-fold>

    private void cierreMouseEntered(java.awt.event.MouseEvent evt) {
        cierre.setBackground(Color.red);
    }

    private void cierreMouseExited(java.awt.event.MouseEvent evt) {
        cierre.setBackground(Color.black);
    }

    private void cierreMouseClicked(java.awt.event.MouseEvent evt) {
        Inicio1 inicioWindow = new Inicio1();
        inicioWindow.setVisible(true);
        this.dispose();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(() -> {
            new HomeBusquedaLotes("Usuario").setVisible(true);
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JLabel cierre;
    private javax.swing.JPanel cierree;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;

```



```

private javax.swing.JPanel jPanel2;
private javax.swing.JPanel menu;
private javax.swing.JPanel ventanas;
// End of variables declaration

private void setBorder(Border createEmptyBorder) {
    throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
}
}

```

BuscarAnimal1

```
package Ganado;
```

```

import com.VerMasClient.VerMasClient;
import com.admin.*;
import com.conexion.Conexion;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.List;
import java.util.ArrayList;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;

```

```
/** @author wicho */
```

```
public class BuscarAnimal1 extends javax.swing.JPanel {
```

```

    private List<Animal> listaAnimales = new ArrayList<>();
    String id = null;
    private int registroActual = 0;

```

```

    public BuscarAnimal1() {
        initComponents();
        ocultarComponentes();
        mostrarRegistrosPorOrden();
        jTextField14.setEditable(false);
    }

```

```

    private void mostrarRegistrosPorOrden() {
        listaAnimales.clear();
        try (Connection conn = Conexion.conectar()) {
            String query = "SELECT id, Numero_economico, Fecha_adquisicion, raza, peso, sexo, salud,
reproductivo, etapa, corporal, sanitario, alimenticio, imagen, precio FROM animales";
            try (Statement stmt = conn.createStatement(); ResultSet rs = stmt.executeQuery(query)) {
                while (rs.next()) {
                    Animal animal = new Animal(
                        rs.getInt("id"),
                        rs.getString("Numero_economico"),
                        rs.getDate("Fecha_adquisicion"),

```

```

        rs.getString("raza"),
        rs.getFloat("peso"),
        rs.getString("sexo"),
        rs.getString("salud"),
        rs.getString("reproductivo"),
        rs.getString("etapa"),
        rs.getString("corporal"),
        rs.getString("sanitario"),
        rs.getString("alimenticio"),
        rs.getBytes("imagen"),
        rs.getString("precio")
    );
    listaAnimales.add(animal);
    Anterior.setVisible(true);
    VerMas.setVisible(true);
    Siguiente.setVisible(true);
}
}
if (!listaAnimales.isEmpty()) {
    registroActual = 0;
    mostrarAnimal(listaAnimales.get(registroActual));
    // Bloquear todos los campos de entrada donde se muestran los datos
    jTextField2.setEditable(false);
    jTextField3.setEditable(false);
    jTextField4.setEditable(false);
    jTextField5.setEditable(false);
    jTextField6.setEditable(false);
    jTextField7.setEditable(false);
    jTextField8.setEditable(false);
    jTextField9.setEditable(false);
    jTextField10.setEditable(false);
    jTextField11.setEditable(false);
    jTextField12.setEditable(false);
    jTextField13.setEditable(false);
} else {
    System.out.println("No se encontraron registros.");
}
} catch (SQLException e) {
    e.printStackTrace();
}
}

private void mostrarAnimal(Animal animal) {
    imagen.setVisible(true);
    // Convertir byte[] a Image y establecer en JLabel
    if (animal.getImagen() != null) {
        ImageIcon imagenIcon = new ImageIcon(animal.getImagen());
        imagen.setIcon(imagenIcon);
    } else {
        imagen.setIcon(null);
    }
}

```



```
ldtxt.setVisible(true);
jTextField2.setVisible(true);
jTextField2.setText(String.valueOf(animal.getId()));
id = jTextField2.getText();

NoTxt.setVisible(true);
jTextField4.setVisible(true);
jTextField4.setText(animal.getNumeroEconomico());

FechaAdqTxt.setVisible(true);
jTextField5.setVisible(true);
jTextField5.setText(animal.getFechaAdquisicion().toString());

RazaTxt.setVisible(true);
jTextField6.setVisible(true);
jTextField6.setText(animal.getRaza());

PesoTxt.setVisible(true);
jTextField3.setVisible(true);
jTextField3.setText(String.valueOf(animal.getPeso()));

SexoTxt.setVisible(true);
jTextField7.setVisible(true);
jTextField7.setText(animal.getSexo());

jLabel8.setVisible(true);
jSeparator1.setVisible(true);

SaludTxt.setVisible(true);
jTextField8.setVisible(true);
jTextField8.setText(animal.getSalud());

ReproductivoTxt.setVisible(true);
jTextField9.setVisible(true);
jTextField9.setText(animal.getReproductivo());

EtapasTxt.setVisible(true);
jTextField10.setVisible(true);
jTextField10.setText(animal.getEtapas());

CorporalTxt.setVisible(true);
jTextField11.setVisible(true);
jTextField11.setText(animal.getCorporal());

SanitarioTxt.setVisible(true);
jTextField12.setVisible(true);
jTextField12.setText(animal.getSanitario());

AlimenticioTxt.setVisible(true);
jTextField13.setVisible(true);
jTextField13.setText(animal.getAlimenticio());
```

```

jLabel8.setVisible(true);
jSeparator1.setVisible(true);
jTextField14.setText(animal.getPrecio());
}

private void ocultarComponentesExclusivos() {
    imagen.setVisible(false);
    Idtxt.setVisible(false);
    jTextField2.setVisible(false);
    NoTxt.setVisible(false);
    jTextField4.setVisible(false);
    FechaAdqTxt.setVisible(false);
    jTextField5.setVisible(false);
    RazaTxt.setVisible(false);
    jTextField6.setVisible(false);
    PesoTxt.setVisible(false);
    jTextField3.setVisible(false);
    SexoTxt.setVisible(false);
    jTextField7.setVisible(false);
    jLabel8.setVisible(false);
    jSeparator1.setVisible(false);
    SaludTxt.setVisible(false);
    jTextField8.setVisible(false);
    ReproductivoTxt.setVisible(false);
    jTextField9.setVisible(false);
    EtapaTxt.setVisible(false);
    jTextField10.setVisible(false);
    CorporalTxt.setVisible(false);
    jTextField11.setVisible(false);
    SanitarioTxt.setVisible(false);
    jTextField12.setVisible(false);
    AlimenticioTxt.setVisible(false);
    jTextField13.setVisible(false);
    Anterior.setVisible(false);
    VerMas.setVisible(false);
    Siguiente.setVisible(false);
}

private void ocultarComponentes() {
    ocultarComponentesExclusivos();
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

```

```

jLabel2 = new javax.swing.JLabel();
imagen = new javax.swing.JLabel();
ldtxt = new javax.swing.JLabel();
FechaAdqTxt = new javax.swing.JLabel();
RazaTxt = new javax.swing.JLabel();
PesoTxt = new javax.swing.JLabel();
jTextField2 = new javax.swing.JTextField();
jTextField3 = new javax.swing.JTextField();
SexoTxt = new javax.swing.JLabel();
NoTxt = new javax.swing.JLabel();
jTextField4 = new javax.swing.JTextField();
jSeparator1 = new javax.swing.JSeparator();
jLabel8 = new javax.swing.JLabel();
SaludTxt = new javax.swing.JLabel();
ReproductivoTxt = new javax.swing.JLabel();
EtapasTxt = new javax.swing.JLabel();
CorporalTxt = new javax.swing.JLabel();
AlimenticioTxt = new javax.swing.JLabel();
SanitarioTxt = new javax.swing.JLabel();
jTextField6 = new javax.swing.JTextField();
jTextField5 = new javax.swing.JTextField();
jTextField7 = new javax.swing.JTextField();
jTextField8 = new javax.swing.JTextField();
jTextField9 = new javax.swing.JTextField();
jTextField10 = new javax.swing.JTextField();
jTextField11 = new javax.swing.JTextField();
jTextField12 = new javax.swing.JTextField();
jTextField13 = new javax.swing.JTextField();
Anterior = new javax.swing.JButton();
VerMas = new javax.swing.JButton();
Siguiente = new javax.swing.JButton();
jTextField14 = new javax.swing.JTextField();
jLabel1 = new javax.swing.JLabel();

setBackground(new java.awt.Color(255, 255, 255));
setLayout(null);

jLabel2.setFont(new java.awt.Font("Roboto Medium", 0, 12)); // NOI18N
jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel2.setText("BUSQUEDA POR ORDEN DE REGISTRO");
add(jLabel2);
jLabel2.setBounds(0, 0, 690, 40);

imagen.setIcon(new javax.swing.ImageIcon(getClass().getResource("/iconos/añadir.png"))); //
NOI18N
imagen.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
add(imagen);
imagen.setBounds(20, 80, 130, 120);

ldtxt.setText("Numero de ID:");
add(ldtxt);
ldtxt.setBounds(190, 70, 90, 20);

```



```
FechaAdqTxt.setText("Fecha de Adquisicion:");  
add(FechaAdqTxt);  
FechaAdqTxt.setBounds(190, 110, 130, 16);
```

```
RazaTxt.setText("Tipo o Raza:");  
add(RazaTxt);  
RazaTxt.setBounds(190, 150, 80, 16);
```

```
PesoTxt.setText("Peso:");  
add(PesoTxt);  
PesoTxt.setBounds(190, 190, 50, 16);  
add(jTextField2);  
jTextField2.setBounds(280, 70, 140, 22);  
add(jTextField3);  
jTextField3.setBounds(240, 190, 160, 22);
```

```
SexoTxt.setText("Sexo:");  
add(SexoTxt);  
SexoTxt.setBounds(410, 190, 50, 16);
```

```
NoTxt.setText("No. Economico Propiedad:");  
add(NoTxt);  
NoTxt.setBounds(430, 70, 160, 20);  
add(jTextField4);  
jTextField4.setBounds(590, 70, 73, 22);
```

```
jSeparator1.setForeground(new java.awt.Color(153, 153, 153));  
add(jSeparator1);  
jSeparator1.setBounds(150, 230, 510, 10);
```

```
jLabel8.setText("Estados o Categorias");  
add(jLabel8);  
jLabel8.setBounds(20, 220, 130, 20);
```

```
SaludTxt.setText("Salud:");  
add(SaludTxt);  
SaludTxt.setBounds(20, 260, 60, 20);
```

```
ReproductivoTxt.setText("Reproductivo:");  
add(ReproductivoTxt);  
ReproductivoTxt.setBounds(230, 260, 90, 20);
```

```
EtapaTxt.setText("Etapa:");  
add(EtapaTxt);  
EtapaTxt.setBounds(470, 260, 50, 20);
```

```
CorporalTxt.setText("Corporal:");  
add(CorporalTxt);  
CorporalTxt.setBounds(20, 310, 70, 20);
```

```
AlimenticioTxt.setText("Alimenticio:");
```



```

add(AlimenticioTxt);
AlimenticioTxt.setBounds(440, 310, 80, 20);

SanitarioTxt.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
SanitarioTxt.setText("Sanitario:");
add(SanitarioTxt);
SanitarioTxt.setBounds(220, 310, 60, 20);
add(jTextField6);
jTextField6.setBounds(320, 150, 340, 22);
add(jTextField5);
jTextField5.setBounds(320, 110, 340, 22);
add(jTextField7);
jTextField7.setBounds(450, 190, 210, 22);
add(jTextField8);
jTextField8.setBounds(70, 260, 150, 22);
add(jTextField9);
jTextField9.setBounds(320, 260, 140, 22);
add(jTextField10);
jTextField10.setBounds(510, 260, 150, 22);
add(jTextField11);
jTextField11.setBounds(80, 310, 140, 22);
add(jTextField12);
jTextField12.setBounds(280, 310, 160, 22);
add(jTextField13);
jTextField13.setBounds(510, 310, 150, 22);

Anterior.setText("ANTERIOR");
Anterior.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        AnteriorMouseClicked(evt);
    }
});
add(Anterior);
Anterior.setBounds(140, 390, 110, 23);

VerMas.setText("VER MÁS");
VerMas.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        VerMasMouseClicked(evt);
    }
});
add(VerMas);
VerMas.setBounds(440, 390, 110, 23);

Siguiente.setText("SIGUIENTE");
Siguiente.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        SiguienteMouseClicked(evt);
    }
});
add(Siguiente);
Siguiente.setBounds(280, 390, 120, 23);

```



```

add(jTextField14);
jTextField14.setBounds(590, 10, 64, 22);

jLabel1.setText("Precio $$$:");
add(jLabel1);
jLabel1.setBounds(510, 10, 70, 20);
} // </editor-fold>

private void SiguienteMouseClicked(java.awt.event.MouseEvent evt) {
    if (registroActual < listaAnimales.size() - 1) {
        registroActual++;
        mostrarAnimal(listaAnimales.get(registroActual));
    } else {
        JOptionPane.showMessageDialog(this, "No hay más registros.", "Fin de la lista",
JOptionPane.INFORMATION_MESSAGE);
    }
}

private void AnteriorMouseClicked(java.awt.event.MouseEvent evt) {
    if (registroActual > 0) {
        registroActual--;
        mostrarAnimal(listaAnimales.get(registroActual));
    } else {
        JOptionPane.showMessageDialog(this, "No hay registros anteriores.", "Inicio de la lista",
JOptionPane.INFORMATION_MESSAGE);
    }
}

private void VerMasMouseClicked(java.awt.event.MouseEvent evt) {
    VerMasClient verWindow = new VerMasClient(id);
    verWindow.setVisible(true);
}

// Variables declaration - do not modify
private javax.swing.JLabel AlimenticioTxt;
private javax.swing.JButton Anterior;
private javax.swing.JLabel CorporalTxt;
private javax.swing.JLabel EtapaTxt;
private javax.swing.JLabel FechaAdqTxt;
private javax.swing.JLabel Idtxt;
private javax.swing.JLabel NoTxt;
private javax.swing.JLabel PesoTxt;
private javax.swing.JLabel RazaTxt;
private javax.swing.JLabel ReproductivoTxt;
private javax.swing.JLabel SaludTxt;
private javax.swing.JLabel SanitarioTxt;
private javax.swing.JLabel SexoTxt;
private javax.swing.JButton Siguiente;
private javax.swing.JButton VerMas;
private javax.swing.JLabel imagen;
private javax.swing.JLabel jLabel1;

```

```

private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel8;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JTextField jTextField10;
private javax.swing.JTextField jTextField11;
private javax.swing.JTextField jTextField12;
private javax.swing.JTextField jTextField13;
private javax.swing.JTextField jTextField14;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField4;
private javax.swing.JTextField jTextField5;
private javax.swing.JTextField jTextField6;
private javax.swing.JTextField jTextField7;
private javax.swing.JTextField jTextField8;
private javax.swing.JTextField jTextField9;
// End of variables declaration
}

```

HomeGanado.java

```
package Ganado;
```

```

import com.frame.Inicio1;
import java.awt.BorderLayout;
import java.awt.Color;
import javax.swing.border.Border;

```

```
/** @author wicho */
```

```
public class HomeGanado extends javax.swing.JFrame {
```

```

    int xMouse, yMouse;
    private String username;

```

```
/**
```

```
 * Creates new form Home
```

```
 */
```

```

public HomeGanado(String username) {
    initComponents();
    this.username = username;
    BuscarAnimal1 buscarAnimal1 = new BuscarAnimal1();
    ventanas.removeAll();
    ventanas.setLayout(new BorderLayout());
    buscarAnimal1.setSize(ventanas.getWidth(), ventanas.getHeight());
    ventanas.add(buscarAnimal1);
    ventanas.revalidate();
    ventanas.repaint();
    buscarAnimal1.setVisible(true);
}

```

```
/**
```

```
 * This method is called from within the constructor to initialize the form.
```



```

* WARNING: Do NOT modify this code. The content of this method is always
* regenerated by the Form Editor.
*/
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    cierree = new javax.swing.JPanel();
    cierre = new javax.swing.JLabel();
    jPanel2 = new javax.swing.JPanel();
    ventanas = new javax.swing.JPanel();
    menu = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel1 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setUndecorated(true);
    getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    cierree.setBackground(new java.awt.Color(0, 0, 0));
    cierree.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            cierreeMouseClicked(evt);
        }
        public void mouseEntered(java.awt.event.MouseEvent evt) {
            cierreeMouseEntered(evt);
        }
        public void mouseExited(java.awt.event.MouseEvent evt) {
            cierreeMouseExited(evt);
        }
    });

    cierre.setFont(new java.awt.Font("Roboto Medium", 0, 18)); // NOI18N
    cierre.setForeground(new java.awt.Color(255, 255, 255));
    cierre.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    cierre.setText("X");

    javax.swing.GroupLayout cierreeLayout = new javax.swing.GroupLayout(cierree);
    cierree.setLayout(cierreeLayout);
    cierreeLayout.setHorizontalGroup(
        cierreeLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                cierreeLayout.createSequentialGroup()
                    .addComponent(cierre, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                    .addGap(10, 10, 10))
    );
    cierreeLayout.setVerticalGroup(
        cierreeLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                cierreeLayout.createSequentialGroup()
                    .addComponent(cierre, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                    .addGap(10, 10, 10))
    );
}

```

```

        .addGap(0, 0, Short.MAX_VALUE)
        .addComponent(cierre, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE))
    );

    getContentPane().add(cierree, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 30, -
1));

    jPanel2.setBackground(new java.awt.Color(0, 0, 0));

    javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
    jPanel2.setLayout(jPanel2Layout);
    jPanel2Layout.setHorizontalGroup(
        jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 840, Short.MAX_VALUE)
    );
    jPanel2Layout.setVerticalGroup(
        jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 31, Short.MAX_VALUE)
    );

    getContentPane().add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 0, 840,
31));

    ventanas.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));

    javax.swing.GroupLayout ventanasLayout = new javax.swing.GroupLayout(ventanas);
    ventanas.setLayout(ventanasLayout);
    ventanasLayout.setHorizontalGroup(
        ventanasLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 688, Short.MAX_VALUE)
    );
    ventanasLayout.setVerticalGroup(
        ventanasLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 468, Short.MAX_VALUE)
    );

    getContentPane().add(ventanas, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 30,
690, 470));

    menu.setBackground(new java.awt.Color(255, 255, 255));
    menu.setPreferredSize(new java.awt.Dimension(870, 500));
    menu.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    jLabel2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO.jpg"))); //
NOI18N
    menu.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 360, -1, 110));

    jLabel3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO-
removebg-preview.png"))); // NOI18N
    menu.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, -1));

```

```

        JLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/campo (1).jpg"))); //
NOI18N
        menu.add(JLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, 470));

        getContentPane().add(menu, new org.netbeans.lib.awtextra.AbsoluteConstraints(690, 30, 180,
480));

        pack();
        setLocationRelativeTo(null);
    } // </editor-fold>

    private void cierreMouseClicked(java.awt.event.MouseEvent evt) {
        cierre.setBackground(Color.red);
    }

    private void cierreMouseExited(java.awt.event.MouseEvent evt) {
        cierre.setBackground(Color.black);
    }

    private void cierreMouseClicked(java.awt.event.MouseEvent evt) {
        Inicio1 inicioWindow = new Inicio1();
        inicioWindow.setVisible(true);
        this.dispose();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(() -> {
            new HomeGanado("Usuario").setVisible(true);
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JLabel cierre;
    private javax.swing.JPanel cierree;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel menu;
    private javax.swing.JPanel ventanas;
    // End of variables declaration

    private void setBorder(Border createEmptyBorder) {
        throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
    }
}

```



Gasto.java

```
package com.Administracion;

import com.frame.Inicio;
import java.awt.BorderLayout;
import java.awt.Color;
import javax.swing.border.Border;

/**
 *
 * @author wicho
 */
public class Gasto extends javax.swing.JFrame {
    int xMouse, yMouse;
    private String username;
    /**
     * Creates new form Home
     */
    public Gasto(String username) {
        initComponents();
        this.username = username;
        System.out.println(username);
        Gastos pruebaPanel = new Gastos(username);
        ventanas.removeAll();
        ventanas.setLayout(new BorderLayout());
        pruebaPanel.setSize(ventanas.getWidth(), ventanas.getHeight());
        ventanas.add(pruebaPanel);
        ventanas.revalidate();
        ventanas.repaint();
        pruebaPanel.setVisible(true);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        cierre = new javax.swing.JPanel();
        cierre = new javax.swing.JLabel();
        jPanel2 = new javax.swing.JPanel();
        ventanas = new javax.swing.JPanel();
        menu = new javax.swing.JPanel();
        jLabel2 = new javax.swing.JLabel();
    }
}
```



```

jLabel3 = new javax.swing.JLabel();
jLabel1 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setUndecorated(true);
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

cierree.setBackground(new java.awt.Color(0, 0, 0));
cierree.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        cierreeMouseClicked(evt);
    }
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        cierreeMouseEntered(evt);
    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        cierreeMouseExited(evt);
    }
});

cierre.setFont(new java.awt.Font("Roboto Medium", 0, 18)); // NOI18N
cierre.setForeground(new java.awt.Color(255, 255, 255));
cierre.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
cierre.setText("X");

javax.swing.GroupLayout cierreeLayout = new javax.swing.GroupLayout(cierree);
cierree.setLayout(cierreeLayout);
cierreeLayout.setHorizontalGroup(
    cierreeLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            cierreeLayout.createSequentialGroup()
                .addComponent(cierre, javax.swing.GroupLayout.DEFAULT_SIZE, 24,
Short.MAX_VALUE)
                .addGap(10, 10, 10))
        .addGroup(cierreeLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(cierreeLayout.createSequentialGroup()
                .addGap(10, 10, 10)
                .addComponent(cierre, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(10, 10, 10))
);

getContentPane().add(cierree, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 30, -
1));

jPanel2.setBackground(new java.awt.Color(0, 0, 0));

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(

```



```

        jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 840, Short.MAX_VALUE)
        );
        jPanel2Layout.setVerticalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGap(0, 31, Short.MAX_VALUE)
            );

        getContentPane().add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 0, 840,
31));

        ventanas.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));

        javax.swing.GroupLayout ventanasLayout = new javax.swing.GroupLayout(ventanas);
        ventanas.setLayout(ventanasLayout);
        ventanasLayout.setHorizontalGroup(
            ventanasLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGap(0, 688, Short.MAX_VALUE)
            );
        ventanasLayout.setVerticalGroup(
            ventanasLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGap(0, 468, Short.MAX_VALUE)
            );

        getContentPane().add(ventanas, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 30,
690, 470));

        menu.setBackground(new java.awt.Color(255, 153, 0));
        menu.setPreferredSize(new java.awt.Dimension(870, 500));
        menu.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabel2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO.jpg"))); //
NOI18N
        menu.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 360, -1, 110));

        jLabel3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO-
removebg-preview.png"))); // NOI18N
        menu.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, -1));

        jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/campo (1).jpg"))); //
NOI18N
        menu.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, 470));

        getContentPane().add(menu, new org.netbeans.lib.awtextra.AbsoluteConstraints(690, 30, 180,
470));

        pack();
        setLocationRelativeTo(null);
    } // </editor-fold>

    private void cierreMouseEntered(java.awt.event.MouseEvent evt) {
        cierre.setBackground(Color.red);
    }

```



```

    }

    private void cierreMouseExited(java.awt.event.MouseEvent evt) {
        cierre.setBackground(Color.black);
    }

    private void cierreMouseClicked(java.awt.event.MouseEvent evt) {
        Inicio inicioWindow = new Inicio(username);
        inicioWindow.setVisible(true);
        this.dispose();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(() -> {
            new Gasto("Usuario").setVisible(true);
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JLabel cierre;
    private javax.swing.JPanel cierree;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel menu;
    private javax.swing.JPanel ventanas;
    // End of variables declaration

    private void setBorder(Border createEmptyBorder) {
        throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
    }
}

```

Gastos.java

```

package com.Administracion;

import com.conexion.Conexion;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

```



```

/** @author wicho */
public class Gastos extends javax.swing.JPanel {

    private double total = 0.0;
    private String username;

    /**
     * Creates new form Precios
     */
    public Gastos(String username) {
        initComponents();
        this.username = username;
        cargarUltimosRegistros();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        Anio = new javax.swing.JTextField();
        Presupuesto = new javax.swing.JTextField();
        Concepto = new javax.swing.JTextField();
        Proveedor = new javax.swing.JTextField();
        Guardar = new javax.swing.JButton();
        Mes = new javax.swing.JComboBox<>();
        Monto = new javax.swing.JTextField();
        jLabel4 = new javax.swing.JLabel();
        GastoTotal = new javax.swing.JTextField();
        jLabel5 = new javax.swing.JLabel();
        Restante = new javax.swing.JTextField();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jScrollPane1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();

        setBackground(new java.awt.Color(255, 255, 255));

        jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel1.setText("Administracion de Gastos");

        jLabel2.setText("Año:");

        jLabel3.setText("Presupuesto Asignado:");

```



```

Concepto.setText("Concepto");
Concepto.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        ConceptoMouseClicked(evt);
    }
});

Provedor.setText("Provedor");
Provedor.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        ProvedorMouseClicked(evt);
    }
});

Guardar.setText("Añadir");
Guardar.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        GuardarMouseClicked(evt);
    }
});

Mes.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "MES", "ENERO",
"FEBRERO", "MARZO", "ABRIL", "MAYO", "JUNIO", "JULIO", "AGOSTO", "SEPTIEMBRE",
"OCTUBRE", "NOVIEMBRE", "DICIEMBRE" }));

Monto.setText("Monto");
Monto.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        MontoMouseClicked(evt);
    }
});

jLabel4.setText("Gasto Total:");

GastoTotal.setText("0.0");

jLabel5.setText("Restante:");

jButton1.setText("Terminar Presupuesto");
jButton1.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jButton1MouseClicked(evt);
    }
});

jButton2.setText("Asignar Presupuesto");
jButton2.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jButton2MouseClicked(evt);
    }
});

```



```

jTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

        },
        new String [] {
            "AÑO", "MES", "CONCEPTO", "PROVEDOR", "MONTO"
        }
    ));
jScrollPane1.setViewportView(jTable1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);
this.setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
            .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 80,
                        javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(GastoTotal, javax.swing.GroupLayout.PREFERRED_SIZE,
                        130, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                    .addComponent(jButton2)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
                        javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jButton1)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 119,
                                javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGap(18, 18, 18)
                            .addComponent(Concepto, javax.swing.GroupLayout.PREFERRED_SIZE,
                                180, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                            .addComponent(Proveedor, javax.swing.GroupLayout.PREFERRED_SIZE,
                                150, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                            .addComponent(Monto, javax.swing.GroupLayout.PREFERRED_SIZE, 73,
                                javax.swing.GroupLayout.PREFERRED_SIZE)
                        )
                    )
                )
            )
        )
    );

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(Guardar))
    .addGroup(layout.createSequentialGroup())
    .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE,
140, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(Presupuesto,
javax.swing.GroupLayout.PREFERRED_SIZE, 126, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE,
64, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(Restante, javax.swing.GroupLayout.PREFERRED_SIZE,
135, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(40, 40, 40)
    .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE,
34, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(Anio))
    .addComponent(jScrollPane1))))
    .addGap(12, 12, 12)))
    .addGap(18, 18, 18))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup())
    .addComponent(jLabel1)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(Anio, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel2)
    .addComponent(Presupuesto, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel3)
    .addComponent(jLabel5)
    .addComponent(Restante, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(Mes, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(Concepto, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(Provedor, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(Monto, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(Guardar))
        .addGap(18, 18, 18)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 320,
Short.MAX_VALUE)
        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel4)
        .addComponent(GastoTotal, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jButton1)
        .addComponent(jButton2))
        .addContainerGap()
    );
} // </editor-fold>

```

```

private void cargarRegistrosPorAnio() {
    String anio = Anio.getText().trim();
    if (anio.isEmpty()) {
        javax.swing.JOptionPane.showMessageDialog(this, "Ingrese un año válido.", "Advertencia",
javax.swing.JOptionPane.WARNING_MESSAGE);
        return;
    }
}

```

```

    String query = "SELECT * FROM gastos WHERE anio = ?"; // Reemplaza "gastos" con el
nombre real de tu tabla
    try (Connection con = Conexion.conectar(); PreparedStatement pst =
con.prepareStatement(query)) {

```

```

        pst.setString(1, anio);
        try (ResultSet rs = pst.executeQuery()) {
            // Limpia la tabla antes de agregar nuevos datos
            javax.swing.table.DefaultTableModel model = (javax.swing.table.DefaultTableModel)
jTable1.getModel();
            model.setRowCount(0);
            double totalMonto = 0;

```

```

            while (rs.next()) {
                String mes = rs.getString("mes");
                String concepto = rs.getString("concepto");
                String proveedor = rs.getString("proveedor");
                double monto = rs.getDouble("monto");
                model.addRow(new Object[]{anio, mes, concepto, proveedor, monto});
                totalMonto += monto;
            }

```

```

GastoTotal.setText(String.format("%.2f", totalMonto));
System.out.println(totalMonto);

```

```

try {
    double presupuesto = Double.parseDouble(Presupuesto.getText().trim());

```



```

        System.out.println(presupuesto);
        double nuevoRestante = presupuesto - totalMonto;
        System.out.println(nuevoRestante);
        Restante.setText(String.format("%.2f", nuevoRestante));
    } catch (NumberFormatException e) {
        javax.swing.JOptionPane.showMessageDialog(this, "Error al calcular el restante: " +
e.getMessage(), "Error", javax.swing.JOptionPane.ERROR_MESSAGE);
    }
}
} catch (SQLException e) {
    e.printStackTrace();
    javax.swing.JOptionPane.showMessageDialog(this, "Error al cargar los registros: " +
e.getMessage(), "Error", javax.swing.JOptionPane.ERROR_MESSAGE);
}
}

private void cargarUltimosRegistros() {
    Connection con = Conexion.conectar();
    String query = "SELECT * FROM presupuesto ORDER BY anio DESC LIMIT 1";

    try (PreparedStatement pst = con.prepareStatement(query); ResultSet rs = pst.executeQuery())
    {

        if (rs.next()) {
            String presupuesto = rs.getString("presupuesto").trim();
            String anio = rs.getString("anio").trim();
            Presupuesto.setText(presupuesto);
            Anio.setText(anio);
            Presupuesto.setEditable(false);
            Anio.setEditable(false);
            Restante.setEditable(false);
            GastoTotal.setEditable(false);
            String gasto = GastoTotal.getText().trim();

            try {
                // Convertir valores a números
                double dineroDouble = Double.parseDouble(presupuesto);
                double gastoDouble = Double.parseDouble(gasto);

                // Verificar que los valores sean válidos
                if (gastoDouble > dineroDouble) {
                    JOptionPane.showMessageDialog(this, "El gasto no puede ser mayor que el
presupuesto.", "Error", JOptionPane.WARNING_MESSAGE);
                    Restante.setText("0");
                    return;
                }

                // Calcular y mostrar el restante
                double restante = dineroDouble - gastoDouble;
                Restante.setText(String.valueOf(restante));
                cargarRegistrosPorAnio();
            } catch (NumberFormatException nfe) {

```

```

        Restante.setText("0");
    }
} else {

}

} catch (Exception e) {
    JOptionPane.showMessageDialog(this, "Error al cargar los últimos registros: " +
e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
}
}

private void jButton2MouseClicked(java.awt.event.MouseEvent evt) {
    String dinero = Presupuesto.getText().trim();
    String anio = Anio.getText().trim();
    String gasto = GastoTotal.getText().trim();

// Validar entradas
    if (dinero.isEmpty() || anio.isEmpty() || gasto.isEmpty()) {
        JOptionPane.showMessageDialog(this, "Por favor, complete todos los campos.", "Error",
JOptionPane.WARNING_MESSAGE);
        return;
    }

    try {
        // Convertir valores a números
        double dineroDouble = Double.parseDouble(dinero);
        double gastoDouble = Double.parseDouble(gasto);

        // Verificar que los valores sean válidos
        if (gastoDouble > dineroDouble) {
            JOptionPane.showMessageDialog(this, "El gasto no puede ser mayor que el
presupuesto.", "Error", JOptionPane.WARNING_MESSAGE);
            return;
        }

        // Conexión a la base de datos
        try (Connection con = Conexion.conectar()) {
            // Insertar el nuevo presupuesto
            String insertSql = "INSERT INTO presupuesto (presupuesto, anio) VALUES (?, ?)";
            PreparedStatement insertStatement = con.prepareStatement(insertSql);
            insertStatement.setDouble(1, dineroDouble);
            insertStatement.setString(2, anio);
            insertStatement.executeUpdate();

            // Recuperar el último presupuesto agregado
            String querySql = "SELECT * FROM presupuesto ORDER BY anio DESC LIMIT 1";
            PreparedStatement queryStatement = con.prepareStatement(querySql);
            ResultSet resultSet = queryStatement.executeQuery();

            if (resultSet.next()) {

```



```

        // Mostrar los valores recuperados en los campos correspondientes
        Presupuesto.setText(String.valueOf(resultSet.getDouble("presupuesto")));
        Anio.setText(resultSet.getString("anio"));

        // Calcular y mostrar el restante
        double restante = resultSet.getDouble("presupuesto") - gastoDouble;
        Restante.setText(String.valueOf(restante));
    } else {
        JOptionPane.showMessageDialog(this, "No se encontraron registros recientes en la
base de datos.", "Información", JOptionPane.INFORMATION_MESSAGE);
    }

    JOptionPane.showMessageDialog(this, "Presupuesto registrado exitosamente y valores
cargados.");
}
} catch (NumberFormatException e) {
    JOptionPane.showMessageDialog(this, "Por favor, ingrese valores numéricos válidos para el
presupuesto y el gasto.", "Error", JOptionPane.ERROR_MESSAGE);
} catch (SQLException ex) {
    Logger.getLogger(getClass().getName()).log(Level.SEVERE, "Error al registrar o recuperar
el presupuesto", ex);
    JOptionPane.showMessageDialog(this, "Error al registrar o recuperar el presupuesto: " +
ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
}

}

private void GuardarMouseClicked(java.awt.event.MouseEvent evt) {
    String anio = Anio.getText().trim();
    String mes = (String) Mes.getSelectedItemAt();
    String concepto = Concepto.getText().trim();
    String proveedor = Proveedor.getText().trim();
    String monto = Monto.getText().trim();

    // Validar que los campos no estén vacíos
    if (anio.isEmpty() || mes.equals("----") || concepto.isEmpty() || proveedor.isEmpty() ||
monto.isEmpty()) {
        JOptionPane.showMessageDialog(this, "Todos los campos son obligatorios.", "Advertencia",
JOptionPane.WARNING_MESSAGE);
        return;
    }

    // Validar que el año y el monto sean números válidos
    try {
        int anioInt = Integer.parseInt(anio); // Validar formato numérico del año
        double montoDouble = Double.parseDouble(monto); // Validar formato numérico del monto

        if (anioInt < 0 || montoDouble < 0) {
            JOptionPane.showMessageDialog(this, "El año y el monto deben ser valores positivos.",
"Advertencia", JOptionPane.WARNING_MESSAGE);
            return;
        }
    }
}

```



```

// Convertir el presupuesto a double
double presupuestoDouble = Double.parseDouble(Presupuesto.getText().trim());

// Calcular el restante
double restante = presupuestoDouble - total;

// Verificar que el monto no sea mayor que el restante
if (montoDouble > restante) {
    JOptionPane.showMessageDialog(this, "El monto no puede ser mayor al restante del presupuesto.", "Advertencia", JOptionPane.WARNING_MESSAGE);
    return;
}

// Guardar en la base de datos
try (Connection con = Conexion.conectar()) {
    String sql = "INSERT INTO gastos (anio, mes, concepto, proveedor, monto) VALUES (?, ?, ?, ?, ?)";
    PreparedStatement statement = con.prepareStatement(sql);
    statement.setInt(1, anioInt);
    statement.setString(2, mes);
    statement.setString(3, concepto);
    statement.setString(4, proveedor);
    statement.setDouble(5, montoDouble);
    statement.executeUpdate();

    total = total + montoDouble;
    GastoTotal.setText(String.valueOf(total));
    restante = presupuestoDouble - total;
    Restante.setText(String.valueOf(restante));
    JOptionPane.showMessageDialog(this, "Registro guardado exitosamente.", "Información",
JOptionPane.INFORMATION_MESSAGE);
    cargarRegistrosPorAnio();
    Mes.setSelectedIndex(0);
    Concepto.setText("Concepto");
    Proveedor.setText("Proveedor");
    Monto.setText("Monto");

} catch (SQLException ex) {
    JOptionPane.showMessageDialog(this, "Error al guardar los datos: " + ex.getMessage(),
"Error", JOptionPane.ERROR_MESSAGE);
}
} catch (NumberFormatException e) {
    JOptionPane.showMessageDialog(this, "El año y el monto deben ser números válidos.",
"Error", JOptionPane.WARNING_MESSAGE);
}
}

private void ConceptoMouseClicked(java.awt.event.MouseEvent evt) {
    Concepto.setText("");
}

```



```

private void ProvedorMouseClicked(java.awt.event.MouseEvent evt) {
    Provedor.setText("");
}

private void MontoMouseClicked(java.awt.event.MouseEvent evt) {
    Monto.setText("");
}

private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
    Presupuesto.setText("");
    Anio.setText("");
    Restante.setText("");
    GastoTotal.setText("0.0");
    Presupuesto.setEditable(true);
    Anio.setEditable(true);
    javax.swing.table.DefaultTableModel model = (javax.swing.table.DefaultTableModel)
jTable1.getModel();
    model.setRowCount(0);
}

// Variables declaration - do not modify
private javax.swing.JTextField Anio;
private javax.swing.JTextField Concepto;
private javax.swing.JTextField GastoTotal;
private javax.swing.JButton Guardar;
private javax.swing.JComboBox<String> Mes;
private javax.swing.JTextField Monto;
private javax.swing.JTextField Presupuesto;
private javax.swing.JTextField Provedor;
private javax.swing.JTextField Restante;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
// End of variables declaration
}

```

Editor.java

```

package com.Editor;

import com.admin.*;
import com.frame.Inicio;
import java.awt.BorderLayout;
import java.awt.Color;

```



```

import javax.swing.border.Border;

/**
 *
 * @author wicho
 */
public class Editor extends javax.swing.JFrame {
    int xMouse, yMouse;

    private String username;
    /**
     * Creates new form Home
     */
    public Editor(String username,String id) {
        initComponents();
        this.username = username;
        EditorAnimal pruebaPanel = new EditorAnimal(username,id);
        ventanas.removeAll();
        ventanas.setLayout(new BorderLayout());
        pruebaPanel.setSize(ventanas.getWidth(), ventanas.getHeight());
        ventanas.add(pruebaPanel);
        ventanas.revalidate();
        ventanas.repaint();
        pruebaPanel.setVisible(true);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        cierre = new javax.swing.JPanel();
        cierre = new javax.swing.JLabel();
        jPanel2 = new javax.swing.JPanel();
        ventanas = new javax.swing.JPanel();
        menu = new javax.swing.JPanel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel1 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setUndecorated(true);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        cierre.setBackground(new java.awt.Color(0, 0, 0));
        cierre.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                cierreMouseClicked(evt);
            }
        });
    }

```

```

    }
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        cierreMouseEntered(evt);
    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        cierreMouseExited(evt);
    }
});

cierre.setFont(new java.awt.Font("Roboto Medium", 0, 18)); // NOI18N
cierre.setForeground(new java.awt.Color(255, 255, 255));
cierre.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
cierre.setText("X");

javax.swing.GroupLayout cierreLayout = new javax.swing.GroupLayout(cierre);
cierre.setLayout(cierreLayout);
cierreLayout.setHorizontalGroup(
    cierreLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
cierreLayout.createSequentialGroup()
            .addComponent(cierre, javax.swing.GroupLayout.DEFAULT_SIZE, 24,
Short.MAX_VALUE)
            .addGap())
        );
cierreLayout.setVerticalGroup(
    cierreLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
cierreLayout.createSequentialGroup()
            .addGap(0, 0, Short.MAX_VALUE)
            .addComponent(cierre, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE))
        );

getContentPane().add(cierre, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 30, -
1));

jPanel2.setBackground(new java.awt.Color(0, 0, 0));

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(
jPanel2Layout.createSequentialGroup()
            .addGap(0, 840, Short.MAX_VALUE)
            )
        );
jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(
jPanel2Layout.createSequentialGroup()
            .addGap(0, 31, Short.MAX_VALUE)
            )
        );

getContentPane().add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 0, 840,
31));

```

```

ventanas.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));

javax.swing.GroupLayout ventanasLayout = new javax.swing.GroupLayout(ventanas);
ventanas.setLayout(ventanasLayout);
ventanasLayout.setHorizontalGroup(
    ventanasLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 688, Short.MAX_VALUE)
);
ventanasLayout.setVerticalGroup(
    ventanasLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 468, Short.MAX_VALUE)
);

getContentPane().add(ventanas, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 30,
690, 470));

menu.setBackground(new java.awt.Color(255, 153, 0));
menu.setPreferredSize(new java.awt.Dimension(870, 500));
menu.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO.jpg"))); //
NOI18N
menu.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 360, -1, 110));

jLabel3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO-
removebg-preview.png"))); // NOI18N
menu.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, -1));

jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/campo (1).jpg"))); //
NOI18N
menu.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, 470));

getContentPane().add(menu, new org.netbeans.lib.awtextra.AbsoluteConstraints(690, 30, 180,
470));

pack();
setLocationRelativeTo(null);
} // </editor-fold>

private void cierreMouseClicked(java.awt.event.MouseEvent evt) {
    cierre.setBackground(Color.red);
}

private void cierreMouseExited(java.awt.event.MouseEvent evt) {
    cierre.setBackground(Color.black);
}

private void cierreMouseClicked(java.awt.event.MouseEvent evt) {
    Inicio inicioWindow = new Inicio(username);
    inicioWindow.setVisible(true);
    this.dispose();
}

```



```

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(() -> {

    });

}

// Variables declaration - do not modify
private javax.swing.JLabel cierre;
private javax.swing.JPanel cierree;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel menu;
private javax.swing.JPanel ventanas;
// End of variables declaration

private void setBorder(Border createEmptyBorder) {
    throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
}
}

```

Precios.java

```
package com.Precios;
```

```

import com.conexion.Conexion;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import javax.swing.JOptionPane;

```

```

/**
 *
 * @author wicho
 */
public class Precios extends javax.swing.JPanel {

    private String username;
    /**
     * Creates new form Precios
     */
    public Precios(String username) {

```



```

    initComponents();
    this.username = username;
    formWindowOpened();
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    PrecioTernero = new javax.swing.JTextField();
    PrecioJoven = new javax.swing.JTextField();
    PrecioAdulto = new javax.swing.JTextField();
    PrecioAnciano = new javax.swing.JTextField();
    Guardar = new javax.swing.JButton();

    setBackground(new java.awt.Color(255, 255, 255));

    jLabel1.setText("Precios segun la etapa");

    jLabel2.setText("Ternero o becerro ");

    jLabel3.setText("Joven o recría");

    jLabel4.setText("Adulto");

    jLabel5.setText("Anciano");

    Guardar.setText("Guardar");
    Guardar.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            GuardarMouseClicked(evt);
        }
    });

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);
    this.setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel1)
                .addComponent(jLabel2)
                .addComponent(jLabel3)
                .addComponent(jLabel4)
                .addComponent(jLabel5)
                .addComponent(PrecioTernero)
                .addComponent(PrecioJoven)
                .addComponent(PrecioAdulto)
                .addComponent(PrecioAnciano)
                .addComponent(Guardar)
            )
            .addContainerGap(43, true)
        )
    );
}

```

```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
    .addComponent(jLabel2,      javax.swing.GroupLayout.DEFAULT_SIZE,      118,
Short.MAX_VALUE)
    .addComponent(jLabel3,      javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(jLabel4,      javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(jLabel5,      javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    .addGap(35, 35, 35)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
    .addComponent(jLabel1,      javax.swing.GroupLayout.DEFAULT_SIZE,      180,
Short.MAX_VALUE)
    .addComponent(PrecioTernero)
    .addComponent(PrecioJoven)
    .addComponent(PrecioAdulto)
    .addComponent(PrecioAnciano)))
    .addGroup(layout.createSequentialGroup())
    .addGap(244, 244, 244)
    .addComponent(Guardar,      javax.swing.GroupLayout.PREFERRED_SIZE,      103,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addContainerGap(314, Short.MAX_VALUE))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup())
    .addGap(14, 14, 14)
    .addComponent(jLabel1)
    .addGap(48, 48, 48)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabel2)
    .addComponent(PrecioTernero,      javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabel3)
    .addComponent(PrecioJoven,      javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabel4)
    .addComponent(PrecioAdulto,      javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabel5)
    .addComponent(PrecioAnciano,      javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,      183,
Short.MAX_VALUE)

```

```

        .addComponent(Guardar)
        .addGap(68, 68, 68))
    );
} // </editor-fold>

private void formWindowOpened() {
    try (Connection con = Conexion.conectar()) {
        if (con != null) {
            String query = "SELECT ternero, joven, adulto, anciano FROM Precios";
            try (PreparedStatement stmt = con.prepareStatement(query);
                ResultSet rs = stmt.executeQuery()) {

                if (rs.next()) {
                    PrecioTernero.setText(String.valueOf(rs.getDouble("ternero")));
                    PrecioJoven.setText(String.valueOf(rs.getDouble("joven")));
                    PrecioAdulto.setText(String.valueOf(rs.getDouble("adulto")));
                    PrecioAnciano.setText(String.valueOf(rs.getDouble("anciano")));
                } else {
                    PrecioTernero.setText("");
                    PrecioJoven.setText("");
                    PrecioAdulto.setText("");
                    PrecioAnciano.setText("");
                }
            }
        }
    } catch (SQLException e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(this, "Error al cargar los precios.");
    }
}

private void GuardarMouseClicked(java.awt.event.MouseEvent evt) {
    try (Connection con = Conexion.conectar()) {
        if (con != null) {
            String checkSql = "SELECT COUNT(*) FROM Precios";
            try (PreparedStatement checkStmt = con.prepareStatement(checkSql);
                ResultSet rs = checkStmt.executeQuery()) {

                rs.next();
                int count = rs.getInt(1);
                if (count > 0) {

                    String updateSql = "UPDATE Precios SET ternero = ?, joven = ?, adulto = ?, anciano =
?";

                    try (PreparedStatement updateStmt = con.prepareStatement(updateSql)) {
                        updateStmt.setDouble(1, Double.parseDouble(PrecioTernero.getText()));
                        updateStmt.setDouble(2, Double.parseDouble(PrecioJoven.getText()));
                        updateStmt.setDouble(3, Double.parseDouble(PrecioAdulto.getText()));
                        updateStmt.setDouble(4, Double.parseDouble(PrecioAnciano.getText()));
                        updateStmt.executeUpdate();
                        JOptionPane.showMessageDialog(this, "Precios actualizados exitosamente");
                    }
                }
            }
        }
    }
}

```

```

    }
    } else {

        String insertSql = "INSERT INTO Precios (ternero, joven, adulto, anciano) VALUES (?,
?, ?, ?)";

        try (PreparedStatement insertStmt = con.prepareStatement(insertSql)) {
            insertStmt.setDouble(1, Double.parseDouble(PrecioTernero.getText()));
            insertStmt.setDouble(2, Double.parseDouble(PrecioJoven.getText()));
            insertStmt.setDouble(3, Double.parseDouble(PrecioAdulto.getText()));
            insertStmt.setDouble(4, Double.parseDouble(PrecioAnciano.getText()));
            insertStmt.executeUpdate();
            JOptionPane.showMessageDialog(this, "Precios guardados exitosamente");
        }
    }
}

String updateAnimalesSql = "UPDATE animales " +
    "SET precio = " +
    "CASE " +
    "WHEN etapa = 'Ternero' THEN (SELECT ternero FROM Precios) * peso "
+
    "WHEN etapa = 'Joven' THEN (SELECT joven FROM Precios) * peso " +
    "WHEN etapa = 'Adulto' THEN (SELECT adulto FROM Precios) * peso " +
+
    "WHEN etapa = 'Anciano' THEN (SELECT anciano FROM Precios) * peso "
+
    "ELSE precio END";
try (PreparedStatement updateAnimalesStmt = con.prepareStatement(updateAnimalesSql))
{
    int rowsUpdated = updateAnimalesStmt.executeUpdate();
    JOptionPane.showMessageDialog(this, "Se actualizaron los precios de " + rowsUpdated +
" animales");
    System.out.println("Nombre de usuario: " + username);
    String insertMovimiento = "INSERT INTO movimientos (fecha, hora, usuario,privilegio,
movimiento, id) VALUES (?, ?, ?, ?, ?, ?)";
    try (PreparedStatement movStmt = con.prepareStatement(insertMovimiento)) {
        LocalDateTime fechaHoraActual = LocalDateTime.now();
        String fecha = fechaHoraActual.toLocalDate().toString();
        String
                                hora
                                =
fechaHoraActual.toLocalTime().format(DateTimeFormatter.ofPattern("HH:mm:ss"));

        movStmt.setString(1, fecha);
        movStmt.setString(2, hora);
        movStmt.setString(3, username);
        movStmt.setString(4, "Administrador");
        movStmt.setString(5, "Actualizacion Precio");
        movStmt.setString(6, "No Aplica");

        movStmt.executeUpdate();
    }
}
}
} catch (SQLException e) {

```



```

        e.printStackTrace();
        JOptionPane.showMessageDialog(this, "Error al guardar o actualizar los datos");
    } catch (NumberFormatException e) {
        JOptionPane.showMessageDialog(this, "Por favor ingresa valores numéricos válidos");
    }
}

// Variables declaration - do not modify
private javax.swing.JButton Guardar;
private javax.swing.JTextField PrecioAdulto;
private javax.swing.JTextField PrecioAnciano;
private javax.swing.JTextField PrecioJoven;
private javax.swing.JTextField PrecioTernero;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
// End of variables declaration
}

```

PreciosV.java

```

package com.Precios;

import com.frame.Inicio;
import java.awt.BorderLayout;
import java.awt.Color;
import javax.swing.border.Border;

/**
 *
 * @author wicho
 */
public class PreciosV extends javax.swing.JFrame {

    int xMouse, yMouse;
    private String username;

    /**
     * Creates new form Home
     */
    public PreciosV(String username) {
        initComponents();
        this.username = username;
        System.out.println(username);

        Precios pruebaPanel = new Precios(username);
        ventanas.removeAll();
        ventanas.setLayout(new BorderLayout());
    }

```



```

pruebaPanel.setSize(ventanas.getWidth(), ventanas.getHeight());
ventanas.add(pruebaPanel);
ventanas.revalidate();
ventanas.repaint();
pruebaPanel.setVisible(true);
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    cierre = new javax.swing.JPanel();
    cierre = new javax.swing.JLabel();
    jPanel2 = new javax.swing.JPanel();
    ventanas = new javax.swing.JPanel();
    menu = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel1 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setUndecorated(true);
    getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    cierre.setBackground(new java.awt.Color(0, 0, 0));
    cierre.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            cierreMouseClicked(evt);
        }
        public void mouseEntered(java.awt.event.MouseEvent evt) {
            cierreMouseEntered(evt);
        }
        public void mouseExited(java.awt.event.MouseEvent evt) {
            cierreMouseExited(evt);
        }
    });

    cierre.setFont(new java.awt.Font("Roboto Medium", 0, 18)); // NOI18N
    cierre.setForeground(new java.awt.Color(255, 255, 255));
    cierre.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    cierre.setText("X");

    javax.swing.GroupLayout cierreLayout = new javax.swing.GroupLayout(cierre);
    cierre.setLayout(cierreLayout);
    cierreLayout.setHorizontalGroup(
        cierreLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
cierreeLayout.createSequentialGroup()
        .addComponent(cierre, javax.swing.GroupLayout.DEFAULT_SIZE, 24,
Short.MAX_VALUE)
        .addContainerGap())
    );
    cierreeLayout.setVerticalGroup(
        cierreeLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
cierreeLayout.createSequentialGroup()
        .addGap(0, 0, Short.MAX_VALUE)
        .addComponent(cierre, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE))
    );

    getContentPane().add(cierree, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 30, -
1));

    jPanel2.setBackground(new java.awt.Color(0, 0, 0));

    javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
    jPanel2.setLayout(jPanel2Layout);
    jPanel2Layout.setHorizontalGroup(
        jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 840, Short.MAX_VALUE)
    );
    jPanel2Layout.setVerticalGroup(
        jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 31, Short.MAX_VALUE)
    );

    getContentPane().add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 0, 840,
31));

    ventanas.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));

    javax.swing.GroupLayout ventanasLayout = new javax.swing.GroupLayout(ventanas);
    ventanas.setLayout(ventanasLayout);
    ventanasLayout.setHorizontalGroup(
        ventanasLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 688, Short.MAX_VALUE)
    );
    ventanasLayout.setVerticalGroup(
        ventanasLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 468, Short.MAX_VALUE)
    );

    getContentPane().add(ventanas, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 30,
690, 470));

    menu.setBackground(new java.awt.Color(255, 153, 0));
    menu.setPreferredSize(new java.awt.Dimension(870, 500));

```

```

        menu.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabel2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO.jpg"))); //
NOI18N
        menu.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 360, -1, 110));

        jLabel3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO-
removebg-preview.png"))); // NOI18N
        menu.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, -1));

        jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/campo (1).jpg"))); //
NOI18N
        menu.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, 470));

        getContentPane().add(menu, new org.netbeans.lib.awtextra.AbsoluteConstraints(690, 30, 180,
470));

        pack();
        setLocationRelativeTo(null);
    } // </editor-fold>

    private void cierreMouseEntered(java.awt.event.MouseEvent evt) {
        cierre.setBackground(Color.red);
    }

    private void cierreMouseExited(java.awt.event.MouseEvent evt) {
        cierre.setBackground(Color.black);
    }

    private void cierreMouseClicked(java.awt.event.MouseEvent evt) {
        Inicio inicioWindow = new Inicio(username);
        inicioWindow.setVisible(true);
        this.dispose();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(() -> {
            new PreciosV("Usuario").setVisible(true);
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JLabel cierre;
    private javax.swing.JPanel cierree;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JPanel jPanel2;

```



```

private javax.swing.JPanel menu;
private javax.swing.JPanel ventanas;
// End of variables declaration

private void setBorder(Border createEmptyBorder) {
    throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
}
}

```

VerAnimal.java

```
package com.VerMas;
```

```

import com.conexion.Conexion;
import com.toedter.calendar.JDateChooser;
import java.awt.Image;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;

```

```
/**
```

```
*
```

```
* @author wicho
```

```
*/
```

```
public class VerAnimal extends javax.swing.JPanel {
```

```

    private String idAnimal;
    private String id;
    private JDateChooser jDateChooser;
    private JDateChooser jDateChooser2;
    private String username;

```

```

    public VerAnimal(String username, String id) {
        this.idAnimal = id;
        this.username = username;
        initComponents();
        cargarDatos(idAnimal);
        System.out.println(idAnimal);
    }

```

```

    private void cargarDatos(String id) {
        Connection con = null;
        try {
            con = Conexion.conectar();
            String query = "SELECT * FROM animales WHERE id = ?";
            PreparedStatement stmt = con.prepareStatement(query);

```

```

stmt.setString(1, id);

ResultSet rs = stmt.executeQuery();
if (rs.next()) {

    jTextField1.setText(rs.getString("id"));
    jTextField17.setText(rs.getString("raza"));
    jTextField3.setText(rs.getString("peso"));
    jTextField9.setText(rs.getString("sexo"));
    jTextField4.setText(rs.getString("numero_economico"));
    jTextField10.setText(rs.getString("salud"));
    jTextField11.setText(rs.getString("reproductivo"));
    jTextField12.setText(rs.getString("etapa"));
    jTextField13.setText(rs.getString("corporal"));
    jTextField15.setText(rs.getString("alimenticio"));
    jTextField14.setText(rs.getString("sanitario"));
    jTextField16.setText(rs.getString("modo_adquisicion"));
    jTextField2.setText(rs.getString("peso_adquirido"));
    jTextField5.setText(rs.getString("peso_destete"));
    jTextField8.setText(rs.getString("raza_padre"));
    jTextField18.setText(rs.getString("raza_madre"));
    jTextField6.setText(rs.getString("fecha_adquisicion"));
    jTextField7.setText(rs.getString("fecha_destete"));
    jTextField19.setText(rs.getString("precio"));

    // Bloquear todos los campos de entrada donde se muestran los datos
    jTextField1.setEditable(false);
    jTextField2.setEditable(false);
    jTextField3.setEditable(false);
    jTextField4.setEditable(false);
    jTextField5.setEditable(false);
    jTextField6.setEditable(false);
    jTextField7.setEditable(false);
    jTextField8.setEditable(false);
    jTextField9.setEditable(false);
    jTextField10.setEditable(false);
    jTextField11.setEditable(false);
    jTextField12.setEditable(false);
    jTextField13.setEditable(false);
    jTextField14.setEditable(false);
    jTextField15.setEditable(false);
    jTextField16.setEditable(false);
    jTextField17.setEditable(false);
    jTextField18.setEditable(false);
    jTextField19.setEditable(false);

    byte[] imageBytes = rs.getBytes("imagen");
    if (imageBytes != null) {
        // Convertir los bytes de la imagen en una imagen
        ImageIcon imagenIcon = new ImageIcon(imageBytes);
        Image image = imagenIcon.getImage().getScaledInstance(150, 150,
Image.SCALE_SMOOTH);

```

```

        imagen.setIcon(new ImageIcon(image));
    } else {
        imagen.setIcon(null);
    }
}

rs.close();
stmt.close();
} catch (SQLException e) {
    e.printStackTrace();
    JOptionPane.showMessageDialog(this, "Error al cargar los datos del animal.");
} finally {
    try {
        if (con != null) {
            con.close(); // Cerrar la conexión
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
}
}

```

/**

* This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

```

jLabel2 = new javax.swing.JLabel();
imagen = new javax.swing.JLabel();
ldtxt = new javax.swing.JLabel();
FechaAdqTxt = new javax.swing.JLabel();
RazaTxt = new javax.swing.JLabel();
PesoTxt = new javax.swing.JLabel();
jTextField1 = new javax.swing.JTextField();
jTextField3 = new javax.swing.JTextField();
SexoTxt = new javax.swing.JLabel();
NoTxt = new javax.swing.JLabel();
jTextField4 = new javax.swing.JTextField();
jSeparator1 = new javax.swing.JSeparator();
jLabel8 = new javax.swing.JLabel();
SaludTxt = new javax.swing.JLabel();
ReproductivoTxt = new javax.swing.JLabel();
EtapaTxt = new javax.swing.JLabel();
CorporalTxt = new javax.swing.JLabel();
AlimenticioTxt = new javax.swing.JLabel();
SanitarioTxt = new javax.swing.JLabel();
jSeparator2 = new javax.swing.JSeparator();

```

```

jLabel15 = new javax.swing.JLabel();
jLabel1 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
jTextField2 = new javax.swing.JTextField();
jLabel4 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
jTextField5 = new javax.swing.JTextField();
jLabel6 = new javax.swing.JLabel();
jLabel7 = new javax.swing.JLabel();
jTextField6 = new javax.swing.JTextField();
jTextField7 = new javax.swing.JTextField();
jTextField8 = new javax.swing.JTextField();
jTextField9 = new javax.swing.JTextField();
jTextField10 = new javax.swing.JTextField();
jTextField11 = new javax.swing.JTextField();
jTextField12 = new javax.swing.JTextField();
jTextField13 = new javax.swing.JTextField();
jTextField14 = new javax.swing.JTextField();
jTextField15 = new javax.swing.JTextField();
jTextField16 = new javax.swing.JTextField();
jTextField17 = new javax.swing.JTextField();
jTextField18 = new javax.swing.JTextField();
jTextField19 = new javax.swing.JTextField();
jLabel9 = new javax.swing.JLabel();
jButton1 = new javax.swing.JButton();

setBackground(new java.awt.Color(255, 255, 255));
setLayout(null);

jLabel2.setFont(new java.awt.Font("Roboto Medium", 0, 12)); // NOI18N
jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel2.setText("INFORMACION DEL GANADO");
add(jLabel2);
jLabel2.setBounds(0, 0, 690, 30);

imagen.setIcon(new javax.swing.ImageIcon(getClass().getResource("/iconos/añadir.png"))); //
NOI18N
imagen.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
add(imagen);
imagen.setBounds(10, 40, 130, 120);

Idtxt.setText("Numero de ID:");
add(Idtxt);
Idtxt.setBounds(180, 50, 90, 16);

FechaAdqTxt.setText("Fecha de Adquisicion:");
add(FechaAdqTxt);
FechaAdqTxt.setBounds(180, 80, 130, 16);

RazaTxt.setText("Tipo o Raza:");
add(RazaTxt);
RazaTxt.setBounds(180, 110, 80, 16);

```



```

PesoTxt.setText("Peso:");
add(PesoTxt);
PesoTxt.setBounds(180, 140, 50, 16);
add(jTextField1);
jTextField1.setBounds(270, 50, 140, 22);
add(jTextField3);
jTextField3.setBounds(230, 140, 160, 22);

SexoTxt.setText("Sexo:");
add(SexoTxt);
SexoTxt.setBounds(400, 140, 50, 16);

NoTxt.setText("No. Economico Propiedad:");
add(NoTxt);
NoTxt.setBounds(420, 50, 160, 20);
add(jTextField4);
jTextField4.setBounds(580, 50, 73, 22);

jSeparator1.setForeground(new java.awt.Color(153, 153, 153));
add(jSeparator1);
jSeparator1.setBounds(140, 180, 510, 10);

jLabel8.setText("Estados o Categorías");
add(jLabel8);
jLabel8.setBounds(10, 170, 130, 20);

SaludTxt.setText("Salud:");
add(SaludTxt);
SaludTxt.setBounds(10, 200, 60, 20);

ReproductivoTxt.setText("Reproductivo:");
add(ReproductivoTxt);
ReproductivoTxt.setBounds(220, 200, 90, 20);

EtapaTxt.setText("Etapa:");
add(EtapaTxt);
EtapaTxt.setBounds(462, 200, 50, 20);

CorporalTxt.setText("Corporal:");
add(CorporalTxt);
CorporalTxt.setBounds(10, 240, 70, 20);

AlimenticioTxt.setText("Alimenticio:");
add(AlimenticioTxt);
AlimenticioTxt.setBounds(430, 240, 80, 20);

SanitarioTxt.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
SanitarioTxt.setText("Sanitario:");
add(SanitarioTxt);
SanitarioTxt.setBounds(210, 240, 60, 20);

```

```

jSeparator2.setForeground(new java.awt.Color(153, 153, 153));
add(jSeparator2);
jSeparator2.setBounds(140, 280, 510, 10);

jLabel15.setText("Datos de Adquisicion");
add(jLabel15);
jLabel15.setBounds(10, 270, 140, 20);

jLabel1.setText("Modo de Adquisicion:");
add(jLabel1);
jLabel1.setBounds(10, 300, 140, 20);

jLabel3.setText("Peso al Adquirir:");
add(jLabel3);
jLabel3.setBounds(346, 300, 100, 20);
add(jTextField2);
jTextField2.setBounds(450, 300, 200, 22);

jLabel4.setText("Fecha al Destete:");
add(jLabel4);
jLabel4.setBounds(10, 340, 100, 20);

jLabel5.setText("Peso al Destete:");
add(jLabel5);
jLabel5.setBounds(342, 340, 100, 20);
add(jTextField5);
jTextField5.setBounds(450, 340, 200, 22);

jLabel6.setText("Raza del Padre:");
add(jLabel6);
jLabel6.setBounds(10, 380, 90, 20);

jLabel7.setText("Raza de la Madre:");
add(jLabel7);
jLabel7.setBounds(340, 380, 110, 20);
add(jTextField6);
jTextField6.setBounds(310, 80, 340, 22);
add(jTextField7);
jTextField7.setBounds(120, 340, 210, 22);
add(jTextField8);
jTextField8.setBounds(310, 110, 340, 22);
add(jTextField9);
jTextField9.setBounds(450, 140, 200, 22);
add(jTextField10);
jTextField10.setBounds(70, 200, 140, 22);
add(jTextField11);
jTextField11.setBounds(310, 200, 140, 22);
add(jTextField12);
jTextField12.setBounds(500, 200, 150, 22);
add(jTextField13);
jTextField13.setBounds(70, 240, 140, 22);
add(jTextField14);

```

```

jTextField14.setBounds(270, 240, 150, 22);
add(jTextField15);
jTextField15.setBounds(500, 240, 150, 22);
add(jTextField16);
jTextField16.setBounds(140, 300, 190, 22);
add(jTextField17);
jTextField17.setBounds(120, 380, 210, 22);
add(jTextField18);
jTextField18.setBounds(450, 380, 200, 22);
add(jTextField19);
jTextField19.setBounds(570, 10, 80, 22);

jLabel9.setText("Precio $$:");
add(jLabel9);
jLabel9.setBounds(500, 10, 60, 20);

jButton1.setText("Siguiente");
jButton1.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jButton1MouseClicked(evt);
    }
});
add(jButton1);
jButton1.setBounds(545, 430, 100, 23);
} // </editor-fold>

```

```

private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
    System.out.println("este es el id " + idAnimal);
    VerMas1 loginWindow = new VerMas1(username,idAnimal);
    System.out.println("este es el id " + idAnimal);
    loginWindow.setVisible(true);
    this.dispose();
}

```

```

// Variables declaration - do not modify
private javax.swing.JLabel AlimenticioTxt;
private javax.swing.JLabel CorporalTxt;
private javax.swing.JLabel EtapaTxt;
private javax.swing.JLabel FechaAdqTxt;
private javax.swing.JLabel Idtxt;
private javax.swing.JLabel NoTxt;
private javax.swing.JLabel PesoTxt;
private javax.swing.JLabel RazaTxt;
private javax.swing.JLabel ReproductivoTxt;
private javax.swing.JLabel SaludTxt;
private javax.swing.JLabel SanitarioTxt;
private javax.swing.JLabel SexoTxt;
private javax.swing.JLabel imagen;
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel1;

```



```

private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JSeparator jSeparator2;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField10;
private javax.swing.JTextField jTextField11;
private javax.swing.JTextField jTextField12;
private javax.swing.JTextField jTextField13;
private javax.swing.JTextField jTextField14;
private javax.swing.JTextField jTextField15;
private javax.swing.JTextField jTextField16;
private javax.swing.JTextField jTextField17;
private javax.swing.JTextField jTextField18;
private javax.swing.JTextField jTextField19;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField4;
private javax.swing.JTextField jTextField5;
private javax.swing.JTextField jTextField6;
private javax.swing.JTextField jTextField7;
private javax.swing.JTextField jTextField8;
private javax.swing.JTextField jTextField9;
// End of variables declaration

private void dispose() {
    throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
}
}

```

VerMas.java

```

package com.VerMas;
import com.admin.Ganado;
import java.awt.BorderLayout;
import java.awt.Color;

```

```

/**

```

```

 *

```

```

 * @author wicho

```

```

 */

```

```

public class VerMas extends javax.swing.JFrame {

```



```

int xMouse, yMouse;
private String username;
private String id;
/**
 * Creates new form Home
 */
public VerMas(String username, String id) {
    initComponents();
    this.username = username;
    this.id = id;

    this.username = username;
    VerAnimal verPanel = new VerAnimal(username,id);
    ventanas.removeAll();
    ventanas.setLayout(new BorderLayout());
    verPanel.setSize(ventanas.getWidth(), ventanas.getHeight());
    ventanas.add(verPanel);
    ventanas.revalidate();
    ventanas.repaint();
    verPanel.setVisible(true);
    this.dispose();
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    cierre = new javax.swing.JPanel();
    cierre = new javax.swing.JLabel();
    jPanel2 = new javax.swing.JPanel();
    ventanas = new javax.swing.JPanel();
    menu = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel1 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setUndecorated(true);
    getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    cierre.setBackground(new java.awt.Color(0, 0, 0));
    cierre.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseDragged(java.awt.event.MouseEvent evt) {
            cierreMouseDragged(evt);
        }
    });
    cierre.addMouseListener(new java.awt.event.MouseAdapter() {

```



```

        public void mouseClicked(java.awt.event.MouseEvent evt) {
            cierreMouseClicked(evt);
        }
        public void mouseEntered(java.awt.event.MouseEvent evt) {
            cierreMouseEntered(evt);
        }
        public void mouseExited(java.awt.event.MouseEvent evt) {
            cierreMouseExited(evt);
        }
        public void mousePressed(java.awt.event.MouseEvent evt) {
            cierreMousePressed(evt);
        }
    }
});

cierre.setFont(new java.awt.Font("Roboto Medium", 0, 18)); // NOI18N
cierre.setForeground(new java.awt.Color(255, 255, 255));
cierre.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
cierre.setText("X");

javax.swing.GroupLayout cierreLayout = new javax.swing.GroupLayout(cierre);
cierre.setLayout(cierreLayout);
cierreLayout.setHorizontalGroup(
    cierreLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
cierreLayout.createSequentialGroup()
            .addComponent(cierre, javax.swing.GroupLayout.DEFAULT_SIZE, 24,
Short.MAX_VALUE)
            .addGap())
        );
cierreLayout.setVerticalGroup(
    cierreLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
cierreLayout.createSequentialGroup()
            .addGap(0, 0, Short.MAX_VALUE)
            .addComponent(cierre, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE))
        );

getContentPane().add(cierre, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 30, -
1));

jPanel2.setBackground(new java.awt.Color(0, 0, 0));

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(
jPanel2Layout.createSequentialGroup()
            .addGap(0, 840, Short.MAX_VALUE)
            )
        );
jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(
jPanel2Layout.createSequentialGroup()
            .addGap(0, 31, Short.MAX_VALUE)
            )
        );

```

```

);

getContentPane().add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 0, 840,
31));

ventanas.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));

javax.swing.GroupLayout ventanasLayout = new javax.swing.GroupLayout(ventanas);
ventanas.setLayout(ventanasLayout);
ventanasLayout.setHorizontalGroup(
    ventanasLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 688, Short.MAX_VALUE)
);
ventanasLayout.setVerticalGroup(
    ventanasLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 468, Short.MAX_VALUE)
);

getContentPane().add(ventanas, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 30,
690, 470));

menu.setBackground(new java.awt.Color(255, 153, 0));
menu.setPreferredSize(new java.awt.Dimension(870, 500));
menu.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO.jpg"))); //
NOI18N
menu.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 360, -1, 110));

jLabel3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO-
removebg-preview.png"))); // NOI18N
menu.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, -1));

jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/campo (1).jpg"))); //
NOI18N
menu.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, 470));

getContentPane().add(menu, new org.netbeans.lib.awtextra.AbsoluteConstraints(690, 30, 180,
470));
pack();
setLocationRelativeTo(null);
} // </editor-fold>

private void cierreMouseEntered(java.awt.event.MouseEvent evt) {
    cierre.setBackground(Color.red);
}

private void cierreMouseExited(java.awt.event.MouseEvent evt) {
    cierre.setBackground(Color.black);
}

private void cierreMouseClicked(java.awt.event.MouseEvent evt) {

```



```

        Ganado inicioWindow = new Ganado(username); // Pasa el nombre del usuario
        inicioWindow.setVisible(true);
        this.dispose();
    }
    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(() -> {
            // La instancia de VerMas debe ser creada desde una ventana anterior donde se pasan los
            valores
            // No se debería crear desde el main de VerMas
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JLabel cierre;
    private javax.swing.JPanel cierree;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel menu;
    private javax.swing.JPanel ventanas;
    // End of variables declaration
}

```

VerMas1.java

```

package com.VerMas;
import java.awt.BorderLayout;
import java.awt.Color;

/**
 *
 * @author wicho
 */
public class VerMas1 extends javax.swing.JFrame {

    int xMouse, yMouse;
    private String username;
    private String idAnimal;
    private String id;
    /**
     * Creates new form Home
     */
    public VerMas1(String username, String id) {
        initComponents();
        this.username = username;
        this.id = id;

        this.username = username;
        VerMasAnimal verPanel = new VerMasAnimal(id);
    }

```



```

        ventanas.removeAll();
        ventanas.setLayout(new BorderLayout());
        verPanel.setSize(ventanas.getWidth(), ventanas.getHeight());
        ventanas.add(verPanel);
        ventanas.revalidate();
        ventanas.repaint();
        verPanel.setVisible(true);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        cierre = new javax.swing.JPanel();
        cierre = new javax.swing.JLabel();
        jPanel2 = new javax.swing.JPanel();
        ventanas = new javax.swing.JPanel();
        menu = new javax.swing.JPanel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel1 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setUndecorated(true);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        cierre.setBackground(new java.awt.Color(0, 0, 0));
        cierre.addMouseMotionListener(new java.awt.event.MouseMotionAdapter() {
            public void mouseDragged(java.awt.event.MouseEvent evt) {
                cierreMouseDragged(evt);
            }
        });
        cierre.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                cierreMouseClicked(evt);
            }
            public void mouseEntered(java.awt.event.MouseEvent evt) {
                cierreMouseEntered(evt);
            }
            public void mouseExited(java.awt.event.MouseEvent evt) {
                cierreMouseExited(evt);
            }
            public void mousePressed(java.awt.event.MouseEvent evt) {
                cierreMousePressed(evt);
            }
        });
    }

```



```

cierre.setFont(new java.awt.Font("Roboto Medium", 0, 18)); // NOI18N
cierre.setForeground(new java.awt.Color(255, 255, 255));
cierre.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
cierre.setText("X");

javax.swing.GroupLayout cierreLayout = new javax.swing.GroupLayout(cierre);
cierre.setLayout(cierreLayout);
cierreLayout.setHorizontalGroup(
    cierreLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
cierreLayout.createSequentialGroup()
        .addComponent(cierre, javax.swing.GroupLayout.DEFAULT_SIZE, 24,
Short.MAX_VALUE)
        .addContainerGap())
    );
cierreLayout.setVerticalGroup(
    cierreLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
cierreLayout.createSequentialGroup()
        .addGap(0, 0, Short.MAX_VALUE)
        .addComponent(cierre, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE))
    );

getContentPane().add(cierre, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 30, -
1));

jPanel2.setBackground(new java.awt.Color(0, 0, 0));

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(0, 840, Short.MAX_VALUE)
        )
    );
jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(0, 31, Short.MAX_VALUE)
        )
    );

getContentPane().add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 0, 840,
31));

ventanas.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));

javax.swing.GroupLayout ventanasLayout = new javax.swing.GroupLayout(ventanas);
ventanas.setLayout(ventanasLayout);
ventanasLayout.setHorizontalGroup(
    ventanasLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(ventanasLayout.createSequentialGroup()
            .addGap(0, 688, Short.MAX_VALUE)
        )
    );
ventanasLayout.setVerticalGroup(

```

```

        ventanasLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 468, Short.MAX_VALUE)
        );

        getContentPane().add(ventanas, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 30,
690, 470));

        menu.setBackground(new java.awt.Color(255, 153, 0));
        menu.setPreferredSize(new java.awt.Dimension(870, 500));
        menu.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabel2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO.jpg"))); //
NOI18N
        menu.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 360, -1, 110));

        jLabel3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO-
removebg-preview.png"))); // NOI18N
        menu.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, -1));

        jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/campo (1).jpg"))); //
NOI18N
        menu.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, 470));

        getContentPane().add(menu, new org.netbeans.lib.awtextra.AbsoluteConstraints(690, 30, 180,
470));

        pack();
        setLocationRelativeTo(null);
    } // </editor-fold>

    private void cierreMouseEntered(java.awt.event.MouseEvent evt) {
        cierre.setBackground(Color.red);
    }

    private void cierreMouseExited(java.awt.event.MouseEvent evt) {
        cierre.setBackground(Color.black);
    }

    private void cierreMouseClicked(java.awt.event.MouseEvent evt) {
        VerMas inicioWindow = new VerMas(username,id); // Pasa el nombre del usuario
        inicioWindow.setVisible(true);

        this.dispose();
    }

    private void cierreMousePressed(java.awt.event.MouseEvent evt) {

    }

    private void cierreMouseDragged(java.awt.event.MouseEvent evt) {

    }

```

```

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(() -> {

    }

    // Variables declaration - do not modify
    private javax.swing.JLabel cierre;
    private javax.swing.JPanel cierree;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel menu;
    private javax.swing.JPanel ventanas;
    // End of variables declaration

}

```

VerMasAnimal.java

```

package com.VerMas;

import com.conexion.Conexion;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author wicho
 */
public class VerMasAnimal extends javax.swing.JPanel {
    private String idAnimal;
    private DefaultTableModel modeloTabla;
    private int registroActual = 0;

    public VerMasAnimal(String id) {
        this.idAnimal = id;
        System.out.println(idAnimal);
        initComponents();
        configurarTabla();
    }

```



```

        cargarDatosHistorial();
    }

    private void configurarTabla() {
        modeloTabla = new DefaultTableModel(new String[]{"Fecha", "Movimiento", "Dato", "Valor Anterior", "Valor Nuevo"}, 0) {
            @Override
            public boolean isCellEditable(int row, int column) {
                return false;
            }
        };
        jTable1.setModel(modeloTabla);
        Dato.addActionListener(e -> filtrarPorDato());
    }

```

```

private void cargarDatosHistorial() {
    String query = "SELECT fecha, movimiento, Dato, valorAnterior, valorNuevo FROM historial WHERE id = ?";
    try (Connection con = Conexion.conectar();
        PreparedStatement stmt = con.prepareStatement(query)) {

        System.out.println("Conexión establecida correctamente.");

        stmt.setString(1, idAnimal);
        ID.setText(idAnimal);
        ID.setEditable(false);
        System.out.println("ID asignado: " + idAnimal);

        try (ResultSet rs = stmt.executeQuery()) {
            System.out.println("Consulta ejecutada.");
            modeloTabla.setRowCount(0);
            while (rs.next()) {
                System.out.println("Datos encontrados: " +
                    rs.getString("fecha") + ", " +
                    rs.getString("movimiento") + ", " +
                    rs.getString("Dato"));
                modeloTabla.addRow(new Object[]{
                    rs.getString("fecha"),
                    rs.getString("movimiento"),
                    rs.getString("Dato"),
                    rs.getString("valorAnterior"),
                    rs.getString("valorNuevo")
                });
            }
        }
    } catch (SQLException e) {
        System.err.println("Error al cargar datos del historial: " + e.getMessage());
        e.printStackTrace();
    }
}

```



```

private void filtrarPorDato() {
    String filtro = Dato.getSelectedItem().toString();

    if (filtro.equals("----") || filtro.isEmpty()) {
        cargarDatosHistorial();
        return;
    }

    String query = "SELECT fecha, movimiento, Dato, valorAnterior, valorNuevo FROM historial
WHERE id = ? AND Dato = ?";
    try (Connection con = Conexion.conectar();
        PreparedStatement stmt = con.prepareStatement(query)) {

        System.out.println("Conexión establecida correctamente para el filtro.");

        stmt.setString(1, idAnimal);
        stmt.setString(2, filtro);
        System.out.println("Filtrando por: " + filtro);

        try (ResultSet rs = stmt.executeQuery()) {
            System.out.println("Consulta filtrada ejecutada.");
            modeloTabla.setRowCount(0);
            while (rs.next()) {
                modeloTabla.addRow(new Object[]{
                    rs.getString("fecha"),
                    rs.getString("movimiento"),
                    rs.getString("Dato"),
                    rs.getString("valorAnterior"),
                    rs.getString("valorNuevo")
                });
            }
        }
    } catch (SQLException e) {
        System.err.println("Error al filtrar los datos del historial: " + e.getMessage());
        e.printStackTrace();
    }
}

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel2 = new javax.swing.JLabel();
    Dato = new javax.swing.JComboBox<>();
    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    ID = new javax.swing.JTextField();
    jLabel1 = new javax.swing.JLabel();

    setBackground(new java.awt.Color(255, 255, 255));

```



```

setLayout(null);

jLabel2.setFont(new java.awt.Font("Roboto Medium", 0, 12)); // NOI18N
jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel2.setText("BUSQUEDA EN EL REGISTRO");
add(jLabel2);
jLabel2.setBounds(0, 0, 470, 30);

Dato.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "---", "raza", "peso",
"sexo", "numeroEconomico", "salud", "reproductivo", "etapa", "corporal", "alimenticio", "sanitario",
"modoAdquisicion", "pesoAdquirido", "pesoDestete", "razaPadre", "razaMadre", "fechaAdquisicion",
"fechaDestete", " " }));
add(Dato);
Dato.setBounds(460, 50, 220, 22);

JTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

        },
    new String [] {
        "Fecha", "Movimiento", "Id", "Dato", "Valor Anterior", "Nuevo Valor"
    }
));
jScrollPane1.setViewportView(JTable1);

add(jScrollPane1);
jScrollPane1.setBounds(10, 90, 670, 360);

ID.setHorizontalAlignment(javax.swing.JTextField.CENTER);
add(ID);
ID.setBounds(100, 50, 170, 22);

jLabel1.setText("Id Animal");
add(jLabel1);
jLabel1.setBounds(10, 50, 80, 20);
} // </editor-fold>

// Variables declaration - do not modify
private javax.swing.JComboBox<String> Dato;
private javax.swing.JTextField ID;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
// End of variables declaration
}

```



VerAnimalClient.java

```
package com.VerMasClient;

import com.conexion.Conexion;
import com.toedter.calendar.JDateChooser;
import java.awt.Image;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;

/**
 *
 * @author wicho
 */
public class VerAnimalClient extends javax.swing.JPanel {

    private String idAnimal;

    private JDateChooser jDateChooser;
    private JDateChooser jDateChooser2;

    public VerAnimalClient(String id) {
        this.idAnimal = id;
        initComponents();
        cargarDatos(idAnimal);
        System.out.println(idAnimal);
    }

    private void cargarDatos(String id) {
        Connection con = null;
        try {
            con = Conexion.conectar();
            String query = "SELECT * FROM animales WHERE id = ?";
            PreparedStatement stmt = con.prepareStatement(query);
            stmt.setString(1, id);

            ResultSet rs = stmt.executeQuery();
            if (rs.next()) {

                jTextField1.setText(rs.getString("id"));
                jTextField17.setText(rs.getString("raza"));
                jTextField3.setText(rs.getString("peso"));
                jTextField9.setText(rs.getString("sexo"));
                jTextField4.setText(rs.getString("numero_economico"));
                jTextField10.setText(rs.getString("salud"));
                jTextField11.setText(rs.getString("reproductivo"));
            }
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(this, ex.getMessage(), "Error",
                JOptionPane.ERROR_MESSAGE);
        }
    }
}
```

```

jTextField12.setText(rs.getString("etapa"));
jTextField13.setText(rs.getString("corporal"));
jTextField15.setText(rs.getString("alimenticio"));
jTextField14.setText(rs.getString("sanitario"));
jTextField16.setText(rs.getString("modo_adquisicion"));
jTextField2.setText(rs.getString("peso_adquirido"));
jTextField5.setText(rs.getString("peso_destete"));
jTextField8.setText(rs.getString("raza_padre"));
jTextField18.setText(rs.getString("raza_madre"));
jTextField6.setText(rs.getString("fecha_adquisicion"));
jTextField7.setText(rs.getString("fecha_destete"));
jTextField19.setText(rs.getString("precio"));

```

```

// Bloquear todos los campos de entrada donde se muestran los datos

```

```

jTextField1.setEditable(false);
jTextField2.setEditable(false);
jTextField3.setEditable(false);
jTextField4.setEditable(false);
jTextField5.setEditable(false);
jTextField6.setEditable(false);
jTextField7.setEditable(false);
jTextField8.setEditable(false);
jTextField9.setEditable(false);
jTextField10.setEditable(false);
jTextField11.setEditable(false);
jTextField12.setEditable(false);
jTextField13.setEditable(false);
jTextField14.setEditable(false);
jTextField15.setEditable(false);
jTextField16.setEditable(false);
jTextField17.setEditable(false);
jTextField18.setEditable(false);
jTextField19.setEditable(false);

```

```

// Cargar la imagen desde la base de datos

```

```

byte[] imageBytes = rs.getBytes("imagen"); // Asegúrate de que la columna en la base de
datos sea tipo BLOB

```

```

if (imageBytes != null) {
    // Convertir los bytes de la imagen en una imagen
    ImagenIcon imagenIcon = new ImagenIcon(imageBytes);
    Image image = imagenIcon.getImage().getScaledInstance(150, 150,
Image.SCALE_SMOOTH); // Ajustar el tamaño
    imagen.setIcon(new ImagenIcon(image)); // Asignar la imagen al JLabel
} else {
    imagen.setIcon(null);
}
}

```

```

rs.close();
stmt.close();
} catch (SQLException e) {

```



```

        e.printStackTrace();
        JOptionPane.showMessageDialog(this, "Error al cargar los datos del animal.");
    } finally {
        try {
            if (con != null) {
                con.close(); // Cerrar la conexión
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
}

```

```
/**
```

```

 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */

```

```

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

```

```

    jLabel2 = new javax.swing.JLabel();
    imagen = new javax.swing.JLabel();
    Idtxt = new javax.swing.JLabel();
    FechaAdqTxt = new javax.swing.JLabel();
    RazaTxt = new javax.swing.JLabel();
    PesoTxt = new javax.swing.JLabel();
    jTextField1 = new javax.swing.JTextField();
    jTextField3 = new javax.swing.JTextField();
    SexoTxt = new javax.swing.JLabel();
    NoTxt = new javax.swing.JLabel();
    jTextField4 = new javax.swing.JTextField();
    jSeparator1 = new javax.swing.JSeparator();
    jLabel8 = new javax.swing.JLabel();
    SaludTxt = new javax.swing.JLabel();
    ReproductivoTxt = new javax.swing.JLabel();
    EtapaTxt = new javax.swing.JLabel();
    CorporalTxt = new javax.swing.JLabel();
    AlimenticioTxt = new javax.swing.JLabel();
    SanitarioTxt = new javax.swing.JLabel();
    jSeparator2 = new javax.swing.JSeparator();
    jLabel15 = new javax.swing.JLabel();
    jLabel1 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jTextField2 = new javax.swing.JTextField();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    jTextField5 = new javax.swing.JTextField();
    jLabel6 = new javax.swing.JLabel();
    jLabel7 = new javax.swing.JLabel();

```

```

jTextField6 = new javax.swing.JTextField();
jTextField7 = new javax.swing.JTextField();
jTextField8 = new javax.swing.JTextField();
jTextField9 = new javax.swing.JTextField();
jTextField10 = new javax.swing.JTextField();
jTextField11 = new javax.swing.JTextField();
jTextField12 = new javax.swing.JTextField();
jTextField13 = new javax.swing.JTextField();
jTextField14 = new javax.swing.JTextField();
jTextField15 = new javax.swing.JTextField();
jTextField16 = new javax.swing.JTextField();
jTextField17 = new javax.swing.JTextField();
jTextField18 = new javax.swing.JTextField();
jTextField19 = new javax.swing.JTextField();
jLabel9 = new javax.swing.JLabel();

setBackground(new java.awt.Color(255, 255, 255));
setLayout(null);

jLabel2.setFont(new java.awt.Font("Roboto Medium", 0, 12)); // NOI18N
jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel2.setText("INFORMACION DEL GANADO");
add(jLabel2);
jLabel2.setBounds(0, 0, 690, 30);

imagen.setIcon(new javax.swing.ImageIcon(getClass().getResource("/iconos/añadir.png"))); //
NOI18N
imagen.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
imagen.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        imagenMouseClicked(evt);
    }
});
add(imagen);
imagen.setBounds(10, 40, 130, 120);

Idtxt.setText("Numero de ID:");
add(Idtxt);
Idtxt.setBounds(180, 50, 90, 16);

FechaAdqTxt.setText("Fecha de Adquisicion:");
add(FechaAdqTxt);
FechaAdqTxt.setBounds(180, 80, 130, 16);

RazaTxt.setText("Tipo o Raza:");
add(RazaTxt);
RazaTxt.setBounds(180, 110, 80, 16);

PesoTxt.setText("Peso:");
add(PesoTxt);
PesoTxt.setBounds(180, 140, 50, 16);
add(jTextField1);

```



```
jTextField1.setBounds(270, 50, 140, 22);
add(jTextField3);
jTextField3.setBounds(230, 140, 160, 22);

SexoTxt.setText("Sexo:");
add(SexoTxt);
SexoTxt.setBounds(400, 140, 50, 16);

NoTxt.setText("No. Economico Propiedad:");
add(NoTxt);
NoTxt.setBounds(420, 50, 160, 20);
add(jTextField4);
jTextField4.setBounds(580, 50, 73, 22);

jSeparator1.setForeground(new java.awt.Color(153, 153, 153));
add(jSeparator1);
jSeparator1.setBounds(140, 180, 510, 10);

jLabel8.setText("Estados o Categorías");
add(jLabel8);
jLabel8.setBounds(10, 170, 130, 20);

SaludTxt.setText("Salud:");
add(SaludTxt);
SaludTxt.setBounds(10, 200, 60, 20);

ReproductivoTxt.setText("Reproductivo:");
add(ReproductivoTxt);
ReproductivoTxt.setBounds(220, 200, 90, 20);

EtapaTxt.setText("Etapa:");
add(EtapaTxt);
EtapaTxt.setBounds(462, 200, 50, 20);

CorporalTxt.setText("Corporal:");
add(CorporalTxt);
CorporalTxt.setBounds(10, 240, 70, 20);

AlimenticioTxt.setText("Alimenticio:");
add(AlimenticioTxt);
AlimenticioTxt.setBounds(430, 240, 80, 20);

SanitarioTxt.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
SanitarioTxt.setText("Sanitario:");
add(SanitarioTxt);
SanitarioTxt.setBounds(210, 240, 60, 20);

jSeparator2.setForeground(new java.awt.Color(153, 153, 153));
add(jSeparator2);
jSeparator2.setBounds(140, 280, 510, 10);

jLabel15.setText("Datos de Adquisición");
```



```
add(jLabel15);
jLabel15.setBounds(10, 270, 140, 20);

jLabel1.setText("Modo de Adquisicion:");
add(jLabel1);
jLabel1.setBounds(10, 300, 140, 20);

jLabel3.setText("Peso al Adquirir:");
add(jLabel3);
jLabel3.setBounds(346, 300, 100, 20);
add(jTextField2);
jTextField2.setBounds(450, 300, 200, 22);

jLabel4.setText("Fecha al Destete:");
add(jLabel4);
jLabel4.setBounds(10, 340, 100, 20);

jLabel5.setText("Peso al Destete:");
add(jLabel5);
jLabel5.setBounds(342, 340, 100, 20);
add(jTextField5);
jTextField5.setBounds(450, 340, 200, 22);

jLabel6.setText("Raza del Padre:");
add(jLabel6);
jLabel6.setBounds(10, 380, 90, 20);

jLabel7.setText("Raza de la Madre:");
add(jLabel7);
jLabel7.setBounds(340, 380, 110, 20);
add(jTextField6);
jTextField6.setBounds(310, 80, 340, 22);
add(jTextField7);
jTextField7.setBounds(120, 340, 210, 22);
add(jTextField8);
jTextField8.setBounds(310, 110, 340, 22);
add(jTextField9);
jTextField9.setBounds(450, 140, 200, 22);
add(jTextField10);
jTextField10.setBounds(70, 200, 140, 22);
add(jTextField11);
jTextField11.setBounds(310, 200, 140, 22);
add(jTextField12);
jTextField12.setBounds(500, 200, 150, 22);
add(jTextField13);
jTextField13.setBounds(70, 240, 140, 22);
add(jTextField14);
jTextField14.setBounds(270, 240, 150, 22);
add(jTextField15);
jTextField15.setBounds(500, 240, 150, 22);
add(jTextField16);
jTextField16.setBounds(140, 300, 190, 22);
```

```

add(jTextField17);
jTextField17.setBounds(120, 380, 210, 22);
add(jTextField18);
jTextField18.setBounds(450, 380, 200, 22);
add(jTextField19);
jTextField19.setBounds(580, 10, 64, 22);

jLabel9.setText("Precio $$:");
add(jLabel9);
jLabel9.setBounds(510, 10, 60, 20);
} // </editor-fold>

private void imagenMouseClicked(java.awt.event.MouseEvent evt) {

}

// Variables declaration - do not modify
private javax.swing.JLabel AlimenticioTxt;
private javax.swing.JLabel CorporalTxt;
private javax.swing.JLabel EtapaTxt;
private javax.swing.JLabel FechaAdqTxt;
private javax.swing.JLabel Idtxt;
private javax.swing.JLabel NoTxt;
private javax.swing.JLabel PesoTxt;
private javax.swing.JLabel RazaTxt;
private javax.swing.JLabel ReproductivoTxt;
private javax.swing.JLabel SaludTxt;
private javax.swing.JLabel SanitarioTxt;
private javax.swing.JLabel SexoTxt;
private javax.swing.JLabel imagen;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JSeparator jSeparator2;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField10;
private javax.swing.JTextField jTextField11;
private javax.swing.JTextField jTextField12;
private javax.swing.JTextField jTextField13;
private javax.swing.JTextField jTextField14;
private javax.swing.JTextField jTextField15;
private javax.swing.JTextField jTextField16;

```

```

private javax.swing.JTextField jTextField17;
private javax.swing.JTextField jTextField18;
private javax.swing.JTextField jTextField19;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField4;
private javax.swing.JTextField jTextField5;
private javax.swing.JTextField jTextField6;
private javax.swing.JTextField jTextField7;
private javax.swing.JTextField jTextField8;
private javax.swing.JTextField jTextField9;
// End of variables declaration
}

```

VerMasClient.java

```
package com.VerMasClient;
```

```

import com.VerMas.*;
import com.Editor.*;
import com.admin.*;
import com.frame.Inicio;
import com.frame.Inicio1;
import java.awt.BorderLayout;
import java.awt.Color;
import javax.swing.JInternalFrame;
import javax.swing.SwingUtilities;
import javax.swing.border.Border;
import javax.swing.plaf.basic.BasicInternalFrameUI;

```

```

/**
 *
 * @author wicho
 */
public class VerMasClient extends javax.swing.JFrame {
    int xMouse, yMouse;
    /**
     * Creates new form Home
     */
    public VerMasClient(String id) {
        initComponents();
        VerAnimalClient verPanel = new VerAnimalClient(id);
        ventanas.removeAll();
        ventanas.setLayout(new BorderLayout());
        verPanel.setSize(ventanas.getWidth(), ventanas.getHeight());
        ventanas.add(verPanel);
        ventanas.revalidate();
        ventanas.repaint();
        verPanel.setVisible(true);
    }
}

```



```

* This method is called from within the constructor to initialize the form.
* WARNING: Do NOT modify this code. The content of this method is always
* regenerated by the Form Editor.
*/
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    cierreee = new javax.swing.JPanel();
    cierre = new javax.swing.JLabel();
    jPanel2 = new javax.swing.JPanel();
    ventanas = new javax.swing.JPanel();
    menu = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel1 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setUndecorated(true);
    getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    cierreee.setBackground(new java.awt.Color(0, 0, 0));
    cierreee.addMouseMotionListener(new java.awt.event.MouseMotionAdapter() {
        public void mouseDragged(java.awt.event.MouseEvent evt) {
            cierreeeMouseDragged(evt);
        }
    });
    cierreee.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            cierreeeMouseClicked(evt);
        }
        public void mouseEntered(java.awt.event.MouseEvent evt) {
            cierreeeMouseEntered(evt);
        }
        public void mouseExited(java.awt.event.MouseEvent evt) {
            cierreeeMouseExited(evt);
        }
        public void mousePressed(java.awt.event.MouseEvent evt) {
            cierreeeMousePressed(evt);
        }
    });

    cierre.setFont(new java.awt.Font("Roboto Medium", 0, 18)); // NOI18N
    cierre.setForeground(new java.awt.Color(255, 255, 255));
    cierre.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    cierre.setText("X");

    javax.swing.GroupLayout cierreeeLayout = new javax.swing.GroupLayout(cierreee);
    cierreee.setLayout(cierreeeLayout);
    cierreeeLayout.setHorizontalGroup(
        cierreeeLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
cierreeLayout.createSequentialGroup()
        .addComponent(cierre, javax.swing.GroupLayout.DEFAULT_SIZE, 24,
Short.MAX_VALUE)
        .addContainerGap())
    );
    cierreeLayout.setVerticalGroup(
        cierreeLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
cierreeLayout.createSequentialGroup()
        .addGap(0, 0, Short.MAX_VALUE)
        .addComponent(cierre, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE))
    );

    getContentPane().add(cierree, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 30, -
1));

    jPanel2.setBackground(new java.awt.Color(0, 0, 0));

    javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
    jPanel2.setLayout(jPanel2Layout);
    jPanel2Layout.setHorizontalGroup(
        jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 840, Short.MAX_VALUE)
    );
    jPanel2Layout.setVerticalGroup(
        jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 31, Short.MAX_VALUE)
    );

    getContentPane().add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 0, 840,
31));

    ventanas.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));

    javax.swing.GroupLayout ventanasLayout = new javax.swing.GroupLayout(ventanas);
    ventanas.setLayout(ventanasLayout);
    ventanasLayout.setHorizontalGroup(
        ventanasLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 688, Short.MAX_VALUE)
    );
    ventanasLayout.setVerticalGroup(
        ventanasLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 468, Short.MAX_VALUE)
    );

    getContentPane().add(ventanas, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 30,
690, 470));

    menu.setBackground(new java.awt.Color(255, 153, 0));
    menu.setPreferredSize(new java.awt.Dimension(870, 500));

```

```

        menu.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabel2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO.jpg"))); //
NOI18N
        menu.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 360, -1, 110));

        jLabel3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO-
removebg-preview.png"))); // NOI18N
        menu.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, -1));

        jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/campo (1).jpg"))); //
NOI18N
        menu.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, 470));

        getContentPane().add(menu, new org.netbeans.lib.awtextra.AbsoluteConstraints(690, 30, 180,
470));

        pack();
        setLocationRelativeTo(null);
    } // </editor-fold>

    private void cierreMouseEntered(java.awt.event.MouseEvent evt) {
        cierre.setBackground(Color.red);
    }

    private void cierreMouseExited(java.awt.event.MouseEvent evt) {
        cierre.setBackground(Color.black);
    }

    private void cierreMouseClicked(java.awt.event.MouseEvent evt) {
        Inicio1 inicioWindow = new Inicio1();
        inicioWindow.setVisible(true);

        // Opcional: cerrar la ventana actual si lo deseas
        this.dispose();
    }

    private void cierreMousePressed(java.awt.event.MouseEvent evt) {

    }

    private void cierreMouseDragged(java.awt.event.MouseEvent evt) {

    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(() -> {
            new VerMasClient("Usuario").setVisible(true);
        });
    }

```

```

    }

    // Variables declaration - do not modify
    private javax.swing.JLabel cierre;
    private javax.swing.JPanel cierree;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel menu;
    private javax.swing.JPanel ventanas;
    // End of variables declaration

    private void setBorder(Border createEmptyBorder) {
        throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
    }
}

```

Animal.java

```
package com.admin;
```

```
import java.util.Date;
```

```
/**
```

```
*
```

```
* @author wicho
```

```
*/
```

```
public class Animal {
```

```
    private int id;
```

```
    private String numeroEconomico;
```

```
    private Date fechaAdquisicion;
```

```
    private String raza;
```

```
    private float peso;
```

```
    private String sexo;
```

```
    private String salud;
```

```
    private String reproductivo;
```

```
    private String etapa;
```

```
    private String corporal;
```

```
    private String sanitario;
```

```
    private String alimenticio;
```

```
    private byte[] imagen;
```

```
    private String precio;
```

```

    public Animal(int id, String numeroEconomico, Date fechaAdquisicion, String raza, float peso,
String sexo, String salud, String reproductivo, String etapa, String corporal, String sanitario, String
alimenticio, byte[] imagen, String precio) {

```

```
        this.id = id;
```

```
        this.numeroEconomico = numeroEconomico;
```

```
        this.fechaAdquisicion = fechaAdquisicion;
```



```

    this.raza = raza;
    this.peso = peso;
    this.sexo = sexo;
    this.salud = salud;
    this.reproductivo = reproductivo;
    this.etapa = etapa;
    this.corporal = corporal;
    this.sanitario = sanitario;
    this.alimenticio = alimenticio;
    this.imagen = imagen;
    this.precio = precio;
}

// Métodos getter para cada atributo
public int getId() {
    return id;
}

public String getNumeroEconomico() {
    return numeroEconomico;
}

public Date getFechaAdquisicion() {
    return fechaAdquisicion;
}

public String getRaza() {
    return raza;
}

public float getPeso() {
    return peso;
}

public String getSexo() {
    return sexo;
}

public String getSalud() {
    return salud;
}

public String getReproductivo() {
    return reproductivo;
}

public String getEtapa() {
    return etapa;
}

public String getCorporal() {
    return corporal;
}

```



```

    }

    public String getSanitario() {
        return sanitario;
    }

    public String getAlimenticio() {
        return alimenticio;
    }

    public byte[] getImagen() {
        return imagen;
    }

    public void setImagen(byte[] imagen) {
        this.imagen = imagen;
    }

    public String getPrecio() {
        return precio;
    }
}

```

BuscarAnimal.java

```
package com.admin;
```

```

import com.Editar.Editar;
import com.VerMas.VerMas;
import com.conexion.Conexion;
import java.awt.BorderLayout;
import java.awt.Component;
import java.awt.FlowLayout;
//import java.awt.List;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.Vector;
import javax.swing.DefaultCellEditor;
import javax.swing.JButton;

```



```

import javax.swing.JCheckBox;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableCellEditor;
import javax.swing.table.TableCellRenderer;
import java.util.List;
import java.util.ArrayList;
import java.util.Arrays;
import javax.swing.AbstractCellEditor;
import javax.swing.ImageIcon;
import javax.swing.JComboBox;
import javax.swing.JPanel;
import javax.swing.table.TableColumn;

/**
 *
 * @author wicho
 */
public class BuscarAnimal extends javax.swing.JPanel {
    private String username;
    private List<Animal> listaAnimales = new ArrayList<>();
    String id = null;
    private static VerMas verWindow = null;
    private int registroActual = 0;

    public BuscarAnimal(String username) {
        initComponents();
        this.username = username;
        System.out.println(username);
        ocultarComponentes();
        configurarEventos();
        jTextField14.setEditable(false);
        // Bloquear todos los campos de entrada donde se muestran los datos
        jTextField2.setEditable(false);
        jTextField3.setEditable(false);
        jTextField4.setEditable(false);
        jTextField5.setEditable(false);
        jTextField6.setEditable(false);
        jTextField7.setEditable(false);
        jTextField8.setEditable(false);
        jTextField9.setEditable(false);
        jTextField10.setEditable(false);
        jTextField11.setEditable(false);
        jTextField12.setEditable(false);
        jTextField13.setEditable(false);

    }

    public void buscarDatos(String query) {
        try {
            Connection con = Conexion.conectar();

```



```

Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery(query);

limpiarTabla();

double totalPrecio = 0;

// Obtener datos de la base de datos y agregarlos a la tabla
while (rs.next()) {
    String[] fila = new String[jTable1.getColumnCount() - 1]; // Resto de las columnas
    for (int i = 0; i < fila.length; i++) {
        fila[i] = rs.getString(i + 1); // Rellenar datos en las celdas
    }
    // Añadir la fila sin el botón
    ((DefaultTableModel) jTable1.getModel()).addRow(fila);

    // Agregar el botón en la columna "Acciones"
    JButton actionButton = createActionButton();
    ((DefaultTableModel) jTable1.getModel()).setValueAt(actionButton, jTable1.getRowCount() -
1, jTable1.getColumnCount() - 1);

    // Acumular el precio de cada animal
    double precio = rs.getDouble("precio");
    totalPrecio += precio;
}

jTextField14.setText(String.valueOf(totalPrecio));

TableColumn column = jTable1.getColumnModel().getColumn(jTable1.getColumnCount() - 1);
column.setCellRenderer(new ButtonRenderer());
column.setCellEditor(new ButtonEditor(new JCheckBox()));

rs.close();
stmt.close();
con.close();
} catch (SQLException e) {
    e.printStackTrace();
    JOptionPane.showMessageDialog(this, "Error al obtener datos de la base de datos.");
}
}

private JButton createActionButton() {
    JButton button = new JButton("Acción");
    button.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            // Crear opciones para el cuadro de diálogo
            Object[] options = {"Editar", "Ver más", "Eliminar"};

            // Mostrar el cuadro de diálogo de opciones
            int choice = JOptionPane.showOptionDialog(

```

```

        null,
        "Selecciona una acción:",
        "Opciones de Acción",
        JOptionPane.YES_NO_CANCEL_OPTION,
        JOptionPane.QUESTION_MESSAGE,
        null,
        options,
        options[0]
    );

    int selectedRow = jTable1.getSelectedRow();
    if (selectedRow != -1) { // Validar que haya una fila seleccionada

        int idColumnIndex = 1; // Aquí suponemos que la columna 'ID' es la primera. Si no es así, cambia
        este valor
        String id = jTable1.getValueAt(selectedRow, idColumnIndex).toString();

        // Realizar la acción basada en la elección del usuario
        switch (choice) {
            case JOptionPane.YES_OPTION: // Editar
                System.out.println("Editar seleccionado.");
                Editar editarWindow = new Editar(username,id);
                editarWindow.setVisible(true);
                break;
            case JOptionPane.NO_OPTION: // Ver más
                System.out.println("Ver más seleccionado.");
                VerMas verWindow = new VerMas(username,id);
                verWindow.setVisible(true);
                break;
            case JOptionPane.CANCEL_OPTION: // Eliminar
                System.out.println("Eliminar seleccionado.");

                // Mostrar una ventana de confirmación
                int confirmation = JOptionPane.showConfirmDialog(
                    null,
                    "¿Estás seguro de que quieres eliminar este registro?",
                    "Confirmar Eliminación",
                    JOptionPane.YES_NO_OPTION,
                    JOptionPane.WARNING_MESSAGE
                );

                if (confirmation == JOptionPane.YES_OPTION) {
                    // Si el usuario confirma, eliminar el registro
                    eliminarAnimal(id); // Llamamos al método para eliminar el animal
                } else {
                    // Si el usuario cancela, no hacemos nada
                    System.out.println("Eliminación cancelada.");
                }
                break;
            default:
                System.out.println("Acción cancelada.");
        }
    }

```



```

        break;
    }
} else {
    JOptionPane.showMessageDialog(null, "Por favor, selecciona una fila primero.");
}

}

});
return button;
}

private void eliminarAnimal(String id) {
    Connection con = null;
    try {
        con = Conexion.conectar();
        String query = "DELETE FROM animales WHERE id = ?";
        PreparedStatement stmt = con.prepareStatement(query);
        stmt.setString(1, id);

        int rowsAffected = stmt.executeUpdate();

        if (rowsAffected > 0) {
            JOptionPane.showMessageDialog(this, "Registro eliminado con éxito.");

            // Llamar a buscarDatos() con el query según la selección actual
            actualizarTablaSegunSeleccion();
        } else {
            JOptionPane.showMessageDialog(this, "No se encontró el registro para eliminar.");
        }

        stmt.close();
    } catch (SQLException e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(this, "Error al eliminar el registro.");
    } finally {
        try {
            if (con != null) {
                con.close();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
}

```

```

// Método para actualizar la tabla según la selección actual
private void actualizarTablaSegunSeleccion() {
    String selectedOption = (String) jComboBox2.getSelectedItem();
    String query = "";

    String valorSeleccionado = (String) jComboBox11.getSelectedItem();
    System.out.println(valorSeleccionado);

```

```

// Determinar la consulta inicial según la opción seleccionada en JComboBox2
switch (selectedOption) {
    case "Busqueda por Raza":
        query = "SELECT Raza, Id, Numero_economico, Fecha_adquisicion, Peso, Sexo FROM animales";
        configurarModeloTabla(new String[]{"Raza", "Id", "Numero_economico", "Fecha_adquisicion", "Peso", "Sexo"});
        break;
    case "Busqueda por Sexo":
        query = "SELECT Sexo, Id, Numero_economico, Fecha_adquisicion, Peso, Raza FROM animales";
        configurarModeloTabla(new String[]{"Sexo", "Id", "Numero_economico", "Fecha_adquisicion", "Peso", "Raza"});
        break;
    case "Busqueda por Salud":
        configurarComboBoxSalud();
        query = "SELECT Salud, Id, Numero_economico, Raza, Peso, Sexo FROM animales";
        configurarModeloTabla(new String[]{"Salud", "Id", "No.Economico", "Raza", "Peso", "Sexo"});
        break;
    case "Busqueda Reproductivo":
        configurarComboBoxReproductivo();
        query = "SELECT Reproductivo, Id, Raza, Numero_economico, Peso, Sexo FROM animales";
        configurarModeloTabla(new String[]{"Reproductivo", "Id", "No.Economico", "Raza", "Peso", "Sexo"});
        break;
    case "Busqueda por Etapa":
        configurarComboBoxEtapa();
        query = "SELECT Etapa, Id, Numero_economico, Raza, Peso, Sexo FROM animales";
        configurarModeloTabla(new String[]{"Etapa", "Id", "No.Economico", "Raza", "Peso", "Sexo"});
        break;
    case "Busqueda Corporal":
        configurarComboBoxCorporal();
        query = "SELECT Corporal, Id, Numero_economico, Raza, Peso, Sexo FROM animales";
        configurarModeloTabla(new String[]{"Corporal", "Id", "No.Economico", "Raza", "Peso", "Sexo"});
        break;
    case "Busqueda Sanitario":
        configurarComboBoxSanitario();
        query = "SELECT Sanitario, Id, Numero_economico, Raza, Peso, Sexo FROM animales";
        configurarModeloTabla(new String[]{"Sanitario", "Id", "No.Economico", "Raza", "Peso", "Sexo"});
        break;
    case "Busqueda Alimenticio":
        configurarComboBoxAlimenticio();
        query = "SELECT Alimenticio, Id, Numero_economico, Raza, Peso, Sexo FROM animales";
        configurarModeloTabla(new String[]{"Alimenticio", "Id", "No.Economico", "Raza", "Peso", "Sexo"});
        break;
    case "Modo de Adquisicion":
        configurarComboBoxModoAdquisicion();

```

```

        query = "SELECT Modo_adquisicion, Id, Numero_economico, Raza, Peso,
Fecha_adquisicion FROM animales";
        configurarModeloTabla(new String[]{"ModoAdqui", "Id", "No.Economico", "Raza", "Peso",
"FechaAdqui"});
        break;
    default:
        jTable1.setVisible(false); // Oculta la tabla si no hay opción válida seleccionada
        limpiarTabla();
        return;
    }

```

```

// Si hay un valor seleccionado en jComboBox11 y no es "----", usamos ese filtro en la consulta
if (valorSeleccionado != null && !valorSeleccionado.equals("----")) {
    // Modificar la consulta según el valor seleccionado en jComboBox11
    switch (selectedOption) {
        case "Busqueda por Raza":
            query = "SELECT Raza, Id, Numero_economico, Fecha_adquisicion, Peso, Sexo FROM
animales WHERE Raza = " + valorSeleccionado + """;
            break;
        case "Busqueda por Sexo":
            query = "SELECT Sexo, Id, Numero_economico, Fecha_adquisicion, Peso, Raza FROM
animales WHERE Sexo = " + valorSeleccionado + """;
            break;
        case "Busqueda por Salud":
            query = "SELECT Salud, Id, Numero_economico, Raza, Peso, Sexo FROM animales
WHERE Salud = " + valorSeleccionado + """;
            break;
        case "Busqueda Reproductivo":
            query = "SELECT Reproductivo, Id, Numero_economico, Raza, Peso, Sexo FROM
animales WHERE Reproductivo = " + valorSeleccionado + """;
            break;
        case "Busqueda por Etapa":
            query = "SELECT Etapa, Id, Numero_economico, Raza, Peso, Sexo FROM animales
WHERE Etapa = " + valorSeleccionado + """;
            break;
        case "Busqueda Corporal":
            query = "SELECT Corporal, Id, Numero_economico, Raza, Peso, Sexo FROM animales
WHERE Corporal = " + valorSeleccionado + """;
            break;
        case "Busqueda Sanitario":
            query = "SELECT Sanitario, Id, Numero_economico, Raza, Peso, Sexo FROM animales
WHERE Sanitario = " + valorSeleccionado + """;
            break;
        case "Busqueda Alimenticio":
            query = "SELECT Alimenticio, Id, Numero_economico, Raza, Peso, Sexo FROM animales
WHERE Alimenticio = " + valorSeleccionado + """;
            break;
        case "Modo de Adquisicion":
            query = "SELECT Modo_adquisicion, Id, Numero_economico, Raza, Peso,
Fecha_adquisicion FROM animales WHERE Modo_adquisicion = " + valorSeleccionado + """;
            break;
        default:

```



```

        limpiarTabla();
        return;
    }
}

// Ejecutar la consulta si no está vacía
if (!query.isEmpty()) {
    jComboBox11.setSelectedItem(valorSeleccionado);
    buscarDatos(query);
}
}

class ButtonRenderer extends JButton implements TableCellRenderer {
    public ButtonRenderer() {
        setOpaque(true);
    }

    @Override
    public Component getTableCellRendererComponent(JTable table, Object value, boolean
isSelected, boolean hasFocus, int row, int column) {
        if (value instanceof JButton) {
            return (JButton) value;
        }
        return this;
    }
}

class ButtonEditor extends DefaultCellEditor {
    protected JButton button;
    private String label;
    private boolean isPushed;

    public ButtonEditor(JCheckBox checkBox) {
        super(checkBox);
        button = new JButton();
        button.setOpaque(true);
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                fireEditingStopped();
            }
        });
    }

    @Override
    public Component getTableCellEditorComponent(JTable table, Object value, boolean isSelected,
int row, int column) {
        if (value instanceof JButton) {
            button = (JButton) value;
        }
        return button;
    }
}

```



```

@Override
public Object getCellEditorValue() {
    return label;
}
}

class ModeloTablaConBotones extends DefaultTableModel {
    public ModeloTablaConBotones(Object[][] data, Object[] columnNames) {
        super(data, columnNames);
    }

    @Override
    public Class<?> getColumnClass(int columnIndex) {
        if (columnIndex == getColumnCount() - 1) { // La columna "Acciones"
            return JButton.class;
        }
        return super.getColumnClass(columnIndex);
    }
}

private void configurarEventos() {
    jComboBox1.setSelectedIndex(0);

    jComboBox1.addItemListener(e -> {
        if (e.getStateChange() == ItemEvent.SELECTED) {
            String selectedOption = (String) jComboBox1.getSelectedItem();
            ocultarComponentes(); // Oculta todos los componentes al cambiar la opción

            switch (selectedOption) {
                case "Busqueda por Categoria":
                    mostrarBusquedaPorCategoria();
                    break;
                case "Busqueda por ID":
                    jTextField1.setText("");
                    mostrarBusquedaPorID();
                    break;
                case "Mostrar Registros por Orden":
                    mostrarRegistrosPorOrden();
                    break;
                default:
                    ocultarComponentes();
                    break;
            }
        }
    });

    jComboBox2.addItemListener(e -> {
        if (e.getStateChange() == ItemEvent.SELECTED) {
            String selectedOption = (String) jComboBox2.getSelectedItem();
            String query = "";

```



```

ocultarComponentesExclusivos();
//jScrollPane1.setVisible(true);
//jTable1.setVisible(true);

switch (selectedOption) {
    case "Busqueda por Raza":
        configurarComboBoxRaza();
        configurarModeloTabla(new String[]{"Raza", "Id", "No.Economico", "FechaAdq",
"Peso", "precio"});
        query = "SELECT Raza, Id, Numero_economico, Fecha_adquisicion, Peso, precio
FROM animales";
        break;
    case "Busqueda por Sexo":
        configurarComboBoxSexo();
        configurarModeloTabla(new String[]{"Sexo", "Id", "No.Economico", "FechaAdq",
"Peso", "precio"});
        query = "SELECT Sexo, Id, Numero_economico, Fecha_adquisicion, Peso, precio
FROM animales";
        break;
    case "Busqueda por Salud":
        configurarComboBoxSalud();
        configurarModeloTabla(new String[]{"Salud", "Id", "No.Economico","Raza", "Peso",
"precio"});
        query = "SELECT Salud, Id, Numero_economico, Raza, Peso, precio FROM
animales";
        break;
    case "Busqueda Reproductivo":
        configurarComboBoxReproductivo();
        configurarModeloTabla(new String[]{"Reproductivo", "Id", "Raza","No.Economico",
"Peso", "precio"});
        query = "SELECT Reproductivo, Id, Raza, Numero_economico, Peso, precio FROM
animales";
        break;
    case "Busqueda por Etapa":
        configurarComboBoxEtapa();
        configurarModeloTabla(new String[]{"Etapa", "Id", "No.Economico", "Raza", "Peso",
"precio"});
        query = "SELECT Etapa, Id, Numero_economico, Raza, Peso, precio FROM
animales";
        break;
    case "Busqueda Corporal":
        configurarComboBoxCorporal();
        configurarModeloTabla(new String[]{"Corporal", "Id", "No.Economico", "Raza",
"Peso", "precio"});
        query = "SELECT Corporal, Id, Numero_economico, Raza, Peso, precio FROM
animales";
        break;
    case "Busqueda Sanitario":
        configurarComboBoxSanitario();
        configurarModeloTabla(new String[]{"Sanitario", "Id", "No.Economico", "Raza",
"Peso", "precio"});

```

```

        query = "SELECT Sanitario, Id, Numero_economico, Raza, Peso, precio FROM
animales";
        break;
        case "Busqueda Alimenticio":
            configurarComboBoxAlimenticio();
            configurarModeloTabla(new String[]{"Alimenticio", "Id", "No.Economico", "Raza",
"Peso", "precio"});
            query = "SELECT Alimenticio, Id, Numero_economico, Raza, Peso, precio FROM
animales";
            break;
        case "Modo de Adquisicion":
            configurarComboBoxModoAdquisicion();
            configurarModeloTabla(new String[]{"ModoAdqui", "Id", "No.Economico",
"FechaAdqui", "Peso", "precio"});
            query = "SELECT Modo_adquisicion, Id, Numero_economico, Fecha_adquisicion,
Peso, precio FROM animales";
            break;
        default:
            jTable1.setVisible(false); // Oculta la tabla si no hay opción válida seleccionada
            limpiarTabla();
            return;
    }

    // Ejecuta la consulta si no está vacía
    if (!query.isEmpty()) {
        buscarDatos(query);
    }
}

});

jComboBox11.addItemListener(e -> {
    if (e.getStateChange() == ItemEvent.SELECTED && jComboBox2.getSelectedIndex() > 0) {
        String valorSeleccionado = (String) jComboBox11.getSelectedItem();
        String selectedOption = (String) jComboBox2.getSelectedItem();

        String query = "";
        switch (selectedOption) {
            case "Busqueda por Raza":
                query = "SELECT Raza, Id, Numero_economico, Fecha_adquisicion, Peso, precio
FROM animales WHERE Raza = " + valorSeleccionado + """;
                break;
            case "Busqueda por Sexo":
                query = "SELECT Sexo, Id, Numero_economico, Fecha_adquisicion, Peso, precio
FROM animales WHERE Sexo = " + valorSeleccionado + """;
                break;
            case "Busqueda por Salud":
                query = "SELECT Salud, Id, Numero_economico, Raza, Peso, precio FROM animales
WHERE Salud = " + valorSeleccionado + """;
                break;
            case "Busqueda Reproductivo":
                query = "SELECT Reproductivo, Id, Numero_economico, Raza, Peso, precio FROM
animales WHERE Reproductivo = " + valorSeleccionado + """;

```

```

        break;
        case "Busqueda por Etapa":
            query = "SELECT Etapa, Id, Numero_economico, Raza, Peso, precio FROM animales
WHERE Etapa = '" + valorSeleccionado + "'";
            break;
        case "Busqueda Corporal":
            query = "SELECT Corporal, Id, Numero_economico, Raza, Peso, precio FROM
animales WHERE Corporal = '" + valorSeleccionado + "'";
            break;
        case "Busqueda Sanitario":
            query = "SELECT Sanitario, Id, Numero_economico, Raza, Peso, precio FROM
animales WHERE Sanitario = '" + valorSeleccionado + "'";
            break;
        case "Busqueda Alimenticio":
            query = "SELECT Alimenticio, Id, Numero_economico, Raza, Peso, precio FROM
animales WHERE Alimenticio = '" + valorSeleccionado + "'";
            break;
        case "Modo de Adquisicion":
            query = "SELECT Modo_adquisicion, Id, Numero_economico, Fecha_adquisicion,
Peso, precio FROM animales WHERE Modo_adquisicion = '" + valorSeleccionado + "'";
            break;
        default:
            limpiarTabla();
            return;
    }

    // Ejecutar la consulta filtrada
    if (!query.isEmpty()) {
        buscarDatos(query);
    }
}
});
}

private void mostrarBusquedaPorCategoria() {
    jComboBox2.setVisible(true);
}

private void mostrarBusquedaPorID() {
    jButton1.setVisible(true);
    jTextField1.setVisible(true);

    // Obtener el texto del campo de ID y asegurarse de que no esté vacío
    String idText = jTextField1.getText().trim();
    if (idText.isEmpty()) {
        System.out.println("Por favor, ingresa un ID.");
        return;
    }

    int id;
    try {
        // Intentar convertir el texto a un número entero

```



```

        id = Integer.parseInt(idText);
    } catch (NumberFormatException e) {
        System.out.println("ID no válido. Asegúrate de ingresar un número.");
        return;
    }

    // Conectar a la base de datos y realizar la consulta
    try (Connection conn = Conexion.conectar()) {
        String query = "SELECT id, Numero_economico, Fecha_adquisicion, raza, peso, sexo, salud,
reproductivo, etapa, corporal, sanitario, alimenticio, imagen, precio FROM animales WHERE id = ?";

        try (PreparedStatement stmt = conn.prepareStatement(query)) {
            stmt.setInt(1, id);
            try (ResultSet rs = stmt.executeQuery()) {
                if (rs.next()) {
                    // Crear un objeto Animal con los datos del registro encontrado
                    Animal animal = new Animal(
                        rs.getInt("id"),
                        rs.getString("Numero_economico"),
                        rs.getDate("Fecha_adquisicion"),
                        rs.getString("raza"),
                        rs.getFloat("peso"),
                        rs.getString("sexo"),
                        rs.getString("salud"),
                        rs.getString("reproductivo"),
                        rs.getString("etapa"),
                        rs.getString("corporal"),
                        rs.getString("sanitario"),
                        rs.getString("alimenticio"),
                        rs.getBytes("imagen"), // Obtener la imagen como byte[]
                        rs.getString("precio")
                    );
                    mostrarAnimal(animal); // Mostrar los datos en la interfaz

                    // Bloquear todos los campos de entrada donde se muestran los datos
                    jTextField2.setEditable(false);
                    jTextField3.setEditable(false);
                    jTextField4.setEditable(false);
                    jTextField5.setEditable(false);
                    jTextField6.setEditable(false);
                    jTextField7.setEditable(false);
                    jTextField8.setEditable(false);
                    jTextField9.setEditable(false);
                    jTextField10.setEditable(false);
                    jTextField11.setEditable(false);
                    jTextField12.setEditable(false);
                    jTextField13.setEditable(false);
                    jTextField14.setEditable(false);
                    jButton3.setVisible(true);
                    jButton5.setVisible(true);
                    jButton6.setVisible(true);
                }
            }
        }
    }

```



```

    } else {
        System.out.println("No se encontró ningún registro con el ID especificado.");
        // Valores predeterminados
        jTextField1.setText("");
        jComboBox1.setSelectedItem("----");
        jTextField3.setText("");
        jComboBox2.setSelectedItem("----");
        jTextField7.setText("");
        jTextField8.setText("");
        jTextField9.setText("");
        jTextField10.setText("----");
        jTextField11.setText("");
        jTextField13.setText("");
        jTextField12.setText("");
        jTextField6.setText("");
        jComboBox11.setSelectedItem("----");
        jTextField4.setText("");
        jTextField2.setText("");
        jTextField5.setText("");
        imagen.setIcon(null);

        // Bloquear los campos
        jTextField2.setEditable(false);
        jTextField3.setEditable(false);
        jTextField4.setEditable(false);
        jTextField5.setEditable(false);
    }
}
} catch (SQLException e) {
    e.printStackTrace();
}
}

private void mostrarRegistrosPorOrden() {
    listaAnimales.clear(); // Limpiar la lista antes de cargar nuevos registros
    try (Connection conn = Conexion.conectar()) { // Usa `conectar()` en lugar de `getConnection()`
        String query = "SELECT id, Numero_economico, Fecha_adquisicion, raza, peso, sexo, salud,
reproductivo, etapa, corporal, sanitario, alimenticio, imagen, precio FROM animales ORDER BY id
ASC";
        try (Statement stmt = conn.createStatement(); ResultSet rs = stmt.executeQuery(query)) {
            while (rs.next()) {
                Animal animal = new Animal(
                    rs.getInt("id"),
                    rs.getString("Numero_economico"),
                    rs.getDate("Fecha_adquisicion"),
                    rs.getString("raza"),
                    rs.getFloat("peso"),
                    rs.getString("sexo"),
                    rs.getString("salud"),
                    rs.getString("reproductivo"),

```

```

        rs.getString("etapa"),
        rs.getString("corporal"),
        rs.getString("sanitario"),
        rs.getString("alimenticio"),
        rs.getBytes("imagen"), // Obtener la imagen como byte[]
        rs.getString("precio")
    );
    listaAnimales.add(animal);

    jButton2.setVisible(true);
    jButton3.setVisible(true);
    jButton4.setVisible(true);
    jButton5.setVisible(true);
    jButton6.setVisible(true);
    }
}
if (!listaAnimales.isEmpty()) {
    registroActual = 0;
    mostrarAnimal(listaAnimales.get(registroActual));
} else {
    System.out.println("No se encontraron registros.");
}
} catch (SQLException e) {
    e.printStackTrace();
}
}

private void mostrarAnimal(Animal animal) {
    imagen.setVisible(true); // Asumiendo que tienes el componente `imagen` para mostrar alguna
    imagen del animal
    // Convertir byte[] a Image y establecer en JLabel
    if (animal.getImagen() != null) {
        ImagenCon imagenCon = new ImagenCon(animal.getImagen());
        imagen.setIcon(imagenCon);
    } else {
        imagen.setIcon(null); // O una imagen por defecto
    }

    IdTxt.setVisible(true);
    jTextField2.setVisible(true);
    jTextField2.setText(String.valueOf(animal.getId()));
    id = jTextField2.getText();

    NoTxt.setVisible(true);
    jTextField4.setVisible(true);
    jTextField4.setText(animal.getNumeroEconomico());

    FechaAdqTxt.setVisible(true);
    jTextField5.setVisible(true);
    jTextField5.setText(animal.getFechaAdquisicion().toString());

    RazaTxt.setVisible(true);

```

```

jTextField6.setVisible(true);
jTextField6.setText(animal.getRaza());

PesoTxt.setVisible(true);
jTextField3.setVisible(true);
jTextField3.setText(String.valueOf(animal.getPeso()));

SexoTxt.setVisible(true);
jTextField7.setVisible(true);
jTextField7.setText(animal.getSexo());

jLabel8.setVisible(true);
jSeparator1.setVisible(true);

SaludTxt.setVisible(true);
jTextField8.setVisible(true);
jTextField8.setText(animal.getSalud());

ReproductivoTxt.setVisible(true);
jTextField9.setVisible(true);
jTextField9.setText(animal.getReproductivo());

EtapaTxt.setVisible(true);
jTextField10.setVisible(true);
jTextField10.setText(animal.getEtapa());

CorporalTxt.setVisible(true);
jTextField11.setVisible(true);
jTextField11.setText(animal.getCorporal());

SanitarioTxt.setVisible(true);
jTextField12.setVisible(true);
jTextField12.setText(animal.getSanitario());

AlimenticioTxt.setVisible(true);
jTextField13.setVisible(true);
jTextField13.setText(animal.getAlimenticio());

jLabel8.setVisible(true);
jSeparator1.setVisible(true);
jTextField14.setText(animal.getPrecio());
}

private void ocultarComponentesExclusivos() {
    imagen.setVisible(false);
    ldtxt.setVisible(false);
    jTextField2.setVisible(false);
    NoTxt.setVisible(false);
    jTextField4.setVisible(false);
    FechaAdqTxt.setVisible(false);
    jTextField5.setVisible(false);

```



```

RazaTxt.setVisible(false);
jTextField6.setVisible(false);
PesoTxt.setVisible(false);
jTextField3.setVisible(false);
SexoTxt.setVisible(false);
jTextField7.setVisible(false);
jLabel8.setVisible(false);
jSeparator1.setVisible(false);
SaludTxt.setVisible(false);
jTextField8.setVisible(false);
ReproductivoTxt.setVisible(false);
jTextField9.setVisible(false);
EtapasTxt.setVisible(false);
jTextField10.setVisible(false);
CorporalTxt.setVisible(false);
jTextField11.setVisible(false);
SanitarioTxt.setVisible(false);
jTextField12.setVisible(false);
AlimenticioTxt.setVisible(false);
jTextField13.setVisible(false);
jScrollPane1.setVisible(false);
jTable1.setVisible(false);
jComboBox11.setVisible(false);
jButton1.setVisible(false);
jButton2.setVisible(false);
jButton3.setVisible(false);
jButton4.setVisible(false);
jButton5.setVisible(false);
jButton6.setVisible(false);
}

private void ocultarComponentes() {
    ocultarComponentesExclusivos();
    jComboBox2.setVisible(false);
    jTextField1.setVisible(false);
}

// Métodos para configurar los ComboBox específicos
private void configurarComboBoxRaza() {
    jComboBox11.setVisible(true);
    jScrollPane1.setVisible(true);
    jTable1.setVisible(true);
    jComboBox11.removeAllItems();
    String[] razas = {"----", "Angus", "Hereford", "Charolais", "Limousin", "Brahman", "Simmental",
"Holstein", "Jersey", "Guernsey", "Pardo Suizo", "Ayrshire", "Shorthorn", "Normando", "Fleckvieh",
"Nellore", "Criollo"};
    for (String raza : razas) {
        jComboBox11.addItem(raza);
    }
}

private void configurarComboBoxSexo() {

```

```

jComboBox11.setVisible(true);
jScrollPane1.setVisible(true);
jTable1.setVisible(true);
jComboBox11.removeAllItems();
jComboBox11.addItem("----");
jComboBox11.addItem("Macho");
jComboBox11.addItem("Hembra");
}

```

```

private void configurarComboBoxSalud() {
jComboBox11.setVisible(true);
jScrollPane1.setVisible(true);
jTable1.setVisible(true);
jComboBox11.removeAllItems();
jComboBox11.addItem("----");
jComboBox11.addItem("Saludable");
jComboBox11.addItem("Enfermo");
jComboBox11.addItem("En recuperación");
jComboBox11.addItem("Lesionado");
}

```

```

private void configurarComboBoxReproductivo() {
jComboBox11.setVisible(true);
jScrollPane1.setVisible(true);
jTable1.setVisible(true);
jComboBox11.removeAllItems();
jComboBox11.addItem("----");
jComboBox11.addItem("Gestante");
jComboBox11.addItem("En celo");
jComboBox11.addItem("Vacío");
jComboBox11.addItem("Parida");
jComboBox11.addItem("Activo");
jComboBox11.addItem("Inactivo");
}

```

```

private void configurarComboBoxEtapa() {
jComboBox11.setVisible(true);
jScrollPane1.setVisible(true);
jTable1.setVisible(true);
jComboBox11.removeAllItems();
jComboBox11.addItem("----");
jComboBox11.addItem("Ternero o becerro");
jComboBox11.addItem("Joven o recría");
jComboBox11.addItem("Adulto");
jComboBox11.addItem("Anciano");
}

```

```

private void configurarComboBoxCorporal() {
jComboBox11.setVisible(true);
jScrollPane1.setVisible(true);
jTable1.setVisible(true);
jComboBox11.removeAllItems();

```



```

jComboBox11.addItem("----");
jComboBox11.addItem("Óptima");
jComboBox11.addItem("Baja condición");
jComboBox11.addItem("Sobrepeso");
}

private void configurarComboBoxSanitario() {
    jComboBox11.setVisible(true);
    jScrollPane1.setVisible(true);
    jTable1.setVisible(true);
    jComboBox11.removeAllItems();
    jComboBox11.addItem("----");
    jComboBox11.addItem("Vacunado");
    jComboBox11.addItem("En cuarentena");
    jComboBox11.addItem("En tratamiento");
    jComboBox11.addItem("Sin tratamiento");
}

private void configurarComboBoxAlimenticio() {
    jComboBox11.setVisible(true);
    jScrollPane1.setVisible(true);
    jTable1.setVisible(true);
    jComboBox11.removeAllItems();
    jComboBox11.addItem("----");
    jComboBox11.addItem("Sano");
    jComboBox11.addItem("Desnutrido");
    jComboBox11.addItem("Obeso");
}

private void configurarComboBoxModoAdquisicion() {
    jComboBox11.setVisible(true);
    jTable1.setVisible(true);
    jScrollPane1.setVisible(true);
    jComboBox11.removeAllItems();
    jComboBox11.addItem("----");
    jComboBox11.addItem("Compra");
    jComboBox11.addItem("Donación");
    jComboBox11.addItem("Intercambio");
}

private void limpiarTabla() {
    DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
    model.setRowCount(0);
}

private void configurarModeloTabla(String[] columnas) {
    // Usar el modelo personalizado con botones
    DefaultTableModel model = new ModeloTablaConBotones(new Object[0][0], columnas);
    model.addColumn("Acciones");

    // Configurar la tabla con el nuevo modelo

```



```

jTable1.setModel(model);
}
/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel2 = new javax.swing.JLabel();
    jComboBox1 = new javax.swing.JComboBox<>();
    jComboBox2 = new javax.swing.JComboBox<>();
    jTextField1 = new javax.swing.JTextField();
    imagen = new javax.swing.JLabel();
    Idtxt = new javax.swing.JLabel();
    FechaAdqTxt = new javax.swing.JLabel();
    RazaTxt = new javax.swing.JLabel();
    PesoTxt = new javax.swing.JLabel();
    jTextField2 = new javax.swing.JTextField();
    jTextField3 = new javax.swing.JTextField();
    SexoTxt = new javax.swing.JLabel();
    NoTxt = new javax.swing.JLabel();
    jTextField4 = new javax.swing.JTextField();
    jSeparator1 = new javax.swing.JSeparator();
    jLabel8 = new javax.swing.JLabel();
    SaludTxt = new javax.swing.JLabel();
    ReproductivoTxt = new javax.swing.JLabel();
    EtapaTxt = new javax.swing.JLabel();
    CorporalTxt = new javax.swing.JLabel();
    AlimenticioTxt = new javax.swing.JLabel();
    SanitarioTxt = new javax.swing.JLabel();
    jTextField6 = new javax.swing.JTextField();
    jTextField5 = new javax.swing.JTextField();
    jTextField7 = new javax.swing.JTextField();
    jTextField8 = new javax.swing.JTextField();
    jTextField9 = new javax.swing.JTextField();
    jTextField10 = new javax.swing.JTextField();
    jTextField11 = new javax.swing.JTextField();
    jTextField12 = new javax.swing.JTextField();
    jTextField13 = new javax.swing.JTextField();
    jButton2 = new javax.swing.JButton();
    jButton3 = new javax.swing.JButton();
    jButton4 = new javax.swing.JButton();
    jButton5 = new javax.swing.JButton();
    jButton6 = new javax.swing.JButton();
    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    jComboBox11 = new javax.swing.JComboBox<>();
    jLabel3 = new javax.swing.JLabel();
    jButton1 = new javax.swing.JButton();

```

```

jTextField14 = new javax.swing.JTextField();

setBackground(new java.awt.Color(255, 255, 255));
setLayout(null);

jLabel2.setFont(new java.awt.Font("Roboto Medium", 0, 12)); // NOI18N
jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel2.setText("BUSQUEDA EN EL REGISTRO");
add(jLabel2);
jLabel2.setBounds(0, 0, 470, 30);

jComboBox1.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Tipo de
Busqueda", "Mostrar Registros por Orden", "Busqueda por ID", "Busqueda por Categoria" }));
add(jComboBox1);
jComboBox1.setBounds(20, 50, 200, 22);

jComboBox2.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Categoria",
"Busqueda por Raza", "Busqueda por Sexo", "Busqueda por Salud", "Busqueda Reproductivo",
"Busqueda por Etapa", "Busqueda Corporal", "Busqueda Sanitario", "Busqueda Alimenticio", "Modo
de Adquisicion" }));
add(jComboBox2);
jComboBox2.setBounds(240, 50, 220, 22);
add(jTextField1);
jTextField1.setBounds(240, 50, 220, 22);

imagen.setIcon(new javax.swing.ImageIcon(getClass().getResource("/iconos/añadir.png"))); //
NOI18N
imagen.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
imagen.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        imagenMouseClicked(evt);
    }
});
add(imagen);
imagen.setBounds(20, 100, 130, 120);

Idtxt.setText("Numero de ID:");
add(Idtxt);
Idtxt.setBounds(190, 110, 90, 16);

FechaAdqTxt.setText("Fecha de Adquisicion:");
add(FechaAdqTxt);
FechaAdqTxt.setBounds(190, 140, 130, 16);

RazaTxt.setText("Tipo o Raza:");
add(RazaTxt);
RazaTxt.setBounds(190, 170, 80, 16);

PesoTxt.setText("Peso:");
add(PesoTxt);
PesoTxt.setBounds(190, 200, 50, 16);
add(jTextField2);

```



```
jTextField2.setBounds(280, 110, 140, 22);
add(jTextField3);
jTextField3.setBounds(240, 200, 160, 22);

SexoTxt.setText("Sexo:");
add(SexoTxt);
SexoTxt.setBounds(410, 200, 50, 16);

NoTxt.setText("No. Economico Propiedad:");
add(NoTxt);
NoTxt.setBounds(430, 110, 160, 20);
add(jTextField4);
jTextField4.setBounds(590, 110, 73, 22);

jSeparator1.setForeground(new java.awt.Color(153, 153, 153));
add(jSeparator1);
jSeparator1.setBounds(150, 240, 510, 10);

jLabel8.setText("Estados o Categorías");
add(jLabel8);
jLabel8.setBounds(20, 230, 130, 20);

SaludTxt.setText("Salud:");
add(SaludTxt);
SaludTxt.setBounds(20, 260, 60, 20);

ReproductivoTxt.setText("Reproductivo:");
add(ReproductivoTxt);
ReproductivoTxt.setBounds(230, 260, 90, 20);

EtapaTxt.setText("Etapa:");
add(EtapaTxt);
EtapaTxt.setBounds(470, 260, 50, 20);

CorporalTxt.setText("Corporal:");
add(CorporalTxt);
CorporalTxt.setBounds(20, 300, 70, 20);

AlimenticioTxt.setText("Alimenticio:");
add(AlimenticioTxt);
AlimenticioTxt.setBounds(440, 300, 80, 20);

SanitarioTxt.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
SanitarioTxt.setText("Sanitario:");
add(SanitarioTxt);
SanitarioTxt.setBounds(220, 300, 60, 20);
add(jTextField6);
jTextField6.setBounds(320, 170, 340, 22);
add(jTextField5);
jTextField5.setBounds(320, 140, 340, 22);
add(jTextField7);
jTextField7.setBounds(450, 200, 210, 22);
```

```

add(jTextField8);
jTextField8.setBounds(70, 260, 150, 22);
add(jTextField9);
jTextField9.setBounds(320, 260, 140, 22);
add(jTextField10);
jTextField10.setBounds(510, 260, 150, 22);
add(jTextField11);
jTextField11.setBounds(80, 300, 140, 22);
add(jTextField12);
jTextField12.setBounds(280, 300, 160, 22);
add(jTextField13);
jTextField13.setBounds(510, 300, 150, 22);

jButton2.setText("ANTERIOR");
jButton2.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jButton2MouseClicked(evt);
    }
});
add(jButton2);
jButton2.setBounds(40, 390, 110, 23);

jButton3.setText("VER MÁS");
jButton3.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jButton3MouseClicked(evt);
    }
});
add(jButton3);
jButton3.setBounds(340, 390, 110, 23);

jButton4.setText("SIGUIENTE");
jButton4.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jButton4MouseClicked(evt);
    }
});
add(jButton4);
jButton4.setBounds(180, 390, 120, 23);

jButton5.setText("EDITAR");
jButton5.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jButton5MouseClicked(evt);
    }
});
add(jButton5);
jButton5.setBounds(460, 390, 90, 23);

jButton6.setText("ELIMINAR");
add(jButton6);
jButton6.setBounds(560, 390, 100, 23);

```



```

jTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
jScrollPane1.setViewportView(jTable1);

add(jScrollPane1);
jScrollPane1.setBounds(10, 90, 670, 360);

jComboBox11.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "----" }));
add(jComboBox11);
jComboBox11.setBounds(480, 50, 200, 22);

jLabel3.setText("Precio $$$:");
add(jLabel3);
jLabel3.setBounds(490, 10, 80, 20);

jButton1.setText("BUSCAR");
jButton1.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jButton1MouseClicked(evt);
    }
});
add(jButton1);
jButton1.setBounds(480, 50, 90, 23);
add(jTextField14);
jTextField14.setBounds(560, 10, 110, 22);
}

private void imagenMouseClicked(java.awt.event.MouseEvent evt) {
}

private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
    mostrarBusquedaPorID();
}

private void jButton4MouseClicked(java.awt.event.MouseEvent evt) {
    if (registroActual < listaAnimales.size() - 1) { // Verifica que no estamos en el último registro
        registroActual++; // Avanza al siguiente registro
        mostrarAnimal(listaAnimales.get(registroActual)); // Muestra el registro actual
    } else {
        JOptionPane.showMessageDialog(this, "No hay más registros.", "Fin de la lista",
JOptionPane.INFORMATION_MESSAGE);

```

```

    }
    }

    private void jButton2MouseClicked(java.awt.event.MouseEvent evt) {
        if (registroActual > 0) { // Verifica que no estamos en el primer registro
            registroActual--; // Retrocede al registro anterior
            mostrarAnimal(listaAnimales.get(registroActual)); // Muestra el registro actual
        } else {
            JOptionPane.showMessageDialog(this, "No hay registros anteriores.", "Inicio de la lista",
JOptionPane.INFORMATION_MESSAGE);
        }
    }

    private void jButton5MouseClicked(java.awt.event.MouseEvent evt) {
        Editar editarWindow = new Editar(username,id);
        editarWindow.setVisible(true);
        this.dispose();
    }

    private void jButton3MouseClicked(java.awt.event.MouseEvent evt) {
        VerMas verWindow = new VerMas(username,id);
        verWindow.setVisible(true);
        this.dispose();
    }

    // Variables declaration - do not modify
    private javax.swing.JLabel AlimenticioTxt;
    private javax.swing.JLabel CorporalTxt;
    private javax.swing.JLabel EtapaTxt;
    private javax.swing.JLabel FechaAdqTxt;
    private javax.swing.JLabel Idtxt;
    private javax.swing.JLabel NoTxt;
    private javax.swing.JLabel PesoTxt;
    private javax.swing.JLabel RazaTxt;
    private javax.swing.JLabel ReproductivoTxt;
    private javax.swing.JLabel SaludTxt;
    private javax.swing.JLabel SanitarioTxt;
    private javax.swing.JLabel SexoTxt;
    private javax.swing.JLabel imagen;
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JButton jButton4;
    private javax.swing.JButton jButton5;
    private javax.swing.JButton jButton6;
    private javax.swing.JComboBox<String> jComboBox1;
    private javax.swing.JComboBox<String> jComboBox11;
    private javax.swing.JComboBox<String> jComboBox2;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel8;

```

```

private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JTable jTable1;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField10;
private javax.swing.JTextField jTextField11;
private javax.swing.JTextField jTextField12;
private javax.swing.JTextField jTextField13;
private javax.swing.JTextField jTextField14;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField4;
private javax.swing.JTextField jTextField5;
private javax.swing.JTextField jTextField6;
private javax.swing.JTextField jTextField7;
private javax.swing.JTextField jTextField8;
private javax.swing.JTextField jTextField9;
// End of variables declaration

```

```

public void dispose() {
    throw new UnsupportedOperationException("Not supported yet.");
}
}

```

EditorAnimal.java

```
package com.admin;
```

```

import com.conexion.Conexion;
import com.frame.Inicio;
import com.toedter.calendar.JDateChooser;
import java.awt.Graphics2D;
import java.awt.Image;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.awt.image.BufferedImage;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.SimpleDateFormat;
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.util.HashMap;
import java.util.Map;
import java.util.logging.Level;

```



```

import java.util.logging.Logger;
import javax.imageio.ImageIO;
import javax.swing.Icon;
import javax.swing.ImageIcon;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.JPopupMenu;

/**
 *
 * @author wicho
 */
public class EditarAnimal extends javax.swing.JPanel {

    private String idAnimal;
    private String username; // Variable para almacenar el nombre de usuario
    private boolean imagenFueModificada = false;
    private Image imagenSeleccionada = null;
    private JDateChooser jDateChooser;
    private JDateChooser jDateChooser2;

    public EditarAnimal(String username, String id) {
        this.idAnimal = id;
        this.username = username; // Guardar el nombre de usuario
        System.out.println(username);
        initComponents();
        cargarDatos(idAnimal); // Cargar los datos del animal utilizando el ID
        System.out.println(idAnimal);
        jTextField1.setEditable(false);
        precio.setEditable(false);
    }

    private void cargarDatos(String id) {
        Connection con = null;
        try {
            con = Conexion.conectar(); // Obtener la conexión de la clase Conexion
            String query = "SELECT * FROM animales WHERE id = ?"; // Consulta SQL para obtener
datos por ID
            PreparedStatement stmt = con.prepareStatement(query);
            stmt.setString(1, id); // Establecer el ID en la consulta

            ResultSet rs = stmt.executeQuery();
            if (rs.next()) {
                // Asignar cada campo obtenido de la base de datos a su componente
                jTextField1.setText(rs.getString("id"));

                jComboBox1.setSelectedItem(rs.getString("raza"));
                jTextField3.setText(rs.getString("peso"));
                jComboBox2.setSelectedItem(rs.getString("sexo"));
                jTextField4.setText(rs.getString("numero_economico"));
                jComboBox3.setSelectedItem(rs.getString("salud"));
            }
        } catch (SQLException ex) {
            Logger.getLogger(EditarAnimal.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}

```

```

jComboBox4.setSelectedItem(rs.getString("reproductivo"));
jComboBox5.setSelectedItem(rs.getString("etapa"));
jComboBox6.setSelectedItem(rs.getString("corporal"));
jComboBox7.setSelectedItem(rs.getString("alimenticio"));
jComboBox8.setSelectedItem(rs.getString("sanitario"));
jComboBox9.setSelectedItem(rs.getString("modo_adquisicion"));
jTextField2.setText(rs.getString("peso_adquirido"));
jTextField5.setText(rs.getString("peso_destete"));
jComboBox10.setSelectedItem(rs.getString("raza_padre"));
jComboBox11.setSelectedItem(rs.getString("raza_madre"));

SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

// Crear y configurar el primer JDateChooser
jDateChooser = new JDateChooser();
jDateChooser.setBounds(312, 80, 340, 21);
jDateChooser.setDate(rs.getDate("fecha_adquisicion"));
this.add(jDateChooser);

// Crear y configurar el segundo JDateChooser
jDateChooser2 = new JDateChooser();
jDateChooser2.setBounds(120, 340, 210, 21);
jDateChooser2.setDate(rs.getDate("fecha_destete"));
this.add(jDateChooser2); // Añadir al contenedor

// Cargar la imagen desde la base de datos
byte[] imageBytes = rs.getBytes("imagen");
if (imageBytes != null) {
    // Convertir los bytes de la imagen en una imagen
    ImageIcon imagelcon = new ImageIcon(imageBytes);
    Image image = imagelcon.getImage().getScaledInstance(150, 150,
Image.SCALE_SMOOTH); // Ajustar el tamaño
    imagen.setIcon(new ImageIcon(image)); // Asignar la imagen al JLabel
} else {
    imagen.setIcon(null); // Si no hay imagen, establecer el icono como nulo
}

precio.setText(rs.getString("precio"));
}

rs.close();
stmt.close();
} catch (SQLException e) {
    e.printStackTrace();
    JOptionPane.showMessageDialog(this, "Error al cargar los datos del animal.");
} finally {
    try {
        if (con != null) {
            con.close(); // Cerrar la conexión
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```



```

    }
}

private void asignarPrecio() {

    double peso = Double.parseDouble(jTextField3.getText());

    String etapa = jComboBox5.getSelectedItem().toString();

    if (!etapa.equals("----")) {

        String sql = "SELECT " + etapa + " FROM precios LIMIT 1"; // Usamos LIMIT 1 para obtener
solo una fila

        try (Connection con = Conexion.conectar(); PreparedStatement pst =
con.prepareStatement(sql)) {

            // Ejecutar la consulta
            ResultSet rs = pst.executeQuery();

            if (rs.next()) {

                double precioPorKilo = rs.getDouble(1);

                // Calcular el precio final basado en el peso
                double precioFinal = precioPorKilo * peso;

                precio.setText(String.format("%.2f", precioFinal)); // Formatear el precio a dos
decimales
            } else {
                JOptionPane.showMessageDialog(this, "No se encontró el precio para esta etapa",
"Error", JOptionPane.ERROR_MESSAGE);
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Error al obtener el precio: " + e.getMessage(),
"Error", JOptionPane.ERROR_MESSAGE);
        }
    } else {
        // Si no se selecciona una etapa válida, limpiar el campo de precio
        precio.setText("");
    }
}

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel2 = new javax.swing.JLabel();
    imagen = new javax.swing.JLabel();
    Idtxt = new javax.swing.JLabel();
    FechaAdqTxt = new javax.swing.JLabel();

```



```

RazaTxt = new javax.swing.JLabel();
PesoTxt = new javax.swing.JLabel();
jTextField1 = new javax.swing.JTextField();
jComboBox1 = new javax.swing.JComboBox<>();
jTextField3 = new javax.swing.JTextField();
SexoTxt = new javax.swing.JLabel();
jComboBox2 = new javax.swing.JComboBox<>();
NoTxt = new javax.swing.JLabel();
jTextField4 = new javax.swing.JTextField();
jSeparator1 = new javax.swing.JSeparator();
jLabel8 = new javax.swing.JLabel();
SaludTxt = new javax.swing.JLabel();
jComboBox3 = new javax.swing.JComboBox<>();
ReproductivoTxt = new javax.swing.JLabel();
jComboBox4 = new javax.swing.JComboBox<>();
EtapasTxt = new javax.swing.JLabel();
jComboBox5 = new javax.swing.JComboBox<>();
CorporalTxt = new javax.swing.JLabel();
jComboBox6 = new javax.swing.JComboBox<>();
AlimenticioTxt = new javax.swing.JLabel();
jComboBox7 = new javax.swing.JComboBox<>();
SanitarioTxt = new javax.swing.JLabel();
jComboBox8 = new javax.swing.JComboBox<>();
jSeparator2 = new javax.swing.JSeparator();
jLabel15 = new javax.swing.JLabel();
jLabel1 = new javax.swing.JLabel();
jComboBox9 = new javax.swing.JComboBox<>();
jLabel3 = new javax.swing.JLabel();
jTextField2 = new javax.swing.JTextField();
jLabel4 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
jTextField5 = new javax.swing.JTextField();
jLabel6 = new javax.swing.JLabel();
jComboBox10 = new javax.swing.JComboBox<>();
jComboBox11 = new javax.swing.JComboBox<>();
jLabel7 = new javax.swing.JLabel();
guardar = new javax.swing.JButton();
cancelar = new javax.swing.JButton();
jLabel9 = new javax.swing.JLabel();
precio = new javax.swing.JTextField();

```

```

setBackground(new java.awt.Color(255, 255, 255));
setLayout(null);

```

```

jLabel2.setFont(new java.awt.Font("Roboto Medium", 0, 12)); // NOI18N
jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel2.setText("EDITAR GANADO");
add(jLabel2);
jLabel2.setBounds(0, 0, 690, 30);

```

```

imagen.setIcon(new javax.swing.ImageIcon(getClass().getResource("/iconos/añadir.png"))); //
NOI18N

```



```

imagen.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
imagen.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        imagenMouseClicked(evt);
    }
});
add(imagen);
imagen.setBounds(10, 40, 130, 120);

Idtxt.setText("Numero de ID:");
add(Idtxt);
Idtxt.setBounds(180, 50, 90, 16);

FechaAdqTxt.setText("Fecha de Adquisicion:");
add(FechaAdqTxt);
FechaAdqTxt.setBounds(180, 80, 130, 16);

RazaTxt.setText("Tipo o Raza:");
add(RazaTxt);
RazaTxt.setBounds(180, 110, 80, 16);

PesoTxt.setText("Peso:");
add(PesoTxt);
PesoTxt.setBounds(180, 140, 50, 16);
add(jTextField1);
jTextField1.setBounds(270, 50, 140, 22);

jComboBox1.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "----",
"Angus", "Hereford", "Charolais", "Limousin", "Brahman", "Simmental", "Holstein", "Jersey",
"Guernsey", "Pardo Suizo", "Ayrshire", "Shorthorn", "Normando", "Fleckvieh", "Nellore", "Criollo" }));
add(jComboBox1);
jComboBox1.setBounds(120, 380, 210, 22);

jTextField3.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseExited(java.awt.event.MouseEvent evt) {
        jTextField3MouseExited(evt);
    }
});
add(jTextField3);
jTextField3.setBounds(230, 140, 160, 22);

SexoTxt.setText("Sexo:");
add(SexoTxt);
SexoTxt.setBounds(400, 140, 50, 16);

jComboBox2.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "----",
"Macho", "Hembra" }));
add(jComboBox2);
jComboBox2.setBounds(440, 140, 210, 22);

NoTxt.setText("No. Economico Propiedad:");
add(NoTxt);

```

```

NoTxt.setBounds(420, 50, 160, 20);
add(jTextField4);
jTextField4.setBounds(580, 50, 73, 22);

jSeparator1.setForeground(new java.awt.Color(153, 153, 153));
add(jSeparator1);
jSeparator1.setBounds(140, 180, 510, 10);

jLabel8.setText("Estados o Categorías");
add(jLabel8);
jLabel8.setBounds(10, 170, 130, 20);

SaludTxt.setText("Salud:");
add(SaludTxt);
SaludTxt.setBounds(10, 200, 60, 20);

jComboBox3.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "----",
"Saludable", "Enfermo", "En recuperación", "Lesionado" }));
add(jComboBox3);
jComboBox3.setBounds(60, 200, 150, 22);

ReproductivoTxt.setText("Reproductivo:");
add(ReproductivoTxt);
ReproductivoTxt.setBounds(220, 200, 90, 20);

jComboBox4.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "----",
"Gestante", "En celo", "Vacío", "Parida", "Activo", "Inactivo" }));
add(jComboBox4);
jComboBox4.setBounds(310, 200, 140, 22);

EtapaTxt.setText("Etapa:");
add(EtapaTxt);
EtapaTxt.setBounds(462, 200, 50, 20);

jComboBox5.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "----",
"Ternero", "Joven", "Adulto", "Anciano" }));
jComboBox5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jComboBox5ActionPerformed(evt);
    }
});
add(jComboBox5);
jComboBox5.setBounds(500, 200, 150, 22);

CorporalTxt.setText("Corporal:");
add(CorporalTxt);
CorporalTxt.setBounds(10, 240, 70, 20);

jComboBox6.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "----",
"Óptima", "Baja condición ", "Sobrepeso" }));
add(jComboBox6);
jComboBox6.setBounds(70, 240, 140, 22);

```



```

AlimenticioTxt.setText("Alimenticio:");
add(AlimenticioTxt);
AlimenticioTxt.setBounds(430, 240, 80, 20);

jComboBox7.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "----", "Bien
alimentado", "En engorda", "Desnutrido", "En suplementación" }));
add(jComboBox7);
jComboBox7.setBounds(500, 240, 150, 22);

SanitarioTxt.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
SanitarioTxt.setText("Sanitario:");
add(SanitarioTxt);
SanitarioTxt.setBounds(210, 240, 60, 20);

jComboBox8.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "----",
"Vacunado", "En cuarentena", "En tratamiento", "Libre de enfermedades" }));
add(jComboBox8);
jComboBox8.setBounds(270, 240, 154, 22);

jSeparator2.setForeground(new java.awt.Color(153, 153, 153));
add(jSeparator2);
jSeparator2.setBounds(140, 280, 510, 10);

jLabel15.setText("Datos de Adquisicion");
add(jLabel15);
jLabel15.setBounds(10, 270, 140, 20);

jLabel1.setText("Modo de Adquisicion:");
add(jLabel1);
jLabel1.setBounds(10, 300, 140, 20);

jComboBox9.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "----",
"Compra", "Reproducción Propia", "Intercambio o Trueque", "Donación o Cesión", "Arrendamiento o
Alquiler", "Préstamo o Comodato", "Herencia o Sucesión" }));
add(jComboBox9);
jComboBox9.setBounds(140, 300, 190, 22);

jLabel3.setText("Peso al Adquirir:");
add(jLabel3);
jLabel3.setBounds(346, 300, 100, 20);
add(jTextField2);
jTextField2.setBounds(450, 300, 200, 22);

jLabel4.setText("Fecha al Destete:");
add(jLabel4);
jLabel4.setBounds(10, 340, 100, 20);

jLabel5.setText("Peso al Destete:");
add(jLabel5);
jLabel5.setBounds(342, 340, 100, 20);
add(jTextField5);

```

```

jTextField5.setBounds(450, 340, 200, 22);

jLabel6.setText("Raza del Padre:");
add(jLabel6);
jLabel6.setBounds(10, 380, 90, 20);

jComboBox10.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "----",
"Angus", "Hereford", "Charolais", "Limousin", "Brahman", "Simmental", "Holstein", "Jersey",
"Guernsey", "Pardo Suizo", "Ayrshire", "Shorthorn", "Normando", "Fleckvieh", "Nellore", "Criollo" }));
add(jComboBox10);
jComboBox10.setBounds(310, 110, 340, 22);

jComboBox11.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "----",
"Angus", "Hereford", "Charolais", "Limousin", "Brahman", "Simmental", "Holstein", "Jersey",
"Guernsey", "Pardo Suizo", "Ayrshire", "Shorthorn", "Normando", "Fleckvieh", "Nellore", "Criollo" }));
add(jComboBox11);
jComboBox11.setBounds(450, 380, 200, 22);

jLabel7.setText("Raza de la Madre:");
add(jLabel7);
jLabel7.setBounds(340, 380, 110, 20);

guardar.setText("Guardar");
guardar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        guardarActionPerformed(evt);
    }
});
add(guardar);
guardar.setBounds(380, 430, 90, 23);

cancelar.setText("Cancelar");
cancelar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cancelarActionPerformed(evt);
    }
});
add(cancelar);
cancelar.setBounds(260, 430, 90, 23);

jLabel9.setText("Precio $$:");
add(jLabel9);
jLabel9.setBounds(520, 10, 70, 16);

precio.setText(jTextField6);
add(precio);
precio.setBounds(590, 10, 71, 22);
} // </editor-fold>

```

```

private void imagenMouseClicked(java.awt.event.MouseEvent evt) {
    JFileChooser fileChooser = new JFileChooser();

```

```

fileChooser.setDialogTitle("Selecciona una imagen");
int resultado = fileChooser.showOpenDialog(this);

if (resultado == JFileChooser.APPROVE_OPTION) {
    File archivoSeleccionado = fileChooser.getSelectedFile();
    try {
        BufferedImage imagenOriginal = ImageIO.read(archivoSeleccionado);
        imagenSeleccionada = imagenOriginal.getScaledInstance(imagen.getWidth(),
imagen.getHeight(), Image.SCALE_SMOOTH);
        imagen.setIcon(new ImageIcon(imagenSeleccionada));

        imagenFueModificada = true;
    } catch (IOException ex) {
        ex.printStackTrace();
        JOptionPane.showMessageDialog(this, "Error al cargar la imagen", "Error",
JOptionPane.ERROR_MESSAGE);
    }
}

private byte[] convertirImagenABYtes() {
    if (imagenSeleccionada != null) {
        try {
            BufferedImage bufferedImage = new BufferedImage(
                imagenSeleccionada.getWidth(null),
                imagenSeleccionada.getHeight(null),
                BufferedImage.TYPE_INT_ARGB);
            bufferedImage.getGraphics().drawImage(imagenSeleccionada, 0, 0, null);

            // Usar try-with-resources para asegurar la liberación del recurso
            try (ByteArrayOutputStream baos = new ByteArrayOutputStream()) {
                ImageIO.write(bufferedImage, "png", baos);
                return baos.toByteArray();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    return null;
}

private void guardarActionPerformed(java.awt.event.ActionEvent evt) {
    Connection con = null;
    try {
        con = Conexion.conectar();
        String id = jTextField1.getText();
        System.out.println("ID del animal: " + id);

        // Obtener los datos actuales del animal
        String queryDatosActuales = "SELECT * FROM animales WHERE id = ?";

```

```

PreparedStatement stmtDatosActuales = con.prepareStatement(queryDatosActuales);
stmtDatosActuales.setString(1, id);
ResultSet rsDatosActuales = stmtDatosActuales.executeQuery();

Map<String, String> datosAnteriores = new HashMap<>();
byte[] imagenAnterior = null;

if (rsDatosActuales.next()) {
    datosAnteriores.put("raza", rsDatosActuales.getString("raza"));
    datosAnteriores.put("peso", rsDatosActuales.getString("peso"));
    datosAnteriores.put("sexo", rsDatosActuales.getString("sexo"));
    datosAnteriores.put("numero_economico",
rsDatosActuales.getString("numero_economico"));
    datosAnteriores.put("salud", rsDatosActuales.getString("salud"));
    datosAnteriores.put("reproductivo", rsDatosActuales.getString("reproductivo"));
    datosAnteriores.put("etapa", rsDatosActuales.getString("etapa"));
    datosAnteriores.put("corporal", rsDatosActuales.getString("corporal"));
    datosAnteriores.put("alimenticio", rsDatosActuales.getString("alimenticio"));
    datosAnteriores.put("sanitario", rsDatosActuales.getString("sanitario"));
    datosAnteriores.put("modo_adquisicion",
rsDatosActuales.getString("modo_adquisicion"));
    datosAnteriores.put("peso_adquirido", rsDatosActuales.getString("peso_adquirido"));
    datosAnteriores.put("peso_destete", rsDatosActuales.getString("peso_destete"));
    datosAnteriores.put("raza_padre", rsDatosActuales.getString("raza_padre"));
    datosAnteriores.put("raza_madre", rsDatosActuales.getString("raza_madre"));
    datosAnteriores.put("fecha_adquisicion",
rsDatosActuales.getString("fecha_adquisicion"));
    datosAnteriores.put("fecha_destete", rsDatosActuales.getString("fecha_destete"));
    datosAnteriores.put("precio", rsDatosActuales.getString("precio"));
    imagenAnterior = rsDatosActuales.getBytes("imagen");
}
stmtDatosActuales.close();

// Actualizar los datos del animal
String query = "UPDATE animales SET raza = ?, peso = ?, sexo = ?, numero_economico =
?, salud = ?, reproductivo = ?, etapa = ?, corporal = ?, alimenticio = ?, sanitario = ?, modo_adquisicion
= ?, peso_adquirido = ?, peso_destete = ?, raza_padre = ?, raza_madre = ?, fecha_adquisicion = ?,
fecha_destete = ?, imagen = ?, precio = ? WHERE id = ?";
PreparedStatement stmt = con.prepareStatement(query);

// Asignar valores nuevos y verificar cambios
String[] columnas = {
    "raza", "peso", "sexo", "numero_economico", "salud", "reproductivo",
    "etapa", "corporal", "alimenticio", "sanitario", "modo_adquisicion",
    "peso_adquirido", "peso_destete", "raza_padre", "raza_madre", "fecha_adquisicion",
    "fecha_destete", "imagen", "precio"
};
String[] nuevosValores = {
    jComboBox1.getSelectedItem().toString(),
    jTextField3.getText(),
    jComboBox2.getSelectedItem().toString(),
    jTextField4.getText(),

```



```

        jComboBox3.getSelectedItem().toString(),
        jComboBox4.getSelectedItem().toString(),
        jComboBox5.getSelectedItem().toString(),
        jComboBox6.getSelectedItem().toString(),
        jComboBox7.getSelectedItem().toString(),
        jComboBox8.getSelectedItem().toString(),
        jComboBox9.getSelectedItem().toString(),
        jTextField2.getText(),
        jTextField5.getText(),
        jComboBox10.getSelectedItem().toString(),
        jComboBox11.getSelectedItem().toString(),
        new java.sql.Date(jDateChooser.getDate().getTime()).toString(),
        new java.sql.Date(jDateChooser2.getDate().getTime()).toString()
    };

    // Asignar los valores de los parámetros de la consulta
    for (int i = 0; i < columnas.length - 3; i++) {
        if (!nuevosValores[i].equals(datosAnteriores.get(columnas[i]))) {
            System.out.println("Campo modificado: " + columnas[i]
                + " | Valor anterior: " + datosAnteriores.get(columnas[i])
                + " | Nuevo valor: " + nuevosValores[i]);
            String insertHistorial = "INSERT INTO historial (fecha, movimiento, id, Dato,
                valorAnterior, valorNuevo) VALUES (?, ?, ?, ?, ?, ?)";

            try (PreparedStatement hisStmt = con.prepareStatement(insertHistorial)) {
                LocalDateTime fechaHoraActual = LocalDateTime.now();
                String fecha = fechaHoraActual.toLocalDate().toString();

                // Configuración de los valores para la inserción.
                hisStmt.setString(1, fecha);
                hisStmt.setString(2, "Modificacion");
                hisStmt.setString(3, id);
                hisStmt.setString(4, columnas[i]);
                hisStmt.setString(5, datosAnteriores.get(columnas[i]));
                hisStmt.setString(6, nuevosValores[i]);

                hisStmt.executeUpdate();
            }

            }
            stmt.setString(i + 1, nuevosValores[i]);
        }

        stmt.setDate(16, new java.sql.Date(jDateChooser.getDate().getTime()));

        stmt.setDate(17, new java.sql.Date(jDateChooser2.getDate().getTime()));

        // Manejo de la imagen
        if (imagenFueModificada) {
            if (imagenSeleccionada != null) {
                byte[] imagenBytes = convertirImagenABYtes();

```



```

        if (imagenBytes != null) {
            System.out.println("Imagen modificada. Guardando nueva imagen.");
            stmt.setBytes(18, imagenBytes);
        } else {
            System.out.println("Error al convertir la imagen modificada. Usando imagen anterior.");
            stmt.setBytes(18, imagenAnterior);
        }
    } else {
        System.out.println("Error al obtener la imagen modificada. Usando imagen anterior.");
        stmt.setBytes(18, imagenAnterior);
    }
} else {
    System.out.println("Imagen no modificada. Usando imagen anterior.");
    stmt.setBytes(18, imagenAnterior);
}

    stmt.setString(19, precio.getText());
    stmt.setString(20, id);

// Ejecutar la actualización
    int rowsUpdated = stmt.executeUpdate();
    if (rowsUpdated > 0) {
        JOptionPane.showMessageDialog(this, "Datos actualizados correctamente.");
    } else {
        JOptionPane.showMessageDialog(this, "No se encontró el animal para actualizar.");
    }
    // Registrar movimiento en la tabla 'movimientos'
    String insertMovimiento = "INSERT INTO movimientos (fecha, hora, usuario,
privilegio, movimiento, id) VALUES (?, ?, ?, ?, ?, ?)";
    try (PreparedStatement movStmt = con.prepareStatement(insertMovimiento)) {
        LocalDateTime fechaHoraActual = LocalDateTime.now();
        String fecha = fechaHoraActual.toLocalDate().toString();
        String hora = fechaHoraActual.toLocalTime().format(DateTimeFormatter.ofPattern("HH:mm:ss"));
        movStmt.setString(1, fecha);
        movStmt.setString(2, hora);
        movStmt.setString(3, username);
        movStmt.setString(4, "Administrador");
        movStmt.setString(5, "Actualizacion");
        movStmt.setString(6, id);

        movStmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    }

    stmt.close();
} catch (SQLException e) {
    e.printStackTrace();
    JOptionPane.showMessageDialog(this, "Error al guardar los datos del animal.");
} finally {

```



```

        try {
            if (con != null) {
                con.close();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

private byte[] getImagenBytes() {
    try {
        Icon icon = imagen.getIcon();
        if (icon != null && icon instanceof ImageIcon) {
            Image image = ((ImageIcon) icon).getImage();
            ByteArrayOutputStream baos = new ByteArrayOutputStream();
            ImageIO.write(toBufferedImage(image), "png", baos); que prefieres
            return baos.toByteArray();
        }
    } catch (IOException e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(this, "Error al convertir la imagen a bytes.");
    }
    return null; // Devuelve null si no hay imagen u ocurrió un error
}

private BufferedImage toBufferedImage(Image img) {
    if (img instanceof BufferedImage) {
        return (BufferedImage) img;
    }
    BufferedImage bufferedImage = new BufferedImage(img.getWidth(null), img.getHeight(null),
BufferedImage.TYPE_INT_ARGB);
    Graphics2D g2d = bufferedImage.createGraphics();
    g2d.drawImage(img, 0, 0, null);
    g2d.dispose();
    return bufferedImage;
}

private void cancelarActionPerformed(java.awt.event.ActionEvent evt) {
    Inicio inicioWindow = new Inicio(username);
    inicioWindow.setVisible(true);
}

private void jComboBox5ActionPerformed(java.awt.event.ActionEvent evt) {
    asignarPrecio();
}

private void jTextField3MouseExited(java.awt.event.MouseEvent evt) {
    asignarPrecio();
}

```



```

// Variables declaration - do not modify
private javax.swing.JLabel AlimenticioTxt;
private javax.swing.JLabel CorporalTxt;
private javax.swing.JLabel EtapaTxt;
private javax.swing.JLabel FechaAdqTxt;
private javax.swing.JLabel Idtxt;
private javax.swing.JLabel NoTxt;
private javax.swing.JLabel PesoTxt;
private javax.swing.JLabel RazaTxt;
private javax.swing.JLabel ReproductivoTxt;
private javax.swing.JLabel SaludTxt;
private javax.swing.JLabel SanitarioTxt;
private javax.swing.JLabel SexoTxt;
private javax.swing.JButton cancelar;
private javax.swing.JButton guardar;
private javax.swing.JLabel imagen;
private javax.swing.JComboBox<String> jComboBox1;
private javax.swing.JComboBox<String> jComboBox10;
private javax.swing.JComboBox<String> jComboBox11;
private javax.swing.JComboBox<String> jComboBox2;
private javax.swing.JComboBox<String> jComboBox3;
private javax.swing.JComboBox<String> jComboBox4;
private javax.swing.JComboBox<String> jComboBox5;
private javax.swing.JComboBox<String> jComboBox6;
private javax.swing.JComboBox<String> jComboBox7;
private javax.swing.JComboBox<String> jComboBox8;
private javax.swing.JComboBox<String> jComboBox9;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JSeparator jSeparator2;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField4;
private javax.swing.JTextField jTextField5;
private javax.swing.JTextField precio;
// End of variables declaration
}

```

Ganado.java

package com.admin;



```

import com.EDITOR.EDITOR;
import com.Precios.PreciosV;
import com.VerMas.VerMas;
import com.frame.Inicio;
import java.awt.BorderLayout;
import java.awt.Color;
import javax.swing.JInternalFrame;
import javax.swing.SwingUtilities;
import javax.swing.border.Border;
import javax.swing.plaf.basic.BasicInternalFrameUI;

/**
 *
 * @author wicho
 */
public class Ganado extends javax.swing.JFrame {
    int xMouse, yMouse;
    private String username; private String id;
    /**
     * Creates new form Home
     */
    public Ganado(String username) {
        initComponents();
        this.username = username;
        BuscarAnimal pruebaPanel = new BuscarAnimal(username);

        ventanas.removeAll();
        ventanas.setLayout(new BorderLayout());
        pruebaPanel.setSize(ventanas.getWidth(), ventanas.getHeight());
        ventanas.add(pruebaPanel);
        ventanas.revalidate(); // Actualiza el contenedor
        ventanas.repaint(); // Redibuja el contenedor

        pruebaPanel.setVisible(true);
        this.dispose();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        cierre = new javax.swing.JPanel();
        cierre = new javax.swing.JLabel();
        jPanel2 = new javax.swing.JPanel();
        ventanas = new javax.swing.JPanel();
        menu = new javax.swing.JPanel();

```



```

jLabel2 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
jLabel1 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setUndecorated(true);
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

cierree.setBackground(new java.awt.Color(0, 0, 0));
cierree.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseDragged(java.awt.event.MouseEvent evt) {
        cierreeMouseDragged(evt);
    }
});
cierree.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        cierreeMouseClicked(evt);
    }
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        cierreeMouseEntered(evt);
    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        cierreeMouseExited(evt);
    }
    public void mousePressed(java.awt.event.MouseEvent evt) {
        cierreeMousePressed(evt);
    }
});

cierre.setFont(new java.awt.Font("Roboto Medium", 0, 18)); // NOI18N
cierre.setForeground(new java.awt.Color(255, 255, 255));
cierre.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
cierre.setText("X");

javax.swing.GroupLayout cierreeLayout = new javax.swing.GroupLayout(cierree);
cierree.setLayout(cierreeLayout);
cierreeLayout.setHorizontalGroup(
    cierreeLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            cierreeLayout.createSequentialGroup()
                .addComponent(cierre, javax.swing.GroupLayout.DEFAULT_SIZE, 24,
                    Short.MAX_VALUE)
                .addGap(31)
        );
cierreeLayout.setVerticalGroup(
    cierreeLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            cierreeLayout.createSequentialGroup()
                .addGap(0, 0, Short.MAX_VALUE)
                .addComponent(cierre, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
        );

```

```

getContentPane().add(cierree, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 30, -
1));

jPanel2.setBackground(new java.awt.Color(0, 0, 0));

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 840, Short.MAX_VALUE)
);
jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 31, Short.MAX_VALUE)
);

getContentPane().add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 0, 840,
31));

ventanas.setBackground(new java.awt.Color(255, 255, 255));

javax.swing.GroupLayout ventanasLayout = new javax.swing.GroupLayout(ventanas);
ventanas.setLayout(ventanasLayout);
ventanasLayout.setHorizontalGroup(
    ventanasLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 690, Short.MAX_VALUE)
);
ventanasLayout.setVerticalGroup(
    ventanasLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 470, Short.MAX_VALUE)
);

getContentPane().add(ventanas, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 30,
690, 470));

menu.setBackground(new java.awt.Color(255, 153, 0));
menu.setPreferredSize(new java.awt.Dimension(870, 500));
menu.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO.jpg"))); //
NOI18N
menu.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 360, -1, 110));

jLabel3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO-
removebg-preview.png"))); // NOI18N
menu.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, -1));

jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/campo (1).jpg"))); //
NOI18N
menu.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, 470));

```

```

        getContentPane().add(menu, new org.netbeans.lib.awtextra.AbsoluteConstraints(690, 30, 180,
470));

        pack();
        setLocationRelativeTo(null);
    } // </editor-fold>

    private void cierreMouseEntered(java.awt.event.MouseEvent evt) {
        cierre.setBackground(Color.red);
    }

    private void cierreMouseExited(java.awt.event.MouseEvent evt) {
        cierre.setBackground(Color.black);
    }

    private void cierreMouseClicked(java.awt.event.MouseEvent evt) {
        Inicio inicioWindow = new Inicio(username);
        inicioWindow.setVisible(true);

        // Opcional: cerrar la ventana actual si lo deseas
        this.dispose();
    }

    private void cierreMousePressed(java.awt.event.MouseEvent evt) {

    }

    private void cierreMouseDragged(java.awt.event.MouseEvent evt) {

    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(() -> {

        });
    }

    // Variables declaration - do not modify
    private javax.swing.JLabel cierre;
    private javax.swing.JPanel cierree;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel menu;
    private javax.swing.JPanel ventanas;
    // End of variables declaration
}

```



RegistrarAnimal.java

```
package com.admin;

import com.conexion.Conexion;
import com.toedter.calendar.JDateChooser;
import java.awt.Image;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.awt.image.BufferedImage;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.SimpleDateFormat;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.JPopupMenu;

/**
 *
 * @author wicho
 */

public class RegistrarAnimal extends javax.swing.JPanel {

    private String username;
    private JDateChooser jDateChooser;
    private JDateChooser jDateChooser2;
    private Image imagenSeleccionada = null;
    private ImageIcon imagenOriginal;

    public RegistrarAnimal(String username) {
        initComponents();
        this.username = username;
        System.out.println(username);
        Connection con = Conexion.conectar();
```



```

        imagenOriginal = (ImageIcon) imagen.getIcon();
        jDateChooser = new JDateChooser();
        jDateChooser.setDateFormatString("dd 'de' MMMM 'del' yyyy");
        jDateChooser.setBounds(312, 80, 340, 21);
        this.add(jDateChooser);
        jDateChooser2 = new JDateChooser();
        jDateChooser2.setDateFormatString("dd 'de' MMMM 'del' yyyy");
        jDateChooser2.setBounds(120, 340, 210, 21);
        this.add(jDateChooser2);
        precio.setEditable(false);
        guardar.setEnabled(false);
        agregarValidaciones();
        this.revalidate();
        this.repaint();
    }

    private void asignarPrecio() {
        double peso = Double.parseDouble(jTextField3.getText());
        String etapa = jComboBox5.getSelectedItem().toString();
        if (!etapa.equals("----")) {
            String sql = "SELECT " + etapa + " FROM precios LIMIT 1";

            try (Connection con = Conexion.conectar();
                PreparedStatement pst = con.prepareStatement(sql)) {
                ResultSet rs = pst.executeQuery();

                if (rs.next()) {
                    double precioPorKilo = rs.getDouble(1);
                    double precioFinal = precioPorKilo * peso;
                    precio.setText(String.format("%.2f", precioFinal));
                } else {
                    JOptionPane.showMessageDialog(this, "No se encontró el precio para esta etapa", "Error",
                        JOptionPane.ERROR_MESSAGE);
                }
            } catch (SQLException e) {
                JOptionPane.showMessageDialog(this, "Error al obtener el precio: " + e.getMessage(),
                    "Error", JOptionPane.ERROR_MESSAGE);
            }
        } else {
            precio.setText("");
        }
    }

    private void capturarFecha() {
        if (jDateChooser.getDate() != null) {
            SimpleDateFormat dateFormat = new SimpleDateFormat("dd 'de' MMMM 'del' yyyy");
            String selectedDate = dateFormat.format(jDateChooser.getDate());
            System.out.println("Fecha de adquisición: " + selectedDate);
        }

        if (jDateChooser2.getDate() != null) {

```



```

        SimpleDateFormat dateFormat = new SimpleDateFormat("dd 'de' MMMM 'del' yyyy");
        String selectedDate2 = dateFormat.format(jDateChooser2.getDate());
        System.out.println("Fecha de destete: " + selectedDate2);
    }
}

```

```
/**
```

```

 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */

```

```

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

```

```

    jLabel2 = new javax.swing.JLabel();
    imagen = new javax.swing.JLabel();
    Idtxt = new javax.swing.JLabel();
    FechaAdqTxt = new javax.swing.JLabel();
    RazaTxt = new javax.swing.JLabel();
    PesoTxt = new javax.swing.JLabel();
    jTextField1 = new javax.swing.JTextField();
    jComboBox1 = new javax.swing.JComboBox<>();
    jTextField3 = new javax.swing.JTextField();
    SexoTxt = new javax.swing.JLabel();
    jComboBox2 = new javax.swing.JComboBox<>();
    NoTxt = new javax.swing.JLabel();
    jTextField4 = new javax.swing.JTextField();
    jSeparator1 = new javax.swing.JSeparator();
    jLabel8 = new javax.swing.JLabel();
    SaludTxt = new javax.swing.JLabel();
    jComboBox3 = new javax.swing.JComboBox<>();
    ReproductivoTxt = new javax.swing.JLabel();
    jComboBox4 = new javax.swing.JComboBox<>();
    EtapaTxt = new javax.swing.JLabel();
    jComboBox5 = new javax.swing.JComboBox<>();
    CorporalTxt = new javax.swing.JLabel();
    jComboBox6 = new javax.swing.JComboBox<>();
    AlimenticioTxt = new javax.swing.JLabel();
    jComboBox7 = new javax.swing.JComboBox<>();
    SanitarioTxt = new javax.swing.JLabel();
    jComboBox8 = new javax.swing.JComboBox<>();
    jSeparator2 = new javax.swing.JSeparator();
    jLabel15 = new javax.swing.JLabel();
    jLabel1 = new javax.swing.JLabel();
    jComboBox9 = new javax.swing.JComboBox<>();
    jLabel3 = new javax.swing.JLabel();
    jTextField2 = new javax.swing.JTextField();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    jTextField5 = new javax.swing.JTextField();

```

```

jLabel6 = new javax.swing.JLabel();
jComboBox10 = new javax.swing.JComboBox<>();
jComboBox11 = new javax.swing.JComboBox<>();
jLabel7 = new javax.swing.JLabel();
guardar = new javax.swing.JButton();
jButton2 = new javax.swing.JButton();
jLabel9 = new javax.swing.JLabel();
precio = new javax.swing.JTextField();

setBackground(new java.awt.Color(255, 255, 255));
setLayout(null);

jLabel2.setFont(new java.awt.Font("Roboto Medium", 0, 12)); // NOI18N
jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel2.setText("REGISTRO DEL GANADO");
add(jLabel2);
jLabel2.setBounds(0, 0, 700, 30);

imagen.setIcon(new javax.swing.ImageIcon(getClass().getResource("/iconos/añadir.png"))); //
NOI18N
imagen.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
imagen.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        imagenMouseClicked(evt);
    }
});
add(imagen);
imagen.setBounds(10, 40, 130, 120);

Idtxt.setText("Numero de ID:");
add(Idtxt);
Idtxt.setBounds(180, 50, 90, 16);

FechaAdqTxt.setText("Fecha de Adquisicion:");
add(FechaAdqTxt);
FechaAdqTxt.setBounds(180, 80, 130, 16);

RazaTxt.setText("Tipo o Raza:");
add(RazaTxt);
RazaTxt.setBounds(180, 110, 80, 16);

PesoTxt.setText("Peso:");
add(PesoTxt);
PesoTxt.setBounds(180, 140, 50, 16);
add(jTextField1);
jTextField1.setBounds(270, 50, 140, 22);

jComboBox1.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "----",
"Angus", "Hereford", "Charolais", "Limousin", "Brahman", "Simmental", "Holstein", "Jersey",
"Guernsey", "Pardo Suizo", "Ayrshire", "Shorthorn", "Normando", "Fleckvieh", "Nellore", "Criollo" }));
add(jComboBox1);
jComboBox1.setBounds(120, 380, 210, 22);

```

```

jTextField3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField3ActionPerformed(evt);
    }
});
add(jTextField3);
jTextField3.setBounds(230, 140, 160, 22);

SexoTxt.setText("Sexo:");
add(SexoTxt);
SexoTxt.setBounds(400, 140, 50, 16);

jComboBox2.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "----",
"Macho", "Hembra" }));
add(jComboBox2);
jComboBox2.setBounds(440, 140, 210, 22);

NoTxt.setText("No. Economico Propiedad:");
add(NoTxt);
NoTxt.setBounds(420, 50, 160, 20);
add(jTextField4);
jTextField4.setBounds(580, 50, 73, 22);

jSeparator1.setForeground(new java.awt.Color(153, 153, 153));
add(jSeparator1);
jSeparator1.setBounds(140, 180, 510, 10);

jLabel8.setText("Estados o Categorías");
add(jLabel8);
jLabel8.setBounds(10, 170, 130, 20);

SaludTxt.setText("Salud:");
add(SaludTxt);
SaludTxt.setBounds(10, 200, 60, 20);

jComboBox3.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "----",
"Saludable", "Enfermo", "En recuperación", "Lesionado" }));
add(jComboBox3);
jComboBox3.setBounds(60, 200, 150, 22);

ReproductivoTxt.setText("Reproductivo:");
add(ReproductivoTxt);
ReproductivoTxt.setBounds(220, 200, 90, 20);

jComboBox4.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "----",
"Gestante", "En celo", "Vacío", "Parida", "Activo", "Inactivo" }));
add(jComboBox4);
jComboBox4.setBounds(310, 200, 140, 22);

EtapaTxt.setText("Etapa:");
add(EtapaTxt);

```



```

Etapatxt.setBounds(462, 200, 50, 20);

jComboBox5.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "----",
"Ternero", "Joven", "Adulto", "Anciano" }));
jComboBox5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jComboBox5ActionPerformed(evt);
    }
});
add(jComboBox5);
jComboBox5.setBounds(500, 200, 150, 22);

CorporalTxt.setText("Corporal:");
add(CorporalTxt);
CorporalTxt.setBounds(10, 240, 70, 20);

jComboBox6.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "----",
"Óptima", "Baja condición ", "Sobrepeso" }));
add(jComboBox6);
jComboBox6.setBounds(70, 240, 140, 22);

AlimenticioTxt.setText("Alimenticio:");
add(AlimenticioTxt);
AlimenticioTxt.setBounds(430, 240, 80, 20);

jComboBox7.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "----", "Bien
alimentado", "En engorda", "Desnutrido", "En suplementación" }));
add(jComboBox7);
jComboBox7.setBounds(500, 240, 150, 22);

SanitarioTxt.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
SanitarioTxt.setText("Sanitario:");
add(SanitarioTxt);
SanitarioTxt.setBounds(210, 240, 60, 20);

jComboBox8.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "----",
"Vacunado", "En cuarentena", "En tratamiento", "Libre de enfermedades" }));
add(jComboBox8);
jComboBox8.setBounds(270, 240, 154, 22);

jSeparator2.setForeground(new java.awt.Color(153, 153, 153));
add(jSeparator2);
jSeparator2.setBounds(140, 280, 510, 10);

jLabel15.setText("Datos de Adquisicion");
add(jLabel15);
jLabel15.setBounds(10, 270, 140, 20);

jLabel1.setText("Modo de Adquisicion:");
add(jLabel1);
jLabel1.setBounds(10, 300, 140, 20);

```

```

jComboBox9.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "----",
"Compra", "Reproducción Propia", "Intercambio o Trueque", "Donación o Cesión", "Arrendamiento o
Alquiler", "Préstamo o Comodato", "Herencia o Sucesión" }));
add(jComboBox9);
jComboBox9.setBounds(140, 300, 190, 22);

jLabel3.setText("Peso al Adquirir:");
add(jLabel3);
jLabel3.setBounds(346, 300, 100, 20);
add(jTextField2);
jTextField2.setBounds(450, 300, 200, 22);

jLabel4.setText("Fecha al Destete:");
add(jLabel4);
jLabel4.setBounds(10, 340, 100, 20);

jLabel5.setText("Peso al Destete:");
add(jLabel5);
jLabel5.setBounds(342, 340, 100, 20);
add(jTextField5);
jTextField5.setBounds(450, 340, 200, 22);

jLabel6.setText("Raza del Padre:");
add(jLabel6);
jLabel6.setBounds(10, 380, 90, 20);

jComboBox10.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "----",
"Angus", "Hereford", "Charolais", "Limousin", "Brahman", "Simmental", "Holstein", "Jersey",
"Guernsey", "Pardo Suizo", "Ayrshire", "Shorthorn", "Normando", "Fleckvieh", "Nellore", "Criollo" }));
add(jComboBox10);
jComboBox10.setBounds(310, 110, 340, 22);

jComboBox11.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "----",
"Angus", "Hereford", "Charolais", "Limousin", "Brahman", "Simmental", "Holstein", "Jersey",
"Guernsey", "Pardo Suizo", "Ayrshire", "Shorthorn", "Normando", "Fleckvieh", "Nellore", "Criollo" }));
add(jComboBox11);
jComboBox11.setBounds(450, 380, 200, 22);

jLabel7.setText("Raza de la Madre:");
add(jLabel7);
jLabel7.setBounds(340, 380, 110, 20);

guardar.setText("Guardar");
guardar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        guardarActionPerformed(evt);
    }
});
add(guardar);
guardar.setBounds(420, 430, 72, 23);

jButton2.setText("Limpiar");

```



```

jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});
add(jButton2);
jButton2.setBounds(320, 430, 72, 23);

jLabel9.setText("Precio:");
add(jLabel9);
jLabel9.setBounds(520, 10, 50, 16);
add(precio);
precio.setBounds(570, 10, 64, 22);
} // </editor-fold>

```

```

private void agregarValidaciones() {
    ActionListener validarCampos = new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {

            validarFormulario();
        }
    };
}

```

```

ItemListener validarComboBox = new ItemListener() {
    @Override
    public void itemStateChanged(ItemEvent e) {
        validarFormulario();
    }
};

```

// Agregar ActionListener a los campos de texto y ItemListener a los ComboBox

```

jTextField1.addActionListener(validarCampos);
jTextField3.addActionListener(validarCampos);
jTextField4.addActionListener(validarCampos);
jTextField2.addActionListener(validarCampos);
jTextField5.addActionListener(validarCampos);

jComboBox1.addItemListener(validarComboBox);
jComboBox2.addItemListener(validarComboBox);
jComboBox3.addItemListener(validarComboBox);
jComboBox4.addItemListener(validarComboBox);
jComboBox5.addItemListener(validarComboBox);
jComboBox6.addItemListener(validarComboBox);
jComboBox7.addItemListener(validarComboBox);
jComboBox8.addItemListener(validarComboBox);
jComboBox9.addItemListener(validarComboBox);
jComboBox10.addItemListener(validarComboBox);
jComboBox11.addItemListener(validarComboBox);
}

```

```

private void validarFormulario() {
    boolean camposLlenos = !jTextField1.getText().isEmpty() &&
        !jTextField3.getText().isEmpty() &&
        !jTextField4.getText().isEmpty() &&
        !jTextField2.getText().isEmpty() &&
        !jTextField5.getText().isEmpty() &&
        jDateChooser.getDate() != null &&
        jDateChooser2.getDate() != null;

    boolean combosValidos = !jComboBox1.getSelectedItem().equals("----") &&
        !jComboBox2.getSelectedItem().equals("----") &&
        !jComboBox3.getSelectedItem().equals("----") &&
        !jComboBox4.getSelectedItem().equals("----") &&
        !jComboBox5.getSelectedItem().equals("----") &&
        !jComboBox6.getSelectedItem().equals("----") &&
        !jComboBox7.getSelectedItem().equals("----") &&
        !jComboBox8.getSelectedItem().equals("----") &&
        !jComboBox9.getSelectedItem().equals("----") &&
        !jComboBox10.getSelectedItem().equals("----") &&
        !jComboBox11.getSelectedItem().equals("----");

    guardar.setEnabled(camposLlenos && combosValidos);
}

//Eleccion de imagen
private void imagenMouseClicked(java.awt.event.MouseEvent evt) {
    JFileChooser fileChooser = new JFileChooser();
    fileChooser.setDialogTitle("Selecciona una imagen");
    int resultado = fileChooser.showOpenDialog(this);

    if (resultado == JFileChooser.APPROVE_OPTION) {
        File archivoSeleccionado = fileChooser.getSelectedFile();
        try {
            BufferedImage imagenOriginal = ImageIO.read(archivoSeleccionado);
            imagenSeleccionada = imagenOriginal.getScaledInstance(imagen.getWidth(),
imagen.getHeight(), Image.SCALE_SMOOTH);
            imagen.setIcon(new ImageIcon(imagenSeleccionada));
        } catch (IOException ex) {
            ex.printStackTrace();
            JOptionPane.showMessageDialog(this, "Error al cargar la imagen", "Error",
JOptionPane.ERROR_MESSAGE);
        }
    }
}

private byte[] convertirImagenABYtes() {
    if (imagenSeleccionada != null) {
        try {
            BufferedImage bufferedImage = new BufferedImage(
                imagenSeleccionada.getWidth(null),
                imagenSeleccionada.getHeight(null),
                BufferedImage.TYPE_INT_RGB);
            bufferedImage.getGraphics().drawImage(imagenSeleccionada, 0, 0, null);

```

```

        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        ImageIO.write(bufferedImage, "jpg", baos);
        return baos.toByteArray();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
return null;
}

```

```

private void guardarActionPerformed(java.awt.event.ActionEvent evt) {
    // Capturar datos
    String id = jTextField1.getText();
    String raza = (String) jComboBox10.getSelectedItem();
    String peso = jTextField3.getText();
    String sexo = (String) jComboBox2.getSelectedItem();
    String numeroEconomico = jTextField4.getText();
    String salud = (String) jComboBox3.getSelectedItem();
    String reproductivo = (String) jComboBox4.getSelectedItem();
    String etapa = (String) jComboBox5.getSelectedItem();
    String corporal = (String) jComboBox6.getSelectedItem();
    String alimenticio = (String) jComboBox7.getSelectedItem();
    String sanitario = (String) jComboBox8.getSelectedItem();
    String modoAdquisicion = (String) jComboBox9.getSelectedItem();
    String pesoAdquirido = jTextField2.getText();
    String pesoDestete = jTextField5.getText();
    String razaPadre = (String) jComboBox1.getSelectedItem();
    String razaMadre = (String) jComboBox11.getSelectedItem();

    SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
    String fechaAdquisicion = dateFormat.format(jDateChooser.getDate());
    String fechaDestete = dateFormat.format(jDateChooser2.getDate());

    byte[] imagenBytes = convertirImagenABYtes();
    String precioA = precio.getText();

    // Guardar en la base de datos
    try (Connection con = Conexion.conectar()) {
        String sql = "INSERT INTO animales (id, raza, peso, sexo, numero_economico, salud, reproductivo, etapa, " +
            "corporal, alimenticio, sanitario, modo_adquisicion, peso_adquirido, peso_destete, raza_padre, " +
            "raza_madre, fecha_adquisicion, fecha_destete, imagen, precio) VALUES (?, ?, ?, ?, " +
            "? , ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";

        PreparedStatement statement = con.prepareStatement(sql);
        statement.setString(1, id);
        statement.setString(2, raza);
        statement.setString(3, peso);
        statement.setString(4, sexo);
    }
}

```



```

        statement.setString(5, numeroEconomico);
        statement.setString(6, salud);
        statement.setString(7, reproductivo);
        statement.setString(8, etapa);
        statement.setString(9, corporal);
        statement.setString(10, alimenticio);
        statement.setString(11, sanitario);
        statement.setString(12, modoAdquisicion);
        statement.setString(13, pesoAdquirido);
        statement.setString(14, pesoDestete);
        statement.setString(15, razaPadre);
        statement.setString(16, razaMadre);
        statement.setString(17, fechaAdquisicion);
        statement.setString(18, fechaDestete);
        statement.setBytes(19, imagenBytes);
        statement.setString(20, precioA);

        statement.executeUpdate();
        JOptionPane.showMessageDialog(this, "Animal registrado exitosamente.");
        String insertHistorial = "INSERT INTO historial (fecha, movimiento, id, Dato, valorAnterior,
valorNuevo) VALUES (?, ?, ?, ?, ?, ?)";

try (PreparedStatement hisStmt = con.prepareStatement(insertHistorial)) {
    LocalDateTime fechaHoraActual = LocalDateTime.now();
    String fecha = fechaHoraActual.toLocalDate().toString();

    hisStmt.setString(1, fecha);
    hisStmt.setString(2, "Creacion");
    hisStmt.setString(3, id);

    String[] datos = {
        "raza", "peso", "sexo", "numeroEconomico", "salud", "reproductivo",
        "etapa", "corporal", "alimenticio", "sanitario", "modoAdquisicion",
        "pesoAdquirido", "pesoDestete", "razaPadre", "razaMadre", "fechaAdquisicion",
        "fechaDestete"
    };
    String[] valores = {
        raza, peso, sexo, numeroEconomico, salud, reproductivo,
        etapa, corporal, alimenticio, sanitario, modoAdquisicion,
        pesoAdquirido, pesoDestete, razaPadre, razaMadre, fechaAdquisicion,
        fechaDestete
    };

    for (int i = 0; i < datos.length; i++) {
        hisStmt.setString(4, datos[i]);
        hisStmt.setString(5, "---");
        hisStmt.setString(6, valores[i]);

        hisStmt.executeUpdate();
    }
}

System.out.println("Nombre de usuario: " + username);

```



```

        String insertMovimiento = "INSERT INTO movimientos (fecha, hora, usuario,privilegio,
movimiento, id) VALUES (?, ?, ?, ?, ?, ?)";
        try (PreparedStatement movStmt = con.prepareStatement(insertMovimiento)) {
            LocalDateTime fechaHoraActual = LocalDateTime.now();
            String fecha = fechaHoraActual.toLocalDate().toString();
            String hora = fechaHoraActual.toLocalTime().format(DateTimeFormatter.ofPattern("HH:mm:ss"));

            movStmt.setString(1, fecha);
            movStmt.setString(2, hora);
            movStmt.setString(3, username);
            movStmt.setString(4, "Administrador");
            movStmt.setString(5, "Registro de un Animal");
            movStmt.setString(6, id);

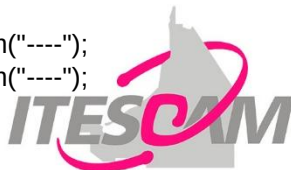
            movStmt.executeUpdate();
        }

//valore predeterminados
precio.setText("");
jTextField1.setText("");
jComboBox1.setSelectedItem("----");
jComboBox2.setSelectedItem("----");
jComboBox3.setSelectedItem("----");
jComboBox4.setSelectedItem("----");
jComboBox5.setSelectedItem("----");
jComboBox6.setSelectedItem("----");
jComboBox7.setSelectedItem("----");
jComboBox8.setSelectedItem("----");
jComboBox9.setSelectedItem("----");
jComboBox10.setSelectedItem("----");
jComboBox11.setSelectedItem("----");
jTextField4.setText("");
jTextField2.setText("");
jTextField5.setText("");
jDateChooser.setDate(null);
jDateChooser2.setDate(null);
imagen.setIcon(imagenOriginal);
jTextField3.setText("");

    } catch (SQLException ex) {
        Logger.getLogger(RegistrarAnimal.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
//valore predeterminados
jTextField1.setText("");
jComboBox1.setSelectedItem("----");
jTextField3.setText("");
jComboBox2.setSelectedItem("----");
jComboBox3.setSelectedItem("----");

```



```

        jComboBox4.setSelectedItem("----");
        jComboBox5.setSelectedItem("----");
        jComboBox6.setSelectedItem("----");
        jComboBox7.setSelectedItem("----");
        jComboBox8.setSelectedItem("----");
        jComboBox9.setSelectedItem("----");
        jComboBox10.setSelectedItem("----");
        jComboBox11.setSelectedItem("----");
        jTextField4.setText("");
        jTextField2.setText("");
        jTextField5.setText("");
        jDateChooser.setDate(null);
        jDateChooser2.setDate(null);
        imagen.setIcon(imagenOriginal);
    }

    private void jComboBox5ActionPerformed(java.awt.event.ActionEvent evt) {
        asignarPrecio();
    }

    private void jTextField3ActionPerformed(java.awt.event.ActionEvent evt) {
        asignarPrecio();
    }

```

```

// Variables declaration - do not modify
private javax.swing.JLabel AlimenticioTxt;
private javax.swing.JLabel CorporalTxt;
private javax.swing.JLabel EtapaTxt;
private javax.swing.JLabel FechaAdqTxt;
private javax.swing.JLabel Idtxt;
private javax.swing.JLabel NoTxt;
private javax.swing.JLabel PesoTxt;
private javax.swing.JLabel RazaTxt;
private javax.swing.JLabel ReproductivoTxt;
private javax.swing.JLabel SaludTxt;
private javax.swing.JLabel SanitarioTxt;
private javax.swing.JLabel SexoTxt;
private javax.swing.JButton guardar;
private javax.swing.JLabel imagen;
private javax.swing.JButton jButton2;
private javax.swing.JComboBox<String> jComboBox1;
private javax.swing.JComboBox<String> jComboBox10;
private javax.swing.JComboBox<String> jComboBox11;
private javax.swing.JComboBox<String> jComboBox2;
private javax.swing.JComboBox<String> jComboBox3;
private javax.swing.JComboBox<String> jComboBox4;
private javax.swing.JComboBox<String> jComboBox5;
private javax.swing.JComboBox<String> jComboBox6;
private javax.swing.JComboBox<String> jComboBox7;
private javax.swing.JComboBox<String> jComboBox8;
private javax.swing.JComboBox<String> jComboBox9;

```

```

private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JSeparator jSeparator2;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField4;
private javax.swing.JTextField jTextField5;
private javax.swing.JTextField precio;
// End of variables declaration
}

```

RegistrarAnimales.java

```
package com.admin;
```

```

import com.frame.Inicio;
import java.awt.BorderLayout;
import java.awt.Color;
import javax.swing.border.Border;

```

```

/**
 *
 * @author wicho
 */
public class RegistrarAnimales extends javax.swing.JFrame {
    int xMouse, yMouse;
    private String username;
    /**
     * Creates new form Home
     */
    public RegistrarAnimales(String username) {
        initComponents();
        this.username = username;
        RegistrarAnimal pruebaPanel = new RegistrarAnimal(username);
        ventanas.removeAll();
        ventanas.setLayout(new BorderLayout());
        pruebaPanel.setSize(ventanas.getWidth(), ventanas.getHeight());
        ventanas.add(pruebaPanel);
        ventanas.revalidate();
        ventanas.repaint();
        pruebaPanel.setVisible(true);
    }
}

```



```

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    cierree = new javax.swing.JPanel();
    cierre = new javax.swing.JLabel();
    jPanel2 = new javax.swing.JPanel();
    ventanas = new javax.swing.JPanel();
    menu = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel1 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setUndecorated(true);
    getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    cierree.setBackground(new java.awt.Color(0, 0, 0));
    cierree.addMouseMotionListener(new java.awt.event.MouseMotionAdapter() {
        public void mouseDragged(java.awt.event.MouseEvent evt) {
            cierreeMouseDragged(evt);
        }
    });
    cierree.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            cierreeMouseClicked(evt);
        }
        public void mouseEntered(java.awt.event.MouseEvent evt) {
            cierreeMouseEntered(evt);
        }
        public void mouseExited(java.awt.event.MouseEvent evt) {
            cierreeMouseExited(evt);
        }
        public void mousePressed(java.awt.event.MouseEvent evt) {
            cierreeMousePressed(evt);
        }
    });

    cierre.setFont(new java.awt.Font("Roboto Medium", 0, 18)); // NOI18N
    cierre.setForeground(new java.awt.Color(255, 255, 255));
    cierre.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    cierre.setText("X");

    javax.swing.GroupLayout cierreeLayout = new javax.swing.GroupLayout(cierree);
    cierree.setLayout(cierreeLayout);
    cierreeLayout.setHorizontalGroup(

```

```

        cierreLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
cierreLayout.createSequentialGroup()
                .addComponent(cierre, javax.swing.GroupLayout.DEFAULT_SIZE, 24,
Short.MAX_VALUE)
                .addContainerGap())
            );
        cierreLayout.setVerticalGroup(
            cierreLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
cierreLayout.createSequentialGroup()
                    .addGap(0, 0, Short.MAX_VALUE)
                    .addComponent(cierre, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE))
                );

        getContentPane().add(cierre, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 30, -
1));

        jPanel2.setBackground(new java.awt.Color(0, 0, 0));

        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGap(0, 840, Short.MAX_VALUE)
        );
        jPanel2Layout.setVerticalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGap(0, 31, Short.MAX_VALUE)
        );

        getContentPane().add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 0, 840,
31));

        javax.swing.GroupLayout ventanasLayout = new javax.swing.GroupLayout(ventanas);
        ventanas.setLayout(ventanasLayout);
        ventanasLayout.setHorizontalGroup(
            ventanasLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGap(0, 690, Short.MAX_VALUE)
        );
        ventanasLayout.setVerticalGroup(
            ventanasLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGap(0, 470, Short.MAX_VALUE)
        );

        getContentPane().add(ventanas, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 30,
690, 470));

        menu.setBackground(new java.awt.Color(255, 153, 0));
        menu.setPreferredSize(new java.awt.Dimension(870, 500));
        menu.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

```

```

        jLabel2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO.jpg"))); //
NOI18N
        menu.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 360, -1, 110));

        jLabel3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO-
removebg-preview.png"))); // NOI18N
        menu.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, -1));

        jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/campo (1).jpg"))); //
NOI18N
        menu.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, 470));

        getContentPane().add(menu, new org.netbeans.lib.awtextra.AbsoluteConstraints(690, 30, 180,
470));

        pack();
        setLocationRelativeTo(null);
    } // </editor-fold>

    private void cierreMouseEntered(java.awt.event.MouseEvent evt) {
        cierre.setBackground(Color.red);
    }

    private void cierreMouseExited(java.awt.event.MouseEvent evt) {
        cierre.setBackground(Color.black);
    }

    private void cierreMouseClicked(java.awt.event.MouseEvent evt) {
        Inicio inicioWindow = new Inicio(username);
        inicioWindow.setVisible(true);
        this.dispose();
    }

    private void cierreMousePressed(java.awt.event.MouseEvent evt) {

    }

    private void cierreMouseDragged(java.awt.event.MouseEvent evt) {

    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(() -> {

        });
    }

```



```

// Variables declaration - do not modify
private javax.swing.JLabel cierre;
private javax.swing.JPanel cierree;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel menu;
private javax.swing.JPanel ventanas;
// End of variables declaration

private void setBorder(Border createEmptyBorder) {
    throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
}
}

```

Registro.java

```
package com.admin;
```

```

/**
 *
 * @author wicho
 */
public class Registro {
    private String id;
    private String campo2;
    private String campo3;
    private String campo4;
    private String fechaAdq;
    private String campo5;
    private String raza;
    private String peso;
    private String sexo;
    private String salud;
    private String reproductivo;
    private String etapa;
    private String corporal;
    private String sanitario;
    private String alimenticio;

    // Constructor y Getters
    public Registro(String id, String campo2, String campo3, String campo4, String fechaAdq, String
campo5,
        String raza, String peso, String sexo, String salud, String reproductivo,
        String etapa, String corporal, String sanitario, String alimenticio) {
        this.id = id;
        this.campo2 = campo2;
        this.campo3 = campo3;
        this.campo4 = campo4;
        this.fechaAdq = fechaAdq;

```



```

        this.campo5 = campo5;
        this.raza = raza;
        this.peso = peso;
        this.sexo = sexo;
        this.salud = salud;
        this.reproductivo = reproductivo;
        this.etapa = etapa;
        this.corporal = corporal;
        this.sanitario = sanitario;
        this.alimenticio = alimenticio;
    }

    public String getId() { return id; }
    public String getCampo2() { return campo2; }
    public String getCampo3() { return campo3; }
    public String getCampo4() { return campo4; }
    public String getFechaAdq() { return fechaAdq; }
    public String getCampo5() { return campo5; }
    public String getRaza() { return raza; }
    public String getPeso() { return peso; }
    public String getSexo() { return sexo; }
    public String getSalud() { return salud; }
    public String getReproductivo() { return reproductivo; }
    public String getEtapa() { return etapa; }
    public String getCorporal() { return corporal; }
    public String getSanitario() { return sanitario; }
    public String getAlimenticio() { return alimenticio; }
}

```

Conexión.java

```
package com.conexion;
```

```
import java.sql.*;
```

```

public class Conexion {
    //base de datos en linea
    //private static String url = "jdbc:mysql://sql5.freemsqldatabase.com:3306/sql5742639";
    //private static String user = "sql5742639";
    //private static String password = "Q4RkVIFnLt";

    //base de datos local
    private static String url = "jdbc:mysql://localhost:3306/mi-granja";
    private static String user = "root";
    private static String password = "";

    public static Connection conectar() {
        Connection con = null;
        try {
            con = DriverManager.getConnection(url, user, password);

```



```

        System.out.println("Conexion exitosa");
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return con;
}

public static boolean insertarUsuario(String username, String password, String masterPassword,
String privilegio) {
    String insertQuery = "INSERT INTO `registros-us` (Usuario, Password, PasswordMaster,
Privilegio) VALUES (?, ?, ?, ?)";

    try (Connection con = conectar(); PreparedStatement stmt =
con.prepareStatement(insertQuery)) {

        stmt.setString(1, username);
        stmt.setString(2, password);
        stmt.setString(3, masterPassword);
        stmt.setString(4, privilegio);
        stmt.executeUpdate();

        return true; // Éxito en la inserción
    } catch (SQLException ex) {
        ex.printStackTrace();
        return false; // Error en la inserción
    }
}

public static Connection getConnection() {
    throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
}
}

```

Inicio.java

```

package com.frame;

import Busqueda.HomeBusqueda;
import BusquedaLotes.HomeBusquedaLotes;
import Ganado.HomeGanado;
import com.Administracion.Gasto;
import com.Precios.PreciosV;
import com.admin.BuscarAnimal;
import com.admin.Ganado;
import com.admin.RegistrarAnimales;
import com.conexion.Conexion;
import com.login.Login;
import com.movimientos.Movimientos;
import com.registrar.RegistrarUsuario;
import java.awt.Color;

```



```

import java.sql.Connection;

/**
 *
 * @author wicho
 */
public class Inicio extends javax.swing.JFrame {
    private String username;
    /**
     * Creates new form Inicio
     */
    public Inicio(String username) {
        initComponents();
        this.username = username;
        System.out.println(username);
        this.setLocationRelativeTo(null);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        bg = new javax.swing.JPanel();
        encabezado = new javax.swing.JPanel();
        exitBtn = new javax.swing.JPanel();
        exitTxt = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        bgIniciarSesion = new javax.swing.JPanel();
        btnIniciarSesion1 = new javax.swing.JLabel();
        txtIniciarSesion = new javax.swing.JLabel();
        bgRegistrarUsuario = new javax.swing.JPanel();
        btnIniciarSesion2 = new javax.swing.JLabel();
        txtRegistrar = new javax.swing.JLabel();
        bgMovimientos = new javax.swing.JPanel();
        btnIniciarSesion3 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        bgGanado = new javax.swing.JPanel();
        btnIniciarSesion = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        bgBusqueda = new javax.swing.JPanel();
        btnIniciarSesion5 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        bgBusquedaLotes = new javax.swing.JPanel();
        btnIniciarSesion4 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
    }

```

```

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setUndecorated(true);

bg.setBackground(new java.awt.Color(255, 255, 255));
bg.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

encabezado.setBackground(new java.awt.Color(0, 0, 0));
encabezado.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

exitBtn.setBackground(new java.awt.Color(0, 0, 0));
exitBtn.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        exitBtnMouseClicked(evt);
    }
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        exitBtnMouseEntered(evt);
    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        exitBtnMouseExited(evt);
    }
});

exitTxt.setFont(new java.awt.Font("Roboto Medium", 0, 24)); // NOI18N
exitTxt.setForeground(new java.awt.Color(255, 255, 255));
exitTxt.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
exitTxt.setText("X");

javax.swing.GroupLayout exitBtnLayout = new javax.swing.GroupLayout(exitBtn);
exitBtn.setLayout(exitBtnLayout);
exitBtnLayout.setHorizontalGroup(
    exitBtnLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            exitBtnLayout.createSequentialGroup()
                .addGap(0, 0, Short.MAX_VALUE)
                .addComponent(exitTxt, javax.swing.GroupLayout.PREFERRED_SIZE, 37,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap()
        );
exitBtnLayout.setVerticalGroup(
    exitBtnLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(exitBtnLayout.createSequentialGroup()
            .addContainerGap()
            .addComponent(exitTxt)
            .addGap(0, 0, Short.MAX_VALUE)
        );

encabezado.add(exitBtn, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, -1));

jLabel7.setFont(new java.awt.Font("Roboto Medium", 0, 24)); // NOI18N
jLabel7.setForeground(new java.awt.Color(255, 255, 255));
jLabel7.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel7.setText("BIENVENIDO AL SISTEMA");
encabezado.add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 30, 870, 44));

```

```

jLabel8.setFont(new java.awt.Font("Roboto Light", 0, 12)); // NOI18N
jLabel8.setForeground(new java.awt.Color(255, 255, 255));
jLabel8.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel8.setText("PORFAVOR ELIJA UNA OPCION");
encabezado.add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 80, 864, -1));

bg.add(encabezado, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, 110));

bgIniciarSesion.setBackground(new java.awt.Color(255, 255, 255));
bgIniciarSesion.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        bgIniciarSesionMouseClicked(evt);
    }
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        bgIniciarSesionMouseEntered(evt);
    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        bgIniciarSesionMouseExited(evt);
    }
});

btnIniciarSesion1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnIniciarSesion1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/iniciar-
sesion.png"))); // NOI18N

txtIniciarSesion.setFont(new java.awt.Font("Roboto Medium", 0, 12)); // NOI18N
txtIniciarSesion.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
txtIniciarSesion.setText("VER GANADO");

javax.swing.GroupLayout bgIniciarSesionLayout = new
javax.swing.GroupLayout(bgIniciarSesion);
bgIniciarSesion.setLayout(bgIniciarSesionLayout);
bgIniciarSesionLayout.setHorizontalGroup(
    bgIniciarSesionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            bgIniciarSesionLayout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(btnIniciarSesion1, javax.swing.GroupLayout.PREFERRED_SIZE, 128,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(15, 15, 15))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            bgIniciarSesionLayout.createSequentialGroup()
                .addGap(0, 0, Short.MAX_VALUE)
                .addComponent(txtIniciarSesion, javax.swing.GroupLayout.PREFERRED_SIZE, 160,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap()
        ));
bgIniciarSesionLayout.setVerticalGroup(
    bgIniciarSesionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(bgIniciarSesionLayout.createSequentialGroup()
            .addGroup(bgIniciarSesionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(bgIniciarSesionLayout.createSequentialGroup()
                    .addContainerGap()

```

```

        .addComponent(btnIniciarSesion1, javax.swing.GroupLayout.PREFERRED_SIZE, 128,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(txtIniciarSesion)
        .addContainerGap()
    );

    bg.add(bgIniciarSesion, new org.netbeans.lib.awtextra.AbsoluteConstraints(60, 120, 160,
160));

    bgRegistrarUsuario.setBackground(new java.awt.Color(255, 255, 255));
    bgRegistrarUsuario.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            bgRegistrarUsuarioMouseClicked(evt);
        }
        public void mouseEntered(java.awt.event.MouseEvent evt) {
            bgRegistrarUsuarioMouseEntered(evt);
        }
        public void mouseExited(java.awt.event.MouseEvent evt) {
            bgRegistrarUsuarioMouseExited(evt);
        }
    });

    btnIniciarSesion2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    btnIniciarSesion2.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/usuariosNuevos.png"))); // NOI18N

    txtRegistrar.setFont(new java.awt.Font("Roboto Medium", 0, 12)); // NOI18N
    txtRegistrar.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    txtRegistrar.setText("REGISTRAR USUARIO");

    javax.swing.GroupLayout bgRegistrarUsuarioLayout = new
javax.swing.GroupLayout(bgRegistrarUsuario);
    bgRegistrarUsuario.setLayout(bgRegistrarUsuarioLayout);
    bgRegistrarUsuarioLayout.setHorizontalGroup(

    bgRegistrarUsuarioLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
bgRegistrarUsuarioLayout.createSequentialGroup()
            .addGap(0, 0, Short.MAX_VALUE)
            .addComponent(txtRegistrar, javax.swing.GroupLayout.PREFERRED_SIZE, 160,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGroup(bgRegistrarUsuarioLayout.createSequentialGroup()
                .addGap(16, 16, 16)
                .addComponent(btnIniciarSesion2, javax.swing.GroupLayout.PREFERRED_SIZE, 128,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );
    bgRegistrarUsuarioLayout.setVerticalGroup(

    bgRegistrarUsuarioLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addGroup(bgRegistrarUsuarioLayout.createSequentialGroup()
            .addContainerGap()
            .addComponent(btnIniciarSesion2, javax.swing.GroupLayout.PREFERRED_SIZE, 128,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtRegistrar)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );

```

```

bg.add(bgRegistrarUsuario, new org.netbeans.lib.awtextra.AbsoluteConstraints(340, 120, 160,
160));

```

```

bgMovimientos.setBackground(new java.awt.Color(255, 255, 255));
bgMovimientos.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        bgMovimientosMouseClicked(evt);
    }
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        bgMovimientosMouseEntered(evt);
    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        bgMovimientosMouseExited(evt);
    }
});

```

```

btnIniciarSesion3.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnIniciarSesion3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/reloj.png"))); //
NOI18N

```

```

jLabel3.setFont(new java.awt.Font("Roboto Medium", 0, 12)); // NOI18N
jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel3.setText("MOVIMIENTOS");

```

```

javax.swing.GroupLayout      bgMovimientosLayout      =      new
javax.swing.GroupLayout(bgMovimientos);
bgMovimientos.setLayout(bgMovimientosLayout);
bgMovimientosLayout.setHorizontalGroup(
    bgMovimientosLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            bgMovimientosLayout.createSequentialGroup()
                .addGroup(bgMovimientosLayout.createParallelGroup()
                    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(btnIniciarSesion3, javax.swing.GroupLayout.PREFERRED_SIZE, 128,
                        javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(15, 15, 15))
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                    bgMovimientosLayout.createSequentialGroup()
                        .addGap(0, 0, Short.MAX_VALUE)
                        .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 160,
                            javax.swing.GroupLayout.PREFERRED_SIZE))
                .addContainerGap()
            )
);
bgMovimientosLayout.setVerticalGroup(
    bgMovimientosLayout.createSequentialGroup()
        .addContainerGap()
        .addGroup(bgMovimientosLayout.createParallelGroup()
            .addGroup(bgMovimientosLayout.createSequentialGroup()
                .add(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .add(btnIniciarSesion3, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(10, 10, 10))
            .addGroup(bgMovimientosLayout.createSequentialGroup()
                .addContainerGap(10, 10, 10)
                .add(btnRegistrarUsuario, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(10, 10, 10))
        )
);

```

```

        .addGroup(bgMovimientosLayout.createSequentialGroup())
        .addContainerGap()
        .addComponent(btnIniciarSesion3, javax.swing.GroupLayout.PREFERRED_SIZE, 128,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jLabel3)
        .addContainerGap()
    );

    bg.add(bgMovimientos, new org.netbeans.lib.awtextra.AbsoluteConstraints(620, 120, 160,
160));

    bgGanado.setBackground(new java.awt.Color(255, 255, 255));
    bgGanado.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            bgGanadoMouseClicked(evt);
        }
        public void mouseEntered(java.awt.event.MouseEvent evt) {
            bgGanadoMouseEntered(evt);
        }
        public void mouseExited(java.awt.event.MouseEvent evt) {
            bgGanadoMouseExited(evt);
        }
    });

    btnIniciarSesion.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    btnIniciarSesion.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/documentacion.png"))); // NOI18N

    jLabel4.setFont(new java.awt.Font("Roboto Medium", 0, 12)); // NOI18N
    jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    jLabel4.setText("PRECIOS");

    javax.swing.GroupLayout bgGanadoLayout = new javax.swing.GroupLayout(bgGanado);
    bgGanado.setLayout(bgGanadoLayout);
    bgGanadoLayout.setHorizontalGroup(
        bgGanadoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(bgGanadoLayout.createSequentialGroup()
                .addGroup(bgGanadoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(bgGanadoLayout.createSequentialGroup()
                        .addGap(16, 16, 16)
                        .addComponent(btnIniciarSesion, javax.swing.GroupLayout.PREFERRED_SIZE, 128,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                    .addGroup(bgGanadoLayout.createSequentialGroup()
                        .addGap(16, 16, 16)
                        .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 160,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addContainerGap())
    );
    bgGanadoLayout.setVerticalGroup(
        bgGanadoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(bgGanadoLayout.createSequentialGroup()
                .addGap(16, 16, 16)
                .addComponent(btnIniciarSesion, javax.swing.GroupLayout.PREFERRED_SIZE, 128,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(16, 16, 16)
                .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 160,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())
    );

```



```

        .addComponent(btnIniciarSesion5, javax.swing.GroupLayout.PREFERRED_SIZE, 128,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jLabel5, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addContainerGap()
    );

    bg.add(bgBusqueda, new org.netbeans.lib.awtextra.AbsoluteConstraints(340, 310, 160, 160));

    bgBusquedaLotes.setBackground(new java.awt.Color(255, 255, 255));
    bgBusquedaLotes.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            bgBusquedaLotesMouseClicked(evt);
        }
        public void mouseEntered(java.awt.event.MouseEvent evt) {
            bgBusquedaLotesMouseEntered(evt);
        }
        public void mouseExited(java.awt.event.MouseEvent evt) {
            bgBusquedaLotesMouseExited(evt);
        }
    });

    btnIniciarSesion4.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    btnIniciarSesion4.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/agrupamiento.png"))); // NOI18N

    jLabel6.setFont(new java.awt.Font("Roboto Medium", 0, 12)); // NOI18N
    jLabel6.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    jLabel6.setText("REGISTRAR GANADO");

    javax.swing.GroupLayout bgBusquedaLotesLayout = new
javax.swing.GroupLayout(bgBusquedaLotes);
    bgBusquedaLotes.setLayout(bgBusquedaLotesLayout);
    bgBusquedaLotesLayout.setHorizontalGroup(

    bgBusquedaLotesLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
bgBusquedaLotesLayout.createSequentialGroup()
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(btnIniciarSesion4, javax.swing.GroupLayout.PREFERRED_SIZE, 128,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(15, 15, 15))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
bgBusquedaLotesLayout.createSequentialGroup()
            .addGap(0, 0, Short.MAX_VALUE)
            .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 160,
javax.swing.GroupLayout.PREFERRED_SIZE))
    );
    bgBusquedaLotesLayout.setVerticalGroup(

    bgBusquedaLotesLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addGroup(bgBusquedaLotesLayout.createSequentialGroup())
        .addContainerGap()
        .addComponent(btnIniciarSesion4, javax.swing.GroupLayout.PREFERRED_SIZE, 128,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jLabel6)
        .addContainerGap()
    );

    bg.add(bgBusquedaLotes, new org.netbeans.lib.awtextra.AbsoluteConstraints(620, 310, 160,
160));

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(bg, javax.swing.GroupLayout.PREFERRED_SIZE, 870,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(0, 0, Short.MAX_VALUE))
            .addGap(0, 0, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(bg, javax.swing.GroupLayout.PREFERRED_SIZE, 500,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(0, 0, Short.MAX_VALUE))
            .addGap(0, 0, Short.MAX_VALUE))
    );

    pack();
} // </editor-fold>

private void bgIniciarSesionMouseEntered(java.awt.event.MouseEvent evt) {
    bgIniciarSesion.setBackground(Color.red);
    txtIniciarSesion.setForeground(Color.white);
}

private void bgRegistrarUsuarioMouseEntered(java.awt.event.MouseEvent evt) {
    bgRegistrarUsuario.setBackground(Color.red);
    txtRegistrar.setForeground(Color.white);
}

private void bgMovimientosMouseEntered(java.awt.event.MouseEvent evt) {
    bgMovimientos.setBackground(Color.red);
    jLabel3.setForeground(Color.white);
}

private void bgGanadoMouseEntered(java.awt.event.MouseEvent evt) {
    bgGanado.setBackground(Color.red);
    jLabel4.setForeground(Color.white);
}

```



```

private void bgBusquedaMouseEntered(java.awt.event.MouseEvent evt) {
    bgBusqueda.setBackground(Color.red);
    jLabel5.setForeground(Color.white);
}

private void bgBusquedaLotesMouseEntered(java.awt.event.MouseEvent evt) {
    bgBusquedaLotes.setBackground(Color.red);
    jLabel6.setForeground(Color.white);
}

private void bgIniciarSesionMouseExited(java.awt.event.MouseEvent evt) {
    bgIniciarSesion.setBackground(Color.white);
    txtIniciarSesion.setForeground(Color.black);
}

private void bgRegistrarUsuarioMouseExited(java.awt.event.MouseEvent evt) {
    bgRegistrarUsuario.setBackground(Color.white);
    txtRegistrar.setForeground(Color.black);
}

private void bgMovimientosMouseExited(java.awt.event.MouseEvent evt) {
    bgMovimientos.setBackground(Color.white);
    jLabel3.setForeground(Color.black);
}

private void bgGanadoMouseExited(java.awt.event.MouseEvent evt) {
    bgGanado.setBackground(Color.white);
    jLabel4.setForeground(Color.black);
}

private void bgBusquedaMouseExited(java.awt.event.MouseEvent evt) {
    bgBusqueda.setBackground(Color.white);
    jLabel5.setForeground(Color.black);
}

private void bgBusquedaLotesMouseExited(java.awt.event.MouseEvent evt) {
    bgBusquedaLotes.setBackground(Color.white);
    jLabel6.setForeground(Color.black);
}

private void exitBtnMouseEntered(java.awt.event.MouseEvent evt) {
    exitBtn.setBackground(Color.red);
    exitTxt.setForeground(Color.black);
}

private void exitBtnMouseExited(java.awt.event.MouseEvent evt) {
    exitBtn.setBackground(Color.black);
    exitTxt.setForeground(Color.white);
}

private void exitBtnMouseClicked(java.awt.event.MouseEvent evt) {

```

```

Login loginWindow = new Login();
loginWindow.setVisible(true);

// Opcional: cerrar la ventana actual si lo deseas
this.dispose();
}

private void bgIniciarSesionMouseClicked(java.awt.event.MouseEvent evt) {
    Ganado buscarAnimal = new Ganado(username);
    System.out.println(username);
    buscarAnimal.setVisible(true);

    this.dispose();
}

private void bgRegistrarUsuarioMouseClicked(java.awt.event.MouseEvent evt) {
    RegistrarUsuario registrarUWindow = new RegistrarUsuario(username);
    registrarUWindow.setVisible(true);

    this.dispose();
}

private void bgMovimientosMouseClicked(java.awt.event.MouseEvent evt) {
    Movimientos movimientosWindows = new Movimientos(username);
    movimientosWindows.setVisible(true);
    this.dispose();
}

private void bgBusquedaMouseClicked(java.awt.event.MouseEvent evt) {

    Gasto gastoWindow= new Gasto(username);
    gastoWindow.setVisible(true);
    System.out.println("click");
    this.dispose();
}

private void bgGanadoMouseClicked(java.awt.event.MouseEvent evt) {

    PreciosV precioVWindow = new PreciosV(username);
    precioVWindow.setVisible(true);
    this.dispose();
}

private void bgBusquedaLotesMouseClicked(java.awt.event.MouseEvent evt) {

    RegistrarAnimales registroWindow = new RegistrarAnimales(username);
    registroWindow.setVisible(true);
    this.dispose();
}

/**
 * @param args the command line arguments

```



```

*/
public static void main(String args[]) {
    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            Connection con = Conexion.conectar();
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JPanel bg;
private javax.swing.JPanel bgBusqueda;
private javax.swing.JPanel bgBusquedaLotes;
private javax.swing.JPanel bgGanado;
private javax.swing.JPanel bgIniciarSesion;
private javax.swing.JPanel bgMovimientos;
private javax.swing.JPanel bgRegistrarUsuario;
private javax.swing.JLabel btnIniciarSesion;
private javax.swing.JLabel btnIniciarSesion1;
private javax.swing.JLabel btnIniciarSesion2;
private javax.swing.JLabel btnIniciarSesion3;
private javax.swing.JLabel btnIniciarSesion4;
private javax.swing.JLabel btnIniciarSesion5;
private javax.swing.JPanel encabezado;
private javax.swing.JPanel exitBtn;
private javax.swing.JLabel exitTxt;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel txtIniciarSesion;
private javax.swing.JLabel txtRegistrar;
// End of variables declaration
}

```

Inicio1.java

```
package com.frame;
```

```

import Busqueda.HomeBusqueda;
import BusquedaLotes.HomeBusquedaLotes;
import Ganado.HomeGanado;
import com.conexion.Conexion;
import com.login.Login;
import java.awt.Color;
import java.sql.Connection;

```

```
/**
```



```

*
* @author wicho
*/
public class Inicio1 extends javax.swing.JFrame {

    /**
     * Creates new form Inicio
     */
    public Inicio1() {
        initComponents();
        this.setLocationRelativeTo(null);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        bg = new javax.swing.JPanel();
        encabezado = new javax.swing.JPanel();
        exitBtn = new javax.swing.JPanel();
        exitTxt = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        bgGanado = new javax.swing.JPanel();
        btnIniciarSesion = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        bgBusqueda = new javax.swing.JPanel();
        btnIniciarSesion5 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        bgBusquedaLotes = new javax.swing.JPanel();
        btnIniciarSesion4 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setUndecorated(true);

        bg.setBackground(new java.awt.Color(255, 255, 255));
        bg.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        encabezado.setBackground(new java.awt.Color(0, 0, 0));
        encabezado.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        exitBtn.setBackground(new java.awt.Color(0, 0, 0));
        exitBtn.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                exitBtnMouseClicked(evt);
            }
        });
    }

```

```

        public void mouseEntered(java.awt.event.MouseEvent evt) {
            exitBtnMouseEntered(evt);
        }
        public void mouseExited(java.awt.event.MouseEvent evt) {
            exitBtnMouseExited(evt);
        }
    });

    exitTxt.setFont(new java.awt.Font("Roboto Medium", 0, 24)); // NOI18N
    exitTxt.setForeground(new java.awt.Color(255, 255, 255));
    exitTxt.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    exitTxt.setText("X");

    javax.swing.GroupLayout exitBtnLayout = new javax.swing.GroupLayout(exitBtn);
    exitBtn.setLayout(exitBtnLayout);
    exitBtnLayout.setHorizontalGroup(
        exitBtnLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                exitBtnLayout.createSequentialGroup()
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                        exitBtnLayout.createSequentialGroup()
                            .addGap(0, 0, Short.MAX_VALUE)
                            .addComponent(exitTxt, javax.swing.GroupLayout.PREFERRED_SIZE,
                                37, javax.swing.GroupLayout.PREFERRED_SIZE)
                        )
                    )
            .addContainerGap()
            .addComponent(exitTxt)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );

    encabezado.add(exitBtn, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, -1));

    jLabel7.setFont(new java.awt.Font("Roboto Medium", 0, 24)); // NOI18N
    jLabel7.setForeground(new java.awt.Color(255, 255, 255));
    jLabel7.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    jLabel7.setText("BIENVENIDO AL SISTEMA");
    encabezado.add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 30, 870, 44));

    jLabel8.setFont(new java.awt.Font("Roboto Light", 0, 12)); // NOI18N
    jLabel8.setForeground(new java.awt.Color(255, 255, 255));
    jLabel8.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    jLabel8.setText("PORFAVOR ELIJA UNA OPCION");
    encabezado.add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 80, 864, -1));

    bg.add(encabezado, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, 110));

    bgGanado.setBackground(new java.awt.Color(255, 255, 255));
    bgGanado.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
    bgGanado.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {

```

```

        bgGanadoMouseClicked(evt);
    }
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        bgGanadoMouseEntered(evt);
    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        bgGanadoMouseExited(evt);
    }
});

btnIniciarSesion.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnIniciarSesion.setIcon(new javax.swing.ImageIcon(getClass().getResource("/documentos/
(2).png"))); // NOI18N

jLabel4.setFont(new java.awt.Font("Roboto Medium", 0, 18)); // NOI18N
jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel4.setText("GANADO");

javax.swing.GroupLayout bgGanadoLayout = new javax.swing.GroupLayout(bgGanado);
bgGanado.setLayout(bgGanadoLayout);
bgGanadoLayout.setHorizontalGroup(
    bgGanadoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            bgGanadoLayout.createSequentialGroup()
                .addContainerGap()
                .addComponent(btnIniciarSesion, javax.swing.GroupLayout.PREFERRED_SIZE, 0,
Short.MAX_VALUE)
                .addContainerGap()
                .addComponent(jLabel4, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 258, Short.MAX_VALUE)
            );
        bgGanadoLayout.setVerticalGroup(
            bgGanadoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(bgGanadoLayout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(btnIniciarSesion, javax.swing.GroupLayout.DEFAULT_SIZE, 260,
Short.MAX_VALUE)
                    .addGap(24, 24, 24)
                    .addComponent(jLabel4)
                    .addContainerGap()
                );
);

bg.add(bgGanado, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 150, 260, 320));

bgBusqueda.setBackground(new java.awt.Color(255, 255, 255));
bgBusqueda.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0,
0)));
bgBusqueda.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        bgBusquedaMouseClicked(evt);
    }
    public void mouseEntered(java.awt.event.MouseEvent evt) {

```

```

        bgBusquedaMouseEntered(evt);
    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        bgBusquedaMouseExited(evt);
    }
});

btnIniciarSesion5.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnIniciarSesion5.setIcon(new javax.swing.ImageIcon(getClass().getResource("/pagina-de-busqueda.png"))); // NOI18N

jLabel5.setFont(new java.awt.Font("Roboto Medium", 0, 18)); // NOI18N
jLabel5.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel5.setText("BUSQUEDA ESPECIFICA");

javax.swing.GroupLayout bgBusquedaLayout = new javax.swing.GroupLayout(bgBusqueda);
bgBusqueda.setLayout(bgBusquedaLayout);
bgBusquedaLayout.setHorizontalGroup(
    bgBusquedaLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jLabel5, javax.swing.GroupLayout.Alignment.TRAILING,
            javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
            Short.MAX_VALUE)
        .addGroup(bgBusquedaLayout.createSequentialGroup()
            .addGap(16, 16, 16)
            .addComponent(btnIniciarSesion5, javax.swing.GroupLayout.PREFERRED_SIZE, 228,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(14, Short.MAX_VALUE))
);
bgBusquedaLayout.setVerticalGroup(
    bgBusquedaLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(bgBusquedaLayout.createSequentialGroup()
            .addContainerGap()
            .addComponent(btnIniciarSesion5, javax.swing.GroupLayout.DEFAULT_SIZE, 274,
                Short.MAX_VALUE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jLabel5)
            .addGap(4, 4, 4))
);

bg.add(bgBusqueda, new org.netbeans.lib.awtextra.AbsoluteConstraints(310, 150, 260, 320));

bgBusquedaLotes.setBackground(new java.awt.Color(255, 255, 255));
bgBusquedaLotes.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));
bgBusquedaLotes.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        bgBusquedaLotesMouseClicked(evt);
    }
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        bgBusquedaLotesMouseEntered(evt);
    }
    public void mouseExited(java.awt.event.MouseEvent evt) {

```

```

        bgBusquedaLotesMouseExited(evt);
    }
});

btnIniciarSesion4.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnIniciarSesion4.setIcon(new javax.swing.ImageIcon(getClass().getResource("/grupo-de-
enfoque.png"))); // NOI18N

jLabel6.setFont(new java.awt.Font("Roboto Medium", 0, 18)); // NOI18N
jLabel6.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel6.setText("BUSQUEDA POR LOTES");

javax.swing.GroupLayout bgBusquedaLotesLayout = new
javax.swing.GroupLayout(bgBusquedaLotes);
bgBusquedaLotes.setLayout(bgBusquedaLotesLayout);
bgBusquedaLotesLayout.setHorizontalGroup(

bgBusquedaLotesLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(btnIniciarSesion4, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 259, Short.MAX_VALUE)
    .addComponent(jLabel6, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
);
bgBusquedaLotesLayout.setVerticalGroup(

bgBusquedaLotesLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(bgBusquedaLotesLayout.createSequentialGroup()
        .addGap(10, 10, 10)
        .addComponent(btnIniciarSesion4, javax.swing.GroupLayout.PREFERRED_SIZE, 273,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(10, 10, 10)
        .addComponent(jLabel6)
        .addGap(10, 10, 10)
    )
);

bg.add(bgBusquedaLotes, new org.netbeans.lib.awtextra.AbsoluteConstraints(590, 150, 261,
320));

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(bg, javax.swing.GroupLayout.PREFERRED_SIZE, 870,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(0, 0, Short.MAX_VALUE)
        )
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()

```

```

        .addComponent(bg,          javax.swing.GroupLayout.PREFERRED_SIZE,          500,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, Short.MAX_VALUE))
    );

    pack();
} // </editor-fold>

private void bgGanadoMouseEntered(java.awt.event.MouseEvent evt) {
    bgGanado.setBackground(Color.red);
    jLabel4.setForeground(Color.white);
}

private void bgBusquedaMouseEntered(java.awt.event.MouseEvent evt) {
    bgBusqueda.setBackground(Color.red);
    jLabel5.setForeground(Color.white);
}

private void bgBusquedaLotesMouseEntered(java.awt.event.MouseEvent evt) {
    bgBusquedaLotes.setBackground(Color.red);
    jLabel6.setForeground(Color.white);
}

private void bgGanadoMouseExited(java.awt.event.MouseEvent evt) {
    bgGanado.setBackground(Color.white);
    jLabel4.setForeground(Color.black);
}

private void bgBusquedaMouseExited(java.awt.event.MouseEvent evt) {
    bgBusqueda.setBackground(Color.white);
    jLabel5.setForeground(Color.black);
}

private void bgBusquedaLotesMouseExited(java.awt.event.MouseEvent evt) {
    bgBusquedaLotes.setBackground(Color.white);
    jLabel6.setForeground(Color.black);
}

private void exitBtnMouseEntered(java.awt.event.MouseEvent evt) {
    exitBtn.setBackground(Color.red);
    exitTxt.setForeground(Color.black);
}

private void exitBtnMouseExited(java.awt.event.MouseEvent evt) {
    exitBtn.setBackground(Color.black);
    exitTxt.setForeground(Color.white);
}

private void exitBtnMouseClicked(java.awt.event.MouseEvent evt) {
    Login inicioWindow = new Login();
    inicioWindow.setVisible(true);
}

```

```

        // Opcional: cerrar la ventana actual si lo deseas
        this.dispose();
    }

    private void bgBusquedaMouseClicked(java.awt.event.MouseEvent evt) {
        String username = null;
        HomeBusqueda buscarHWindow= new HomeBusqueda(username);
        buscarHWindow.setVisible(true);
        System.out.println("click");
        this.dispose();
    }

    private void bgGanadoMouseClicked(java.awt.event.MouseEvent evt) {
        String username = null;
        HomeGanado homeGWindow = new HomeGanado(username);
        homeGWindow.setVisible(true);
        this.dispose();
    }

    private void bgBusquedaLotesMouseClicked(java.awt.event.MouseEvent evt) {
        String username = null;
        HomeBusquedaLotes homeBLWindow = new HomeBusquedaLotes(username);
        homeBLWindow.setVisible(true);
        this.dispose();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Inicio1().setVisible(true);
                Connection con = Conexion.conectar();
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JPanel bg;
    private javax.swing.JPanel bgBusqueda;
    private javax.swing.JPanel bgBusquedaLotes;
    private javax.swing.JPanel bgGanado;
    private javax.swing.JLabel btnIniciarSesion;
    private javax.swing.JLabel btnIniciarSesion4;
    private javax.swing.JLabel btnIniciarSesion5;
    private javax.swing.JPanel encabezado;
    private javax.swing.JPanel exitBtn;
    private javax.swing.JLabel exitTxt;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;

```



```

private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
// End of variables declaration
}

```

Login.java

```

package com.login;

import com.conexion.Conexion;
import com.frame.Inicio;
import com.frame.Inicio1;
import com.registrar.RegistrarUsuario;
import java.awt.Color;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;

/**
 * @author wicho
 */
public class Login extends javax.swing.JFrame {
    private String username;
    int xMouse, yMouse;

    public Login() {
        initComponents();
        this.setLocationRelativeTo(null);
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        bg = new javax.swing.JPanel();
        logo = new javax.swing.JLabel();
        fondo = new javax.swing.JLabel();
        logo2 = new javax.swing.JLabel();
        favicon = new javax.swing.JLabel();
        userLabel = new javax.swing.JLabel();
        userTxt = new javax.swing.JTextField();
        jSeparator1 = new javax.swing.JSeparator();
        passLabel = new javax.swing.JLabel();
        jSeparator2 = new javax.swing.JSeparator();
        passTxt = new javax.swing.JPasswordField();
        header = new javax.swing.JPanel();
        exitBtn = new javax.swing.JPanel();
    }

```

```

exitTxt = new javax.swing.JLabel();
loginBtn = new javax.swing.JPanel();
loginBtnTxt = new javax.swing.JLabel();
jButton1 = new javax.swing.JButton();
jButton2 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setUndecorated(true);
setResizable(false);

bg.setBackground(new java.awt.Color(255, 255, 255));
bg.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

logo.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
logo.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGO2BUENO-removebg-preview.png"))); // NOI18N
bg.add(logo, new org.netbeans.lib.awtextra.AbsoluteConstraints(550, 100, 320, 210));

fondo.setIcon(new javax.swing.ImageIcon(getClass().getResource("/campo.jpg"))); // NOI18N
bg.add(fondo, new org.netbeans.lib.awtextra.AbsoluteConstraints(550, 0, 320, 500));

logo2.setFont(new java.awt.Font("Roboto Black", 0, 24)); // NOI18N
logo2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO-removebg-preview.png"))); // NOI18N
bg.add(logo2, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 70, -1, -1));

favicon.setFont(new java.awt.Font("Roboto Black", 0, 24)); // NOI18N
favicon.setText("INICIAR SESION");
bg.add(favicon, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 210, -1, -1));

userLabel.setFont(new java.awt.Font("Roboto Light", 1, 14)); // NOI18N
userLabel.setText("Usuario");
bg.add(userLabel, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 260, -1, -1));

userTxt.setFont(new java.awt.Font("Roboto", 0, 12)); // NOI18N
userTxt.setBorder(null);
userTxt.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mousePressed(java.awt.event.MouseEvent evt) {
        userTxtMousePressed(evt);
    }
});
bg.add(userTxt, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 290, 330, 30));

jSeparator1.setForeground(new java.awt.Color(0, 0, 0));
bg.add(jSeparator1, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 320, 330, -1));

passLabel.setFont(new java.awt.Font("Roboto Light", 1, 14)); // NOI18N
passLabel.setText("Contraseña");
bg.add(passLabel, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 340, -1, -1));

jSeparator2.setForeground(new java.awt.Color(0, 0, 0));
bg.add(jSeparator2, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 400, 330, -1));

```

```

passTxt.setBorder(null);
passTxt.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mousePressed(java.awt.event.MouseEvent evt) {
        passTxtMousePressed(evt);
    }
});
bg.add(passTxt, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 370, 330, 30));

header.setBackground(new java.awt.Color(255, 255, 255));
header.addMouseMotionListener(new java.awt.event.MouseMotionAdapter() {
    public void mouseDragged(java.awt.event.MouseEvent evt) {
        headerMouseDragged(evt);
    }
});
header.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mousePressed(java.awt.event.MouseEvent evt) {
        headerMousePressed(evt);
    }
});

exitBtn.setBackground(new java.awt.Color(255, 255, 255));

exitTxt.setFont(new java.awt.Font("Roboto Medium", 0, 24)); // NOI18N
exitTxt.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
exitTxt.setText("X");
exitTxt.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        exitTxtMouseClicked(evt);
    }
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        exitTxtMouseEntered(evt);
    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        exitTxtMouseExited(evt);
    }
});

javax.swing.GroupLayout exitBtnLayout = new javax.swing.GroupLayout(exitBtn);
exitBtn.setLayout(exitBtnLayout);
exitBtnLayout.setHorizontalGroup(
    exitBtnLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(exitTxt, javax.swing.GroupLayout.PREFERRED_SIZE,
            35,
            javax.swing.GroupLayout.PREFERRED_SIZE)
);
exitBtnLayout.setVerticalGroup(
    exitBtnLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(exitBtnLayout.createSequentialGroup()
            .addComponent(exitTxt)
            .addGap(0, 6, Short.MAX_VALUE))
);

```

```

javax.swing.GroupLayout headerLayout = new javax.swing.GroupLayout(header);
header.setLayout(headerLayout);
headerLayout.setHorizontalGroup(
    headerLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(headerLayout.createSequentialGroup()
            .addComponent(exitBtn, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(0, 835, Short.MAX_VALUE))
        );
headerLayout.setVerticalGroup(
    headerLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(headerLayout.createSequentialGroup()
            .addComponent(exitBtn, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(0, 0, Short.MAX_VALUE))
        );

bg.add(header, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 870, 30));

loginBtn.setBackground(new java.awt.Color(0, 153, 51));

loginBtnTxt.setFont(new java.awt.Font("Roboto Medium", 0, 18)); // NOI18N
loginBtnTxt.setForeground(new java.awt.Color(255, 255, 255));
loginBtnTxt.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
loginBtnTxt.setText("ENTRAR");
loginBtnTxt.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
loginBtnTxt.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        loginBtnTxtMouseClicked(evt);
    }
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        loginBtnTxtMouseEntered(evt);
    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        loginBtnTxtMouseExited(evt);
    }
});

javax.swing.GroupLayout loginBtnLayout = new javax.swing.GroupLayout(loginBtn);
loginBtn.setLayout(loginBtnLayout);
loginBtnLayout.setHorizontalGroup(
    loginBtnLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(loginBtnTxt, javax.swing.GroupLayout.DEFAULT_SIZE, 130,
            Short.MAX_VALUE)
        );
loginBtnLayout.setVerticalGroup(
    loginBtnLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(loginBtnTxt, javax.swing.GroupLayout.DEFAULT_SIZE, 40,
            Short.MAX_VALUE)
        );

bg.add(loginBtn, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 430, 130, 40));

```



```

jButton1.setBackground(new java.awt.Color(0, 102, 255));
jButton1.setFont(new java.awt.Font("Roboto Medium", 0, 14)); // NOI18N
jButton1.setForeground(new java.awt.Color(255, 255, 255));
jButton1.setText("REGISTRAR");
jButton1.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jButton1MouseClicked(evt);
    }
});
bg.add(jButton1, new org.netbeans.lib.awtextra.AbsoluteConstraints(200, 430, 130, 40));

jButton2.setBackground(new java.awt.Color(255, 102, 102));
jButton2.setFont(new java.awt.Font("Roboto Medium", 0, 14)); // NOI18N
jButton2.setForeground(new java.awt.Color(255, 255, 255));
jButton2.setText("INVITADO");
jButton2.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jButton2MouseClicked(evt);
    }
});
bg.add(jButton2, new org.netbeans.lib.awtextra.AbsoluteConstraints(350, 430, 100, 40));

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(bg, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(bg, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 491, javax.swing.GroupLayout.PREFERRED_SIZE)
);

pack();
} // </editor-fold>

private void headerMousePressed(java.awt.event.MouseEvent evt) {
    xMouse = evt.getX();
    yMouse = evt.getY();
}

private void headerMouseDragged(java.awt.event.MouseEvent evt) {
    int x = evt.getXOnScreen();
    int y = evt.getYOnScreen();
    this.setLocation(x - xMouse, y - yMouse);
}

private void exitTxtMouseClicked(java.awt.event.MouseEvent evt) {
    System.exit(0);
}

```

```

}

private void exitTxtMouseEntered(java.awt.event.MouseEvent evt) {
    exitBtn.setBackground(Color.red);
    exitTxt.setForeground(Color.white);
}

private void exitTxtMouseExited(java.awt.event.MouseEvent evt) {
    exitBtn.setBackground(Color.white);
    exitTxt.setForeground(Color.black);
}

private void loginBtnTxtMouseEntered(java.awt.event.MouseEvent evt) {
    loginBtn.setBackground(new Color(115, 255, 150));
    loginBtnTxt.setForeground(Color.BLACK);
}

private void loginBtnTxtMouseExited(java.awt.event.MouseEvent evt) {
    loginBtn.setBackground(new Color(0, 153, 51));
    loginBtnTxt.setForeground(Color.white);
}

private void userTxtMousePressed(java.awt.event.MouseEvent evt) {
    if (userTxt.getText().equals("Nombre de Usuario")) {
        userTxt.setText("");
        userTxt.setForeground(Color.black);
    }
    if (String.valueOf(passTxt.getPassword()).isEmpty()) {
        passTxt.setText("*****");
        passTxt.setForeground(Color.gray);
    }
}

private void passTxtMousePressed(java.awt.event.MouseEvent evt) {
    if (userTxt.getText().isEmpty()) {
        userTxt.setText("Nombre de Usuario");
        userTxt.setForeground(Color.gray);
    }
    if (String.valueOf(passTxt.getPassword()).equals("*****")) {
        passTxt.setText("");
        passTxt.setForeground(Color.black);
    }
}

private void loginBtnTxtMouseClicked(java.awt.event.MouseEvent evt) {

    String username = userTxt.getText();
    String password = String.valueOf(passTxt.getPassword());

    try (Connection con = Conexion.conectar()) {
        if (con != null) {

```



```

        String query = "SELECT Usuario, Password, privilegio FROM `registros-us` WHERE
        BINARY Usuario = ?";
        try (PreparedStatement stmt = con.prepareStatement(query)) {
            stmt.setString(1, username);

            try (ResultSet rs = stmt.executeQuery()) {
                if (rs.next()) {
                    String dbPassword = rs.getString("Password");
                    String privilegio = rs.getString("privilegio");

                    if (dbPassword.equals(password)) {
                        javax.swing.JOptionPane.showMessageDialog(this,
                            "Bienvenido, " + username + "!",
                            "LOGIN EXITOSO",
                            javax.swing.JOptionPane.INFORMATION_MESSAGE);

                        String insertMovimiento = "INSERT INTO movimientos (fecha, hora, usuario,
                        privilegio, movimiento) VALUES (?, ?, ?, ?, ?)";
                        try (PreparedStatement movStmt = con.prepareStatement(insertMovimiento)) {
                            LocalDateTime fechaHoraActual = LocalDateTime.now();
                            String fecha = fechaHoraActual.toLocalDate().toString();
                            String hora = fechaHoraActual.toLocalTime().format(DateTimeFormatter.ofPattern("HH:mm:ss"));
                            =

                            movStmt.setString(1, fecha);
                            movStmt.setString(2, hora);
                            movStmt.setString(3, username);
                            movStmt.setString(4, privilegio);
                            movStmt.setString(5, "Inicio de sesión");

                            movStmt.executeUpdate();
                        } catch (SQLException e) {
                            e.printStackTrace();
                        }

                        if ("Administrador".equalsIgnoreCase(privilegio)) {
                            Inicio inicio = new Inicio(username);
                            inicio.setVisible(true);
                            this.dispose();
                        } else if ("Usuario".equalsIgnoreCase(privilegio)) {
                            Inicio1 inicioCli = new Inicio1();
                            inicioCli.setVisible(true);
                            this.dispose();
                        } else {
                            javax.swing.JOptionPane.showMessageDialog(this,
                                "Privilegio desconocido para el usuario: " + username,
                                "ERROR DE PRIVILEGIOS",
                                javax.swing.JOptionPane.ERROR_MESSAGE);
                            return;
                        }
                    }

                    this.dispose();
                }
            }
        }
    }
}

```



```

        } else {
            javax.swing.JOptionPane.showMessageDialog(this,
                "Contraseña incorrecta para el usuario: " + username,
                "ERROR DE LOGIN",
                javax.swing.JOptionPane.ERROR_MESSAGE);
        }
    } else {
        javax.swing.JOptionPane.showMessageDialog(this,
            "El usuario " + username + " no está registrado.",
            "USUARIO NO REGISTRADO",
            javax.swing.JOptionPane.ERROR_MESSAGE);
    }
}
} else {
    javax.swing.JOptionPane.showMessageDialog(this,
        "Error de conexión a la base de datos",
        "ERROR",
        javax.swing.JOptionPane.ERROR_MESSAGE);
}
} catch (SQLException ex) {
    ex.printStackTrace();
}

// Restablecer los campos de usuario y contraseña
userTxt.setText("Nombre de Usuario");
userTxt.setForeground(Color.gray);
passTxt.setText("*****");
passTxt.setForeground(Color.gray);

}

private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
    RegistrarUsuario registrar = new RegistrarUsuario(username);
    registrar.setVisible(true);
    this.dispose();
}

private void jButton2MouseClicked(java.awt.event.MouseEvent evt) {
    Inicio1 inicioWindow = new Inicio1();
    inicioWindow.setVisible(true);
    this.dispose();
}

public static void main(String args[]) {

    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Login().setVisible(true);
        }
    });
}
}

```



```

// Variables declaration - do not modify
private javax.swing.JPanel bg;
private javax.swing.JPanel exitBtn;
private javax.swing.JLabel exitTxt;
private javax.swing.JLabel favicon;
private javax.swing.JLabel fondo;
private javax.swing.JPanel header;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JSeparator jSeparator2;
private javax.swing.JPanel loginBtn;
private javax.swing.JLabel loginBtnTxt;
private javax.swing.JLabel logo;
private javax.swing.JLabel logo2;
private javax.swing.JLabel passLabel;
private javax.swing.JPasswordField passTxt;
private javax.swing.JLabel userLabel;
private javax.swing.JTextField userTxt;
// End of variables declaration
}

```

Movimientos.java

```
package com.movimientos;
```

```

import com.conexion.Conexion;
import com.frame.Inicio;
import java.awt.Color;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.table.DefaultTableCellRenderer;
import javax.swing.table.DefaultTableModel;

```

```

/**
 *
 * @author wicho
 */
public class Movimientos extends javax.swing.JFrame {
    private String username;
    public Movimientos(String username) {
        initComponents();
        this.username = username;
        cargarMovimientos();
        this.setLocationRelativeTo(null);
    }
    private void cargarMovimientos() {
        DefaultTableModel modelo = new DefaultTableModel();

```



```

        modelo.addColumn("FECHA");
        modelo.addColumn("HORA");
        modelo.addColumn("USUARIO");
        modelo.addColumn("PRIVILEGIO");
        modelo.addColumn("MOVIMIENTO");
        modelo.addColumn("ID DEL GANADO");

        jTable1.setModel(modelo);

        String query = "SELECT fecha, hora, usuario,privilegio, movimiento, id FROM movimientos
ORDER BY fecha DESC, hora DESC";

        try (Connection con = Conexion.conectar();
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery(query)) {

            while (rs.next()) {
                String fecha = rs.getString("fecha");
                String hora = rs.getString("hora");
                String usuario = rs.getString("usuario");
                String privilegio = rs.getString("privilegio");
                String movimiento = rs.getString("movimiento");
                String id = rs.getString("id");

                modelo.addRow(new Object[]{fecha, hora, usuario,privilegio, movimiento,id});
            }

            centrarCeldas(jTable1);

        } catch (SQLException e) {
            e.printStackTrace();
            javax.swing.JOptionPane.showMessageDialog(this,
                "Error al cargar movimientos desde la base de datos.",
                "ERROR",
                javax.swing.JOptionPane.ERROR_MESSAGE);
        }
    }

    private void centrarCeldas(javax.swing.JTable table) {
        DefaultTableCellRenderer centrarRenderer = new DefaultTableCellRenderer();
        centrarRenderer.setHorizontalAlignment(javax.swing.JLabel.CENTER);

        for (int i = 0; i < table.getColumnCount(); i++) {
            table.getColumnModel().getColumn(i).setCellRenderer(centrarRenderer);
        }
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        bg = new javax.swing.JPanel();

```



```

encabe1 = new javax.swing.JPanel();
cierre = new javax.swing.JPanel();
jLabel1 = new javax.swing.JLabel();
logo = new javax.swing.JLabel();
logo1 = new javax.swing.JLabel();
jLabel2 = new javax.swing.JLabel();
jScrollPane1 = new javax.swing.JScrollPane();
jTable1 = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setUndecorated(true);

bg.setBackground(new java.awt.Color(255, 255, 255));
bg.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

encabe1.setBackground(new java.awt.Color(255, 153, 0));
encabe1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

cierre.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        cierreMouseClicked(evt);
    }
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        cierreMouseEntered(evt);
    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        cierreMouseExited(evt);
    }
});
cierre.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel1.setFont(new java.awt.Font("Roboto Medium", 0, 24)); // NOI18N
jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel1.setText("X");
cierre.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 30, 30));

encabe1.add(cierre, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 30, 30));

logo.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO-removebg-preview.png"))); // NOI18N
encabe1.add(logo, new org.netbeans.lib.awtextra.AbsoluteConstraints(690, 0, 220, 120));

logo1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO-removebg-preview.png"))); // NOI18N
encabe1.add(logo1, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 0, 190, 120));

jLabel2.setFont(new java.awt.Font("Roboto Medium", 3, 36)); // NOI18N
jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel2.setText("MIS MOVIMIENTOS");
encabe1.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, -2, 600, 120));

bg.add(encabe1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 880, 120));

```

```

jTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "FECHA", "HORA", "USUARIO", "MOVIMIENTO"
    }
));
jTable1.setFocusable(false);
jTable1.setRowHeight(25);
jTable1.setSelectionBackground(new java.awt.Color(232, 57, 95));
jTable1.getTableHeader().setReorderingAllowed(false);
jScrollPane1.setViewportView(jTable1);

bg.add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 120, 880, 380));

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(bg, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(bg, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);

pack();
} // </editor-fold>

private void cierreMouseEntered(java.awt.event.MouseEvent evt) {
    cierre.setBackground(Color.red);
    cierre.setForeground(Color.black);
}

private void cierreMouseExited(java.awt.event.MouseEvent evt) {
    cierre.setBackground(Color.black);
    cierre.setForeground(Color.white);
}

private void cierreMouseClicked(java.awt.event.MouseEvent evt) {
    Inicio inicioWindow = new Inicio(username);
    inicioWindow.setVisible(true);

    this.dispose();
}

```



```

public static void main(String args[]) {

    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {

        }
    });
}

// Variables declaration - do not modify
private javax.swing.JPanel bg;
private javax.swing.JPanel cierre;
private javax.swing.JPanel encabe1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
private javax.swing.JLabel logo;
private javax.swing.JLabel logo1;
// End of variables declaration
}

```

RegistrarUsuario.java

```
package com.registrar;
```

```

import com.conexion.Conexion;
import com.login.Login;
import java.awt.Color;

```

```

/**
 * @author wicho
 */
public class RegistrarUsuario extends javax.swing.JFrame {
    private String username;
    int xMouse, yMouse;

    public RegistrarUsuario(String username) {
        initComponents();
        this.username = username;
        this.setLocationRelativeTo(null);
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        bg = new javax.swing.JPanel();

```



```

logo = new javax.swing.JLabel();
fondo = new javax.swing.JLabel();
logo2 = new javax.swing.JLabel();
favicon = new javax.swing.JLabel();
userLabel = new javax.swing.JLabel();
userTxt = new javax.swing.JTextField();
jSeparator1 = new javax.swing.JSeparator();
passLabel = new javax.swing.JLabel();
jSeparator2 = new javax.swing.JSeparator();
passTxt = new javax.swing.JPasswordField();
header = new javax.swing.JPanel();
exitBtn = new javax.swing.JPanel();
exitTxt = new javax.swing.JLabel();
loginBtn = new javax.swing.JPanel();
loginBtnTxt = new javax.swing.JLabel();
passLabel1 = new javax.swing.JLabel();
jSeparator3 = new javax.swing.JSeparator();
passTxt1 = new javax.swing.JPasswordField();
adminRadioButton = new javax.swing.JRadioButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setUndecorated(true);
setResizable(false);

bg.setBackground(new java.awt.Color(255, 255, 255));
bg.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

logo.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
logo.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGO2BUENO-removebg-preview.png"))); // NOI18N
bg.add(logo, new org.netbeans.lib.awtextra.AbsoluteConstraints(550, 70, 320, 240));

fondo.setIcon(new javax.swing.ImageIcon(getClass().getResource("/campo.jpg"))); // NOI18N
bg.add(fondo, new org.netbeans.lib.awtextra.AbsoluteConstraints(550, 0, 320, 500));

logo2.setFont(new java.awt.Font("Roboto Black", 0, 24)); // NOI18N
logo2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGOBUENO-removebg-preview.png"))); // NOI18N
bg.add(logo2, new org.netbeans.lib.awtextra.AbsoluteConstraints(60, 20, -1, -1));

favicon.setFont(new java.awt.Font("Roboto Black", 0, 24)); // NOI18N
favicon.setText("Registrar Usuario");
bg.add(favicon, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 130, -1, -1));

userLabel.setFont(new java.awt.Font("Roboto Light", 1, 14)); // NOI18N
userLabel.setText("Usuario");
bg.add(userLabel, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 170, -1, -1));

userTxt.setFont(new java.awt.Font("Roboto", 0, 12)); // NOI18N
userTxt.setBorder(null);
userTxt.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mousePressed(java.awt.event.MouseEvent evt) {

```

```

        userTxtMousePressed(evt);
    }
});
bg.add(userTxt, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 200, 330, 30));

jSeparator1.setForeground(new java.awt.Color(0, 0, 0));
bg.add(jSeparator1, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 230, 330, -1));

passLabel.setFont(new java.awt.Font("Roboto Light", 1, 14)); // NOI18N
passLabel.setText("Contraseña");
bg.add(passLabel, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 250, -1, -1));

jSeparator2.setForeground(new java.awt.Color(0, 0, 0));
bg.add(jSeparator2, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 310, 330, -1));

passTxt.setFont(new java.awt.Font("SansSerif", 0, 12)); // NOI18N
passTxt.setBorder(null);
passTxt.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mousePressed(java.awt.event.MouseEvent evt) {
        passTxtMousePressed(evt);
    }
});
bg.add(passTxt, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 280, 330, 30));

header.setBackground(new java.awt.Color(255, 255, 255));
header.addMouseMotionListener(new java.awt.event.MouseMotionAdapter() {
    public void mouseDragged(java.awt.event.MouseEvent evt) {
        headerMouseDragged(evt);
    }
});
header.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mousePressed(java.awt.event.MouseEvent evt) {
        headerMousePressed(evt);
    }
});

exitBtn.setBackground(new java.awt.Color(255, 255, 255));
exitBtn.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        exitBtnMouseClicked(evt);
    }
});

exitTxt.setFont(new java.awt.Font("Roboto Medium", 0, 24)); // NOI18N
exitTxt.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
exitTxt.setText("X");
exitTxt.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        exitTxtMouseClicked(evt);
    }
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        exitTxtMouseEntered(evt);
    }
});

```



```

    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        exitTxtMouseExited(evt);
    }
});

javax.swing.GroupLayout exitBtnLayout = new javax.swing.GroupLayout(exitBtn);
exitBtn.setLayout(exitBtnLayout);
exitBtnLayout.setHorizontalGroup(
    exitBtnLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(exitTxt, javax.swing.GroupLayout.PREFERRED_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE)
);
exitBtnLayout.setVerticalGroup(
    exitBtnLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(exitBtnLayout.createSequentialGroup()
            .addComponent(exitTxt)
            .addGap(0, 6, Short.MAX_VALUE))
);

javax.swing.GroupLayout headerLayout = new javax.swing.GroupLayout(header);
header.setLayout(headerLayout);
headerLayout.setHorizontalGroup(
    headerLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(headerLayout.createSequentialGroup()
            .addComponent(exitBtn,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(0, 835, Short.MAX_VALUE))
);
headerLayout.setVerticalGroup(
    headerLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(headerLayout.createSequentialGroup()
            .addComponent(exitBtn,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(0, 0, Short.MAX_VALUE))
);

bg.add(header, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 870, 30));

loginBtn.setBackground(new java.awt.Color(0, 153, 51));

loginBtnTxt.setFont(new java.awt.Font("Roboto Medium", 0, 18)); // NOI18N
loginBtnTxt.setForeground(new java.awt.Color(255, 255, 255));
loginBtnTxt.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
loginBtnTxt.setText("ENTRAR");
loginBtnTxt.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
loginBtnTxt.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        loginBtnTxtMouseClicked(evt);
    }
});
public void mouseEntered(java.awt.event.MouseEvent evt) {
    loginBtnTxtMouseEntered(evt);
}

```

35,

```

    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        loginBtnTxtMouseExited(evt);
    }
});

javax.swing.GroupLayout loginBtnLayout = new javax.swing.GroupLayout(loginBtn);
loginBtn.setLayout(loginBtnLayout);
loginBtnLayout.setHorizontalGroup(
    loginBtnLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(loginBtnTxt, javax.swing.GroupLayout.DEFAULT_SIZE, 130,
Short.MAX_VALUE)
);
loginBtnLayout.setVerticalGroup(
    loginBtnLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(loginBtnTxt, javax.swing.GroupLayout.DEFAULT_SIZE, 40,
Short.MAX_VALUE)
);

bg.add(loginBtn, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 430, 130, 40));

passLabel1.setFont(new java.awt.Font("Roboto Light", 1, 14)); // NOI18N
passLabel1.setText("Contraseña Maestra");
bg.add(passLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 330, -1, -1));

jSeparator3.setForeground(new java.awt.Color(0, 0, 0));
bg.add(jSeparator3, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 390, 330, -1));

passTxt1.setFont(new java.awt.Font("SansSerif", 0, 12)); // NOI18N
passTxt1.setForeground(new java.awt.Color(204, 204, 204));
passTxt1.setText("*****");
passTxt1.setBorder(null);
passTxt1.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mousePressed(java.awt.event.MouseEvent evt) {
        passTxt1MousePressed(evt);
    }
});
bg.add(passTxt1, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 360, 330, 30));

adminRadioButton.setFont(new java.awt.Font("Roboto Medium", 0, 12)); // NOI18N
adminRadioButton.setText("Administrador");
bg.add(adminRadioButton, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 400, -1, -1));

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(bg, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addComponent(bg,                                javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 491, javax.swing.GroupLayout.PREFERRED_SIZE)
    );

    pack();
} // </editor-fold>

private void headerMousePressed(java.awt.event.MouseEvent evt) {
    xMouse = evt.getX();
    yMouse = evt.getY();
}

private void headerMouseDragged(java.awt.event.MouseEvent evt) {
    int x = evt.getXOnScreen();
    int y = evt.getYOnScreen();
    this.setLocation(x - xMouse, y - yMouse);
}

private void exitTxtMouseClicked(java.awt.event.MouseEvent evt) {
    Login inicioWindow = new Login();
    inicioWindow.setVisible(true);
}

private void exitTxtMouseEntered(java.awt.event.MouseEvent evt) {
    exitBtn.setBackground(Color.red);
    exitTxt.setForeground(Color.white);
}

private void exitTxtMouseExited(java.awt.event.MouseEvent evt) {
    exitBtn.setBackground(Color.white);
    exitTxt.setForeground(Color.black);
}

private void loginBtnTxtMouseEntered(java.awt.event.MouseEvent evt) {
    loginBtn.setBackground(new Color(115, 255, 150));
    loginBtnTxt.setForeground(Color.BLACK);
}

private void loginBtnTxtMouseExited(java.awt.event.MouseEvent evt) {
    loginBtn.setBackground(new Color(0, 153, 51));
    loginBtnTxt.setForeground(Color.white);
}

private void userTxtMousePressed(java.awt.event.MouseEvent evt) {

    if (String.valueOf(passTxt1.getPassword()).isEmpty()) {
        passTxt1.setText("*****");
        passTxt1.setForeground(Color.gray);
    }
}

private void passTxtMousePressed(java.awt.event.MouseEvent evt) {

```



```

        if (String.valueOf(passTxt1.getPassword()).isEmpty()) {
            passTxt1.setText("*****");
            passTxt1.setForeground(Color.gray);
        }
    }

    private void loginBtnTxtMouseClicked(java.awt.event.MouseEvent evt) {
        // Obtener el usuario y la contraseña ingresados
        String username = userTxt.getText();
        String passwordMas = String.valueOf(passTxt1.getPassword());
        String password = String.valueOf(passTxt.getPassword());

        // Definir la contraseña maestra para validación
        String masterPassword = "M@STER99";

        if (passwordMas.equals(masterPassword)) {
            String privilegio = adminRadioButton.isSelected() ? "Administrador" : "Usuario";

            boolean registroExitoso = Conexion.insertarUsuario(username, password, masterPassword,
            privilegio);

            if (registroExitoso) {
                javax.swing.JOptionPane.showMessageDialog(this,
                    "Registro Correcto. ¡Usuario agregado exitosamente!",
                    "REGISTRO",
                    javax.swing.JOptionPane.INFORMATION_MESSAGE);

                Login inicioWindow = new Login();
                inicioWindow.setVisible(true);
                this.dispose();
            } else {
                javax.swing.JOptionPane.showMessageDialog(this,
                    "Error al intentar registrar el usuario. Por favor, intente nuevamente.",
                    "ERROR DE REGISTRO",
                    javax.swing.JOptionPane.ERROR_MESSAGE);
            }
        } else {
            javax.swing.JOptionPane.showMessageDialog(this,
                "Contraseña Maestra incorrecta. No tienes permiso para registrar nuevos usuarios.",
                "ERROR DE REGISTRO",
                javax.swing.JOptionPane.ERROR_MESSAGE);

            passTxt1.setText("*****");
            passTxt1.setForeground(Color.gray);
            adminRadioButton.setSelected(false);
        }
    }

    private void passTxt1MousePressed(java.awt.event.MouseEvent evt) {

```



```

        if (String.valueOf(passTxt1.getPassword()).equals("*****")) {
            passTxt1.setText("");
            passTxt1.setForeground(Color.black);
        }
    }

    private void exitBtnMouseClicked(java.awt.event.MouseEvent evt) {
        Login inicioWindow = new Login();
        inicioWindow.setVisible(true);

        this.dispose();
    }

    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JRadioButton adminRadioButton;
    private javax.swing.JPanel bg;
    private javax.swing.JPanel exitBtn;
    private javax.swing.JLabel exitTxt;
    private javax.swing.JLabel favicon;
    private javax.swing.JLabel fondo;
    private javax.swing.JPanel header;
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JSeparator jSeparator2;
    private javax.swing.JSeparator jSeparator3;
    private javax.swing.JPanel loginBtn;
    private javax.swing.JLabel loginBtnTxt;
    private javax.swing.JLabel logo;
    private javax.swing.JLabel logo2;
    private javax.swing.JLabel passLabel;
    private javax.swing.JLabel passLabel1;
    private javax.swing.JPasswordField passTxt;
    private javax.swing.JPasswordField passTxt1;
    private javax.swing.JLabel userLabel;
    private javax.swing.JTextField userTxt;
    // End of variables declaration
}

```

