

“Sistema de asistencia inteligente para el cuidado optimizado del cultivo de hortalizas en ambientes urbanos”

TESIS PARA TITULACIÓN INTEGRAL

INGENIERÍA EN SISTEMAS COMPUTACIONALES

**EGRESADO:
JUAN PABLO NAVARRO DELGADO**

**DIRECTOR Y CO-DIRECTOR:
DR. Gonzalo Miguel Quetz Aguirre
DR. Felipe Ángel Álvarez Salgado**

Calkiní, Campeche, México 2025



Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO



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/03/25/635

INGENIERÍA EN SISTEMAS COMPUTACIONALES

CONSTANCIA DE NO INCONVENIENCIA

MODALIDAD: TESIS

C. JUAN PABLO NAVARRO DELGADO 7501
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 DR. GONZALO MIGUEL QUETZ AGUIRRE, y el co-director DR. FELIPE ANGEL ALVAREZ SALGADO, respectivamente, este Instituto consta de no tener inconveniencia para la realización del acto protocolario de titulación integral de su trabajo profesional titulado: **"SISTEMA DE ASISTENCIA INTELIGENTE PARA EL CUIDADO OPTIMIZADO DEL CULTIVO DE HORTALIZAS EN AMBIENTES URBANOS"**.

ATENTAMENTE

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

Dr. Jorge Alfonso Nau 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 S/N por Carretera Federal, Calkiní, Campeche,
C.P. 24900 Tel. 996-8134870,
e-mail: ee_dcalkini@tecnm.mx | itescam.edu.mx



Agradecimientos

Quiero expresar mi más profundo reconocimiento a todas las personas que contribuyeron al desarrollo de esta tesis, tanto en el ámbito académico como en lo personal.

En primer lugar, mi más sincero agradecimiento al Dr. Gonzalo Miguel Quetz Aguirre y a la Dra. Marlene Méndez Moreno, quienes jugaron un papel fundamental en el desarrollo de mi tesis y el proyecto. Su dedicación, paciencia y vasto conocimiento académico fueron pilares esenciales en la elaboración de esta tesis de igual forma al Dr. Felipe Ángel Álvarez Salgado quien formo parte de esta tesis como co-director siendo un pilar importante para el desarrollo de esta tesis. Sus enseñanzas no solo enriquecieron el contenido de esta tesis, sino que también dejaron una huella imborrable en mi desarrollo profesional y personal.

Agradezco también a todos los profesores que participaron en mi formación en especial a el Dr. Jose Luis Lira Turriza, la Dra. Yaqueline Pech Hu y el Dr. Jose Manuel Lira Turriza quienes, a lo largo de mi carrera, con su esfuerzo y compromiso sentaron las bases para que hoy pueda alcanzar este logro. Cada clase, consejo y orientación contribuyeron a moldear el profesional que estoy llegando a ser.

Quiero dedicar un agradecimiento especial a mis padres Iris Mercedes Delgado Hernández y Juan Navarro Villa, quienes han sido el motor y la inspiración constante en mi vida. Su amor incondicional, apoyo incansable y ejemplo de fortaleza han sido fundamentales para que hoy alcance este logro. Este trabajo no habría sido posible sin su confianza y sacrificio, y por eso les estaré eternamente agradecido.

A mis hermanos Jesús Santiago y Gabriela Alejandra, por ser mis compañeros de vida, quienes siempre estuvieron ahí para alentarme, apoyarme y recordarme el valor de la unión familiar. Su cariño y motivación han sido una fuente de energía en los momentos más desafiantes.

A mis abuelos que ya no están con nosotros, quienes con su ejemplo de sabiduría, amor y valores dejaron un legado imborrable en mi vida. Aunque su ausencia física se siente profundamente, su memoria vive en mi corazón y me inspira a ser mejor cada día. Este logro lo dedico también a ellos, como un tributo a todo lo que me enseñaron y al amor eterno que siempre me dieron.

A mi familia en general, por el amor y el respaldo que siempre me han brindado, por creer en mí y celebrar cada uno de mis pasos hacia adelante. Gracias por ser mi refugio y mi fortaleza en cada etapa de este camino.

A mis amigos y amigas, quienes han sido una familia elegida en este camino, por su apoyo incondicional, su motivación y las risas que aligeraron incluso los momentos más difíciles. Gracias por compartir este viaje conmigo, por su lealtad y por hacer que cada día sea más llevadero con su compañía y alegría.

Con todo mi corazón, este logro también es suyo, porque sin ustedes nunca habría llegado hasta aquí.

Resumen

El presente estudio se centra en la implementación y evaluación de un sistema de asistencia inteligente, AgroLeaf, diseñado para optimizar el uso de recursos esenciales (agua, luz y nutrientes) en el cultivo de cilantro en entornos urbanos. La investigación responde a la pregunta: ¿Cómo puede un sistema de asistencia inteligente optimizar el cuidado de cultivos de hortalizas en entornos urbanos en términos de consumo de recursos (agua, luz y nutrientes)?

Para responder a esta interrogante, se realizó un experimento comparativo entre dos grupos de cultivo: uno que utilizó el sistema inteligente y otro que empleó métodos tradicionales. Se midieron tres variables principales: consumo de agua, exposición a la luz y uso de fertilizantes. Los resultados obtenidos muestran que el sistema inteligente contribuyó a una mejora significativa en la eficiencia del uso de estos recursos. En comparación con los métodos tradicionales, AgroLeaf logró reducir el consumo de agua en un 8.33%, optimizar las horas de luz recomendadas en un 37.5% y disminuir en un 20% el uso de fertilizantes. Estos hallazgos evidencian que la implementación de tecnologías avanzadas puede mejorar la sostenibilidad de la agricultura urbana, aumentando al mismo tiempo la productividad y rentabilidad de los cultivos.

A pesar de los resultados positivos, se identificaron algunas limitaciones, como variaciones climáticas que pudieron haber afectado el desempeño del sistema, errores en la entrada de datos y una duración limitada del estudio. Se recomienda realizar investigaciones adicionales para evaluar los efectos a largo plazo y mejorar la personalización de las recomendaciones del sistema. Además, se sugiere la capacitación de los usuarios y la integración del sistema con otras tecnologías para maximizar su eficiencia.

Este estudio aporta evidencia sobre el impacto positivo de los sistemas inteligentes en la gestión de cultivos urbanos, sentando las bases para futuras investigaciones y aplicaciones tecnológicas en la agricultura sostenible. La optimización del uso de recursos mediante AgroLeaf representa un paso hacia la transformación digital de la agricultura, facilitando una producción más eficiente, ecológica y rentable en entornos urbanos.

Índice

Índice de Ilustraciones	7
Índice de Tablas.....	7
CAPITULO I. INTRODUCCION	8
1.1 Objetivos del Estudio	9
1.2 Hipótesis	9
CAPÍTULO II. MARCO TEÓRICO	9
2.1 Marco teórico.....	9
2.2 Antecedentes.....	10
2.3 Tecnologías para la optimización de recursos:	12
1. Agricultura Urbana y Sostenibilidad	12
2. Optimización de Recursos en Agricultura	13
3. Sistemas Inteligentes para la Optimización de Recursos.....	13
4. Comparación con los Métodos Tradicionales	13
5. Tecnologías en la Agricultura Urbana: Casos y Aplicaciones.....	14
CAPÍTULO III. MÉTODOS Y MATERIALES	14
3.1 Materiales y métodos.	14
3.1.2 Lenguaje de programación.....	14
3.1.3 Entorno de desarrollo	15
3.2 Implementación de la metodología del proyecto	16
3.2.1 Diseño de la base de datos.....	17
3.3 Metodología	19
Enfoque experimental	19
1. Grupo Experimental (Con la Aplicación Agroleaf)	19
2. Grupo de Control (Métodos Tradicionales).....	20
CAPÍTULO IV. RESULTADOS Y DISCUSIÓN	21
4.1 Pantallas de la aplicación.....	22
4.2 Pruebas de sistema:.....	39
Pruebas de Sistema.....	39
Pruebas de Compatibilidad	42
Pruebas de Rendimiento	43

4.3 Selección de cultivos	45
4.4 Instrumentos y herramientas	45
Recolección de datos.....	45
4.5 Resultados	46
4.5.1 Presentación de los datos	46
4.5.2 Análisis descriptivo	47
4.5.3 Análisis Comparativo.....	47
4.6 Discusión	50
4.6.1 Limitaciones del estudio.....	50
CAPITULO V. CONCLUSIONES Y RECOMENDACIONES	51
5.1 Conclusiones	51
5.2 Recomendaciones	52
Referencias Bibliográficas	53
Anexos	55
Paquete Login	56
Login	56
Main	71
Registro	80
Paquete App	93
Vista Admin	93
Vista Cliente.....	103
Paquete AdminViews	117
Vista Agregar Hortaliza Admin	117
Vista Usuarios Admin	141
Vista Perfil Admin	148
Paquete Views	161
Vista Principal Cliente.....	161
Vista Calendario Cliente	170
Vista Perfil Cliente	184
Vista Reportes Cliente	196
Vista Clima Cliente.....	202
Vista de Hortalizas.....	211

Índice de Ilustraciones

Ilustración 1 Modelo Relacional	19
Ilustración 2. Vista Principal.....	22
Ilustración 3.Vista de registro	23
Ilustración 4.Vista de inicio de sesión.....	24
Ilustración 5. Vista de inicio de sesión exitoso	25
Ilustración 6. Vista principal cliente	25
Ilustración 7. Vista principal cliente - Búsqueda por nombre	26
Ilustración 8.Vista principal cliente - Búsqueda por región	26
Ilustración 9.Vista principal cliente - Hortaliza	27
Ilustración 10.Vista Calendario Cliente	28
Ilustración 11.Vista Calendario Cliente - ingresar datos.....	28
Ilustración 12.Vista Calendario Cliente - Registro	29
Ilustración 13.Vista Calendario Cliente - ver registros por fecha	30
Ilustración 14.Vista Perfil Cliente	31
Ilustración 15.Vista Perfil Cliente - Cambiar información	31
Ilustración 16.Vista Perfil Cliente - Información Actualizada	32
Ilustración 17.Vista Reportes Cliente.....	32
Ilustración 18.Vista Reportes Cliente.....	33
Ilustración 19.Vista Perfil Cliente - Búsqueda de ciudad	33
Ilustración 20.Vista Principal Administrador.....	34
Ilustración 21.Vista Principal Administrador - Registro de hortaliza	34
Ilustración 22.Vista Principal Administrador - Eliminar Hortaliza.....	35
Ilustración 23.Vista Usuarios Administrador	35
Ilustración 24.Vista Usuarios Administrador - Eliminar Usuario	36
Ilustración 25.Vista Usuarios Administrador - Usuario Eliminado	36
Ilustración 26.Vista Usuarios Administrador - Tabla actualizada.....	37
Ilustración 27. Vista Perfil Administrador	37
Ilustración 28.Vista Perfil Administrador - Cambiar Información	38
Ilustración 29.Vista Perfil Administrador - Información Actualizada	38

Índice de Tablas

Tabla 1 Equipo Scrum.....	17
Tabla 2 Pruebas del módulo app y sus submódulos	39
Tabla 3 Pruebas del módulo Login y sus submódulos	40
Tabla 4 Pruebas del módulo AdminViews y sus submódulos	40
Tabla 5 Pruebas del módulo views y sus submódulos	42
Tabla 6 Pruebas de compatibilidad de los módulos	43
Tabla 7 Pruebas de rendimiento de módulos.....	44
Tabla 8 Uso de recursos	47
Tabla 9 Comparación de recursos usados	48

CAPITULO I. INTRODUCCION

El estado de Campeche, ubicado en el sureste de México, cuenta con condiciones climáticas y geográficas que permiten el cultivo de una variedad de hortalizas. Sin embargo, estas condiciones también presentan desafíos, como la variabilidad en las lluvias y las altas temperaturas, que pueden afectar negativamente el rendimiento de los cultivos si no se gestionan adecuadamente. El auge de la agricultura urbana en la región ofrece una oportunidad para aprovechar espacios reducidos en áreas urbanas y periurbanas para la producción de alimentos, pero para que esta actividad sea sostenible, es necesario optimizar el uso de recursos y mejorar las prácticas de cultivo.

El proyecto "Sistema de asistencia inteligente para el cuidado optimizado del cultivo de hortalizas en ambientes urbanos en Campeche" surge como una respuesta a la necesidad de proporcionar a los agricultores urbanos de Campeche una herramienta tecnológica que les permita mejorar el cuidado de sus cultivos de hortalizas. Esta herramienta, una aplicación de escritorio basada en inteligencia artificial, estará diseñada específicamente para las condiciones agroclimáticas de Campeche, proporcionando recomendaciones personalizadas en cuanto al riego, fertilización y otros cuidados necesarios, basándose en datos locales históricos y en tiempo real.

La aplicación optimizará el cultivo de hortalizas típicas de la región, tales como el tomate, el chile habanero, la lechuga y la calabaza, entre otras, que se adaptan bien a las condiciones de

Campeche. Además, estará diseñada para manejar cultivos bajo técnicas de agricultura orgánica y en ambientes controlados, lo que permitirá a los agricultores urbanos sacar el mayor provecho de sus espacios disponibles. Con la implementación de este sistema, se espera reducir el desperdicio de recursos, mejorar la productividad y contribuir al desarrollo de una agricultura más sostenible en la región.

1.1 Objetivos del Estudio

La agricultura urbana ha adquirido un papel fundamental en la seguridad alimentaria de las ciudades, especialmente en áreas donde el espacio es limitado y la eficiencia en el uso de recursos es crucial. En este contexto, las tecnologías avanzadas como los sistemas de asistencia inteligente tienen el potencial de optimizar el manejo de recursos en los cultivos urbanos.

La pregunta central de esta investigación es:

¿Cómo puede un sistema de asistencia inteligente optimizar el cuidado de cultivos de hortalizas en entornos urbanos en términos de consumo de recursos (agua, luz y nutrientes)?

Esta investigación se enfoca en evaluar el impacto de la aplicación Agroleaf, un sistema de asistencia inteligente diseñado para proporcionar recomendaciones sobre el cuidado de cultivos, en particular cilantro, en entornos urbanos. A través de un enfoque experimental, se busca comparar el uso de recursos y la productividad de los cultivos gestionados con la ayuda de la aplicación frente a los métodos tradicionales de cultivo.

1.2 Hipótesis

El uso de la aplicación de asistencia inteligente optimiza de manera más eficiente el uso de recursos (agua, luz y nutrientes) en los cultivos urbanos en comparación con los métodos tradicionales de cultivo, mejorando así la sostenibilidad.

Se espera que el sistema inteligente, al ajustar automáticamente el riego, la fertilización y la exposición a la luz según las necesidades específicas de cada cultivo, logre una utilización más eficiente de los recursos, resultando en menores consumos de agua y fertilizantes sin comprometer la calidad de los cultivos.

CAPÍTULO II. MARCO TEÓRICO

2.1 Marco teórico

Para realizar el desarrollo del sistema de asistencia inteligente, es necesario conocer previamente algunos aspectos teóricos relacionados con las variables clave del proyecto, como son las definiciones del software y el ciclo de vida de la metodología seleccionada.

En este capítulo, se abordarán los diferentes temas relacionados con el desarrollo del sistema de asistencia inteligente. Asimismo, se explicará la metodología seleccionada para

llevar a cabo el proyecto, asegurando que todos los elementos necesarios para su implementación y optimización sean considerados.

2.2 Antecedentes.

Sánchez y Pérez (2019), en su artículo "Paradigma IoT: desde su conceptualización hacia su aplicación en la agricultura", destacan la importancia de los sistemas inteligentes en la optimización del manejo agrícola. Según estos autores, la implementación de tecnologías que permiten el monitoreo y control de cultivos en tiempo real ha mejorado considerablemente la eficiencia en la agricultura. En regiones como Campeche, donde los agricultores enfrentan condiciones climáticas variables, los sistemas inteligentes pueden ayudar a ajustar el riego y la fertilización según las necesidades de cada cultivo, lo que resulta en un mejor uso de los recursos y un aumento en la productividad.

López, Martínez y Torres (2020), en su estudio "Sistema IoT para el monitoreo de variables climatológicas en cultivos de agricultura urbana", señalan que la integración de datos geoespecíficos es clave para la efectividad de los sistemas inteligentes en la agricultura. Estos autores argumentan que la personalización de las recomendaciones agrícolas basadas en las condiciones climáticas locales es esencial para maximizar la eficiencia. En Campeche, donde las condiciones varían significativamente entre zonas, un sistema que utilice datos geoespecíficos puede proporcionar recomendaciones específicas y precisas, lo que mejora el rendimiento de los cultivos.

González y Pérez (2018), en su revisión "Sistemas de riego inteligentes en zonas con escasez de agua: una revisión", subrayan la importancia de los sistemas de riego inteligentes en zonas con escasez de agua. Según su investigación, la implementación de tecnologías que ajusten el riego en función de las necesidades del cultivo puede reducir el uso de agua hasta en un 30%, sin afectar la productividad. En un contexto como el de Campeche, donde el acceso al agua es limitado, este tipo de tecnología puede ser fundamental para asegurar una producción agrícola sostenible.

Rodríguez y Muñoz (2019), en su artículo "Accesibilidad de sistemas inteligentes para pequeños agricultores: diseño de interfaces simplificadas", destacan la necesidad de que los sistemas inteligentes sean accesibles para los pequeños agricultores. Según estos autores, la simplicidad en el diseño de la interfaz es crucial para que los usuarios sin conocimientos técnicos puedan beneficiarse de la tecnología. En el caso de los agricultores urbanos de Campeche, un sistema que sea fácil de usar, pero que integre recomendaciones personalizadas, puede aumentar la adopción de tecnologías avanzadas en la agricultura.

Ramos y Gómez (2020), en su estudio "Integración de datos históricos y en tiempo real en sistemas inteligentes agrícolas", argumentan que la combinación de datos históricos y en tiempo real en los sistemas inteligentes proporciona a los agricultores una ventaja competitiva. Estos autores indican que el análisis de datos históricos sobre las condiciones del suelo y el clima, junto con el monitoreo en tiempo real, permite una toma de decisiones más informada. En Campeche, donde los ciclos climáticos son irregulares, la capacidad de anticiparse a las necesidades de los cultivos mediante el uso de estos datos puede mejorar significativamente la productividad.

García y Hernández (2021), en su investigación "Reducción del uso de fertilizantes y pesticidas mediante sistemas inteligentes con sensores de monitoreo", indican que los sistemas inteligentes que integran sensores de monitoreo pueden ayudar a reducir el uso de fertilizantes y pesticidas, mejorando la sostenibilidad de los cultivos. Según su investigación, el monitoreo en tiempo real permite a los agricultores aplicar insumos agrícolas solo cuando es necesario, lo que reduce el desperdicio y el impacto ambiental. En regiones como Campeche, donde la gestión adecuada de los recursos es crítica, estas tecnologías pueden contribuir al desarrollo de una agricultura más sostenible.

Martínez y Sánchez (2021), en su artículo "Flexibilidad de sistemas inteligentes para adaptarse a diferentes tipos de cultivos", señalan que los sistemas inteligentes deben ser flexibles para adaptarse a los diferentes tipos de cultivos. En Campeche, donde se cultivan hortalizas con diferentes necesidades, como el chile habanero y el tomate, la capacidad de personalizar las recomendaciones según el tipo de cultivo es esencial para asegurar su efectividad. Los sistemas que permiten esta flexibilidad pueden proporcionar a los agricultores recomendaciones precisas y específicas, mejorando el rendimiento general.

Rodríguez y Muñoz (2018), en su estudio "Adopción de sistemas inteligentes en la agricultura: la importancia de interfaces de usuario simples", destacan que la adopción de sistemas inteligentes depende en gran medida de la simplicidad de la interfaz de usuario. En su investigación, subrayan que los agricultores con poca experiencia en tecnología necesitan sistemas fáciles de usar, pero que a la vez ofrezcan recomendaciones detalladas basadas en datos. En Campeche, un sistema con una interfaz accesible podría facilitar la adopción de tecnologías avanzadas por parte de los agricultores urbanos.

La Organización de las Naciones Unidas para la Alimentación y la Agricultura (FAO) (2020), en su informe "La agricultura climáticamente inteligente", subraya que la sostenibilidad en la agricultura es una prioridad global, y que la

implementación de sistemas inteligentes es una de las soluciones clave para optimizar el uso de insumos agrícolas y reducir el impacto ambiental. En Campeche, donde el uso eficiente del agua y otros recursos es fundamental, estos sistemas pueden ayudar a los agricultores a gestionar mejor sus cultivos, reduciendo los costos y mejorando la productividad.

Torres y Martínez (2021), en su artículo "Mejora de la resiliencia climática de los cultivos mediante sistemas inteligentes con datos en tiempo real", señalan que los sistemas inteligentes que utilizan datos en tiempo real pueden mejorar la resiliencia climática de los cultivos. En su estudio, los autores indican que la capacidad de ajustarse rápidamente a los cambios en las condiciones climáticas es esencial para reducir las pérdidas de cosechas debido a eventos climáticos extremos. En Campeche, donde las variaciones climáticas pueden ser significativas, un sistema que integre datos en tiempo real puede ayudar a los agricultores a adaptarse mejor a estos cambios.

2.3 Tecnologías para la optimización de recursos:

1. Agricultura Urbana y Sostenibilidad

La agricultura urbana se define como la práctica de cultivar, procesar y distribuir alimentos dentro de los límites de las ciudades. Con el crecimiento poblacional y la expansión urbana, la agricultura urbana se ha convertido en una actividad crucial para garantizar la seguridad alimentaria en áreas urbanas densamente pobladas. Según la FAO (Organización de las Naciones Unidas para la Alimentación y la Agricultura), las ciudades representan más del 70% de la población mundial, lo que aumenta la demanda de alimentos frescos y sostenibles.

Uno de los mayores desafíos de la agricultura urbana es el uso eficiente de los recursos, especialmente en entornos donde el espacio, el agua y otros recursos naturales son limitados. En este contexto, la sostenibilidad juega un papel clave. La sostenibilidad en la agricultura urbana implica no solo la producción de alimentos, sino también el uso responsable de los recursos naturales, reduciendo el desperdicio y maximizando la eficiencia del consumo de agua, energía y nutrientes. Las tecnologías innovadoras, como los sistemas inteligentes, ofrecen una solución para mejorar esta eficiencia.

2. Optimización de Recursos en Agricultura

La optimización de recursos se refiere a la gestión eficiente de los recursos disponibles para maximizar los beneficios y minimizar el desperdicio. En la agricultura, los principales recursos que se deben optimizar son:

Agua: El riego adecuado es fundamental para el crecimiento de las plantas, pero el uso excesivo de agua en la agricultura es un problema creciente. Según el Banco Mundial, la agricultura es responsable de aproximadamente el 70% del consumo mundial de agua. En la agricultura urbana, donde el acceso al agua puede ser limitado o costoso, optimizar el uso del agua es crucial.

Luz: Las plantas requieren luz para realizar la fotosíntesis, y la cantidad de luz que reciben influye directamente en su crecimiento. Sin embargo, en muchas áreas urbanas, la luz puede no ser adecuada para los cultivos, lo que puede afectar la productividad. El uso adecuado de la luz artificial o la optimización de la luz solar es fundamental para mejorar los rendimientos de los cultivos urbanos.

Nutrientes: Los fertilizantes son esenciales para el crecimiento de las plantas, pero su uso excesivo puede llevar a la contaminación del agua y al desperdicio de recursos. La optimización de los nutrientes implica aplicar la cantidad justa de fertilizantes según las necesidades específicas de las plantas, reduciendo el impacto ambiental y mejorando la eficiencia de los cultivos.

3. Sistemas Inteligentes para la Optimización de Recursos

Monitoreo en Tiempo Real y Control Automático: La capacidad de monitorear las condiciones ambientales en tiempo real permite una gestión precisa de los recursos. Las plataformas basadas en inteligencia artificial pueden realizar ajustes de manera autónoma y en tiempo real, lo que minimiza el desperdicio y maximiza la eficiencia en el uso de recursos.

4. Comparación con los Métodos Tradicionales

Los métodos tradicionales de cultivo se basan en la experiencia y el conocimiento local de los agricultores, quienes aplican recursos según la intuición o las prácticas convencionales. Sin embargo, este enfoque no siempre es eficiente, ya que no tiene en cuenta las variaciones en las condiciones ambientales, el tipo de cultivo o las necesidades específicas de los mismos. Los métodos tradicionales también suelen implicar un uso excesivo de recursos, como el riego y la fertilización, debido a la falta de un monitoreo preciso y en tiempo real.

Los principales desafíos de los métodos tradicionales incluyen:

Uso ineficiente de agua: Los agricultores a menudo aplican más agua de la necesaria debido a la falta de información precisa sobre la humedad del suelo.

Desperdicio de fertilizantes: Los fertilizantes se aplican sin tener en cuenta las necesidades específicas de cada planta, lo que puede llevar a un uso excesivo de productos químicos y al deterioro del medio ambiente.

Falta de adaptabilidad: Los métodos tradicionales no son capaces de adaptarse rápidamente a los cambios en las condiciones climáticas o en las necesidades del cultivo.

En contraste, el sistema inteligente que se propone en esta tesis optimiza estos procesos al proporcionar datos precisos y recomendaciones personalizadas, adaptándose a las condiciones cambiantes del entorno y mejorando la eficiencia en el uso de los recursos.

5. Tecnologías en la Agricultura Urbana: Casos y Aplicaciones

Numerosos estudios han demostrado que los sistemas inteligentes son una solución viable y eficaz para la optimización de recursos en la agricultura urbana. Investigaciones recientes han evaluado el uso de plataformas de monitoreo en tiempo real en cultivos urbanos, mostrando resultados positivos en la reducción del consumo de agua y fertilizantes, así como en el aumento de la productividad de los cultivos.

El estudio de casos de éxito en la implementación de estas tecnologías, como proyectos en ciudades como Singapur, Barcelona y San Francisco, puede proporcionar información valiosa sobre las mejores prácticas y lecciones aprendidas en el contexto de la agricultura urbana inteligente.

CAPÍTULO III. MÉTODOS Y MATERIALES

3.1 Materiales y métodos.

Para el desarrollo del sistema de asistencia inteligente para el cuidado optimizado del cultivo de hortalizas en ambientes urbanos, se utilizaron diversas herramientas de software, metodologías ágiles y lenguajes de programación. Estas herramientas fueron seleccionadas cuidadosamente para facilitar y optimizar el proceso de planificación, diseño, codificación y pruebas.

En esta sección, se describen las tecnologías y enfoques utilizados, así como los pasos seguidos para garantizar que el sistema desarrollado cumpla con los objetivos planteados.

3.1.2 Lenguaje de programación

Java

Java es un lenguaje de programación de propósito general, orientado a objetos, ampliamente utilizado en el desarrollo de aplicaciones de escritorio, móviles y web. En este proyecto, Java fue seleccionado como lenguaje principal debido a sus características versátiles y robustas.

Ventajas de Java para este Proyecto

1. Multiplataforma

Permite la ejecución en cualquier sistema operativo con soporte para la máquina virtual Java (JVM), facilitando la portabilidad del sistema.

2. Ecosistema Extenso

Java cuenta con una amplia gama de bibliotecas y frameworks que simplifican el desarrollo de aplicaciones complejas, como el manejo de bases de datos y la integración con APIs externas.

3. Orientación a Objetos

Ofrece una estructura modular que facilita la reutilización del código y la organización de componentes, lo que es clave en sistemas escalables.

4. Seguridad Integrada

Proporciona mecanismos para manejar excepciones, validar entradas y proteger el sistema contra vulnerabilidades comunes.

5. Comunidad y Documentación

Al ser uno de los lenguajes más utilizados, Java tiene una comunidad extensa que asegura soporte, actualizaciones y recursos para resolver problemas durante el desarrollo.

3.1.3 Entorno de desarrollo

NetBeans IDE 20

NetBeans 20 fue la herramienta seleccionada para llevar a cabo el desarrollo del sistema debido a sus características avanzadas que optimizan el flujo de trabajo en proyectos Java.

Ventajas de NetBeans IDE 20

1. Facilidad de Uso

NetBeans ofrece una interfaz intuitiva que simplifica la escritura, depuración y ejecución del código Java, reduciendo el tiempo necesario para resolver problemas técnicos.

2. Integración con Maven

Permite gestionar dependencias, organizar el proyecto y automatizar procesos como la construcción y el empaquetado del sistema.

3. Soporte para Diseño de Interfaces Gráficas (GUI)

Incluye herramientas visuales para desarrollar interfaces con JavaFX y Swing, lo que permite crear una experiencia de usuario funcional y atractiva.

4. Depuración Avanzada

Proporciona herramientas para identificar y corregir errores en tiempo real, mejorando la eficiencia durante el desarrollo.

5. Gestión de Pruebas

Integra JUnit y otras herramientas para realizar pruebas unitarias, asegurando que cada módulo del sistema funcione como se espera.

6. Extensibilidad

Su arquitectura modular permite agregar plugins para personalizar el entorno según las necesidades del proyecto, incluyendo soporte para control de versiones como Git.

3.2 Implementación de la metodología del proyecto

La metodología seleccionada para el desarrollo de este proyecto fue **Scrum**, una metodología ágil que permitió organizar y controlar las fases de desarrollo de manera eficiente, facilitando la entrega de funcionalidades de forma incremental y funcional. Gracias a Scrum, se logró mantener un flujo de trabajo ordenado, flexible y adaptable a los cambios necesarios durante el desarrollo del sistema.

Aplicación de Scrum en el Proyecto

1. Control y Organización del Proyecto

Scrum proporcionó una estructura que permitió trabajar de manera secuencial y organizada en cada funcionalidad del sistema.

Se siguió un cronograma diario de actividades, en el cual las tareas programadas eran revisadas y analizadas por el encargado del proyecto para asegurar el cumplimiento de los objetivos planteados.

2. Trabajo en Equipo

Las reuniones periódicas permitieron revisar el progreso, identificar posibles bloqueos y realizar ajustes en función de los objetivos del proyecto.

3. Flexibilidad y Adaptación

Durante el desarrollo, Scrum permitió realizar cambios en los requerimientos según las necesidades emergentes, asegurando que el producto final se ajustara a las expectativas del cliente y a los objetivos establecidos.

4. Definición de Roles del Equipo Scrum

Antes de comenzar con el levantamiento de los requerimientos, fue necesario definir los roles de los miembros del equipo Scrum. A continuación, se presenta una tabla con las responsabilidades asignadas:

Roles	Descripción	Persona
Product Owner	Titular de la coordinación	Dr. Gonzalo Miguel Quetz Aguirre
Scrum Master	Director del proyecto	Dra. Marlene Mendez Moreno
Development Team	Desarrollador del proyecto	Juan Pablo Navarro Delgado

Tabla 1 Equipo Scrum

3.2.1 Diseño de la base de datos

Para almacenar y gestionar los datos de la aplicación, se implementó una base de datos relacional, diseñada para satisfacer las necesidades de los usuarios y administradores. La base de datos se desarrolló en MySQL, una de las herramientas más utilizadas y confiables para gestionar grandes volúmenes de datos de manera eficiente.

Especificaciones de la Base de Datos:

1. Tecnología Utilizada:

La base de datos está construida sobre MySQL, una tecnología conocida por su facilidad de uso, robustez y amplia compatibilidad con diferentes plataformas.

- Permite un acceso rápido y seguro a la información almacenada.

2. Gestión de la Base de Datos:

Para administrar y gestionar la base de datos, se utilizaron herramientas como MySQL Workbench y conexiones desde el sistema mediante APIs JDBC integradas en el programa desarrollado en Java.

La comunicación entre la aplicación y la base de datos está optimizada para garantizar transacciones seguras y rápidas.

3. Estructura de la Base de Datos:

Tabla users:

- Almacena la información de los usuarios registrados en el sistema.
- Campos importantes: username, password, role (para diferenciar entre clientes y administradores), y email.

Tabla hortalizas:

- Contiene información relacionada con las hortalizas, como su nombre, temporada de siembra, requerimientos de riego, adaptabilidad climática, y una imagen asociada.

Tabla recordatorios:

- Almacena eventos y recordatorios creados por los usuarios.
- Campos importantes: user_id (clave foránea que relaciona con la tabla users), fecha, titulo, y descripcion.

4. Relaciones:

Relación entre usuarios y recordatorios:

- Un usuario puede tener múltiples recordatorios asociados gracias a la clave foránea user_id en la tabla recordatorios.

Relación independiente de hortalizas:

- La tabla de hortalizas está diseñada para almacenar datos de manera independiente, garantizando que los administradores puedan gestionar esta información sin depender de otras tablas.

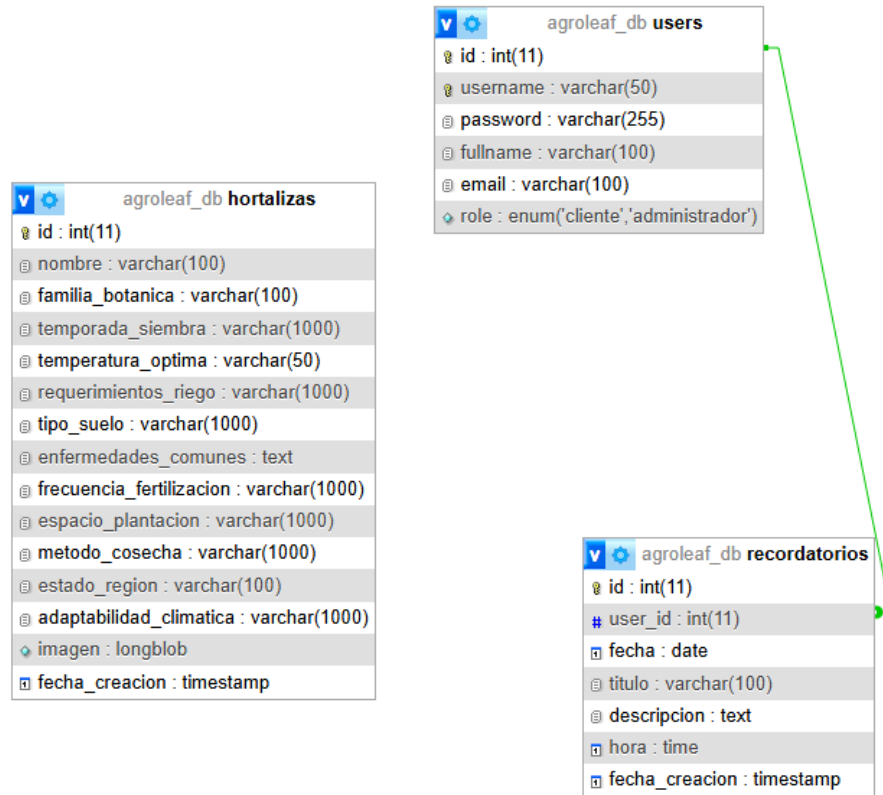


Ilustración 1 Modelo Relacional

3.3 Metodología

Enfoque experimental

El diseño experimental de esta investigación se centra en la evaluación de la eficiencia en el uso de recursos (agua, luz y nutrientes) en el cultivo de cilantro mediante el uso de la aplicación Agroleaf, que proporciona recomendaciones personalizadas basadas en los datos ingresados por los usuarios. Para evaluar el impacto de la aplicación en comparación con los métodos tradicionales, se implementarán dos grupos de cultivos urbanos:

1. Grupo Experimental (Con la Aplicación Agroleaf)

Este grupo de cultivos de cilantro utilizará la aplicación Agroleaf para optimizar el uso de recursos. Los usuarios ingresarán información básica sobre el cultivo (por ejemplo, tipo de suelo, condiciones climáticas locales, tipo de cultivo) y la aplicación generará

recomendaciones automáticas para gestionar el riego, la fertilización y la exposición a la luz según las necesidades del cilantro.

Recomendaciones de la Aplicación:

Riego: La aplicación dará información sobre la cantidad de agua necesaria para el cilantro, tomando en cuenta las condiciones climáticas y el tipo de suelo.

Fertilización: La aplicación proporcionará recomendaciones sobre la cantidad y tipo de fertilizante que el cilantro necesita en cada etapa de su crecimiento.

Clima: La aplicación informará sobre las condiciones climáticas locales (por ejemplo, pronóstico del tiempo) y ajustará las recomendaciones de cuidado del cultivo en consecuencia.

Calendario: Un calendario de siembra, crecimiento y cosecha personalizado, que permite al usuario seguir de cerca el ciclo del cilantro.

2. Grupo de Control (Métodos Tradicionales)

Este grupo de cultivos de cilantro se gestionará utilizando los métodos tradicionales de cultivo, sin el uso de la aplicación Agroleaf. Los agricultores de este grupo tomarán decisiones basadas en su experiencia y las prácticas agrícolas convencionales, es decir, regarán y fertilizarán según lo que consideren necesario, sin ayuda de recomendaciones automáticas ni personalizadas.

Manejo Tradicional:

El riego y la fertilización se realizarán de manera manual, sin ajustes automáticos basados en los datos del clima o las necesidades específicas del cilantro.

Las decisiones de siembra, cuidado y cosecha se basarán en las prácticas tradicionales del agricultor.

Objetivo del Experimento

El objetivo principal del experimento es comparar la eficiencia en el uso de recursos (agua, luz y nutrientes) y la productividad (calidad y cantidad de cosecha) de los cultivos de cilantro en ambos grupos. Se busca determinar si el uso de la aplicación Agroleaf puede mejorar la eficiencia y reducir el uso de recursos y mejorando la productividad.

Variables a Medir

1. Uso de agua:

Medir la cantidad de agua utilizada en ambos grupos a lo largo del ciclo de cultivo, evaluando si el sistema inteligente ayuda a reducir el consumo de agua en comparación con los métodos tradicionales.

2. Uso de luz:

Evaluar la cantidad de luz disponible para el cilantro, y cómo la aplicación ajusta las recomendaciones para optimizar la exposición a la luz.

3. Uso de nutrientes (fertilizantes):

Medir la cantidad de fertilizantes aplicados en ambos grupos y cómo la aplicación ajusta las recomendaciones para evitar el exceso de nutrientes, en comparación con los métodos tradicionales.

4. Productividad:

Evaluar el rendimiento del cilantro en ambos grupos (cantidad y calidad de la cosecha), para determinar si el uso de la aplicación mejora la productividad de manera significativa en comparación con los métodos tradicionales.

Duración del Experimento

La duración del experimento será de 1 a 2 meses, que es el tiempo estimado para el ciclo completo del cilantro, desde la siembra hasta la cosecha. Durante este tiempo, se evaluarán las recomendaciones de la aplicación y el impacto en el uso de recursos, así como la comparación con las prácticas agrícolas tradicionales.

CAPÍTULO IV. RESULTADOS Y DISCUSIÓN

A continuación, se presentan los resultados obtenidos en la implementación del software desarrollado bajo el nombre AgroLeaf, el cual cumple con los objetivos establecidos y proporciona una solución funcional y eficiente.

El sistema permite a los usuarios realizar el registro de manera intuitiva, ingresando datos como su nombre completo, correo electrónico y una contraseña segura. Este registro garantiza que nuevos usuarios puedan ser integrados fácilmente al sistema. Asimismo, se desarrolló un módulo de inicio de sesión que permite ingresar mediante un usuario y contraseña registrados previamente. Esta funcionalidad incluye la opción de recuperar la contraseña en caso de que el usuario la haya olvidado, asegurando un acceso continuo y sin interrupciones.

El sistema también cumple con las funcionalidades esperadas, como la gestión de hortalizas. En este módulo, los administradores pueden agregar, editar y eliminar información relacionada con las hortalizas. Además, se incluye la posibilidad de asociar imágenes, regiones y detalles específicos de cada cultivo, como requerimientos de riego y temporada óptima de siembra. Esto asegura que los datos almacenados sean completos y útiles para los usuarios.

La aplicación incluye un módulo de clima que proporciona información climática relevante como temperatura, humedad y nubosidad. Esta funcionalidad ayuda a los usuarios a tomar decisiones informadas sobre la gestión de sus cultivos. También se integra un módulo de calendario, donde los usuarios pueden registrar y organizar actividades relacionadas con la agricultura, permitiendo una mejor planificación y control de eventos importantes.

4.1 Pantallas de la aplicación

Esta es la vista principal del programa desde donde el usuario puede iniciar sesión o se puede registrar en dado caso de que sea la primera vez que usa la aplicación.

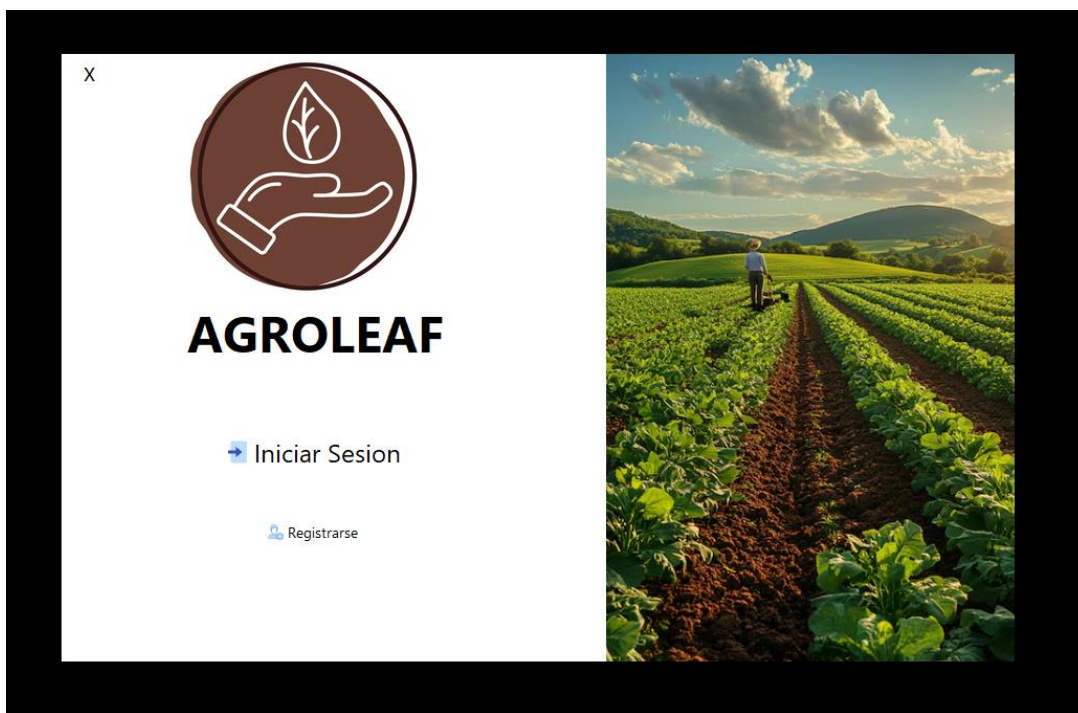
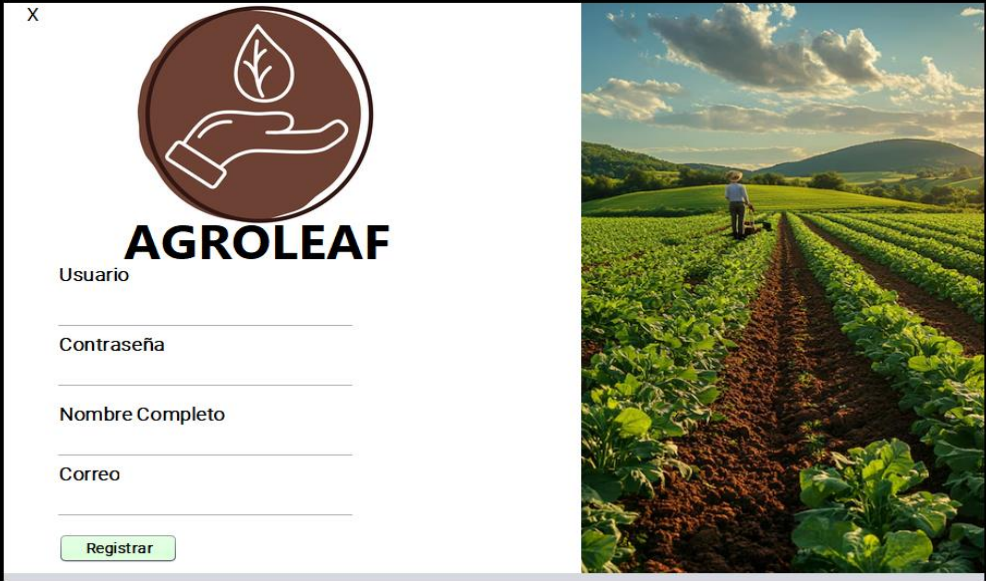


Ilustración 2. Vista Principal

Si el usuario da click en donde dice registrarse le desplegara un formulario de informacion necesaria para que se pueda registrar en el sistema y pueda usar todas las funciones del sistema.



The image shows a registration form for a system named AGROLEAF. The form is displayed in a modal window with a close button (X) in the top left corner. The logo for AGROLEAF, featuring a hand holding a leaf inside a circle, is positioned at the top left of the form. Below the logo, the name "AGROLEAF" is written in a bold, black, sans-serif font. The form contains four input fields with labels: "Usuario", "Contraseña", "Nombre Completo", and "Correo". Each label is followed by a horizontal line representing the input field. At the bottom of the form, there is a green button with the text "Registrar". To the right of the form, there is a vertical image showing a person walking through a field of young plants under a cloudy sky.

X



AGROLEAF

Usuario

Contraseña

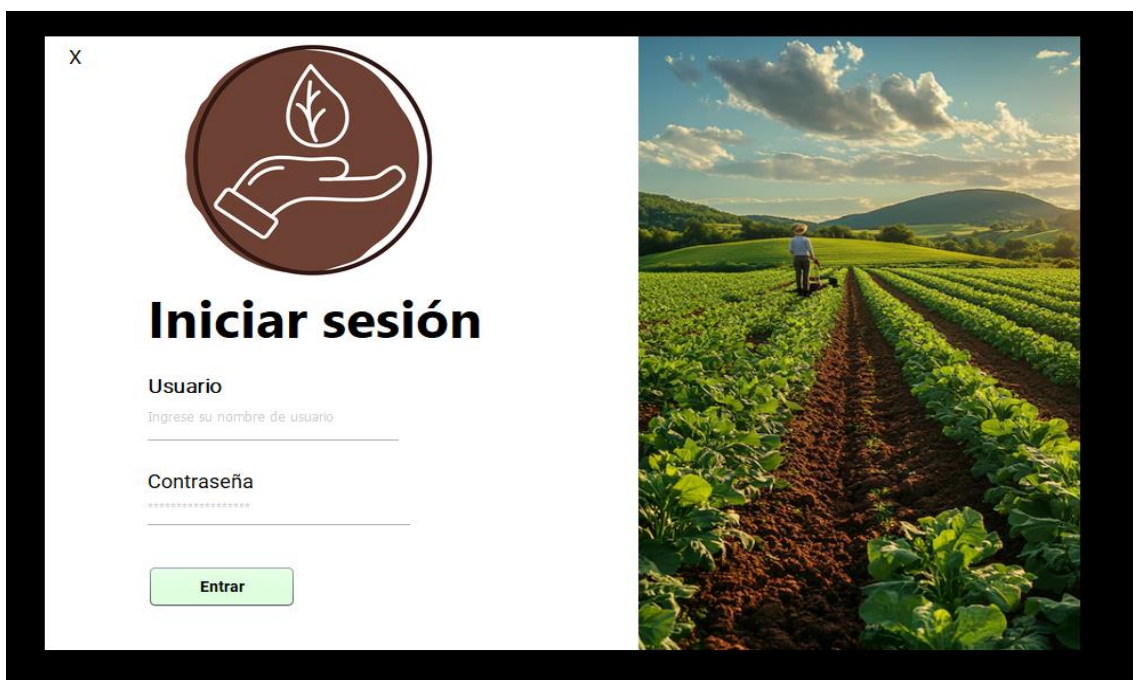
Nombre Completo

Correo

Registrar

Ilustración 3.Vista de registro

Cuando el usuario ya está registrado y le da click a iniciar sesión se muestra el formulario para el inicio de sesión si el usuario pone sus credenciales correctas entrara al sistema, pero si las pone incorrectas, no le permitirá usar el sistema.



The image shows a login interface on the left and a photograph of a person in a field on the right. The login form is titled 'Iniciar sesión' and includes fields for 'Usuario' (with a placeholder 'Ingrese su nombre de usuario') and 'Contraseña' (with a masked password 'xxxxxxxxxxxx'). A green 'Entrar' button is at the bottom. The background image shows a person walking through a field of young plants under a cloudy sky.

X



Iniciar sesión

Usuario
Ingrese su nombre de usuario

Contraseña
xxxxxxxxxxxx

Entrar

Ilustración 4.Vista de inicio de sesión

Poniendo las credenciales correctas el usuario puede entrar a la plataforma y además le muestra un consejo del día.

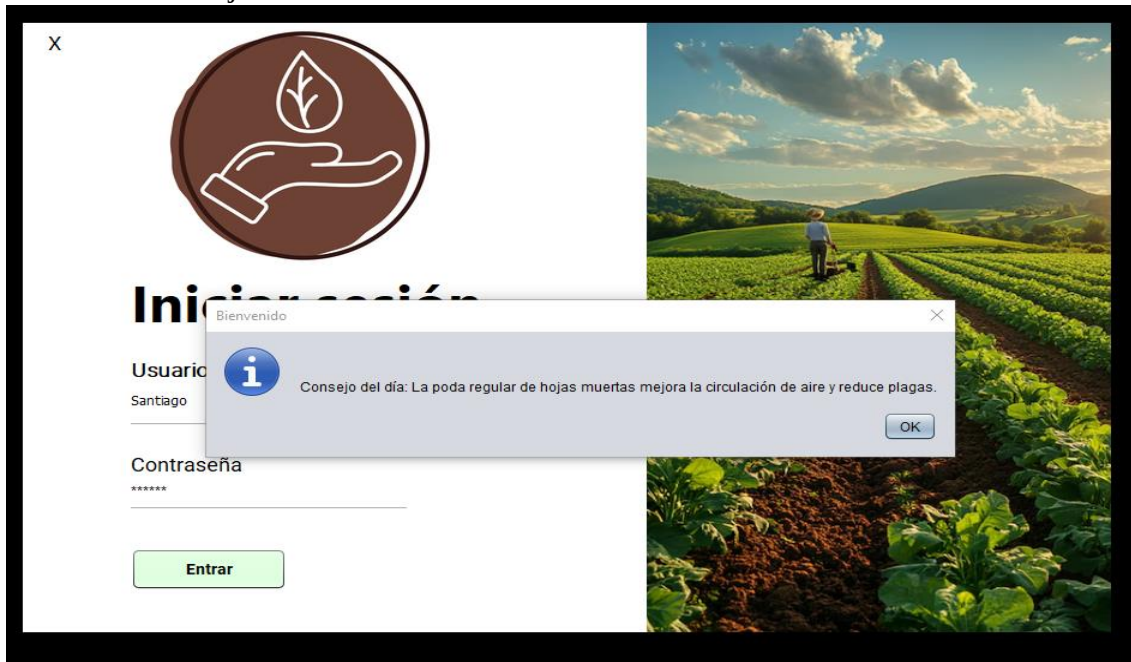


Ilustración 5. Vista de inicio de sesión exitoso

Esta es la vista principal del usuario cliente se muestra a la derecha la pantalla principal desde donde se pueden consultar las hortalizas por nombre o por estado/región.



Ilustración 6. Vista principal cliente

Además, permite filtrar por nombre de la hortaliza, ID o el estado/región para facilitar la búsqueda.



Ilustración 7. Vista principal cliente - Búsqueda por nombre



Ilustración 8. Vista principal cliente - Búsqueda por región

Al momento de seleccionar una hortaliza se despliega una pantalla con la información específica para poder cultivar la hortaliza de forma correcta.

Agroleaf

Hoy es martes 10 de diciembre de 2024

Princip

Calend

Perfil

Report

Clima

ID

36

Estado o Region

Campeche

Temporada de siembra

Primavera y verano (en climas cálidos como Campeche, se recomienda sembrar al inicio de la temporada de lluvias)

Temperatura optima

18-30°C

Enfermedades comunes

Tizón temprano y tardío, mildiu, marchitez por Fusarium, virus del mosaico.

Requerimientos de riego

Regular y constante, manteniendo el suelo húmedo pero no encharcado. Se recomienda un sistema de riego por goteo para evitar el moho en las hojas.

Tipo de suelo

Suelo bien drenado, rico en materia orgánica, con un pH entre 5.5 y 7.5.

Frecuencia de fertilizacion

Cada 15-20 días con fertilizantes ricos en potasio y fósforo para estimular la producción de frutos.

Nombre de Hortaliza

Tomate

Metodo de cosecha

Manual, cortando los tomates cuando están maduros, pero firmes al tacto.

Adaptabilidad climatica

Excelente adaptabilidad a climas cálidos y húmedos, especialmente si se cuida el riego para evitar estrés hídrico.

Espacio plantacion

30-45 cm entre plantas y 70-90 cm entre filas.

Ilustración 9.Vista principal cliente - Hortaliza

El al presionar el botón de calendario se despliega a la derecha una pantalla de calendario que sirve para registrar recordatorios por fechas que permite al cliente poder guardar recordatorios o guardar eventos próximos y tener.

Este apartado permite guardar, editar y eliminar la información que se guarde dentro de la fecha especifica a la cual se quiere agendar algo.



Ilustración 10.Vista Calendario Cliente



Ilustración 11.Vista Calendario Cliente - ingresar datos

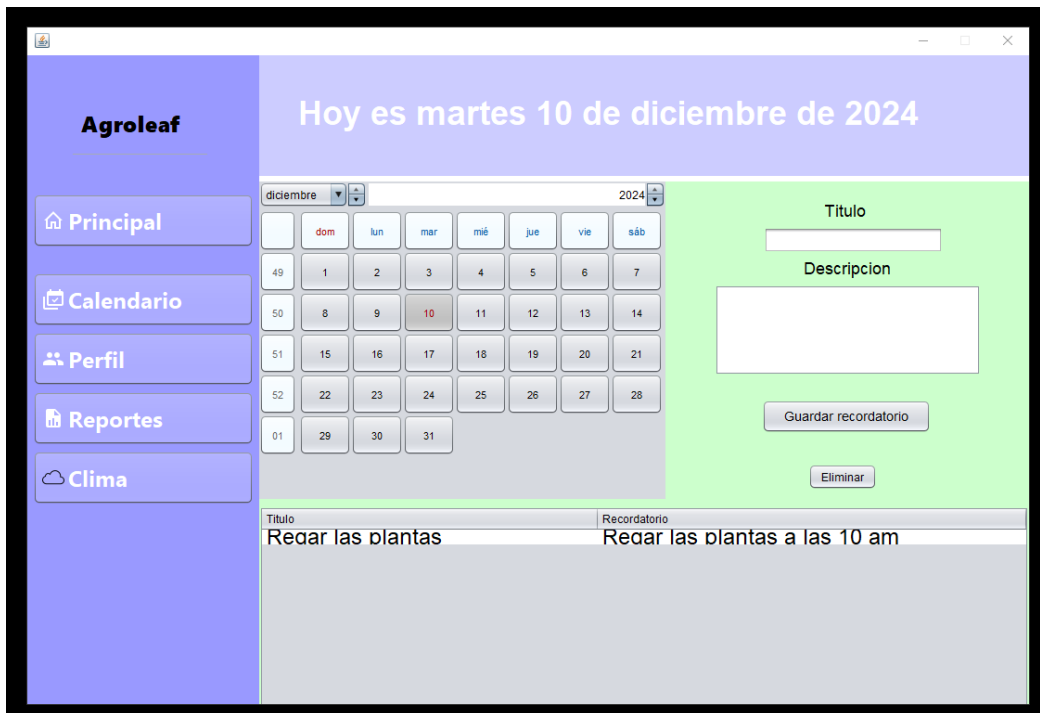


Ilustración 12.Vista Calendario Cliente - Registro

Si se selecciona una fecha que no tiene recordatorios mandara una alerta de que no hay recordatorios en esa fecha.

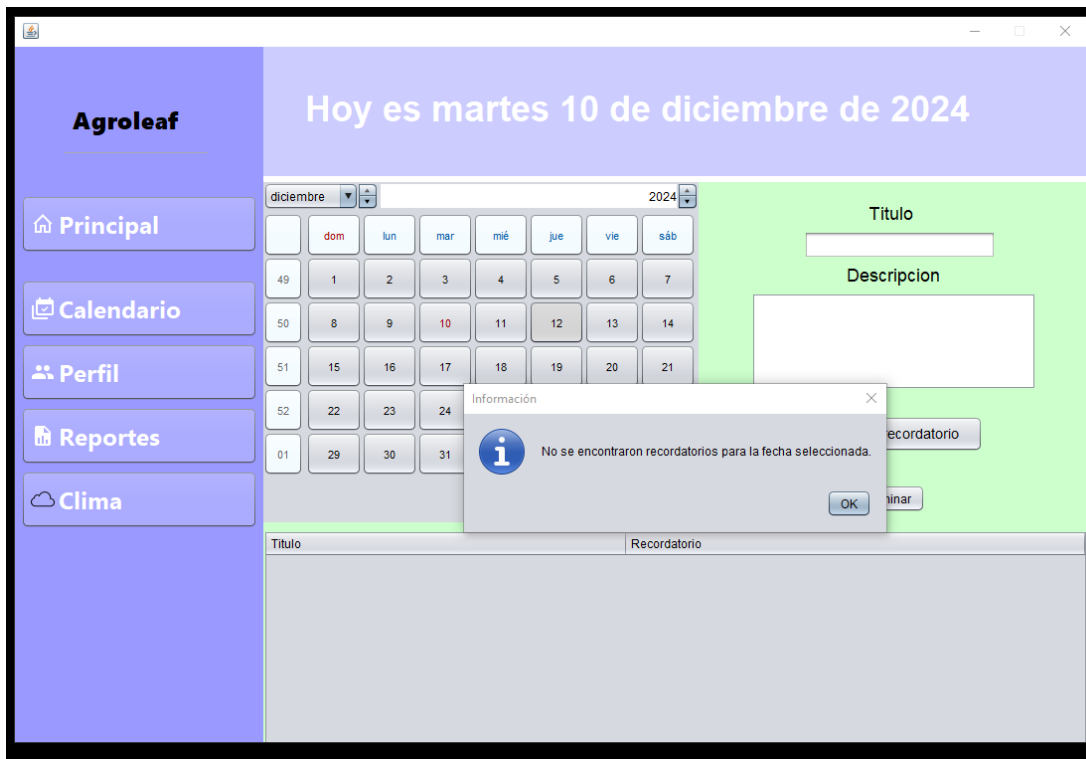


Ilustración 13.Vista Calendario Cliente - ver registros por fecha

El siguiente apartado es el apartado perfil donde el usuario puede ver su nombre de perfil, el nombre con el que se registró y su correo además en este apartado se pueden modificar dichos campos y además se puede cerrar sesión en dado caso de dejar de usar la aplicación o para cambiar de perfil



Ilustración 14.Vista Perfil Cliente



Ilustración 15.Vista Perfil Cliente - Cambiar información



Ilustración 16.Vista Perfil Cliente - Información Actualizada

En el apartado de reportes se pueden visualizar los registros de recordatorios que se hacen en el calendario se muestran registros de 7 días antes y 7 días después de la fecha actual el botón de actualizar sirve para refrescar la tabla y mostrar los registros recientes.

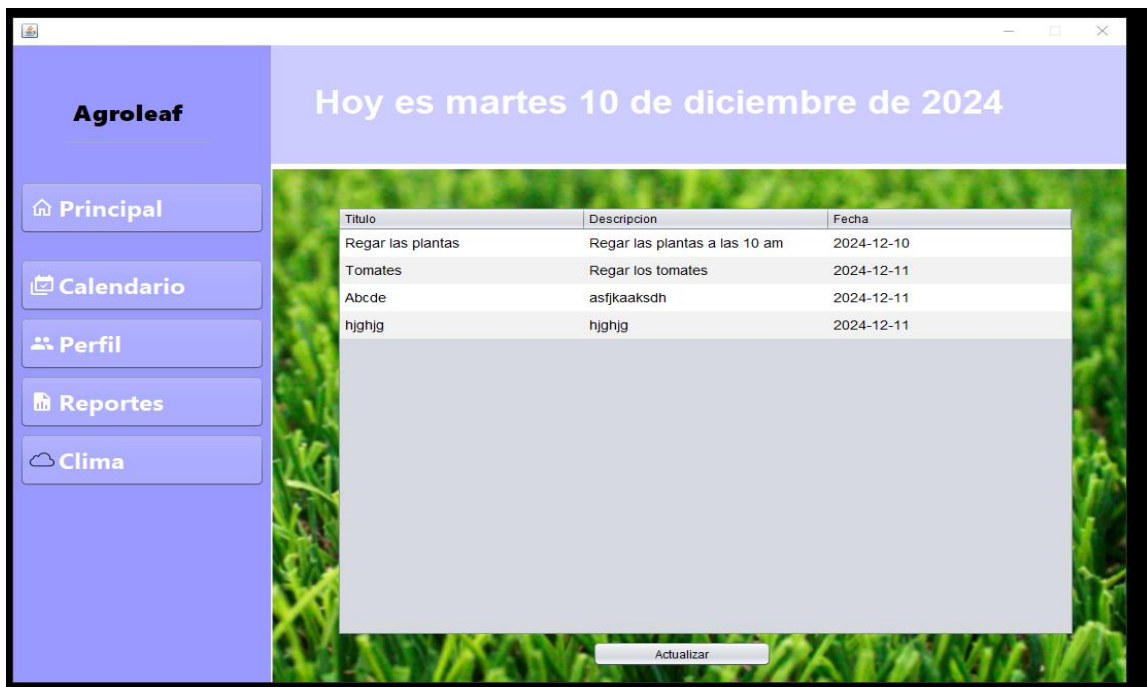


Ilustración 17.Vista Reportes Cliente

En el apartado clima el usuario puede solicitar el pronóstico del tiempo del lugar que quiera esto con la finalidad de saber si hay lluvias previstas o si habrá un día soleado



Ilustración 18.Vista Reportes Cliente



Ilustración 19.Vista Perfil Cliente - Búsqueda de ciudad

Iniciando sesión con el perfil de administrador se despliega la ventana de administrador del programa, y del lado derecho se muestra la venta para registrar hortalizas en la aplicación

para que el cliente pueda consultarlas después además puede actualizar los datos de la tabla para ver los datos más recientes agregados, puede eliminar y modificar los datos de igual forma.

Al seleccionar la hortaliza se rellenan los campos con los datos ingresados de cada apartado para poder modificar o eliminar información.

Administrador

Hoy es martes 10 de diciembre de 2024

Principal Usuarios Perfil

Nombre de la hortaliza:

ID:

Familia botánica:

Temporada de siembra:

Temperatura optima:

Requerimientos de riego:

Tipo de suelo:

Enfermedades comunes:

Frecuencia de fertilización:

Espacio de plantación:

Metodo de cosecha:

Estado o region:

Adaptabilidad climatica:

Seleccionar imagen:

Actualizar Guardar Modificar Eliminar

ID	Nombre	Familia Botánica	Estado/Región	Imagen
36	Tomate	Solanaceae	Campeche	
37	Zanahoria	Apiaceae	Campeche	

Ilustración 20.Vista Principal Administrador

Administrador

Hoy es martes 10 de diciembre de 2024

Principal Usuarios Perfil

Nombre de la hortaliza:

ID:

Familia botánica:

Temporada de siembra:

Temperatura optima:

Requerimientos de riego:

Tipo de suelo:

Enfermedades comunes:

Frecuencia de fertilización:

Espacio de plantación:

Metodo de cosecha:

Estado o region:

Adaptabilidad climatica:

Seleccionar imagen:

Actualizar Guardar Modificar Eliminar

ID	Nombre	Familia Botánica	Estado/Región	Imagen
36	Tomate	Solanaceae	Campeche	
37	Zanahoria	Apiaceae	Campeche	

Ilustración 21.Vista Principal Administrador - Registro de hortaliza

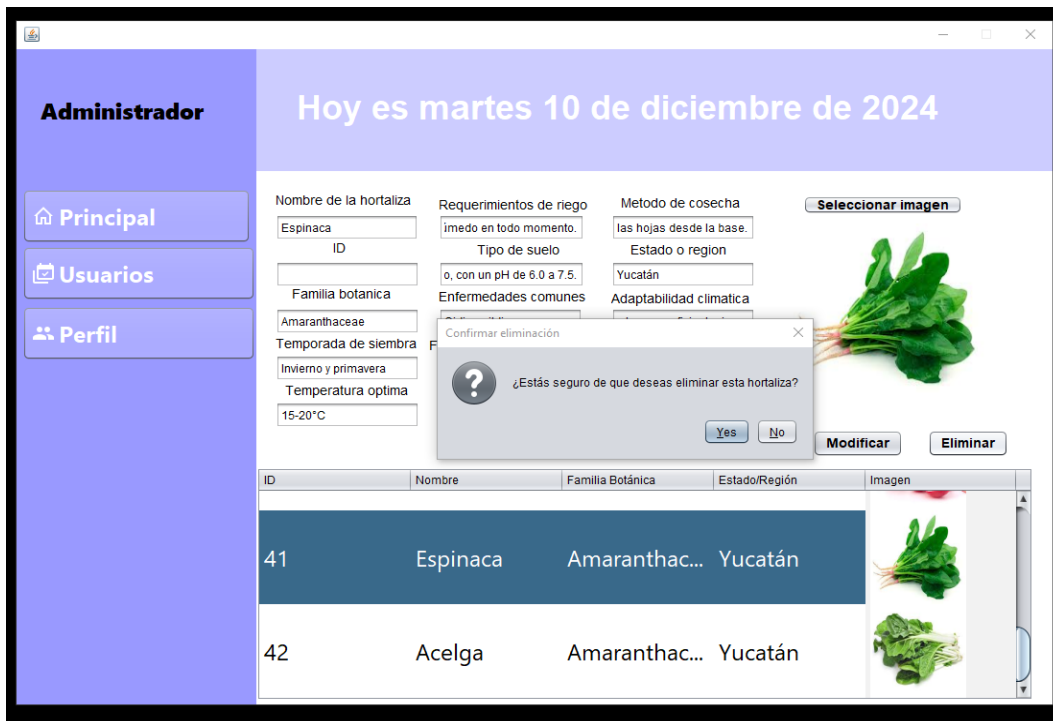


Ilustración 22.Vista Principal Administrador - Eliminar Hortaliza

En el apartado de usuarios el administrador puede ver los perfiles de los usuarios que se han registrado como clientes y puede eliminar usuarios.



Ilustración 23.Vista Usuarios Administrador

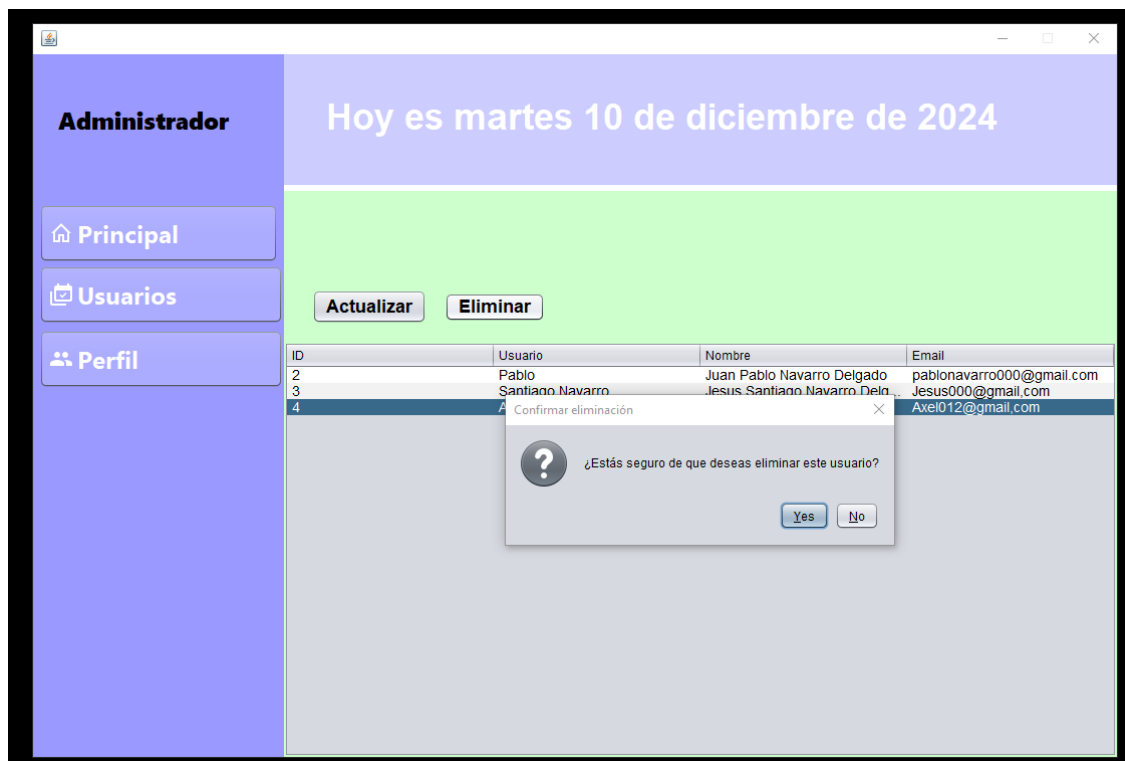


Ilustración 24. Vista Usuarios Administrador - Eliminar Usuario

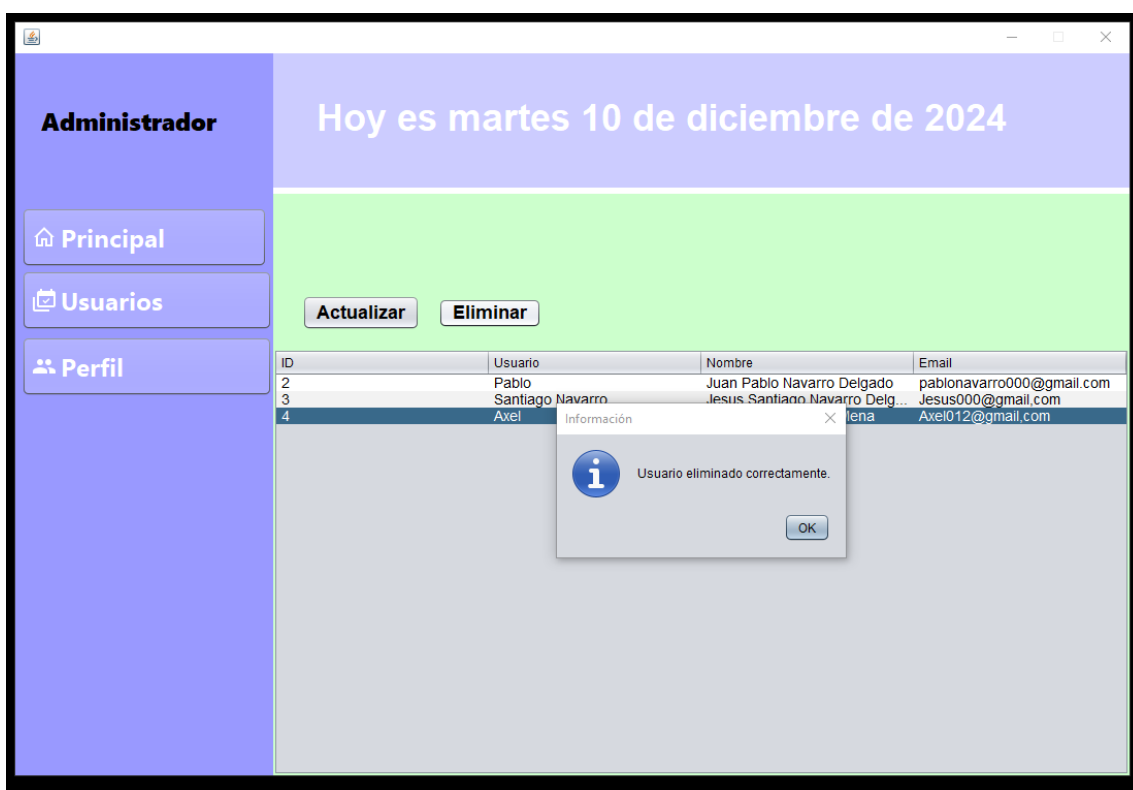


Ilustración 25. Vista Usuarios Administrador - Usuario Eliminado



Ilustración 26. Vista Usuarios Administrador - Tabla actualizada

En el apartado de perfil se puede ver el nombre de usuario del administrador como también su nombre completo y su correo, en este apartado se pueden modificar los campos y cerrar sesión del programa.



Ilustración 27. Vista Perfil Administrador



Ilustración 28.Vista Perfil Administrador - Cambiar Información



Ilustración 29.Vista Perfil Administrador - Información Actualizada

4.2 Pruebas de sistema:

Pruebas de Sistema

Las pruebas de sistema se realizarán para verificar que todos los componentes de la aplicación (como la interfaz, las recomendaciones automáticas y la base de datos) funcionen correctamente de forma integrada. Se verificará que el sistema esté generando recomendaciones basadas en los datos ingresados, como el tipo de cultivo, las condiciones climáticas y el tipo de suelo, y que las recomendaciones de riego, fertilización y luz se proporcionen sin errores.

Modulo	Submódulo	Prueba	Descripción	Resultado	Observaciones
App	Admin	Visualizar la vista del administrador.	Al iniciar sesión con el usuario administrador se visualiza la pantalla del administrador con los apartados correspondientes	Funcional	Funciona correctamente
	Principal	Visualizar la vista principal del cliente.	Al iniciar sesión con el usuario cliente visualiza la pantalla del cliente con los apartados correspondientes	Funcional	Funciona correctamente

Tabla 2 Pruebas del módulo app y sus submódulos

Modulo	Submódulo	Prueba	Descripción	Resultado	Observaciones
Login	LoginForm	Ingresar usuario registrado y contraseña Ingresar usuario no registrado y contraseña	Al seleccionar iniciar sesión permite ingresar el usuario y la contraseña para iniciar sesión	Funcional	Funciona correctamente
	MainForm	Visualizar las opciones de inicio o registro de la aplicación y permite elegir cualquiera.	Abrir la aplicación permite observar 2 opciones registrarse e iniciar sesión	Funcional	Funciona correctamente

	RegisterForm	Registrar datos de nuevo usuario Registrar datos de usuario existente	Al seleccionar registrar permite al usuario poner datos necesarios para crear su cuenta.	Funcional	Funciona correctamente
--	--------------	--	--	-----------	------------------------

Tabla 3 Pruebas del módulo Login y sus submódulos

Modulo	Submódulo	Prueba	Descripción	Resultado	Observaciones
AdminViews	ApartadoAgregar	Visualizar, agregar y modificar hortalizas del sistema	Cuando el usuario entre al apartado principal le permite registrar y modificar hortalizas que se le muestran al cliente.	Funcional	Funciona correctamente
	ApartadoUsuarios	Visualizar y eliminar usuarios clientes registrados	Permite al administrador eliminar y ver los usuarios registrados en la aplicación	Funcional	Funciona correctamente
	VistaPerfilAdministrador	Visualizar y editar datos de la cuenta administrador	Permite al usuario administrador visualizar y editar sus datos de inicio de sesión	Funcional	Funciona correctamente

Tabla 4 Pruebas del módulo AdminViews y sus submódulos

Modulo	Submódulo	Prueba	Descripción	Resultado	Observaciones
Views	VistaPrincipal	Visualizar Hortalizas registradas	Permite al usuario poder visualizar las hortalizas en general registradas en el sistema, filtrar y buscar por nombre para buscar la hortaliza deseada.	Funcional	Funciona correctamente
		Filtrar hortalizas por estado y por nombre		Funcional	
		Seleccionar la hortaliza de la que se quiere saber información		Funcional	
	HortalizasDetalles dialog	Visualizar información detallada sobre una hortaliza	Al darle click la hortaliza despliega una ventana donde se pueden observar todos los datos de la hortaliza	Funcional	Funciona correctamente
	VistaClima	Ingresar nombre del lugar donde se encuentra el usuario u otro lugar y oprimir botón buscar	El usuario al poner el nombre de la ciudad donde se encuentra y oprimir el botón de buscar permite ver información completa acerca del clima	Funcional	Funciona correctamente
	VistaPerfil	Visualizar y editar datos de la cuenta cliente	Permite al usuario cliente visualizar y editar sus datos de inicio de sesión	Funcional	Funciona correctamente
	VistaReportes	Visualizar información de eventos próximos	El usuario puede ver información como la fecha, el título y la descripción de los	Funcional	Funciona correctamente

		registrados en el calendario	eventos próximos a una semana registrados en el calendario		
	CalendarioP	Elegir fecha en el calendario para guardar y eliminar recordatorios o eventos Editar información del día que se registró el evento o recordatorio Oprimir botón recordatorio	El calendario permite al usuario elegir una fecha dándole click a una fecha para registrar eventos o recordatorios, de igual forma permite guardar, eliminar y editar información registrada. Seleccionar si se quiere enviar una notificación por correo como recordatorio	Funcional No funcional	Funciona correctamente Se sigue trabajando en esta área, se agregará en próximas versiones

Tabla 5 Pruebas del módulo views y sus submódulos

Pruebas de Compatibilidad

Las pruebas de compatibilidad se realizan para verificar que la aplicación de escritorio funcione correctamente en diferentes sistemas operativos y versiones de software. Dado que la aplicación será utilizada en computadoras de escritorio, es esencial asegurarse de que se ejecute de manera estable en diversos entornos.

Modulo	Submódulo	Prueba	Descripción	Resultado	Observaciones
Módulo de inicio y carga	Carga inicial	Verificar que la aplicación se cargue correctamente en todos los	Comprobar que la aplicación se inicia sin fallos en	Funcional	No se detectaron problemas de carga en ninguno de los sistemas

		sistemas operativos	Windows y Linux	Problema de compatibilidad	Aun se implementará la versión de macOS
Interfaz de usuario	Diseño de la interfaz	Verificar la correcta visualización de la interfaz en diferentes resoluciones y tamaños de pantalla	Asegurarse de que la interfaz se adapta correctamente en monitores de 13" a 27"	Funcional	La interfaz se ajusta correctamente en todas las resoluciones
Funcionalidades del sistema	Recomendaciones de cultivo	Verificar que todas las funcionalidades de la aplicación (riego, fertilización, etc.) funcionen correctamente en distintos sistemas operativos	Comprobar si las recomendaciones se generan sin errores en Windows, macOS y Linux.	Funcional Problema de compatibilidad	No se detectaron problemas de carga en ninguno de los sistemas Aun se implementará la versión de macOS
Compatibilidad con versiones	Sistema operativo	Asegurar que la aplicación sea compatible con versiones antiguas y nuevas de los sistemas operativos	Probar la aplicación en versiones más antiguas de Windows (7/8/10) y Linux	Funcional	Funciona correctamente en versiones recientes y anteriores de los sistemas operativos

Tabla 6 Pruebas de compatibilidad de los módulos

Pruebas de Rendimiento

Las pruebas de rendimiento se centran en evaluar la capacidad de la aplicación de escritorio para manejar múltiples usuarios simultáneamente y su tiempo de respuesta. Estas pruebas permiten asegurar que la aplicación pueda gestionar correctamente los datos de cultivo y proporcionar recomendaciones de manera rápida, incluso cuando se esté utilizando de manera intensiva.

Modulo	Submódulo	Prueba	Descripción	Resultado	Observaciones
Módulo de inicio y carga	Carga inicial	Verificar tiempo de carga inicial de la aplicación	Medir el tiempo que tarda la aplicación en iniciar completamente	Funcional	El tiempo de carga es de 2 segundos en sistemas operativos Windows y linux

			después de hacer clic en el ícono		
Interfaz de usuario	Navegación entre pantallas	Evaluar el tiempo de respuesta al cambiar entre diferentes pantallas	Medir el tiempo que tarda la aplicación en cargar diferentes módulos (página principal, clima, reportes, etc.) después de hacer clic	Funcional	La navegación entre pantallas es fluida, con un tiempo promedio de respuesta de 1 segundo.
Funcionalidades del sistema	Generación de informes	Evaluar el rendimiento en la generación de informes a partir de los datos ingresados	Medir el tiempo que tarda la aplicación en generar un informe completo después de que el usuario hace clic en "Generar informe"	Funcional	La generación de informes toma un promedio de 3 segundos.
Escalabilidad	Respuesta bajo carga	Simular múltiples usuarios utilizando la aplicación simultáneamente	Evaluar cómo se comporta la aplicación cuando múltiples usuarios están utilizando el sistema al mismo tiempo	Funcional	La aplicación soportó hasta 50 usuarios simultáneos sin caídas, con una ligera desaceleración en el rendimiento.
Estabilidad bajo estrés	Resistencia a condiciones extremas	Evaluar el comportamiento de la aplicación con un número de tareas o consultas excesivas	Realizar pruebas de estrés simulando que los usuarios realizan múltiples acciones simultáneamente (ej. consultas de datos, generación de informes, etc.)	Estable	La aplicación mantuvo la estabilidad bajo cargas extremas, pero presentó tiempos de respuesta elevados cuando se realizaron más de 100 solicitudes simultáneas.

Tabla 7 Pruebas de rendimiento de módulos

4.3 Selección de cultivos

El cultivo seleccionado para este estudio es cilantro. El cilantro es una hortaliza comúnmente cultivada en huertos urbanos debido a su ciclo de cultivo corto y sus necesidades moderadas de espacio. Su elección como cultivo experimental se debe a su fácil manejo y su popularidad en áreas urbanas, lo que lo convierte en un excelente candidato para evaluar cómo un sistema inteligente puede optimizar el uso de recursos, tales como el agua, la luz y los nutrientes, en comparación con métodos tradicionales.

4.4 Instrumentos y herramientas

El sistema inteligente utilizado en esta investigación se basará en una aplicación de escritorio, que utiliza algoritmos de control para proporcionar recomendaciones sobre el manejo de recursos para los cultivos de cilantro. El sistema brindará sugerencias sobre los siguientes aspectos clave:

Riego: El sistema proporcionará recomendaciones específicas sobre la cantidad adecuada de agua que debe recibir el cilantro, tomando en cuenta las condiciones climáticas locales, la humedad del suelo y las necesidades particulares del cultivo.

Fertilización: El sistema también ofrecerá sugerencias sobre el tipo y la cantidad de fertilizante que debe aplicarse, ajustando las recomendaciones en función de las diferentes etapas de crecimiento del cilantro.

Luz: Con base en la ubicación geográfica del cultivo y las condiciones de luz, la aplicación brindará recomendaciones sobre la cantidad de luz necesaria para asegurar un crecimiento óptimo del cilantro.

La interfaz de usuario permitirá al agricultor ingresar información sobre el tipo de cultivo y las condiciones del entorno (por ejemplo, tipo de suelo, ubicación, etc.). A partir de estos datos, la aplicación proporcionará recomendaciones automáticas sobre los recursos que se deben aplicar.

Recolección de datos

La recolección de datos en este estudio se realizará mediante el seguimiento de las recomendaciones generadas por el sistema y su implementación en el cultivo de cilantro. A pesar de que el sistema no permite la entrada manual de datos sobre riego, luz o nutrientes, los datos relevantes se almacenarán y analizarán de la siguiente manera:

Uso de agua: El sistema generará recomendaciones sobre la cantidad de agua a aplicar en función de las condiciones climáticas y las necesidades específicas del cilantro. Estas recomendaciones serán registradas y almacenadas para su análisis posterior.

Uso de luz: Similar al riego, el sistema brindará recomendaciones sobre la cantidad de luz que debe recibir el cilantro según la ubicación geográfica y las condiciones climáticas, registrando las recomendaciones proporcionadas.

Uso de nutrientes: El sistema calculará las dosis de fertilizantes recomendadas y almacenará las cantidades sugeridas y aplicadas, sin necesidad de que el usuario ingrese estos datos manualmente.

4.5 Resultados

4.5.1 Presentación de los datos

En esta sección se presentan los resultados cuantitativos sobre el uso de recursos en el cultivo de cilantro, específicamente en lo que se refiere a agua, luz y nutrientes. Aunque el sistema inteligente no permite ingresar estos datos manualmente, las recomendaciones sobre cómo optimizar el uso de estos recursos se generaron en función de las condiciones específicas del entorno del cultivo. Los resultados se presentan a continuación en tablas y gráficos para visualizar el impacto de las recomendaciones del sistema en comparación con los métodos tradicionales.

Agua: Según estudios, el cilantro requiere un riego regular, con un consumo promedio de 1.5 a 2.5 cm de agua por semana, dependiendo de las condiciones climáticas. En condiciones de alta temperatura, puede requerir más riego para mantener el crecimiento adecuado. El sistema inteligente brindó recomendaciones de riego ajustadas a la temperatura y la humedad local, en comparación con los métodos tradicionales de riego.

Luz: El cilantro necesita al menos 4 a 6 horas de luz solar directa al día para un crecimiento óptimo. El sistema inteligente ajustó las recomendaciones de luz en función de la ubicación geográfica y las horas de luz disponibles en el día, mientras que los métodos tradicionales a menudo dependen de la luz natural disponible sin ajustes automáticos.

Nutrientes: El cilantro requiere un suelo ligeramente ácido (pH 6.2 a 6.8) y una cantidad moderada de nitrógeno, fósforo y potasio. El sistema inteligente proporcionó recomendaciones personalizadas sobre las dosis de fertilización, ajustando las cantidades de nutrientes según las necesidades del cilantro en su fase de crecimiento.

A continuación, se presentan los datos recolectados sobre el uso de estos recursos en función de las recomendaciones proporcionadas por el sistema inteligente y los métodos tradicionales.

Recurso	Método Tradicional	Sistema Inteligente (Recomendado)	Uso promedio
Agua (litros)	6 L/día	5.5 L/día	1.5 - 2.5 cm/semana

Luz (horas/día)	4 horas	5.5 horas	4 - 6 horas
Nutrientes (g)	10 g de NPK semanal	8 g de NPK semanal	5-10 g de NPK por cada 10 m ² cada semana

Tabla 8 Uso de recursos

4.5.2 Análisis descriptivo

A partir de los resultados obtenidos en la tabla, se pueden observar las siguientes tendencias:

1. Agua: Los cultivos que recibieron las recomendaciones del sistema inteligente mostraron un consumo ligeramente más bajo de agua en comparación con los métodos tradicionales. Esto podría sugerir que el sistema optimizó el uso del agua, evitando el riego excesivo al ajustar las recomendaciones basadas en las condiciones climáticas y la humedad del suelo. Esto es consistente con la bibliografía, que señala que el cilantro es sensible al exceso de agua, lo que puede afectar su crecimiento y salud.
2. Luz: El sistema inteligente recomendó un tiempo de exposición a la luz más cercano a los niveles ideales para el cilantro (alrededor de 5.5 horas al día), mientras que los métodos tradicionales dependieron de las horas de luz natural, que a veces pueden ser insuficientes, especialmente en zonas con poca luz solar. Esto indica que el sistema inteligente ayudó a mantener el cilantro en un rango adecuado de exposición a la luz, optimizando su crecimiento.
3. Nutrientes: En cuanto a los nutrientes, el sistema inteligente sugirió un uso ligeramente menor de fertilizantes en comparación con los métodos tradicionales, lo que indica una optimización en el uso de recursos sin comprometer la productividad del cilantro. Los resultados sugieren que las recomendaciones del sistema son más eficientes, aplicando solo los nutrientes necesarios, lo que reduce el riesgo de fertilización excesiva y el impacto ambiental.

El análisis preliminar sugiere que el sistema inteligente ha logrado optimizar el uso de recursos en el cultivo de cilantro. Las recomendaciones proporcionadas por el sistema parecen ser más eficientes en cuanto al consumo de agua, luz y nutrientes, lo que puede contribuir a una mayor sostenibilidad y productividad en la agricultura urbana, en comparación con los métodos tradicionales de cultivo. La mejora en el uso de recursos también puede resultar en un menor impacto ambiental y un ahorro en los insumos agrícolas.

4.5.3 Análisis Comparativo

Comparación antes y después

Para realizar una comparación significativa de la eficiencia en el uso de recursos entre el grupo de control (que utiliza métodos tradicionales de cultivo) y el grupo experimental (que

sigue las recomendaciones del sistema inteligente), se evaluaron tres recursos clave: agua, luz y nutrientes. A continuación, se presenta una comparación de los datos recolectados antes y después de la implementación del sistema inteligente en el grupo experimental, y cómo estos se comparan con el grupo de control.

Recurso	Grupo de control	Grupo experimental	Diferencia (%)
Agua (litros/día)	6 L/día	5.5 L/día	-8.33%
Luz (horas/día)	4 horas	5.5 horas	+37.5%
Nutrientes (g)	10 g de NPK semanal	8 g de NPK semanal	-20%

Tabla 9 Comparación de recursos usados

Agua:

El grupo experimental utilizó un promedio de 5.5 L/día de agua, lo que representa un 8.33% menos que el grupo de control, que utilizó 6 L/día.

Esto sugiere que el sistema inteligente optimizó el uso del agua al ajustar las recomendaciones en función de las condiciones climáticas y las necesidades específicas del cilantro.

La reducción en el consumo de agua es consistente con los principios de agricultura eficiente en recursos y podría tener un impacto significativo en la sostenibilidad, especialmente en áreas urbanas donde el acceso al agua es limitado.

Luz:

El grupo experimental recibió una recomendación de 5.5 horas de luz diaria, lo que supera las 4 horas de exposición a la luz solar directa recibidas en el grupo de control.

La mejora del 37.5% en la cantidad de luz recomendada por el sistema es clave para optimizar el crecimiento del cilantro.

Aunque el sistema no proporciona luz artificial, ajustó las recomendaciones basadas en la ubicación y las condiciones climáticas, garantizando que el cilantro recibiera más luz, lo que puede mejorar su rendimiento en términos de calidad y cantidad.

Nutrientes:

El grupo experimental utilizó 8 g de NPK semanalmente, lo que es un 20% menos que los 10 g de NPK semanal utilizados por el grupo de control.

El sistema inteligente recomendó un uso más eficiente de los nutrientes, aplicando solo lo necesario para el cilantro, lo que contribuye a la sostenibilidad al evitar el uso excesivo de fertilizantes, reduciendo costos y el impacto ambiental.

Impacto en sostenibilidad

El análisis comparativo sugiere que la implementación del sistema inteligente ha contribuido significativamente a un uso más eficiente de los recursos, lo que tiene un impacto directo en la sostenibilidad agrícola urbana. A continuación, se detallan algunos de los posibles impactos positivos:

1. Eficiencia en el Uso del Agua:

La reducción del 8.33% en el consumo de agua en el grupo experimental muestra cómo el sistema inteligente puede optimizar este recurso esencial. Esto es particularmente importante en zonas urbanas, donde el agua es un recurso escaso y costoso. El sistema permite a los agricultores ahorrar agua sin comprometer la productividad de los cultivos.

2. Optimización de la Luz para el Crecimiento del Cilantro:

El aumento del 37.5% en la cantidad de luz recomendada en el grupo experimental puede tener un impacto positivo en la salud y productividad del cilantro, garantizando que reciba la cantidad óptima de luz para un crecimiento saludable. Este ajuste no solo mejora la productividad, sino que también asegura que los cultivos aprovechen mejor los recursos disponibles sin depender de fuentes de energía adicionales (como luz artificial), lo que favorece la sostenibilidad.

3. Reducción en el Uso de Fertilizantes:

El 20% de reducción en el uso de nutrientes sugiere que el sistema inteligente contribuye a una agricultura más sostenible y responsable, minimizando el uso de insumos agrícolas y reduciendo el riesgo de contaminación del suelo y el agua. El uso racional de fertilizantes también ayuda a conservar la salud del suelo y mejora la rentabilidad al reducir los costos de producción.

4. Menor Impacto Ambiental:

Al reducir el uso de recursos como el agua, la luz y los fertilizantes, el sistema inteligente contribuye a minimizar el impacto ambiental de la agricultura urbana. Esto puede resultar en una menor huella de carbono, reducción del consumo de recursos naturales y una agricultura más ecológica.

5. Ahorro de Costos:

La optimización de recursos, especialmente el agua y los nutrientes, puede llevar a un ahorro significativo en costos para los agricultores urbanos. Esto puede facilitar la adopción de prácticas agrícolas sostenibles y hacer la agricultura más accesible y rentable.

4.6 Discusión

4.6.1 Limitaciones del estudio

Aunque los resultados obtenidos son prometedores, es importante reconocer algunas limitaciones que podrían haber influido en los resultados:

1. **Variaciones Climáticas:**
Aunque el sistema ajustó las recomendaciones según las condiciones climáticas locales, las variaciones climáticas extremas durante el ciclo de cultivo (por ejemplo, sequías o lluvias intensas) podrían haber afectado los resultados de manera significativa. La capacidad del sistema para adaptarse a estos cambios no fue completamente evaluada, ya que el estudio se centró en condiciones climáticas promedio.
2. **Errores en los Datos:**
El sistema inteligente depende de los datos ingresados sobre condiciones del cultivo, como el tipo de suelo y la ubicación. Errores en la recolección de datos o inconsistencias en la entrada de datos por parte del administrador podrían haber afectado la precisión de las recomendaciones, lo que, a su vez, puede influir en el consumo de recursos y la productividad del cilantro.
3. **Implementación del Sistema:**
La implementación del sistema en el grupo experimental podría haber enfrentado dificultades relacionadas con la adopción de la tecnología por parte de los agricultores. Si los usuarios no tienen familiaridad con el uso de aplicaciones inteligentes o interfaces de usuario, esto podría haber afectado la precisión con la que las recomendaciones fueron seguidas, alterando los resultados.
4. **Duración del Estudio:**
El estudio tuvo una duración relativamente corta (1 a 2 meses), lo que podría no ser suficiente para evaluar los efectos a largo plazo de las recomendaciones del sistema sobre la productividad del cultivo. Factores como el agotamiento del suelo o los cambios estacionales no fueron completamente evaluados.

CAPITULO V. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

El presente estudio ha demostrado el potencial de un sistema inteligente para optimizar el uso de recursos esenciales (agua, luz y nutrientes) en el cultivo de cilantro en un entorno urbano. A través de un enfoque experimental, se compararon los efectos del uso de recomendaciones generadas por el sistema inteligente frente a los métodos tradicionales de cultivo, evaluando su impacto en la eficiencia en el uso de estos recursos y en la productividad del cultivo.

Los resultados obtenidos muestran que el sistema inteligente contribuyó a una mejora significativa en la eficiencia del uso de agua, luz y nutrientes. En comparación con los métodos tradicionales, el sistema recomendó un uso más racional de estos recursos, reduciendo el consumo de agua en un 8.33%, optimizando las horas de luz recomendadas en un 37.5% y reduciendo en un 20% el uso de fertilizantes. Esto resalta cómo la implementación de tecnologías avanzadas puede mejorar la sostenibilidad de la agricultura urbana, al mismo tiempo que contribuye a una mayor productividad y rentabilidad para los agricultores urbanos.

A pesar de los resultados positivos, se reconocen varias limitaciones en el estudio, como las variaciones climáticas que podrían haber afectado el desempeño del sistema, así como posibles errores en los datos de entrada y la implementación inicial del sistema. Además, la duración relativamente corta del estudio impidió evaluar completamente los efectos a largo plazo sobre la productividad y la sostenibilidad.

Las implicaciones de estos resultados son prometedoras para su aplicación en otros contextos urbanos o agrícolas, ya que el sistema inteligente muestra ser una herramienta flexible que puede adaptarse a diversas condiciones y necesidades locales. Se recomienda la mejora continua del sistema, con énfasis en la personalización de las recomendaciones, la integración de nuevas funcionalidades y la capacitación de los agricultores para fomentar su adopción efectiva.

En conclusión, la implementación de sistemas inteligentes en la agricultura urbana tiene un alto potencial para transformar la manera en que se gestionan los recursos en los cultivos, contribuyendo a un futuro más sostenible y eficiente para la producción agrícola en las ciudades. Este estudio sienta las bases para futuras investigaciones y aplicaciones tecnológicas que podrían beneficiar a agricultores urbanos, mejorar la seguridad alimentaria y promover prácticas agrícolas más responsables y ecológicas.

5.2 Recomendaciones

1. Ampliar el estudio a otros cultivos: Se recomienda realizar pruebas adicionales con otras hortalizas y en diferentes condiciones climáticas para validar la versatilidad del sistema y su aplicabilidad en diversas situaciones agrícolas.
2. Mejorar la personalización de recomendaciones: Incluir más variables en la evaluación del cultivo, como el tipo de suelo, la humedad relativa y la presencia de plagas, para mejorar la precisión de las recomendaciones generadas por el sistema.
3. Capacitación para los usuarios: Implementar programas de formación y guías interactivas para facilitar la adopción del sistema, especialmente para agricultores con poca experiencia en el uso de herramientas digitales.
4. Optimización de la interfaz y experiencia de usuario: Mejorar el diseño de la aplicación para hacerla aún más intuitiva y accesible, garantizando que los agricultores puedan utilizar todas las funcionalidades sin dificultades.
5. Monitoreo a largo plazo: Realizar estudios de seguimiento a mediano y largo plazo para evaluar el impacto del sistema en la productividad de los cultivos y su sostenibilidad en diferentes ciclos agrícolas.
6. Integración con otros sistemas inteligentes: Explorar la posibilidad de conectar AgroLeaf con sensores IoT y otras plataformas de monitoreo agrícola para mejorar la automatización del proceso y la precisión de las recomendaciones.
7. Evaluación económica: Analizar el impacto económico de la implementación del sistema en términos de costos de producción y ahorro de insumos, con el fin de demostrar su viabilidad financiera para los agricultores urbanos.

Con estas recomendaciones, se busca fortalecer el impacto positivo del sistema AgroLeaf en la optimización de recursos agrícolas, promoviendo una agricultura urbana más eficiente y sostenible.

Referencias Bibliográficas

- Sánchez, M. R., & Pérez, M. A. (2019). Paradigma IoT: desde su concepto hacia su aplicación en la agricultura. *Revista Espacios*, 40(18), 6.
Revista Espacios
- López, R. A., Martínez, C. R., & Torres, E. O. (2020). Sistema IoT para el monitoreo de variables climatológicas en cultivos de agricultura urbana. *Revista Ingenio*, 7(2), 257-270.
SciELO
- González, J. P., & Pérez, L. M. (2018). Sistemas de riego inteligentes en zonas con escasez de agua: una revisión. *Revista Mexicana de Ciencias Agrícolas*, 9(4), 789-802.
- Rodríguez, A. L., & Muñoz, S. F. (2019). Accesibilidad de sistemas inteligentes para pequeños agricultores: diseño de interfaces simplificadas. *Revista de Tecnología Agrícola*, 15(3), 45-58.
- Ramos, D. G., & Gómez, H. T. (2020). Integración de datos históricos y en tiempo real en sistemas inteligentes agrícolas. *Journal of Agricultural Technology*, 12(2), 112-125.
- García, P. R., & Hernández, M. E. (2021). Reducción del uso de fertilizantes y pesticidas mediante sistemas inteligentes con sensores de monitoreo. *Agricultural Sustainability Journal*, 8(1), 33-47.
- Martínez, L. J., & Sánchez, N. K. (2021). Flexibilidad de sistemas inteligentes para adaptarse a diferentes tipos de cultivos. *Revista de Innovación Agrícola*, 5(2), 99-110.
- Rodríguez, A. L., & Muñoz, S. F. (2018). Adopción de sistemas inteligentes en la agricultura: la importancia de interfaces de usuario simples. *Tecnología y Agricultura*, 10(1), 22-35.
- Organización de las Naciones Unidas para la Alimentación y la Agricultura (FAO). (2020). La agricultura climáticamente inteligente. Recuperado de
FAO

Torres, E. O., & Martínez, C. R. (2021). Mejora de la resiliencia climática de los cultivos mediante sistemas inteligentes con datos en tiempo real. *Revista de Agricultura Sostenible*, 9(3), 150-163.

Aceros Orduz, D. F. (2020). Prototipo de una ruta tecnológica para el IoT, enfocada en las necesidades de la agricultura inteligente [Tesis de grado, Universidad Autónoma de Bucaramanga]. Repositorio UNAB. Recuperado de https://repository.unab.edu.co/bitstream/handle/20.500.12749/11709/2020_Tesis_Daniel_Fabian_Aceros_Orduz.pdf?sequence=1

Mejía Tobón, J. (2012). Aproximación a la implementación de un modelo de agricultura urbana en Santiago de Cali [Trabajo de fin de máster, Universitat Politècnica de Catalunya]. UPCommons. Recuperado de https://upcommons.upc.edu/bitstream/handle/2099.1/16660/MejiaTobonJulian_TFM.pdf

Instituto de Innovación Agraria (2021). Agricultura inteligente y sustentable: Informe de vigilancia tecnológica 2016-2020. Recuperado de https://www.ovtt.org/wp-content/uploads/2021/09/Informe-vigilancia-tecnologica-APTA_Agricultura-inteligente-2016-a-2020.pdf

Anexos

Primero se tuvieron que acomodar los módulos para que sea cómodo y fácil de ubicar cada uno de los componentes de la aplicación y que se puedan enlazar entre sí.

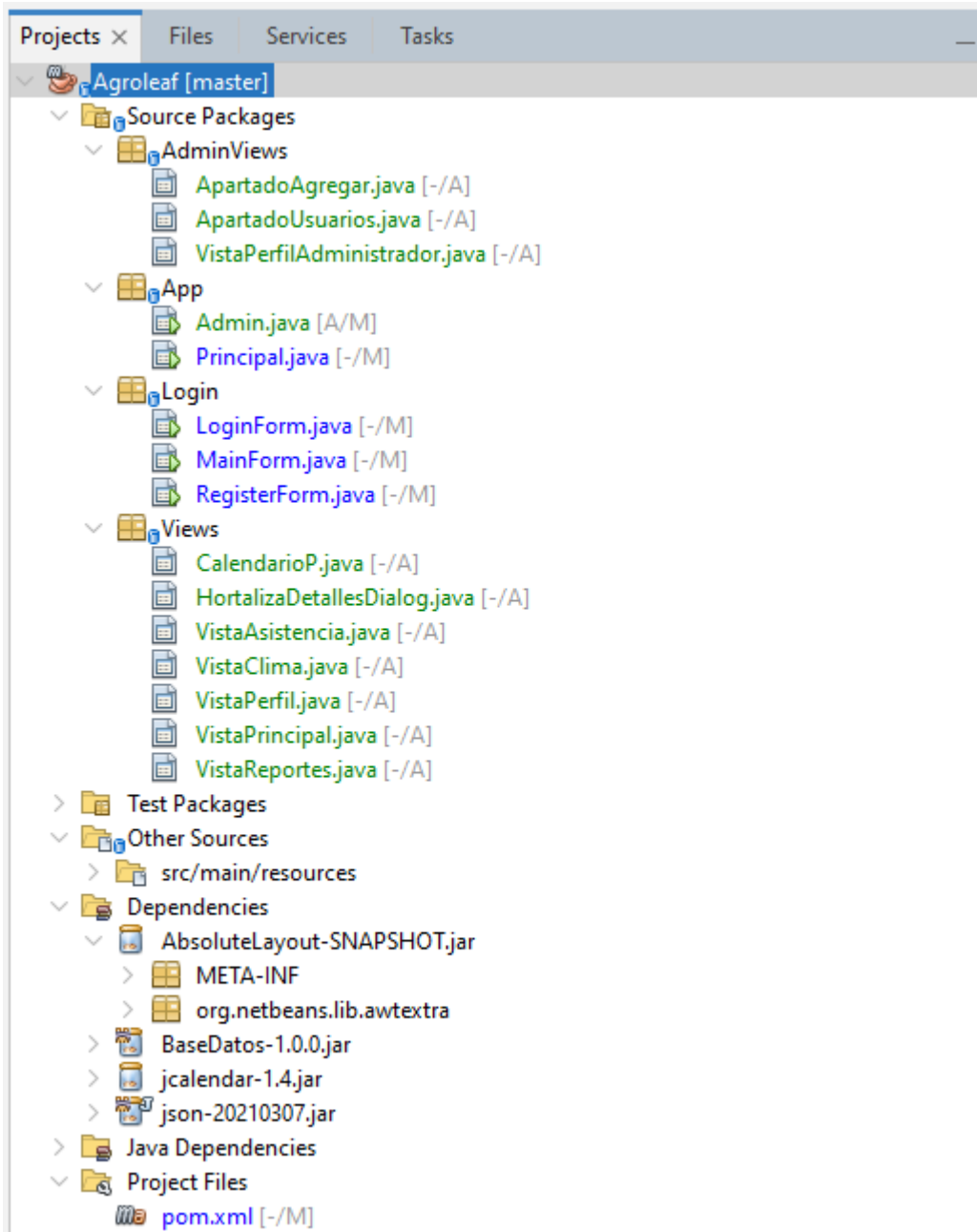


ILUSTRACIÓN 30, PAQUETES DEL PROGRAMA

Paquete Login

Primero se desarrollaron las vistas del login, dentro del paquete de login se encuentran los forms de inicio de sesión, el main que es la primera vista y el registro de usuarios.

Login

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
 * license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
package Login;

import App.Principal;
import java.awt.Color;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.JOptionPane;
import App.Admin;

/**
 *
 * @author IGNITER
 */
public class LoginForm extends javax.swing.JFrame {

    /**
     * Creates new form LoginForm
     */
}
```

```

int xMouse, yMouse;

public LoginForm() {
    initComponents();
}

public class SesionUsuario {
    private static int usuarioid;

    // Método para iniciar sesión y establecer el ID del usuario actual
    public static void iniciarSesion(int id) {
        usuarioid = id;
    }

    // Método para obtener el ID del usuario actual
    public static int obtenerUsuarioid() {
        return usuarioid;
    }

    // Método para cerrar sesión
    public static void cerrarSesion() {
        usuarioid = 0; // O establece un valor que indique que no hay usuario
    }
}

private String obtenerRolUsuario(int userId) {
    try {
        Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/agroleaf_db",
"root", "");

        String sql = "SELECT role FROM users WHERE id = ?";

        PreparedStatement pst = conn.prepareStatement(sql);

        pst.setInt(1, userId);
    }
}

```

```

        ResultSet rs = pst.executeQuery();

        if (rs.next()) {
            return rs.getString("role"); // Devuelve el rol
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}

private int loginUser(String username, String password) {
    try {
        // Conexión a la base de datos
        Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/agroleaf_db",
"root", "");

        String sql = "SELECT id, role FROM users WHERE username = ? AND password = ?";
        PreparedStatement pst = conn.prepareStatement(sql);
        pst.setString(1, username);
        pst.setString(2, password);
        ResultSet rs = pst.executeQuery();

        // Si las credenciales son correctas, devuelve el ID del usuario
        if (rs.next()) {
            int userId = rs.getInt("id"); // Obtén el ID del usuario
            SesionUsuario.iniciarSesion(userId); // Guarda el ID en la sesión
            return userId; // Devuelve el ID del usuario
        } else {
            return -1; // Usuario o contraseña incorrectos
        }
    }
}

```

```

    }
} catch (Exception e) {
    e.printStackTrace();
    return -1;
}
}
}

```

```

/**
 * 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() {

    jPanel1 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();
    jTextFieldUsername = new javax.swing.JTextField();
    jLabel2 = new javax.swing.JLabel();
    jPasswordField = new javax.swing.JPasswordField();
    jButtonLogin = new javax.swing.JButton();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    jSeparator1 = new javax.swing.JSeparator();
    jSeparator2 = new javax.swing.JSeparator();

```

```

jPanel2 = new javax.swing.JPanel();

jButtonClose = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setLocationByPlatform(true);
setUndecorated(true);
setPreferredSize(new java.awt.Dimension(907, 580));
setResizable(false);

jPanel1.setBackground(new java.awt.Color(255, 255, 255));
jPanel1.setPreferredSize(new java.awt.Dimension(907, 580));
jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel1.setFont(new java.awt.Font("Roboto Light", 1, 18)); // NOI18N
jLabel1.setText("Usuario");
jPanel1.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 320, -1, -1));

jTextFieldUsername.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jTextFieldUsername.setForeground(new java.awt.Color(204, 204, 204));
jTextFieldUsername.setText("Ingresa su nombre de usuario");
jTextFieldUsername.setBorder(null);
jTextFieldUsername.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mousePressed(java.awt.event.MouseEvent evt) {
        jTextFieldUsernameMousePressed(evt);
    }
});

jTextFieldUsername.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextFieldUsernameActionPerformed(evt);
    }
});

```

```

    }
});

jPanel1.add(jTextFieldUsername, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 340,
290, 40));

jLabel2.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N
jLabel2.setText("Contraseña");
jPanel1.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 410, -1, -1));

jPasswordField.setForeground(new java.awt.Color(204, 204, 204));
jPasswordField.setText("Ingresa contraseña");
jPasswordField.setBorder(null);
jPasswordField.setCursor(new java.awt.Cursor(java.awt.Cursor.TEXT_CURSOR));
jPasswordField.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mousePressed(java.awt.event.MouseEvent evt) {
        jPasswordFieldMousePressed(evt);
    }
});

jPasswordField.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jPasswordFieldActionPerformed(evt);
    }
});

jPanel1.add(jPasswordField, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 430, 240,
30));

jButtonLogin.setBackground(new java.awt.Color(204, 255, 204));
jButtonLogin.setFont(new java.awt.Font("Roboto", 1, 14)); // NOI18N
jButtonLogin.setText("Entrar");
jButtonLogin.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));

```

```

jButtonLogin.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonLoginActionPerformed(evt);
    }
});

jPanel1.add(jButtonLogin, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 500, 130,
40));

jLabel3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Plantio.PNG"))); //
NOI18N
jLabel3.setText("jLabel1");
jPanel1.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(520, 0, 390, 580));

jLabel4.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/2714_R1NUIENBUiAyNDQ3LTM2 (1).jpg"))); //
NOI18N
jLabel4.setText("jLabel3");
jPanel1.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 0, 250, 235));

jLabel5.setFont(new java.awt.Font("Segoe UI", 1, 48)); // NOI18N
jLabel5.setText("Iniciar sesión");
jPanel1.add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 230, 360, 70));

jSeparator1.setForeground(new java.awt.Color(0, 0, 0));
jPanel1.add(jSeparator1, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 380, 220,
10));

jSeparator2.setForeground(new java.awt.Color(0, 0, 0));
jPanel1.add(jSeparator2, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 460, 230,
20));

```

```
jPanel2.setOpaque(false);

jPanel2.addMouseMotionListener(new java.awt.event.MouseMotionAdapter() {

    public void mouseDragged(java.awt.event.MouseEvent evt) {

        jPanel2MouseDragged(evt);

    }

});

jPanel2.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mousePressed(java.awt.event.MouseEvent evt) {

        jPanel2MousePressed(evt);

    }

});
```

```
jButtonClose.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N

jButtonClose.setText("X");

jButtonClose.setBorder(null);

jButtonClose.setBorderPainted(false);

jButtonClose.setContentAreaFilled(false);

jButtonClose.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));

jButtonClose.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseClicked(java.awt.event.MouseEvent evt) {

        jButtonCloseMouseClicked(evt);

    }

    public void mouseEntered(java.awt.event.MouseEvent evt) {

        jButtonCloseMouseEntered(evt);

    }

    public void mousePressed(java.awt.event.MouseEvent evt) {

        jButtonCloseMousePressed(evt);

    }

});
```

```

jButtonClose.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonCloseActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addComponent(jButtonClose, javax.swing.GroupLayout.PREFERRED_SIZE, 53,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(0, 857, Short.MAX_VALUE))
        );
jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(0, 0, Short.MAX_VALUE)
            .addComponent(jButtonClose, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
                javax.swing.GroupLayout.PREFERRED_SIZE))
        );

jPanel1.add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 910, 40));

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addComponent(jPanel1, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 910, Short.MAX_VALUE)

    );

    layout.setVerticalGroup(

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(jPanel1, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

    );

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

```

```

private void jButtonLoginActionPerformed(java.awt.event.ActionEvent evt) {

```

```

    String username = jTextFieldUsername.getText();

```

```

    // Obtén el texto del campo de contraseña

```

```

    String password = new String(jPasswordField.getPassword());

```

```

    int userId = loginUser(username, password);

```

```

    if (userId != -1) {

```

```

        String[] consejos = {

```

```

            "Recuerda regar tus hortalizas temprano en la mañana para evitar la evaporación del agua.",

```

```

            "La rotación de cultivos ayuda a prevenir plagas y enfermedades en tus hortalizas.",

```

```

            "Agrega compost a tu suelo para mejorar la calidad y aumentar los nutrientes disponibles.",

```

```

            "Evita el riego excesivo, ya que puede provocar enfermedades en las raíces de las plantas.",

```

```

            "Plantar flores cerca de tus hortalizas puede atraer insectos polinizadores.",

```

```

            "El acolchado con paja o hojas secas ayuda a mantener la humedad del suelo.",

```

```

            "Usa fertilizantes orgánicos para mantener tus cultivos saludables y ecológicos.",

```

"La poda regular de hojas muertas mejora la circulación de aire y reduce plagas.",
"Las hortalizas de temporada son más resistentes y requieren menos cuidados específicos.",
"Recuerda inspeccionar tus plantas regularmente para detectar plagas o enfermedades a tiempo."

};

// Selecciona un mensaje aleatorio

int indiceAleatorio = (int) (Math.random() * consejos.length);

String consejoDelDia = consejos[indiceAleatorio];

// Muestra el consejo del día junto con el mensaje de inicio de sesión

JOptionPane.showMessageDialog(null, "\nConsejo del día: " + consejoDelDia, "Bienvenido",
JOptionPane.INFORMATION_MESSAGE);

// Verificar el rol y abrir la vista correspondiente

String role = obtenerRolUsuario(userId); // Crea el método obtenerRolUsuario si aún no existe

if (role.equals("administrador")) {

new Admin().setVisible(true); // Abre la vista de administrador

} else if (role.equals("cliente")) {

new Principal().setVisible(true); // Abre la vista de cliente

}

this.dispose(); // Cerrar la ventana de login

} else {

JOptionPane.showMessageDialog(null, "Usuario o contraseña incorrectos.");

}

}

```
private void jPasswordFieldActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
}
```

```
private void jTextFieldUsernameActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

```
private void jButtonCloseActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

```
private void jPanel2MousePressed(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    xMouse = evt.getX();  
    yMouse = evt.getY();  
  
}
```

```
private void jPanel2MouseDragged(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    int x = evt.getXOnScreen();  
    int y = evt.getYOnScreen();  
    this.setLocation(x - xMouse, y - yMouse);  
}
```

```
private void jButtonCloseMousePressed(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:
```

```
}
```

```
private void jButtonCloseMouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    System.exit(0);  
}
```

```
private void jButtonCloseMouseEntered(java.awt.event.MouseEvent evt) {  
    jButtonClose.setBackground(Color.red);  
}
```

```
private void jTextFieldUsernameMousePressed(java.awt.event.MouseEvent evt) {  
    if(jTextFieldUsername.getText().equals("Ingrese su nombre de usuario")){  
        jTextFieldUsername.setText("");  
        jTextFieldUsername.setForeground(Color.BLACK);  
    }
```

```
        jPasswordField.setText("*****");  
        jPasswordField.setForeground(Color.GRAY);  
    }
```

```
private void jPasswordFieldMousePressed(java.awt.event.MouseEvent evt) {  
  
    jPasswordField.setText("");  
    jPasswordField.setForeground(Color.BLACK);  
  
    if(jTextFieldUsername.getText().isEmpty()){  
        jTextFieldUsername.setText("Ingrese su nombre de usuario");  
    }
```

```

        jTextFieldUsername.setForeground(Color.BLACK);
    }
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

        java.util.logging.Logger.getLogger(LoginForm.class.getName()).log(java.util.logging.Level.SEVERE,
            null, ex);

    } catch (InstantiationException ex) {

        java.util.logging.Logger.getLogger(LoginForm.class.getName()).log(java.util.logging.Level.SEVERE,
            null, ex);

    } catch (IllegalAccessException ex) {

```

```
java.util.logging.Logger.getLogger(LoginForm.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
```

```
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
```

```
java.util.logging.Logger.getLogger(LoginForm.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
```

```
    }
```

```
//</editor-fold>
```

```
/* Create and display the form */
```

```
java.awt.EventQueue.invokeLater(new Runnable() {
```

```
    public void run() {
```

```
        new LoginForm().setVisible(true);
```

```
    }
```

```
});
```

```
}
```

```
// Variables declaration - do not modify
```

```
private javax.swing.JButton jButtonClose;
```

```
private javax.swing.JButton jButtonLogin;
```

```
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.JPanel jPanel1;
```

```
private javax.swing.JPanel jPanel2;
```

```
private javax.swing.JPasswordField jPasswordField;
```

```
private javax.swing.JSeparator jSeparator1;
```

```
private javax.swing.JSeparator jSeparator2;  
private javax.swing.JTextField jTextFieldUsername;  
// End of variables declaration  
}
```

Main

```
/*  
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this  
license  
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template  
 */
```

```
package Login;
```

```
import java.awt.Color;
```

```
/**  
 *  
 * @author IGNITER  
 */  
public class MainForm extends javax.swing.JFrame {  
  
    /**  
     * Creates new form MainForm
```

```

*/
int xMouse, yMouse;
public MainForm() {
    initComponents();
}

/**
 * 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() {

    jPanel1 = new javax.swing.JPanel();
    btnLogin = new javax.swing.JButton();
    btnRegister = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jPanel2 = new javax.swing.JPanel();
    jButtonClose = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setBackground(new java.awt.Color(51, 171, 0));
    setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
    setLocationByPlatform(true);
    setUndecorated(true);

```

```

setResizable(false);

jPanel1.setBackground(new java.awt.Color(255, 255, 255));
jPanel1.setForeground(new java.awt.Color(255, 255, 255));
jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

btnLogin.setFont(new java.awt.Font("Segoe UI", 0, 24)); // NOI18N
btnLogin.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icons8-entrar-24.png"))); // NOI18N
btnLogin.setText("Iniciar Sesion");
btnLogin.setContentAreaFilled(false);
btnLogin.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
btnLogin.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnLoginActionPerformed(evt);
    }
});
jPanel1.add(btnLogin, new org.netbeans.lib.awtextra.AbsoluteConstraints(110, 350, 260, 60));

btnRegister.setFont(new java.awt.Font("Segoe UI", 0, 14)); // NOI18N
btnRegister.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icons8-añadir-usuario-masculino-15.png"))); // NOI18N
btnRegister.setText("Registrarse");
btnRegister.setContentAreaFilled(false);
btnRegister.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
btnRegister.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnRegisterActionPerformed(evt);
    }
});

```

```
jPanel1.add(btnRegister, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 430, 160, 52));
```

```
jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Plantio.PNG"))); // NOI18N
```

```
jLabel1.setText("jLabel1");
```

```
jPanel1.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(520, 0, 390, 580));
```

```
jLabel3.setIcon(new  
javax.swing.ImageIcon(getClass().getResource("/2714_R1NUIENBUiAyNDQ3LTM2 (1).jpg"))); // NOI18N
```

```
jLabel3.setText("jLabel3");
```

```
jPanel1.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 0, 250, 235));
```

```
jLabel2.setFont(new java.awt.Font("Segoe UI", 1, 48)); // NOI18N
```

```
jLabel2.setText("AGROLEAF");
```

```
jPanel1.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(120, 230, 290, 70));
```

```
jPanel2.setOpaque(false);
```

```
jPanel2.addMouseListener(new java.awt.event.MouseAdapter() {
```

```
    public void mouseDragged(java.awt.event.MouseEvent evt) {
```

```
        jPanel2MouseDragged(evt);
```

```
    }
```

```
});
```

```
jPanel2.addMouseListener(new java.awt.event.MouseAdapter() {
```

```
    public void mousePressed(java.awt.event.MouseEvent evt) {
```

```
        jPanel2MousePressed(evt);
```

```
    }
```

```
});
```

```

jButtonClose.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N
jButtonClose.setText("X");
jButtonClose.setBorder(null);
jButtonClose.setBorderPainted(false);
jButtonClose.setContentAreaFilled(false);
jButtonClose.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
jButtonClose.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jButtonCloseMouseClicked(evt);
    }
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        jButtonCloseMouseEntered(evt);
    }
    public void mousePressed(java.awt.event.MouseEvent evt) {
        jButtonCloseMousePressed(evt);
    }
});
jButtonClose.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonCloseActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jButtonClose, javax.swing.GroupLayout.DEFAULT_SIZE, 80, true)
            .addContainerGap(100, true))
);

```

```

        .addComponent(jButtonClose, javax.swing.GroupLayout.PREFERRED_SIZE, 53,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(0, 857, Short.MAX_VALUE))

    );

    jPanel2Layout.setVerticalGroup(

        jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup())

        .addGap(0, 0, Short.MAX_VALUE)

        .addComponent(jButtonClose, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))

    );

    jPanel1.add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 910, 40));

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);

    layout.setHorizontalGroup(

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(jPanel1, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

    );

    layout.setVerticalGroup(

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(jPanel1, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

    );

    pack();

```

```
}// </editor-fold>
```

```
private void btnRegisterActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here: // Código para abrir el formulario de registro  
    RegisterForm registerForm = new RegisterForm();  
    registerForm.setVisible(true);  
    this.dispose(); // Cerrar la ventana principal  
}
```

```
private void btnLoginActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your han // Código para abrir el formulario de inicio de sesión  
    LoginForm loginForm = new LoginForm();  
    loginForm.setVisible(true);  
    this.dispose(); // Cerrar la ventana principal  
}
```

```
private void jButtonCloseMouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    System.exit(0);  
}
```

```
private void jButtonCloseMouseEntered(java.awt.event.MouseEvent evt) {  
    jButtonClose.setBackground(Color.red);  
}
```

```
private void jButtonCloseMousePressed(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
}
```

```
private void jButtonCloseActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

```
private void jPanel2MouseDragged(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    int x = evt.getXOnScreen();  
    int y = evt.getYOnScreen();  
    this.setLocation(x - xMouse, y - yMouse);  
}
```

```
private void jPanel2MousePressed(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    xMouse = evt.getX();  
    yMouse = evt.getY();  
  
}
```

```
/**  
 * @param args the command line arguments  
 */  
public static void main(String args[]) {  
    /* Set the Nimbus look and feel */  
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">  
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.  
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html  
     */  
    try {
```

```

        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(MainForm.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(MainForm.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(MainForm.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(MainForm.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

        }
    }
}
//</editor-fold>

```

```

/* Create and display the form */

```

```

java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new MainForm().setVisible(true);
    }
});

```

```
}
```

```
// Variables declaration - do not modify  
private javax.swing.JButton btnLogin;  
private javax.swing.JButton btnRegister;  
private javax.swing.JButton jButtonClose;  
private javax.swing.JLabel jLabel1;  
private javax.swing.JLabel jLabel2;  
private javax.swing.JLabel jLabel3;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JPanel jPanel2;  
// End of variables declaration  
}
```

Registro

```
/*
```

```
* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this  
license
```

```
* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
```

```
*/
```

```
package Login;
```

```
import java.awt.Color;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.PreparedStatement;
```

```
import javax.swing.JOptionPane;
```

```

/**
 *
 * @author IGNITER
 */
public class RegisterForm extends javax.swing.JFrame {

    /**
     * Creates new form RegisterForm
     */
    int xMouse, yMouse;
    public RegisterForm() {
        initComponents();
    }

    /**
     * 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() {

        jPanel1 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jTextFieldUsername = new javax.swing.JTextField();
        jLabel2 = new javax.swing.JLabel();

```

```
jPasswordField = new javax.swing.JPasswordField();

jLabel3 = new javax.swing.JLabel();

jTextFieldFullName = new javax.swing.JTextField();

jLabel4 = new javax.swing.JLabel();

jTextFieldEmail = new javax.swing.JTextField();

jButtonRegister = new javax.swing.JButton();

jLabel5 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

jLabel7 = new javax.swing.JLabel();

jSeparator1 = new javax.swing.JSeparator();

jSeparator2 = new javax.swing.JSeparator();

jSeparator3 = new javax.swing.JSeparator();

jSeparator4 = new javax.swing.JSeparator();

jPanel2 = new javax.swing.JPanel();

jButtonClose = new javax.swing.JButton();


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

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

setLocationByPlatform(true);

setUndecorated(true);

setPreferredSize(new java.awt.Dimension(900, 580));

setResizable(false);

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


jPanel1.setBackground(new java.awt.Color(255, 255, 255));

jPanel1.setMinimumSize(new java.awt.Dimension(907, 580));

jPanel1.setPreferredSize(new java.awt.Dimension(907, 580));

jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
```

```
jLabel1.setFont(new java.awt.Font("Roboto Light", 1, 18)); // NOI18N
jLabel1.setText("Usuario");
jPanel1.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 270, -1, -1));

jTextFieldUsername.setBorder(null);
jTextFieldUsername.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
jTextFieldUsername.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextFieldUsernameActionPerformed(evt);
    }
});

jPanel1.add(jTextFieldUsername, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 300,
270, 30));

jLabel2.setFont(new java.awt.Font("Roboto Light", 1, 18)); // NOI18N
jLabel2.setText("Contraseña");
jPanel1.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 340, -1, -1));

jPasswordField.setBorder(null);
jPasswordField.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
jPanel1.add(jPasswordField, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 360, 275,
28));

jLabel3.setFont(new java.awt.Font("Roboto Light", 1, 18)); // NOI18N
jLabel3.setText("Nombre Completo");
jPanel1.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 410, -1, -1));

jTextFieldFullName.setBorder(null);
jTextFieldFullName.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
jTextFieldFullName.addActionListener(new java.awt.event.ActionListener() {
```

```

        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jTextFieldFullNameActionPerformed(evt);
        }
    });

    jPanel1.add(jTextFieldFullName, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 430,
275, 30));

    jLabel4.setFont(new java.awt.Font("Roboto Light", 1, 18)); // NOI18N
    jLabel4.setText("Correo");
    jPanel1.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 470, -1, -1));

    jTextFieldEmail.setBorder(null);
    jTextFieldEmail.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
    jTextFieldEmail.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jTextFieldEmailActionPerformed(evt);
        }
    });

    jPanel1.add(jTextFieldEmail, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 490, 275,
30));

    jButtonRegister.setBackground(new java.awt.Color(204, 255, 204));
    jButtonRegister.setFont(new java.awt.Font("Roboto Light", 1, 14)); // NOI18N
    jButtonRegister.setText("Registrar");
    jButtonRegister.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
    jButtonRegister.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButtonRegisterActionPerformed(evt);
        }
    });

```

```

jPanel1.add(jButtonRegister, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 540,
110, 30));

jLabel5.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Plantio.PNG"))); //
NOI18N

jLabel5.setText("jLabel1");

jPanel1.add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(530, 0, 380, 580));

jLabel6.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/2714_R1NUIENBUiAyNDQ3LTM2 (1).jpg"))); //
NOI18N

jLabel6.setText("jLabel3");

jPanel1.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 10, 270, 220));

jLabel7.setFont(new java.awt.Font("Segoe UI", 1, 48)); // NOI18N
jLabel7.setText("AGROLEAF");

jPanel1.add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(110, 210, 290, 70));

jSeparator1.setBackground(new java.awt.Color(0, 0, 0));

jPanel1.add(jSeparator1, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 330, 270,
10));

jSeparator2.setBackground(new java.awt.Color(0, 0, 0));

jPanel1.add(jSeparator2, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 390, 270,
30));

jSeparator3.setBackground(new java.awt.Color(0, 0, 0));

jPanel1.add(jSeparator3, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 460, 270,
10));

jSeparator4.setBackground(new java.awt.Color(0, 0, 0));

```

```
jPanel1.add(jSeparator4, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 520, 270, 30));
```

```
jPanel2.setOpaque(false);
```

```
jPanel2.addMouseMotionListener(new java.awt.event.MouseMotionAdapter() {  
    public void mouseDragged(java.awt.event.MouseEvent evt) {  
        jPanel2MouseDragged(evt);  
    }  
});
```

```
jPanel2.addMouseListener(new java.awt.event.MouseAdapter() {  
    public void mousePressed(java.awt.event.MouseEvent evt) {  
        jPanel2MousePressed(evt);  
    }  
});
```

```
jButtonClose.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N
```

```
jButtonClose.setText("X");
```

```
jButtonClose.setBorder(null);
```

```
jButtonClose.setBorderPainted(false);
```

```
jButtonClose.setContentAreaFilled(false);
```

```
jButtonClose.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
```

```
jButtonClose.addMouseListener(new java.awt.event.MouseAdapter() {  
    public void mouseClicked(java.awt.event.MouseEvent evt) {  
        jButtonCloseMouseClicked(evt);  
    }  
}
```

```
    public void mouseEntered(java.awt.event.MouseEvent evt) {  
        jButtonCloseMouseEntered(evt);  
    }  
}
```

```
    public void mousePressed(java.awt.event.MouseEvent evt) {
```

```

        jButtonCloseMousePressed(evt);
    }
});

jButtonClose.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonCloseActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addComponent(jButtonClose, javax.swing.GroupLayout.PREFERRED_SIZE, 53,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(0, 857, Short.MAX_VALUE))
        );
jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(0, 0, Short.MAX_VALUE)
            .addComponent(jButtonClose, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
                javax.swing.GroupLayout.PREFERRED_SIZE))
        );

jPanel1.add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 910, 40));

```

```
getContentPane().add(jPanel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, -8, 910, 580));
```

```
pack();
```

```
}// </editor-fold>
```

```
private void jButtonRegisterActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    // TODO add yprivate void jButtonRegisterActionPerformed(java.awt.event.ActionEvent evt) {
```

```
String username = jTextFieldUsername.getText();
```

```
String password = new String(jPasswordField.getPassword());
```

```
String fullname = jTextFieldFullName.getText();
```

```
String email = jTextFieldEmail.getText();
```

```
registerUser(username, password, fullname, email);
```

```
}
```

```
private void registerUser(String username, String password, String fullname, String email) {
```

```
    try {
```

```
        Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/agroleaf_db",  
"root", "");
```

```
String sql = "INSERT INTO users (username, password, fullname, email) VALUES (?, ?, ?, ?)";
```

```
PreparedStatement pst = conn.prepareStatement(sql);
```

```
pst.setString(1, username);
```

```
pst.setString(2, password);
```

```
pst.setString(3, fullname);
```

```
pst.setString(4, email);
```

```
int rowAffected = pst.executeUpdate();
```

```
System.out.println("Fila afectada: " + rowAffected);
```

```
if (rowAffected > 0) {
```

```
    JOptionPane.showMessageDialog(this, "Usuario registrado exitosamente.");
```

```
    new MainForm().setVisible(true);
```

```
    this.dispose(); // Cerrar la ventana de registro
```

```

    } else {

        JOptionPane.showMessageDialog(this, "Error al registrar el usuario.");

    }

    conn.close();
} catch (Exception e) {
    e.printStackTrace();
    JOptionPane.showMessageDialog(this, "Error: " + e.getMessage());
}
}

private void jTextFieldFullNameActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jTextFieldEmailActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jTextFieldUsernameActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jButtonCloseMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    System.exit(0);
}

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

```

```

        jButtonClose.setBackground(Color.red);
    }

    private void jButtonCloseMousePressed(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
    }

    private void jButtonCloseActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jPanel2MouseDragged(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:

        int x = evt.getXOnScreen();
        int y = evt.getYOnScreen();
        this.setLocation(x - xMouse, y - yMouse);
    }

    private void jPanel2MousePressed(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:

        xMouse = evt.getX();
        yMouse = evt.getY();

    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {

```

```

/* Set the Nimbus look and feel */

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
   * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
   */
try {
    for (javax.swing.UIManager.LookAndFeelInfo info :
        javax.swing.UIManager.getInstalledLookAndFeels()) {
        if ("Nimbus".equals(info.getName())) {
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
            break;
        }
    }
} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(RegisterForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(RegisterForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(RegisterForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(RegisterForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    }
}
//</editor-fold>

```

```
/* Create and display the form */  
java.awt.EventQueue.invokeLater(new Runnable() {  
    public void run() {  
        new RegisterForm().setVisible(true);  
    }  
});  
}
```

```
// Variables declaration - do not modify  
private javax.swing.JButton jButtonClose;  
private javax.swing.JButton jButtonRegister;  
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.JLabel jLabel6;  
private javax.swing.JLabel jLabel7;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JPanel jPanel2;  
private javax.swing.JPasswordField jPasswordField;  
private javax.swing.JSeparator jSeparator1;  
private javax.swing.JSeparator jSeparator2;  
private javax.swing.JSeparator jSeparator3;  
private javax.swing.JSeparator jSeparator4;  
private javax.swing.JTextField jTextFieldEmail;  
private javax.swing.JTextField jTextFieldFullName;  
private javax.swing.JTextField jTextFieldUsername;  
// End of variables declaration
```

```
}
```

Paquete App

Dentro del paquete de App se encuentra la vista principal de cliente y la vista principal de administrador que se ejecutan después de que se inicia sesión correctamente.

Vista Admin

```
/*  
  
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this  
 license  
  
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template  
  
 */  
  
package App;  
  
import AdminViews.*;  
  
import AdminViews.VistaPerfilAdministrador;  
  
import java.awt.BorderLayout;  
  
import java.time.LocalDate;  
  
import java.time.format.DateTimeFormatter;  
  
import java.util.Locale;  
  
import javax.swing.JPanel;  
  
  
/**  
  
 *  
  
 * @author IGNITER  
  
 */
```

```

public class Admin extends javax.swing.JFrame {

    /**
     * Creates new form Principal
     */
    public Admin() {
        initComponents();
        SetDate();
        initContent();
    }

    private void SetDate(){

        LocalDate now = LocalDate.now();
        Locale spanishLocale = new Locale("es", "ES");
        Date.setText(now.format(DateTimeFormatter.ofPattern("'Hoy es' EEEE dd 'de' MMMM 'de'
yyyy'", spanishLocale)));
    }

    private void initContent() {
        ShowJPanel(new ApartadoAgregar());
    }

    private void ShowJPanel (JPanel p){
        p.setSize(830, 563);
        p.setLocation(0,0);

        content.removeAll();
        content.add(p, BorderLayout.CENTER);
    }
}

```

```

        content.revalidate();

        content.repaint();
    }

    /**
     * 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() {

        BackGround = new javax.swing.JPanel();
        Menu = new javax.swing.JPanel();
        jSeparator1 = new javax.swing.JSeparator();
        Agroleaf = new javax.swing.JLabel();
        ApartadoUsuarios = new javax.swing.JButton();
        PerfilAdmin = new javax.swing.JButton();
        Principal = new javax.swing.JButton();
        Header = new javax.swing.JPanel();
        Date = new javax.swing.JLabel();
        content = new javax.swing.JPanel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setResizable(false);

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

```

```
Menu.setBackground(new java.awt.Color(153, 153, 255));
```

```
Menu.setPreferredSize(new java.awt.Dimension(270, 640));
```

```
Agroleaf.setFont(new java.awt.Font("Segoe UI Black", 0, 24)); // NOI18N
```

```
Agroleaf.setText("Administrador");
```

```
ApartadoUsuarios.setBackground(new java.awt.Color(153, 153, 255));
```

```
ApartadoUsuarios.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N
```

```
ApartadoUsuarios.setForeground(new java.awt.Color(255, 255, 255));
```

```
ApartadoUsuarios.setIcon(new javax.swing.ImageIcon(getClass().getResource("/calendar-  
multiple-check.png"))); // NOI18N
```

```
ApartadoUsuarios.setText("Usuarios");
```

```
ApartadoUsuarios.setBorder(javax.swing.BorderFactory.createMatteBorder(1, 10, 1, 1, new  
java.awt.Color(0, 0, 0)));
```

```
ApartadoUsuarios.setBorderPainted(false);
```

```
ApartadoUsuarios.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
```

```
ApartadoUsuarios.setHideActionText(true);
```

```
ApartadoUsuarios.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
```

```
ApartadoUsuarios.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        ApartadoUsuariosActionPerformed(evt);
```

```
    }
```

```
});
```

```
PerfilAdmin.setBackground(new java.awt.Color(153, 153, 255));
```

```
PerfilAdmin.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N
```

```
PerfilAdmin.setForeground(new java.awt.Color(255, 255, 255));
```

```
PerfilAdmin.setIcon(new javax.swing.ImageIcon(getClass().getResource("/account-  
multiple.png"))); // NOI18N
```

```

    PerfilAdmin.setText("Perfil");

    PerfilAdmin.setBorder(javax.swing.BorderFactory.createMatteBorder(1, 10, 1, 1, new
java.awt.Color(0, 0, 0)));

    PerfilAdmin.setBorderPainted(false);

    PerfilAdmin.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));

    PerfilAdmin.setHideActionText(true);

    PerfilAdmin.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);

    PerfilAdmin.addActionListener(new java.awt.event.ActionListener() {

        public void actionPerformed(java.awt.event.ActionEvent evt) {

            PerfilAdminActionPerformed(evt);

        }

    });

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

    Principal.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N

    Principal.setForeground(new java.awt.Color(255, 255, 255));

    Principal.setIcon(new javax.swing.ImageIcon(getClass().getResource("/home-outline.png")));
// NOI18N

    Principal.setText("Principal");

    Principal.setBorder(javax.swing.BorderFactory.createMatteBorder(1, 10, 1, 1, new
java.awt.Color(0, 0, 0)));

    Principal.setBorderPainted(false);

    Principal.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));

    Principal.setHideActionText(true);

    Principal.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);

    Principal.addActionListener(new java.awt.event.ActionListener() {

        public void actionPerformed(java.awt.event.ActionEvent evt) {

            PrincipalActionPerformed(evt);

        }

    });

```

```

javax.swing.GroupLayout MenuLayout = new javax.swing.GroupLayout(Menu);
Menu.setLayout(MenuLayout);
MenuLayout.setHorizontalGroup(
    MenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(MenuLayout.createSequentialGroup()

            .addGroup(MenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(MenuLayout.createSequentialGroup()
                    .addGap(25, 25, 25)
                    .addComponent(Agroleaf, javax.swing.GroupLayout.PREFERRED_SIZE, 224,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(MenuLayout.createSequentialGroup()
                    .addContainerGap()

            .addGroup(MenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(MenuLayout.createSequentialGroup()
                    .addComponent(PerfilAdmin, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(ApartadoUsuarios, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 145,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())
            .addGroup(MenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(MenuLayout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(Principal, javax.swing.GroupLayout.PREFERRED_SIZE, 238,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addContainerGap(168, Short.MAX_VALUE)))
        );

```

```

MenuLayout.setVerticalGroup(
    MenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(MenuLayout.createSequentialGroup()
            .addGap(40, 40, 40)

.addGroup(MenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(Agroleaf, javax.swing.GroupLayout.PREFERRED_SIZE, 51,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(119, 119, 119)
    .addComponent(ApartadoUsuarios, javax.swing.GroupLayout.PREFERRED_SIZE, 58,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(PerfilAdmin, javax.swing.GroupLayout.PREFERRED_SIZE, 58,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
.addGroup(MenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(MenuLayout.createSequentialGroup()
        .addGap(149, 149, 149)
        .addComponent(Principal, javax.swing.GroupLayout.PREFERRED_SIZE, 58,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(492, Short.MAX_VALUE)))
);

```

```
Header.setBackground(new java.awt.Color(204, 204, 255));
```

```
Date.setFont(new java.awt.Font("Arial", 1, 36)); // NOI18N
```

```
Date.setForeground(new java.awt.Color(255, 255, 255));
```

```
javax.swing.GroupLayout HeaderLayout = new javax.swing.GroupLayout(Header);
```

```

Header.setLayout(HeaderLayout);

HeaderLayout.setHorizontalGroup(
    HeaderLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(HeaderLayout.createSequentialGroup()
            .addGap(42, 42, 42)
            .addComponent(Date, javax.swing.GroupLayout.PREFERRED_SIZE, 739,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );

HeaderLayout.setVerticalGroup(
    HeaderLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(HeaderLayout.createSequentialGroup()
            .addGap(30, 30, 30)
            .addComponent(Date, javax.swing.GroupLayout.PREFERRED_SIZE, 67,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(33, Short.MAX_VALUE))
        );

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

javax.swing.GroupLayout contentLayout = new javax.swing.GroupLayout(content);
content.setLayout(contentLayout);
contentLayout.setHorizontalGroup(
    contentLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 830, Short.MAX_VALUE)
    );

contentLayout.setVerticalGroup(
    contentLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 563, Short.MAX_VALUE)
    );

```

```

javax.swing.GroupLayout BackGroundLayout = new javax.swing.GroupLayout(BackGround);
BackGround.setLayout(BackGroundLayout);
BackGroundLayout.setHorizontalGroup(
    BackGroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(BackGroundLayout.createSequentialGroup()
            .addComponent(Menu, javax.swing.GroupLayout.PREFERRED_SIZE, 250,
                javax.swing.GroupLayout.PREFERRED_SIZE)

            .addGroup(BackGroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(Header, javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(content, javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
        );
BackGroundLayout.setVerticalGroup(
    BackGroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(Menu, javax.swing.GroupLayout.DEFAULT_SIZE, 699, Short.MAX_VALUE)
        .addGroup(BackGroundLayout.createSequentialGroup()
            .addComponent(Header, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(content, javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addComponent(BackGround, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

    );

    layout.setVerticalGroup(

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(BackGround, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

    );

    pack();

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

private void ApartadoUsuariosActionPerformed(java.awt.event.ActionEvent evt) {
ShowJPanel(new ApartadoUsuarios());
}

private void PerfilAdminActionPerformed(java.awt.event.ActionEvent evt) {
    ShowJPanel(new VistaPerfilAdministrador());
}

private void PrincipalActionPerformed(java.awt.event.ActionEvent evt) {
    ShowJPanel(new ApartadoAgregar());
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */

```

```

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new Admin().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JLabel Agroleaf;
private javax.swing.JButton ApartadoUsuarios;
private javax.swing.JPanel BackGround;
private javax.swing.JLabel Date;
private javax.swing.JPanel Header;
private javax.swing.JPanel Menu;
private javax.swing.JButton PerfilAdmin;
private javax.swing.JButton Principal;
private javax.swing.JPanel content;
private javax.swing.JSeparator jSeparator1;
// End of variables declaration
}

```

Vista Cliente

```

/*
* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
this license
* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this
template
*/

```

```

package App;

import Views.*;
import java.awt.BorderLayout;
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.util.Locale;
import javax.swing.JPanel;

/**
 *
 * @author IGNITER
 */
public class Principal extends javax.swing.JFrame {

    /**
     * Creates new form Principal
     */
    public Principal() {
        initComponents();
        SetDate();
        initContent();
    }

    private void SetDate(){

        LocalDate now = LocalDate.now();

        Locale spanishLocale = new Locale("es", "ES");

        Date.setText(now.format(DateTimeFormatter.ofPattern("'Hoy es' EEEE dd 'de'
MMMM 'de' yyyy", spanishLocale)));

```

```

    }

    private void initContent() {
        ShowJPanel(new VistaPrincipal());

    }

    private void ShowJPanel (JPanel p){
        p.setSize(830, 563);
        p.setLocation(0,0);

        content.removeAll();
        content.add(p, BorderLayout.CENTER);
        content.revalidate();
        content.repaint();
    }

    /**
     * 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() {

        BackGround = new javax.swing.JPanel();
        Menu = new javax.swing.JPanel();
        jSeparator1 = new javax.swing.JSeparator();

```

```

Agroleaf = new javax.swing.JLabel();
ButtonCalendario = new javax.swing.JButton();
ButtonPerfil = new javax.swing.JButton();
ButtonPrincipal = new javax.swing.JButton();
ButtonReportes = new javax.swing.JButton();
ButtonClima = new javax.swing.JButton();
Header = new javax.swing.JPanel();
Date = new javax.swing.JLabel();
content = new javax.swing.JPanel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setResizable(false);

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

Menu.setBackground(new java.awt.Color(153, 153, 255));
Menu.setPreferredSize(new java.awt.Dimension(270, 640));

Agroleaf.setFont(new java.awt.Font("Segoe UI Black", 1, 24)); // NOI18N
Agroleaf.setText("Agroleaf");

ButtonCalendario.setBackground(new java.awt.Color(153, 153, 255));
ButtonCalendario.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N
ButtonCalendario.setForeground(new java.awt.Color(255, 255, 255));
ButtonCalendario.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/calendar-multiple-check.png"))); //
NOI18N
ButtonCalendario.setText("Calendario");

```

```
    ButtonCalendario.setBorder(javax.swing.BorderFactory.createMatteBorder(1, 10, 1, 1,
new java.awt.Color(0, 0, 0)));

    ButtonCalendario.setBorderPainted(false);

    ButtonCalendario.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));

    ButtonCalendario.setHideActionText(true);

    ButtonCalendario.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);

    ButtonCalendario.addActionListener(new java.awt.event.ActionListener() {

        public void actionPerformed(java.awt.event.ActionEvent evt) {

            ButtonCalendarioActionPerformed(evt);

        }

    });
```

```
    ButtonPerfil.setBackground(new java.awt.Color(153, 153, 255));

    ButtonPerfil.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N

    ButtonPerfil.setForeground(new java.awt.Color(255, 255, 255));

    ButtonPerfil.setIcon(new javax.swing.ImageIcon(getClass().getResource("/account-
multiple.png"))); // NOI18N

    ButtonPerfil.setText("Perfil");

    ButtonPerfil.setBorder(javax.swing.BorderFactory.createMatteBorder(1, 10, 1, 1, new
java.awt.Color(0, 0, 0)));

    ButtonPerfil.setBorderPainted(false);

    ButtonPerfil.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));

    ButtonPerfil.setHideActionText(true);

    ButtonPerfil.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);

    ButtonPerfil.addActionListener(new java.awt.event.ActionListener() {

        public void actionPerformed(java.awt.event.ActionEvent evt) {

            ButtonPerfilActionPerformed(evt);

        }

    });
```

```
ButtonPrincipal.setBackground(new java.awt.Color(153, 153, 255));

ButtonPrincipal.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N

ButtonPrincipal.setForeground(new java.awt.Color(255, 255, 255));

ButtonPrincipal.setIcon(new javax.swing.ImageIcon(getClass().getResource("/home-
outline.png"))); // NOI18N

ButtonPrincipal.setText("Principal");

ButtonPrincipal.setBorder(javax.swing.BorderFactory.createMatteBorder(1, 10, 1, 1,
new java.awt.Color(0, 0, 0)));

ButtonPrincipal.setBorderPainted(false);

ButtonPrincipal.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));

ButtonPrincipal.setHideActionText(true);

ButtonPrincipal.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);

ButtonPrincipal.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        ButtonPrincipalActionPerformed(evt);

    }

});
```

```
ButtonReportes.setBackground(new java.awt.Color(153, 153, 255));

ButtonReportes.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N

ButtonReportes.setForeground(new java.awt.Color(255, 255, 255));

ButtonReportes.setIcon(new javax.swing.ImageIcon(getClass().getResource("/file-
chart.png"))); // NOI18N

ButtonReportes.setText("Reportes");

ButtonReportes.setBorder(javax.swing.BorderFactory.createMatteBorder(1, 10, 1, 1,
new java.awt.Color(0, 0, 0)));

ButtonReportes.setBorderPainted(false);

ButtonReportes.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));

ButtonReportes.setHideActionText(true);
```

```
ButtonReportes.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
```

```
ButtonReportes.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        ButtonReportesActionPerformed(evt);
```

```
    }
```

```
});
```

```
ButtonClima.setBackground(new java.awt.Color(153, 153, 255));
```

```
ButtonClima.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N
```

```
ButtonClima.setForeground(new java.awt.Color(255, 255, 255));
```

```
ButtonClima.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icons8-nube-24.png"))); // NOI18N
```

```
ButtonClima.setText("Clima");
```

```
ButtonClima.setBorder(javax.swing.BorderFactory.createMatteBorder(1, 10, 1, 1, new java.awt.Color(0, 0, 0)));
```

```
ButtonClima.setBorderPainted(false);
```

```
ButtonClima.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
```

```
ButtonClima.setHideActionText(true);
```

```
ButtonClima.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
```

```
ButtonClima.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        ButtonClimaActionPerformed(evt);
```

```
    }
```

```
});
```

```
javax.swing.GroupLayout MenuLayout = new javax.swing.GroupLayout(Menu);
```

```
Menu.setLayout(MenuLayout);
```

```
MenuLayout.setHorizontalGroup(
```

```
    MenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```

        .addGroup(MenuLayout.createSequentialGroup())

        .addGap(49, 49, 49)

        .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE,
145, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addContainerGap(56, Short.MAX_VALUE))

        .addGroup(MenuLayout.createSequentialGroup())

        .addContainerGap()

        .addGroup(MenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(ButtonCalendario,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

            .addComponent(ButtonPerfil, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

            .addComponent(ButtonReportes, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

            .addComponent(ButtonClima, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

            .addContainerGap())

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
MenuLayout.createSequentialGroup()

            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

            .addComponent(Agroleaf, javax.swing.GroupLayout.PREFERRED_SIZE, 159,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addGap(33, 33, 33))

        .addGroup(MenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(MenuLayout.createSequentialGroup()

                .addContainerGap()

```

```

        .addComponent(ButtonPrincipal, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addContainerGap()))

);

MenuLayout.setVerticalGroup(

MenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(MenuLayout.createSequentialGroup()

.addGap(48, 48, 48)

.addComponent(Agroleaf, javax.swing.GroupLayout.PREFERRED_SIZE, 51,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(119, 119, 119)

.addComponent(ButtonCalendario,
javax.swing.GroupLayout.PREFERRED_SIZE, 58,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(ButtonPerfil, javax.swing.GroupLayout.PREFERRED_SIZE,
58, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(ButtonReportes, javax.swing.GroupLayout.PREFERRED_SIZE,
58, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(ButtonClima, javax.swing.GroupLayout.PREFERRED_SIZE,
58, javax.swing.GroupLayout.PREFERRED_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))

.addGroup(MenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(MenuLayout.createSequentialGroup()

```

```

        .addGap(149, 149, 149)

        .addComponent(ButtonPrincipal,
javax.swing.GroupLayout.PREFERRED_SIZE, 58,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addContainerGap(473, Short.MAX_VALUE)))

);

Header.setBackground(new java.awt.Color(204, 204, 255));

Date.setFont(new java.awt.Font("Arial", 1, 36)); // NOI18N
Date.setForeground(new java.awt.Color(255, 255, 255));

javax.swing.GroupLayout HeaderLayout = new javax.swing.GroupLayout(Header);
Header.setLayout(HeaderLayout);
HeaderLayout.setHorizontalGroup(

HeaderLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(HeaderLayout.createSequentialGroup()
        .addGap(42, 42, 42)
        .addComponent(Date, javax.swing.GroupLayout.PREFERRED_SIZE, 747,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );
HeaderLayout.setVerticalGroup(

HeaderLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(HeaderLayout.createSequentialGroup()
        .addGap(30, 30, 30)
        .addComponent(Date, javax.swing.GroupLayout.PREFERRED_SIZE, 67,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addContainerGap(33, Short.MAX_VALUE))
    );

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

    javax.swing.GroupLayout contentLayout = new javax.swing.GroupLayout(content);
    content.setLayout(contentLayout);
    contentLayout.setHorizontalGroup(

contentLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 830, Short.MAX_VALUE)
    );
    contentLayout.setVerticalGroup(

contentLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 563, Short.MAX_VALUE)
    );

    javax.swing.GroupLayout BackGroundLayout = new
    javax.swing.GroupLayout(BackGround);
    BackGround.setLayout(BackGroundLayout);
    BackGroundLayout.setHorizontalGroup(

BackGroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(BackGroundLayout.createSequentialGroup()
            .addComponent(Menu, javax.swing.GroupLayout.PREFERRED_SIZE, 250,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGroup(BackGroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.
                LEADING)

```

```

        .addComponent(Header, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(content, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))

    );

    BackgroundLayout.setVerticalGroup(

BackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(Menu, javax.swing.GroupLayout.DEFAULT_SIZE, 699,
Short.MAX_VALUE)

        .addGroup(BackgroundLayout.createSequentialGroup())

        .addComponent(Header, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(content, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

    );

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);

    layout.setHorizontalGroup(

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(Background, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

    );

    layout.setVerticalGroup(

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(Background, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

    );

```

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

```
private void ButtonCalendarioActionPerformed(java.awt.event.ActionEvent evt) {  
    ShowJPanel(new CalendarioP());  
}
```

```
private void ButtonPerfilActionPerformed(java.awt.event.ActionEvent evt) {  
    ShowJPanel(new VistaPerfil());  
}
```

```
private void ButtonPrincipalActionPerformed(java.awt.event.ActionEvent evt) {  
    ShowJPanel(new VistaPrincipal());  
}
```

```
private void ButtonReportesActionPerformed(java.awt.event.ActionEvent evt) {  
    ShowJPanel(new VistaReportes());  
}
```

```
private void ButtonClimaActionPerformed(java.awt.event.ActionEvent evt) {  
    ShowJPanel(new VistaClima());  
}
```

```
/**  
 * @param args the command line arguments  
 */
```

```

public static void main(String args[]) {
    /* Set the Nimbus look and feel */

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Principal().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JLabel Agroleaf;
private javax.swing.JPanel BackGround;
private javax.swing.JButton ButtonCalendario;
private javax.swing.JButton ButtonClima;
private javax.swing.JButton ButtonPerfil;
private javax.swing.JButton ButtonPrincipal;
private javax.swing.JButton ButtonReportes;
private javax.swing.JLabel Date;
private javax.swing.JPanel Header;
private javax.swing.JPanel Menu;
private javax.swing.JPanel content;
private javax.swing.JSeparator jSeparator1;

// End of variables declaration
}

```

Paquete AdminViews

En el paquete admin views se encuentran los apartados que puede ver el administrador que son agregar hortalizas, el apartado de usuarios y el de perfil.

Vista Agregar Hortaliza Admin

```
/*
```

```
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change  
this license
```

```
 * Click nbfs://nbhost/SystemFileSystem/Templates/GuiForms/JPanel.java to edit this  
template
```

```
*/
```

```
package AdminViews;
```

```
import java.awt.Component;
```

```
import java.awt.Image;
```

```
import java.io.File;
```

```
import java.io.FileInputStream;
```

```
import java.io.IOException;
```

```
import javax.swing.ImageIcon;
```

```
import javax.swing.JFileChooser;
```

```
import javax.swing.filechooser.FileNameExtensionFilter;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.ResultSet;
```

```
import java.sql.SQLException;
```

```
import javax.swing.JOptionPane;
```

```
import javax.swing.JTable;
```

```
import javax.swing.table.DefaultTableCellRenderer;
```

```
import javax.swing.table.DefaultTableModel;
```

```

/**
 *
 * @author IGNITER
 */

public class ApartadoAgregar extends javax.swing.JPanel {

    private File imagenSeleccionada;

    /**
     * Creates new form ApartadoAgregar
     */
    public ApartadoAgregar() {
        initComponents();
        TablaHortalizas.setRowHeight(100);
    }

    /**
     * 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();
    }

```

```
NombreHortaliza = new javax.swing.JTextField();
ID = new javax.swing.JTextField();
FamiliaBotanica = new javax.swing.JTextField();
TemporadaSiembra = new javax.swing.JTextField();
jLabel5 = new javax.swing.JLabel();
TemperaturaOptima = new javax.swing.JTextField();
jLabel6 = new javax.swing.JLabel();
jLabel7 = new javax.swing.JLabel();
jLabel8 = new javax.swing.JLabel();
jLabel9 = new javax.swing.JLabel();
jLabel10 = new javax.swing.JLabel();
jLabel11 = new javax.swing.JLabel();
jLabel12 = new javax.swing.JLabel();
RequerimientosDeRiego = new javax.swing.JTextField();
TipoDeSuelo = new javax.swing.JTextField();
EnfermedadesComunes = new javax.swing.JTextField();
FrecuenciaDeFertilizacion = new javax.swing.JTextField();
EspacioDePlantacion = new javax.swing.JTextField();
MetodoDeCosecha = new javax.swing.JTextField();
EstadoORegion = new javax.swing.JTextField();
jLabel14 = new javax.swing.JLabel();
AdaptabilidadClimatica = new javax.swing.JTextField();
SeleccionarImagen = new javax.swing.JButton();
Lblimagen = new javax.swing.JLabel();
jScrollPane1 = new javax.swing.JScrollPane();
TablaHortalizas = new javax.swing.JTable();
JGuardar = new javax.swing.JButton();
JModificar = new javax.swing.JButton();
```

```
JEliminar = new javax.swing.JButton();
ActualizarButton = new javax.swing.JButton();

setBackground(new java.awt.Color(255, 255, 255));
setPreferredSize(new java.awt.Dimension(830, 563));
setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel1.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
jLabel1.setText("Nombre de la hortaliza");
add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 10, 160, 31));

jLabel2.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
jLabel2.setText("ID");
add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(80, 60, 37, 32));

jLabel3.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
jLabel3.setText("Familia botanica");
add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(37, 117, 121, -1));

jLabel4.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
jLabel4.setText("Temporada de siembra");
add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 170, -1, -1));

NombreHortaliza.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        NombreHortalizaActionPerformed(evt);
    }
});
```

```
add(NombreHortaliza, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 40, 149,
-1));
add(ID, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 90, 150, -1));
add(FamiliaBotanica, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 140,
150, -1));
add(TemporadaSiembra, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 190,
150, -1));
```

```
jLabel5.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
jLabel5.setText("Temperatura optima");
add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 220, -1, -1));
add(TemperaturaOptima, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 240,
150, -1));
```

```
jLabel6.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
jLabel6.setText("Requerimientos de riego");
add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(190, 20, -1, 20));
```

```
jLabel7.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
jLabel7.setText("Tipo de suelo");
add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(230, 70, -1, -1));
```

```
jLabel8.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
jLabel8.setText("Enfermedades comunes");
add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(190, 120, -1, -1));
```

```
jLabel9.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
jLabel9.setText("Frecuencia de fertilizacion");
add(jLabel9, new org.netbeans.lib.awtextra.AbsoluteConstraints(180, 170, -1, 20));
```

```

jLabel10.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
jLabel10.setText("Espacio de plantacion");
add(jLabel10, new org.netbeans.lib.awtextra.AbsoluteConstraints(200, 220, -1, -1));

jLabel11.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
jLabel11.setText("Metodo de cosecha");
add(jLabel11, new org.netbeans.lib.awtextra.AbsoluteConstraints(380, 20, -1, -1));

jLabel12.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
jLabel12.setText("Estado o region");
add(jLabel12, new org.netbeans.lib.awtextra.AbsoluteConstraints(390, 70, -1, -1));
add(RequerimientosDeRiego, new org.netbeans.lib.awtextra.AbsoluteConstraints(190,
40, 152, -1));
add(TipoDeSuelo, new org.netbeans.lib.awtextra.AbsoluteConstraints(190, 90, 152, -
1));
add(EnfermedadesComunes, new org.netbeans.lib.awtextra.AbsoluteConstraints(190,
140, 150, -1));
add(FrecuenciaDeFertilizacion, new
org.netbeans.lib.awtextra.AbsoluteConstraints(190, 190, 150, -1));
add(EspacioDePlantacion, new org.netbeans.lib.awtextra.AbsoluteConstraints(190,
240, 150, -1));
add(MetodoDeCosecha, new org.netbeans.lib.awtextra.AbsoluteConstraints(370, 40,
150, -1));
add(EstadoORegion, new org.netbeans.lib.awtextra.AbsoluteConstraints(370, 90, 150,
-1));

jLabel14.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
jLabel14.setText("Adaptabilidad climatica");
add(jLabel14, new org.netbeans.lib.awtextra.AbsoluteConstraints(370, 120, -1, 20));

AdaptabilidadClimatica.addActionListener(new java.awt.event.ActionListener() {

```

```

        public void actionPerformed(java.awt.event.ActionEvent evt) {
            AdaptabilidadClimaticaActionPerformed(evt);
        }
    });

    add(AdaptabilidadClimatica, new org.netbeans.lib.awtextra.AbsoluteConstraints(370,
140, 150, -1));

```

```

    SeleccionarImagen.setFont(new java.awt.Font("Arial", 1, 14)); // NOI18N
    SeleccionarImagen.setText("Seleccionar imagen");
    SeleccionarImagen.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            SeleccionarImagenActionPerformed(evt);
        }
    });

    add(SeleccionarImagen, new org.netbeans.lib.awtextra.AbsoluteConstraints(570, 20, -
1, 20));

    add(Lblimagen, new org.netbeans.lib.awtextra.AbsoluteConstraints(560, 50, 200,
180));

```

```

    TablaHortalizas.setFont(new java.awt.Font("Segoe UI", 0, 24)); // NOI18N
    TablaHortalizas.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {
            {},
            {},
            {},
            {}
        },
        new String [] {

```

```
    }  
    ));
```

```
TablaHortalizas.setAutoResizeMode(javax.swing.JTable.AUTO_RESIZE_NEXT_COLUMN);
```

```
    TablaHortalizas.addMouseListener(new java.awt.event.MouseAdapter() {  
        public void mouseClicked(java.awt.event.MouseEvent evt) {  
            TablaHortalizasMouseClicked(evt);  
        }  
    });  
jScrollPane1.setViewportViewView(TablaHortalizas);
```

```
add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 310, 810, 249));
```

```
JGuardar.setFont(new java.awt.Font("Arial", 1, 14)); // NOI18N  
JGuardar.setText("Guardar");  
JGuardar.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        JGuardarActionPerformed(evt);  
    }  
});  
add(JGuardar, new org.netbeans.lib.awtextra.AbsoluteConstraints(470, 270, -1, -1));
```

```
JModificar.setFont(new java.awt.Font("Arial", 1, 14)); // NOI18N  
JModificar.setText("Modificar");  
JModificar.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        JModificarActionPerformed(evt);  
    }  
});
```

```

    }
});
add(JModificar, new org.netbeans.lib.awtextra.AbsoluteConstraints(580, 270, -1, -1));

JEliminar.setFont(new java.awt.Font("Arial", 1, 14)); // NOI18N
JEliminar.setText("Eliminar");
JEliminar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        JEliminarActionPerformed(evt);
    }
});
add(JEliminar, new org.netbeans.lib.awtextra.AbsoluteConstraints(700, 270, -1, -1));

ActualizarButton.setFont(new java.awt.Font("Arial", 1, 14)); // NOI18N
ActualizarButton.setText("Actualizar");
ActualizarButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ActualizarButtonActionPerformed(evt);
    }
});
add(ActualizarButton, new org.netbeans.lib.awtextra.AbsoluteConstraints(350, 270,
100, -1));
} // </editor-fold>

public class ImageRenderer extends DefaultTableCellRenderer {
    @Override
    public Component getTableCellRendererComponent(JTable table, Object value, boolean
isSelected, boolean hasFocus, int row, int column) {
        if (value instanceof ImageIcon) {
            setIcon((ImageIcon) value);

```

```

        setText(null); // Elimina cualquier texto en la celda
    } else {
        setIcon(null);
        setText(value != null ? value.toString() : ""); // Muestra el texto si no es una imagen
    }
    return this;
}
}

private void cargarDatosDesdeBD(int id) {
    try {
        // Establece la conexión con la base de datos
        Connection conn =
        DriverManager.getConnection("jdbc:mysql://localhost:3306/agroleaf_db", "root", "");

        // Consulta SQL para obtener todos los datos del registro con el ID especificado
        String sql = "SELECT * FROM hortalizas WHERE id = ?";
        PreparedStatement pst = conn.prepareStatement(sql);
        pst.setInt(1, id);
        ResultSet rs = pst.executeQuery();

        // Si se encuentra el registro, llena los campos
        if (rs.next()) {
            NombreHortaliza.setText(rs.getString("nombre"));
            FamiliaBotanica.setText(rs.getString("familia_botanica"));
            TemporadaSiembra.setText(rs.getString("temporada_siembra"));
            TemperaturaOptima.setText(rs.getString("temperatura_optima"));
            RequerimientosDeRiego.setText(rs.getString("requerimientos_riego"));
            TipoDeSuelo.setText(rs.getString("tipo_suelo"));
            EnfermedadesComunes.setText(rs.getString("enfermedades_comunes"));
        }
    }
}

```

```

FrecuenciaDeFertilizacion.setText(rs.getString("frecuencia_fertilizacion"));
EspacioDePlantacion.setText(rs.getString("espacio_plantacion"));
MetodoDeCosecha.setText(rs.getString("metodo_cosecha"));
EstadoORegion.setText(rs.getString("estado_region"));
AdaptabilidadClimatica.setText(rs.getString("adaptabilidad_climatica"));

// Carga la imagen desde la base de datos y la muestra en el JLabel
byte[] imgBytes = rs.getBytes("imagen");
if (imgBytes != null) {
    ImageIcon icon = new ImageIcon(new
    ImageIcon(imgBytes).getImage().getScaledInstance(Lblimagen.getWidth(),
    Lblimagen.getHeight(), Image.SCALE_SMOOTH));
    Lblimagen.setIcon(icon);
} else {
    Lblimagen.setIcon(null); // Limpia el JLabel si no hay imagen
}
}

// Cierra la conexión
rs.close();
pst.close();
conn.close();

} catch (SQLException e) {
    e.printStackTrace();
    System.out.println("Error al cargar datos desde la base de datos: " + e.getMessage());
}
}

```

```

public void cargarDatosEnTabla() {
    // Modelo de la tabla con columnas específicas

    DefaultTableModel model = new DefaultTableModel(new String[]{"ID", "Nombre",
"Familia Botánica", "Estado/Región", "Imagen"}, 0);

    TablaHortalizas.setModel(model); // Aplica el modelo a TablaHortalizas

    try {
        // Establece la conexión con la base de datos

        Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/agroleaf_db", "root", "");

        // Consulta SQL para obtener los datos deseados

        String sql = "SELECT id, nombre, familia_botanica, estado_region, imagen FROM
hortalizas";

        PreparedStatement pst = conn.prepareStatement(sql);

        ResultSet rs = pst.executeQuery();

        // Itera sobre el ResultSet y agrega cada fila al modelo de la tabla
        while (rs.next()) {
            int id = rs.getInt("id");

            String nombre = rs.getString("nombre");

            String familiaBotanica = rs.getString("familia_botanica");

            String estadoRegion = rs.getString("estado_region");

            // Convierte la imagen desde la base de datos a ImageIcon

            byte[] imgBytes = rs.getBytes("imagen");

            ImageIcon icon = null;

            if (imgBytes != null) {

                Image img = new ImageIcon(imgBytes).getImage().getScaledInstance(100, 100,
Image.SCALE_SMOOTH);

```

```

        icon = new ImageIcon(img);
    }

    // Agrega la fila al modelo de la tabla
    model.addRow(new Object[]{id, nombre, familiaBotanica, estadoRegion, icon});
}

// Cierra la conexión
rs.close();
pst.close();
conn.close();

} catch (SQLException e) {
    e.printStackTrace();
    System.out.println("Error al cargar datos en la tabla: " + e.getMessage());
}

// Configura el renderizador de la columna de imagen
TablaHortalizas.getColumnModel().getColumn(4).setCellRenderer(new
ImageRenderer());
}

private void seleccionarImagen() {
    JFileChooser fileChooser = new JFileChooser();

    fileChooser.setFileFilter(new FileNameExtensionFilter("Imágenes", "jpg", "jpeg",
"png", "gif"));

    int resultado = fileChooser.showOpenDialog(this);

    if (resultado == JFileChooser.APPROVE_OPTION) {
        imagenSeleccionada = fileChooser.getSelectedFile();
    }
}

```

```

        // Muestra la imagen seleccionada en el JLabel Lblimagen

        ImageIcon icon = new ImageIcon(new
        ImageIcon(imagenSeleccionada.getAbsolutePath())

                .getImage().getScaledInstance(Lblimagen.getWidth(), Lblimagen.getHeight(),
        Image.SCALE_SMOOTH));

        Lblimagen.setIcon(icon);

    }

}

public void guardarHortaliza() {

    // Verifica si los campos requeridos están completos

    if (!validarCampos()) {

        return; // Si falta algún campo, detiene el proceso de guardado

    }

    try {

        // Establece la conexión con la base de datos

        Connection conn =
        DriverManager.getConnection("jdbc:mysql://localhost:3306/agroleaf_db", "root", "");

        // Define la consulta SQL para insertar una nueva hortaliza

        String sql = "INSERT INTO hortalizas (nombre, familia_botanica,
        temporada_siembra, "
                + "temperatura_optima, requerimientos_riego, tipo_suelo, "
                + "enfermedades_comunes, frecuencia_fertilizacion, espacio_plantacion, "
                + "metodo_cosecha, estado_region, adaptabilidad_climatica, imagen) "
                + "VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";

        // Prepara la consulta

        PreparedStatement pst = conn.prepareStatement(sql);

```

```
// Asigna valores desde los campos de texto
pst.setString(1, NombreHortaliza.getText());
pst.setString(2, FamiliaBotanica.getText());
pst.setString(3, TemporadaSiembra.getText());
pst.setString(4, TemperaturaOptima.getText());
pst.setString(5, RequerimientosDeRiego.getText());
pst.setString(6, TipoDeSuelo.getText());
pst.setString(7, EnfermedadesComunes.getText());
pst.setString(8, FrecuenciaDeFertilizacion.getText());
pst.setString(9, EspacioDePlantacion.getText());
pst.setString(10, MetodoDeCosecha.getText());
pst.setString(11, EstadoORegion.getText());
pst.setString(12, AdaptabilidadClimatica.getText());

// Carga la imagen si hay una seleccionada
if (imagenSeleccionada != null) {
    FileInputStream fis = new FileInputStream(imagenSeleccionada);
    pst.setBinaryStream(13, fis, (int) imagenSeleccionada.length());
}

// Ejecuta la consulta y obtiene el número de filas afectadas
int rowAffected = pst.executeUpdate();
System.out.println("Fila afectada: " + rowAffected);

// Cierra la conexión y la consulta
pst.close();
conn.close();
```

```

        if (rowAffected > 0) {
            System.out.println("Hortaliza guardada correctamente.");
        } else {
            System.out.println("No se guardaron los datos.");
        }

    } catch (SQLException | IOException e) {
        System.out.println("Error al guardar en la base de datos: " + e.getMessage());
        e.printStackTrace();
    }
}

public void modificarHortaliza(int id) {
    try {
        Connection conn =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/agroleaf_db", "root", "");

        String sql;
        PreparedStatement pst;

        if (imagenSeleccionada != null) {
            // Si hay una imagen nueva seleccionada, incluye el campo de imagen en la
            // actualización

            sql = "UPDATE hortalizas SET nombre = ?, familia_botanica = ?,
temporada_siembra = ?, "
                + "temperatura_optima = ?, requerimientos_riego = ?, tipo_suelo = ?, "
                + "enfermedades_comunes = ?, frecuencia_fertilizacion = ?, espacio_plantacion =
?, "
                + "metodo_cosecha = ?, estado_region = ?, adaptabilidad_climatica = ?, imagen
= ? "
                + "WHERE id = ?";

```

```

pst = conn.prepareStatement(sql);

// Asigna valores desde los campos de texto
pst.setString(1, NombreHortaliza.getText());
pst.setString(2, FamiliaBotanica.getText());
pst.setString(3, TemporadaSiembra.getText());
pst.setString(4, TemperaturaOptima.getText());
pst.setString(5, RequerimientosDeRiego.getText());
pst.setString(6, TipoDeSuelo.getText());
pst.setString(7, EnfermedadesComunes.getText());
pst.setString(8, FrecuenciaDeFertilizacion.getText());
pst.setString(9, EspacioDePlantacion.getText());
pst.setString(10, MetodoDeCosecha.getText());
pst.setString(11, EstadoORegion.getText());
pst.setString(12, AdaptabilidadClimatica.getText());

// Agrega la imagen al parámetro 13
FileInputStream fis = new FileInputStream(imagenSeleccionada);
pst.setBinaryStream(13, fis, (int) imagenSeleccionada.length());

// El ID es el parámetro 14
pst.setInt(14, id);
} else {
    // Si no hay una imagen nueva seleccionada, omite el campo de imagen
    sql = "UPDATE hortalizas SET nombre = ?, familia_botanica = ?,
temporada_siembra = ?, "
        + "temperatura_optima = ?, requerimientos_riego = ?, tipo_suelo = ?, "
        + "enfermedades_comunes = ?, frecuencia_fertilizacion = ?, espacio_plantacion =
?, "

```

```

        + "metodo_cosecha = ?, estado_region = ?, adaptabilidad_climatica = ? "
        + "WHERE id = ?";

pst = conn.prepareStatement(sql);

// Asigna valores desde los campos de texto
pst.setString(1, NombreHortaliza.getText());
pst.setString(2, FamiliaBotanica.getText());
pst.setString(3, TemporadaSiembra.getText());
pst.setString(4, TemperaturaOptima.getText());
pst.setString(5, RequerimientosDeRiego.getText());
pst.setString(6, TipoDeSuelo.getText());
pst.setString(7, EnfermedadesComunes.getText());
pst.setString(8, FrecuenciaDeFertilizacion.getText());
pst.setString(9, EspacioDePlantacion.getText());
pst.setString(10, MetodoDeCosecha.getText());
pst.setString(11, EstadoORegion.getText());
pst.setString(12, AdaptabilidadClimatica.getText());

// El ID es el parámetro 13 cuando no hay imagen
pst.setInt(13, id);
}

// Ejecuta la consulta
int rowAffected = pst.executeUpdate();
System.out.println("Fila afectada: " + rowAffected);

// Cierra la conexión
pst.close();

```

```
conn.close();

if (rowAffected > 0) {
    System.out.println("Hortaliza modificada correctamente.");
} else {
    System.out.println("No se modificaron los datos.");
}

} catch (SQLException | IOException e) {
    System.out.println("Error al modificar en la base de datos: " + e.getMessage());
    e.printStackTrace();
}
}
```

```
public void eliminarHortaliza(int id) {
    try {
        // Establece la conexión con la base de datos
        Connection conn =
        DriverManager.getConnection("jdbc:mysql://localhost:3306/agroleaf_db", "root", "");

        // Consulta SQL para eliminar la hortaliza
        String sql = "DELETE FROM hortalizas WHERE id = ?";
        PreparedStatement pst = conn.prepareStatement(sql);
        pst.setInt(1, id); // Asigna el ID al parámetro

        // Ejecuta la consulta
        int rowAffected = pst.executeUpdate();
    }
}
```

```

        System.out.println("Filas afectadas: " + rowAffected);

        // Cierra la conexión
        pst.close();
        conn.close();

        // Muestra un mensaje de éxito si la operación fue exitosa
        if (rowAffected > 0) {
            System.out.println("Hortaliza eliminada correctamente.");
        } else {
            System.out.println("No se encontró el registro para eliminar.");
        }

    } catch (SQLException e) {
        System.out.println("Error al eliminar de la base de datos: " + e.getMessage());
        e.printStackTrace();
    }
}

private boolean validarCampos() {
    if (NombreHortaliza.getText().trim().isEmpty()) {
        JOptionPane.showMessageDialog(null, "El campo 'Nombre de la Hortaliza' es obligatorio.", "Error", JOptionPane.ERROR_MESSAGE);
        return false;
    }

    if (FamiliaBotanica.getText().trim().isEmpty()) {
        JOptionPane.showMessageDialog(null, "El campo 'Familia Botánica' es obligatorio.", "Error", JOptionPane.ERROR_MESSAGE);
        return false;
    }
}

```

```

if (EstadoORegion.getText().trim().isEmpty()) {
    JOptionPane.showMessageDialog(null, "El campo 'Estado/Región' es obligatorio.",
    "Error", JOptionPane.ERROR_MESSAGE);
    return false;
}
if (Lblimagen.getIcon() == null) {
    JOptionPane.showMessageDialog(null, "Debe seleccionar una imagen para la
    hortaliza.", "Error", JOptionPane.ERROR_MESSAGE);
    return false;
}
return true;
}

```

```

private void SeleccionarImagenActionPerformed(java.awt.event.ActionEvent evt) {
    seleccionarImagen();
}

```

```

private void TablaHortalizasMouseClicked(java.awt.event.MouseEvent evt) {
    int selectedRow = TablaHortalizas.getSelectedRow(); // Obtiene la fila seleccionada
    // Verifica que haya una fila seleccionada
    if (selectedRow != -1) {
        int id = (int) TablaHortalizas.getValueAt(selectedRow, 0); // Obtén el ID de la fila
        seleccionada
        cargarDatosDesdeBD(id); // Llama al método para cargar todos los datos desde la BD
    }
}

```

```
}
```

```
private void NombreHortalizaActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

```
private void AdaptabilidadClimaticaActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

```
private void JGuardarActionPerformed(java.awt.event.ActionEvent evt) {  
guardarHortaliza();  
cargarDatosEnTabla();  
}
```

```
private void JModificarActionPerformed(java.awt.event.ActionEvent evt) {  
int id = Integer.parseInt(ID.getText()); // Obtén el ID desde el campo de texto  
modificarHortaliza(id); // Llama al método de modificación usando el ID actual  
cargarDatosEnTabla(); // Refresca la tabla después de modificar  
}
```

```
private void JEliminarActionPerformed(java.awt.event.ActionEvent evt) {  
// Verifica que hay una fila seleccionada  
int selectedRow = TablaHortalizas.getSelectedRow();  
if (selectedRow != -1) {  
    int id = (int) TablaHortalizas.getValueAt(selectedRow, 0); // Obtén el ID de la  
    hortaliza seleccionada
```

```
    // Confirmación de eliminación
```

```

        int confirm = JOptionPane.showConfirmDialog(null, "¿Estás seguro de que deseas
eliminar esta hortaliza?", "Confirmar eliminación", JOptionPane.YES_NO_OPTION);

        if (confirm == JOptionPane.YES_OPTION) {

            eliminarHortaliza(id); // Llama al método de eliminación

            cargarDatosEnTabla(); // Refresca la tabla después de eliminar

        }
    } else {

        JOptionPane.showMessageDialog(null, "Selecciona una hortaliza para eliminar.");

    }

}

```

```

private void ActualizarButtonActionPerformed(java.awt.event.ActionEvent evt) {

    // Llama al método para cargar los datos en la tabla

    cargarDatosEnTabla();

}

```

```

// Variables declaration - do not modify

private javax.swing.JButton ActualizarButton;

private javax.swing.JTextField AdaptabilidadClimatica;

private javax.swing.JTextField EnfermedadesComunes;

private javax.swing.JTextField EspacioDePlantacion;

private javax.swing.JTextField EstadoORegion;

private javax.swing.JTextField FamiliaBotanica;

private javax.swing.JTextField FrecuenciaDeFertilizacion;

private javax.swing.JTextField ID;

private javax.swing.JButton JEliminar;

private javax.swing.JButton JGuardar;

private javax.swing.JButton JModificar;

```

```
private javax.swing.JLabel Lblimagen;
private javax.swing.JTextField MetodoDeCosecha;
private javax.swing.JTextField NombreHortaliza;
private javax.swing.JTextField RequerimientosDeRiego;
private javax.swing.JButton SeleccionarImagen;
private javax.swing.JTable TablaHortalizas;
private javax.swing.JTextField TemperaturaOptima;
private javax.swing.JTextField TemporadaSiembra;
private javax.swing.JTextField TipoDeSuelo;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel14;
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.JScrollPane jScrollPane1;
// End of variables declaration
}
```

Vista Usuarios Admin

/*

* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

* Click nbfs://nbhost/SystemFileSystem/Templates/GuiForms/JPanel.java to edit this template

*/

package AdminViews;

import Views.*;

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.DefaultCellEditor;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableColumn;

/**
 *
 * @author IGNITER
 */
public class ApartadoUsuarios extends javax.swing.JPanel {

    /**
     * Creates new form VistaPrincipal
     */
    public ApartadoUsuarios() {
        initComponents();
        cargarUsuarios(); // Cargar los usuarios al iniciar
    }

    private void cargarUsuarios() {
        DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
        model.setRowCount(0); // Limpiar la tabla antes de cargar nuevos datos

        try (Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/agroleaf_db", "root", "")) {

```

```

String sql = "SELECT id, username, fullname, email, role FROM users";

PreparedStatement pst = conn.prepareStatement(sql);

ResultSet rs = pst.executeQuery();

while (rs.next()) {
    int id = rs.getInt("id");

    String username = rs.getString("username");

    String fullname = rs.getString("fullname");

    String email = rs.getString("email");

    String role = rs.getString("role");

    model.addRow(new Object[]{id, username, fullname, email, role});
}
} catch (Exception e) {
    e.printStackTrace();

    JOptionPane.showMessageDialog(this, "Error al cargar los usuarios: " +
e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
}
}

/**
 * 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() {

```

```

jScrollPane1 = new javax.swing.JScrollPane();
jTable1 = new javax.swing.JTable();
actualizarButton = new javax.swing.JButton();
EliminarButton = new javax.swing.JButton();

setBackground(new java.awt.Color(204, 255, 204));
setPreferredSize(new java.awt.Dimension(830, 563));

jTable1.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
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 [] {
        "ID", "Usuario", "Nombre", "Email"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});

```

```
jScrollPane1.setViewportView(jTable1);
```

```
actualizarButton.setFont(new java.awt.Font("Arial", 1, 18)); // NOI18N
```

```
actualizarButton.setText("Actualizar");
```

```
actualizarButton.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        actualizarButtonActionPerformed(evt);
```

```
    }
```

```
});
```

```
EliminarButton.setFont(new java.awt.Font("Arial", 1, 18)); // NOI18N
```

```
EliminarButton.setText("Eliminar");
```

```
EliminarButton.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        EliminarButtonActionPerformed(evt);
```

```
    }
```

```
});
```

```
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);
```

```
this.setLayout(layout);
```

```
layout.setHorizontalGroup(
```

```
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 830,  
Short.MAX_VALUE)
```

```
        .addGroup(layout.createSequentialGroup()
```

```
            .addGap(28, 28, 28)
```

```
            .addComponent(actualizarButton)
```

```
            .addGap(18, 18, 18)
```

```
            .addComponent(EliminarButton)
```

```

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addGap(0, 103, Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(actualizarButton)
            .addComponent(EliminarButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
413, javax.swing.GroupLayout.PREFERRED_SIZE))
    );
} // </editor-fold>

```

```

private void actualizarButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```

```

private void EliminarButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // Verifica si hay una fila seleccionada
    int selectedRow = jTable1.getSelectedRow();
    if (selectedRow == -1) {
        JOptionPane.showMessageDialog(this, "Por favor, selecciona un usuario para
eliminar.", "Advertencia", JOptionPane.WARNING_MESSAGE);
        return;
    }
}

```

```

    }

    // Obtén el ID del usuario seleccionado
    int id = (int) jTable1.getValueAt(selectedRow, 0);

    // Confirmación de eliminación

    int confirm = JOptionPane.showConfirmDialog(this, "¿Estás seguro de que deseas
eliminar este usuario?", "Confirmar eliminación", JOptionPane.YES_NO_OPTION);

    if (confirm != JOptionPane.YES_OPTION) {
        return; // Si el usuario cancela, no se hace nada
    }

    // Elimina al usuario de la base de datos

    try (Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/agroleaf_db", "root", "")) {

        String sql = "DELETE FROM users WHERE id = ?";
        PreparedStatement pst = conn.prepareStatement(sql);
        pst.setInt(1, id);

        int filasAfectadas = pst.executeUpdate();

        if (filasAfectadas > 0) {
            JOptionPane.showMessageDialog(this, "Usuario eliminado correctamente.",
"Información", JOptionPane.INFORMATION_MESSAGE);

            cargarUsuarios(); // Actualiza la tabla después de eliminar
        } else {
            JOptionPane.showMessageDialog(this, "No se pudo eliminar el usuario. Inténtalo
nuevamente.", "Error", JOptionPane.ERROR_MESSAGE);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }

```

```
        JOptionPane.showMessageDialog(this, "Error al eliminar el usuario: " +  
e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);  
    }  
}
```

```
// Variables declaration - do not modify  
private javax.swing.JButton EliminarButton;  
private javax.swing.JButton actualizarButton;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JTable jTable1;  
// End of variables declaration  
}
```

Vista Perfil Admin

```
/*  
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change  
this license  
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JPanel.java to edit this  
template  
 */  
package AdminViews;  
  
import Views.*;
```

```
import Login.LoginForm;
import Login.LoginForm.SesionUsuario;
import java.awt.Window;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.JOptionPane;
import javax.swing.SwingUtilities;

/**
 *
 * @author IGNITER
 */
public class VistaPerfilAdministrador extends javax.swing.JPanel {
    private int clienteId; // Variable para almacenar el ID del cliente
    private boolean editando = false; // Variable para saber si está en modo edición


    /**
     * Creates new form VistaPrincipal
     */
    public VistaPerfilAdministrador() {
        initComponents();
        cargarDatosPerfil(); // Cargar datos al inicializar la vista
        bloquearCampos(); // Bloquear campos inicialmente
        configurarEventos(); // Llama al método para configurar eventos
    }
}
```

```

}

// Método para configurar los eventos
private void configurarEventos() {
    // Evento para el botón de cerrar sesión
    BotonCerrar.addActionListener(evt -> cerrarSesion());

    // Evento para el botón de editar perfil
    EditarPerfil.addActionListener(evt -> {
        if (editando) {
            guardarCambiosPerfil(); // Guarda cambios si ya está en modo edición
        } else {
            editando = true;
            desbloquearCampos(); // Permite la edición
            EditarPerfil.setText("Guardar Cambios"); // Cambia el texto del botón
        }
    });
}

// Método para cargar los datos del perfil del usuario actual
private void cargarDatosPerfil() {
    int usuarioId = SesionUsuario.obtenerUsuarioId();
    if (usuarioId == 0) {
        JOptionPane.showMessageDialog(this, "No se ha encontrado el usuario en la sesión.", "Error", JOptionPane.ERROR_MESSAGE);
        return;
    }

    try (Connection conn =
        DriverManager.getConnection("jdbc:mysql://localhost:3306/agroleaf_db", "root", "")) {

```

```

String sql = "SELECT username, fullname, email FROM users WHERE id = ?";
PreparedStatement pst = conn.prepareStatement(sql);
pst.setInt(1, usuarioId);
ResultSet rs = pst.executeQuery();

if (rs.next()) {
    NombrePerfil.setText(rs.getString("username"));
    NombreUser.setText(rs.getString("fullname"));
    CorreoField.setText(rs.getString("email"));
} else {
    JOptionPane.showMessageDialog(this, "No se encontraron datos para el
usuario.", "Información", JOptionPane.INFORMATION_MESSAGE);
}

} catch (Exception e) {
    e.printStackTrace();
    JOptionPane.showMessageDialog(this, "Error al cargar los datos del perfil: " +
e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
}
}

// Método para bloquear los campos de edición
private void bloquearCampos() {
    NombrePerfil.setEditable(false);
    NombreUser.setEditable(false);
    CorreoField.setEditable(false);
}

// Método para desbloquear los campos de edición

```

```

private void desbloquearCampos() {
    NombrePerfil.setEditable(true);
    NombreUser.setEditable(true);
    CorreoField.setEditable(true);
}

// Método para guardar los cambios en el perfil del usuario
private void guardarCambiosPerfil() {
    int usuarioId = SesionUsuario.obtenerUsuarioId();
    String nuevoNombre = NombrePerfil.getText().trim();
    String nuevoNombreuser = NombreUser.getText().trim();
    String nuevoCorreo = CorreoField.getText().trim();

    try (Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/agroleaf_db", "root", "")) {
        String sql = "UPDATE users SET username = ?, fullname = ?, email = ? WHERE id
= ?";
        PreparedStatement pst = conn.prepareStatement(sql);
        pst.setString(1, nuevoNombre);
        pst.setString(2, nuevoNombreuser);
        pst.setString(3, nuevoCorreo);
        pst.setInt(4, usuarioId);

        int filasAfectadas = pst.executeUpdate();
        if (filasAfectadas > 0) {
            JOptionPane.showMessageDialog(this, "Perfil actualizado correctamente.");
            bloquearCampos(); // Bloquea los campos después de guardar
            editando = false; // Cambia el estado de edición
        }
    }
}

```

```

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

        JOptionPane.showMessageDialog(this, "Error al actualizar el perfil: " +
e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    }
}

```

```

private void cerrarSesion() {
    // Cierra la sesión del usuario
    SesionUsuario.cerrarSesion();

    // Muestra un mensaje de confirmación
    JOptionPane.showMessageDialog(this, "Sesión cerrada.");

    // Abre la ventana de LoginForm
    LoginForm loginForm = new LoginForm();
    loginForm.setVisible(true);

    // Cierra la ventana actual
    // Primero verificamos si el contenedor es un JFrame
    javax.swing.SwingUtilities.getWindowAncestor(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() {

    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    BotonCerrar = new javax.swing.JButton();
    Imagenperfillbl = new javax.swing.JLabel();
    CorreoField = new javax.swing.JTextField();
    NombreUser = new javax.swing.JTextField();
    NombrePerfil = new javax.swing.JTextField();
    EditarPerfil = new javax.swing.JButton();

    setBackground(new java.awt.Color(255, 255, 255));
    setPreferredSize(new java.awt.Dimension(830, 563));

    jLabel1.setFont(new java.awt.Font("Arial", 1, 36)); // NOI18N
    jLabel1.setText("Perfil:");

    jLabel2.setFont(new java.awt.Font("Arial", 0, 18)); // NOI18N
    jLabel2.setText("Nombre:");

    jLabel3.setFont(new java.awt.Font("Arial", 0, 18)); // NOI18N
    jLabel3.setText("Correo:");

    BotonCerrar.setFont(new java.awt.Font("Arial", 1, 18)); // NOI18N
```

```
BotonCerrar.setText("Cerrar Sesion");

BotonCerrar.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        BotonCerrarActionPerformed(evt);

    }

});
```

```
Imagenperfilbl.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/iconoperfil (1).png"))); // NOI18N

Imagenperfilbl.setPreferredSize(new java.awt.Dimension(35, 35));

Imagenperfilbl.setRequestFocusEnabled(false);
```

```
CorreoField.setFont(new java.awt.Font("Arial", 0, 18)); // NOI18N

CorreoField.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        CorreoFieldActionPerformed(evt);

    }

});
```

```
NombreUser.setFont(new java.awt.Font("Arial", 0, 18)); // NOI18N

NombreUser.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        NombreUserActionPerformed(evt);

    }

});
```

```
NombrePerfil.setFont(new java.awt.Font("Arial", 0, 36)); // NOI18N

NombrePerfil.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        NombrePerfilActionPerformed(evt);  
    }  
});
```

```
EditorPerfil.setFont(new java.awt.Font("Arial", 1, 18)); // NOI18N  
EditorPerfil.setText("Editor Perfil");  
EditorPerfil.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        EditorPerfilActionPerformed(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()  
            .addGap(19, 19, 19)  
            .addComponent(Imagenperfilbl, javax.swing.GroupLayout.PREFERRED_SIZE,  
231, javax.swing.GroupLayout.PREFERRED_SIZE)  
            .addGap(18, 18, 18)  
  
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,  
false)  
                .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 164,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
                .addGroup(layout.createSequentialGroup()  
                    .addGap(2, 2, 2)  
  
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```

        .addGroup(layout.createSequentialGroup())

        .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 107,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(NombrePerfil))

        .addGroup(layout.createSequentialGroup())

        .addComponent(jLabel2,
javax.swing.GroupLayout.PREFERRED_SIZE, 86,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(NombreUser,
javax.swing.GroupLayout.PREFERRED_SIZE, 297,
javax.swing.GroupLayout.PREFERRED_SIZE))))

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()

        .addComponent(jLabel3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(CorreoField,
javax.swing.GroupLayout.PREFERRED_SIZE, 301,
javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()

        .addGap(30, 30, 30)

        .addComponent(EditarPerfil, javax.swing.GroupLayout.PREFERRED_SIZE,
139, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

```

```

        .addComponent(BotonCerrar)
        .addGap(49, 49, 49))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(29, 29, 29)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 47,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(NombrePerfil,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(33, 33, 33)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel2)
            .addComponent(NombreUser,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(36, 36, 36)

```

```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(CorreoField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel3)))
.addComponent(Imagenperfilbl,
javax.swing.GroupLayout.PREFERRED_SIZE, 217,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
276, Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(EditarPerfil, javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(BotonCerrar, javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(32, 32, 32))
);
} // </editor-fold>

```

```

private void EditarPerfilActionPerformed(java.awt.event.ActionEvent evt) {

}

```

```

private void BotonCerrarActionPerformed(java.awt.event.ActionEvent evt) {
    JOptionPane.showMessageDialog(this, "Sesión cerrada.");
}

```

```

private void NombrePerfilActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```

```
}
```

```
private void NombreUserActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    // TODO add your handling code here:
```

```
}
```

```
private void CorreoFieldActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    // TODO add your handling code here:
```

```
}
```

```
// Variables declaration - do not modify
```

```
private javax.swing.JButton BotonCerrar;
```

```
private javax.swing.JTextField CorreoField;
```

```
private javax.swing.JButton EditarPerfil;
```

```
private javax.swing.JLabel Imagenperfillbl;
```

```
private javax.swing.JTextField NombrePerfil;
```

```
private javax.swing.JTextField NombreUser;
```

```
private javax.swing.JLabel jLabel1;
```

```
private javax.swing.JLabel jLabel2;
```

```
private javax.swing.JLabel jLabel3;
```

```
private javax.swing.JLabel jLabel4;
```

```
// End of variables declaration
```

```
}
```

Paquete Views

En el paquete de views se encuentran las vistas del cliente que son la vista principal, calendario, perfil, reportes y clima se organizaron aparte para no confundir las rutas y paquetes que se importan.

Vista Principal Cliente

/*

* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JPanel.java to edit this template

*/

package Views;

import java.awt.Component;

import java.awt.Image;

import java.sql.Connection;

import java.sql.DriverManager;

import static java.sql.JDBCType.BLOB;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.swing.ImageIcon;

```

import javax.swing.JDialog;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableCellRenderer;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author IGNITER
 */
public class VistaPrincipal extends javax.swing.JPanel {

    /**
     * Creates new form VistaPrincipal
     */
    public VistaPrincipal() {
        initComponents();
        cargarEstados();
        cargarHortalizas("", ""); // Cargar todos los datos inicialmente
        configurarRenderizadoImagenes(); // Configurar el renderizador para la columna de
        imágenes
        // Ajustar el tamaño de las celdas
        TablaHortalizas.setRowHeight(100); // Ajusta la altura de las filas (por ejemplo, 100
        píxeles)
        TablaHortalizas.getColumnModel().getColumn(3).setPreferredWidth(100); // Ajusta el
        ancho de la columna de la imagen
    }

    private void cargarEstados() {
        Estados.removeAllItems();
    }

```

```
Estados.addItem("Todos");
Estados.addItem("Campeche");
Estados.addItem("Yucatán");
Estados.addItem("Quintana Roo");
}
```

```
// Método para configurar el renderizado de imágenes en la columna de la tabla
private void configurarRenderizadoImagenes() {
    TablaHortalizas.getColumnModel().getColumn(3).setCellRenderer(new
DefaultTableCellRenderer() {
        @Override
        public Component getTableCellRendererComponent(JTable table, Object value,
boolean isSelected, boolean hasFocus, int row, int column) {
            if (value instanceof ImageIcon) {
                setIcon((ImageIcon) value);
                setText(null); // No mostrar texto en la celda
            } else {
                setIcon(null);
                setText("Sin imagen"); // Texto por defecto si no hay imagen
            }
            return this;
        }
    });
}
```

```
// Método para cargar hortalizas en la tabla con filtro de estado y nombre
private void cargarHortalizas(String estado, String nombre) {
    DefaultTableModel model = (DefaultTableModel) TablaHortalizas.getModel();
    model.setRowCount(0); // Limpiar la tabla
}
```

```

try (Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/agroleaf_db", "root", "")) {

    String sql = "SELECT id, nombre, estado_region, imagen FROM hortalizas
WHERE (? = " OR estado_region = ?) AND (? = " OR nombre LIKE ?)";

    PreparedStatement pst = conn.prepareStatement(sql);

    pst.setString(1, estado);
    pst.setString(2, estado);
    pst.setString(3, nombre);
    pst.setString(4, "%" + nombre + "%");

    ResultSet rs = pst.executeQuery();
    while (rs.next()) {
        int id = rs.getInt("id");

        String nombreHortaliza = rs.getString("nombre");
        String estadoRegion = rs.getString("estado_region");

        // Convertir bytes de imagen a ImageIcon
        byte[] imagenBytes = rs.getBytes("imagen");
        ImageIcon imagenIcono = null;
        if (imagenBytes != null) {
            Image imagenEscalada = new
ImageIcon(imagenBytes).getImage().getScaledInstance(100, 100,
Image.SCALE_SMOOTH);
            imagenIcono = new ImageIcon(imagenEscalada);
        }

        // Agregar fila a la tabla con la imagen
        model.addRow(new Object[]{id, nombreHortaliza, estadoRegion,
imagenIcono});
    }
}

```

```

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

        JOptionPane.showMessageDialog(this, "Error al cargar hortalizas: " +
e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    }
}

```

// Convierte el array de bytes en una imagen (ejemplo básico)

```

/**
 * 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() {

    Estados = new javax.swing.JComboBox<>();
    Buscar = new javax.swing.JTextField();
    jScrollPane1 = new javax.swing.JScrollPane();
    TablaHortalizas = new javax.swing.JTable();

```

```
setBackground(new java.awt.Color(204, 255, 204));  
setPreferredSize(new java.awt.Dimension(830, 563));
```

```
Estados.setFont(new java.awt.Font("Arial", 0, 18)); // NOI18N
```

```
Estados.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {  
"Todos", "Campeche", "Yucatan", "Quintana Roo" }));
```

```
Estados.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        EstadosActionPerformed(evt);  
    }  
});
```

```
Buscar.setFont(new java.awt.Font("Arial", 0, 18)); // NOI18N
```

```
Buscar.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        BuscarActionPerformed(evt);  
    }  
});
```

```
TablaHortalizas.setFont(new java.awt.Font("Arial", 0, 18)); // NOI18N
```

```
TablaHortalizas.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 [] {  
        "ID", "Hortaliza", "Estado/Region", "Imagen"
```

```

    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});

TablaHortalizas.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        TablaHortalizasMouseClicked(evt);
    }
});

jScrollPane1.setViewportViewView(TablaHortalizas);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);
this.setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 830,
Short.MAX_VALUE)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addGap(81, 81, 81)
            .addComponent(Buscar, javax.swing.GroupLayout.PREFERRED_SIZE, 209,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

```

```

        .addComponent(Estados, javax.swing.GroupLayout.PREFERRED_SIZE, 129,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(78, 78, 78))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(34, 34, 34)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(Estados, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(Buscar, javax.swing.GroupLayout.PREFERRED_SIZE, 36,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
107, Short.MAX_VALUE)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
386, javax.swing.GroupLayout.PREFERRED_SIZE))
    );
} // </editor-fold>

```

```

private void BuscarActionPerformed(java.awt.event.ActionEvent evt) {
    String estado = Estados.getSelectedItem().toString();
    if (estado.equals("Todos")) estado = "";
    String nombre = Buscar.getText().trim();
    cargarHortalizas(estado, nombre);
}

```

```

private void EstadosActionPerformed(java.awt.event.ActionEvent evt) {
    if (Estados.getSelectedItem() != null) {

```

```

        String estadoSeleccionado = Estados.getSelectedItem().toString();
        cargarHortalizas(estadoSeleccionado, Buscar.getText().trim());
    } else {
        System.out.println("No hay ningún estado seleccionado en el ComboBox.");
    }
}

```

```

private void TablaHortalizasMouseClicked(java.awt.event.MouseEvent evt) {
    // Obtén la fila seleccionada
    int selectedRow = TablaHortalizas.getSelectedRow();
    if (selectedRow != -1) {
        int id = (int) TablaHortalizas.getValueAt(selectedRow, 0);

        // Crea una instancia de HortalizaDetallesDialog
        HortalizaDetallesDialog detallesDialog = new HortalizaDetallesDialog();

        // Carga los detalles de la hortaliza seleccionada
        detallesDialog.cargarDetalles(id);

        // Crear un nuevo JDialog para mostrar los detalles
        JDialog dialog = new JDialog();
        dialog.setContentPane(detallesDialog); // Establecer el contenido del JDialog
        dialog.setSize(1010, 650); // Ajustar el tamaño del diálogo
        dialog.setLocationRelativeTo(this); // Centrar el diálogo en la ventana actual
        dialog.setModal(true); // Hacer que el diálogo sea modal
        dialog.setVisible(true); // Mostrar el diálogo
    } else {
        JOptionPane.showMessageDialog(this, "Seleccione una hortaliza para ver los detalles.", "Sin selección", JOptionPane.WARNING_MESSAGE);
    }
}

```

```

    }
}

// Variables declaration - do not modify
private javax.swing.JTextField Buscar;
private javax.swing.JComboBox<String> Estados;
private javax.swing.JTable TablaHortalizas;
private javax.swing.JScrollPane jScrollPane1;
// End of variables declaration
}

```

Vista Calendario Cliente

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
 this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JPanel.java to edit this
 template
 */
package Views;

```

```

import Login.LoginForm.SesionUsuario;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Time;
import java.util.Date;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author IGNITER
 */
public class CalendarioP extends javax.swing.JPanel {

    /**
     * Creates new form VistaPrincipal
     */
    public CalendarioP() {
        initComponents();

        TablaRecordatorios.getSelectionModel().addListSelectionListener(event -> {
            int selectedRow = TablaRecordatorios.getSelectedRow();
            if (selectedRow != -1) { // Verifica que se haya seleccionado una fila
                String titulo = (String) TablaRecordatorios.getValueAt(selectedRow, 0);
                String descripcion = (String) TablaRecordatorios.getValueAt(selectedRow, 1);
            }
        });
    }

```

```

        // Si el título y descripción no son nulos, mostrar en los campos
        Titulo.setText(titulo != null ? titulo : "");
        Descripcion.setText(descripcion != null ? descripcion : "");
    }
});

jCalendar1.getDayChooser().addPropertyChangeListener("day", evt -> {
    Date fechaSeleccionada = jCalendar1.getDate();
    Titulo.setText("");
    Descripcion.setText("");
    cargarRecordatoriosPorFecha(fechaSeleccionada);
});
}

private int obtenerUsuarioIdActual() {
    return SesionUsuario.obtenerUsuarioId();
}

private void cargarRecordatoriosPorFecha(Date fechaSeleccionada) {
    int usuarioId = obtenerUsuarioIdActual();
    System.out.println("Usuario ID: " + usuarioId); // Línea de depuración

    if (usuarioId == 0) {
        JOptionPane.showMessageDialog(this, "No se ha encontrado el usuario en la sesión.",
            "Error", JOptionPane.ERROR_MESSAGE);
        return;
    }

    System.out.println("Fecha seleccionada: " + fechaSeleccionada); // Línea de depuración

    DefaultTableModel model = (DefaultTableModel) TablaRecordatorios.getModel();
    model.setRowCount(0); // Limpia la tabla antes de agregar nuevos datos

```

```

try (Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/agroleaf_db", "root", "")) {

    String sql = "SELECT titulo, descripcion FROM recordatorios WHERE fecha = ?
AND user_id = ?";

    PreparedStatement pst = conn.prepareStatement(sql);

    pst.setDate(1, new java.sql.Date(fechaSeleccionada.getTime()));

    pst.setInt(2, usuarioId);

    ResultSet rs = pst.executeQuery();

    if (!rs.isBeforeFirst()) { // Verifica si hay resultados

        System.out.println("No se encontraron recordatorios para la fecha y usuario
seleccionados.");

        JOptionPane.showMessageDialog(this, "No se encontraron recordatorios para la
fecha seleccionada.", "Información", JOptionPane.INFORMATION_MESSAGE);

        return;
    }

    // Itera sobre los resultados y agrégalos a la tabla
    while (rs.next()) {

        String titulo = rs.getString("titulo");

        String descripcion = rs.getString("descripcion");

        model.addRow(new Object[]{titulo, descripcion});

    }

} catch (SQLException e) {

    e.printStackTrace();

    JOptionPane.showMessageDialog(this, "Error al cargar los recordatorios: " +
e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);

}

```

```
}
```

```
private void guardarRecordatorio(int usuarioId, Date fecha, String titulo, String
descripcion) {
    try (Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/agroleaf_db", "root", "")) {
        String sql;
        int selectedRow = TablaRecordatorios.getSelectedRow();

        if (selectedRow != -1) { // Si hay una fila seleccionada, actualiza el recordatorio
            String tituloOriginal = (String) TablaRecordatorios.getValueAt(selectedRow, 0);
            sql = "UPDATE recordatorios SET titulo = ?, descripcion = ? WHERE user_id = ?
AND fecha = ? AND titulo = ?";
            PreparedStatement pst = conn.prepareStatement(sql);
            pst.setString(1, titulo);
            pst.setString(2, descripcion);
            pst.setInt(3, usuarioId);
            pst.setDate(4, new java.sql.Date(fecha.getTime()));
            pst.setString(5, tituloOriginal);

            int filasAfectadas = pst.executeUpdate();
            if (filasAfectadas > 0) {
                JOptionPane.showMessageDialog(this, "Recordatorio actualizado
correctamente.");
            }
        } else { // Si no hay fila seleccionada, inserta un nuevo recordatorio
```

```

        sql = "INSERT INTO recordatorios (user_id, fecha, titulo, descripcion) VALUES
        (?, ?, ?, ?)";

        PreparedStatement pst = conn.prepareStatement(sql);

        pst.setInt(1, usuarioId);

        pst.setDate(2, new java.sql.Date(fecha.getTime()));

        pst.setString(3, titulo);

        pst.setString(4, descripcion);


        int filasAfectadas = pst.executeUpdate();

        if (filasAfectadas > 0) {

            JOptionPane.showMessageDialog(this, "Recordatorio guardado correctamente.");

        }

    }

} catch (SQLException e) {

    e.printStackTrace();

    JOptionPane.showMessageDialog(this, "Error al guardar el recordatorio: " +
    e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);

}

cargarRecordatoriosPorFecha(fecha); // Refresca la tabla

}

```

/**

* 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() {

    jLocaleChooser1 = new com.toedter.components.JLocaleChooser();
    jCalendar1 = new com.toedter.calendar.JCalendar();
    jLabel1 = new javax.swing.JLabel();
    ButtonGuardar = new javax.swing.JButton();
    Titulo = new javax.swing.JTextField();
    jScrollPane1 = new javax.swing.JScrollPane();
    Descripcion = new javax.swing.JTextArea();
    jLabel2 = new javax.swing.JLabel();
    jScrollPane2 = new javax.swing.JScrollPane();
    TablaRecordatorios = new javax.swing.JTable();
    EliminarButton = new javax.swing.JButton();

    setBackground(new java.awt.Color(204, 255, 204));
    setPreferredSize(new java.awt.Dimension(830, 563));

    jLabel1.setFont(new java.awt.Font("Arial", 0, 18)); // NOI18N
    jLabel1.setText("Titulo");

    ButtonGuardar.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
    ButtonGuardar.setText("Guardar recordatorio");
    ButtonGuardar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {

```

```
        ButtonGuardarActionPerformed(evt);
    }
});
```

```
Titulo.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        TituloActionPerformed(evt);
    }
});
```

```
Descripcion.setColumns(20);
Descripcion.setRows(5);
jScrollPane1.setViewportViewView(Descripcion);
```

```
jLabel2.setFont(new java.awt.Font("Arial", 0, 18)); // NOI18N
jLabel2.setText("Descripcion");
```

```
TablaRecordatorios.setFont(new java.awt.Font("Arial", 0, 24)); // NOI18N
TablaRecordatorios.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

        },
    new String [] {
        "Titulo", "Recordatorio"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false
    };
});
```

```
});

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});

TablaRecordatorios.setToolTipText("");
TablaRecordatorios.setSelectionBackground(new java.awt.Color(255, 255, 255));
jScrollPane2.setViewportView(TablaRecordatorios);
if (TablaRecordatorios.getColumnModel().getColumnCount() > 0) {
    TablaRecordatorios.getColumnModel().getColumn(0).setResizable(false);
    TablaRecordatorios.getColumnModel().getColumn(0).setPreferredWidth(200);
    TablaRecordatorios.getColumnModel().getColumn(1).setResizable(false);
    TablaRecordatorios.getColumnModel().getColumn(1).setPreferredWidth(300);
}

EliminarButton.setText("Eliminar");
EliminarButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        EliminarButtonActionPerformed(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()
```

```

        .addGap(544, 544, 544)

        .addComponent(Titulo, javax.swing.GroupLayout.PREFERRED_SIZE, 193,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 830,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGroup(layout.createSequentialGroup())

        .addGap(609, 609, 609)

        .addComponent(jLabel1))

        .addGroup(layout.createSequentialGroup())

        .addComponent(jCalendar1, javax.swing.GroupLayout.PREFERRED_SIZE, 438,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(layout.createSequentialGroup()

                .addGap(104, 104, 104)

                .addComponent(ButtonGuardar,
javax.swing.GroupLayout.PREFERRED_SIZE, 182,
javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGroup(layout.createSequentialGroup()

                .addGap(149, 149, 149)

                .addComponent(jLabel2))

            .addGroup(layout.createSequentialGroup()

                .addGap(53, 53, 53)

                .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 287,
javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGroup(layout.createSequentialGroup()

                .addGap(154, 154, 154)

                .addComponent(EliminarButton))))

    );

    layout.setVerticalGroup(

```

```

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jCalendar1, javax.swing.GroupLayout.PREFERRED_SIZE, 342,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(0, 0, Short.MAX_VALUE))
            .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(49, 49, 49)
        .addComponent(Titulo, javax.swing.GroupLayout.PREFERRED_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup()
            .addGap(21, 21, 21)
            .addComponent(jLabel1)))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
            22, Short.MAX_VALUE)
        .addComponent(jLabel2)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(26, 26, 26)
        .addComponent(ButtonGuardar, javax.swing.GroupLayout.PREFERRED_SIZE,
            36, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(33, 33, 33)
        .addComponent(EliminarButton)
        .addGap(18, 18, 18)
        .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE,
            220, javax.swing.GroupLayout.PREFERRED_SIZE))

```

```
);  
} // </editor-fold>
```

```
private void TituloActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

```
private void ButtonGuardarActionPerformed(java.awt.event.ActionEvent evt) {  
String titulo = Titulo.getText().trim();  
String descripcion = Descripcion.getText().trim();  
Date fechaSeleccionada = jCalendar1.getDate(); // Captura la fecha seleccionada  
  
// Verificar que el título y la fecha no estén vacíos  
if (titulo.isEmpty() || fechaSeleccionada == null) {  
    JOptionPane.showMessageDialog(this, "El título y la fecha son obligatorios.", "Error",  
JOptionPane.ERROR_MESSAGE);  
    return;  
}
```

```
int usuarioId = obtenerUsuarioIdActual(); // Obtiene el ID del usuario actual
```

```
// Asegúrate de que usuarioId no es 0  
if (usuarioId == 0) {  
    JOptionPane.showMessageDialog(this, "No se ha encontrado el usuario en la sesión.",  
"Error", JOptionPane.ERROR_MESSAGE);  
    return;  
}
```

```
// Guardar el recordatorio en la base de datos
```

```

guardarRecordatorio(usuarioId, fechaSeleccionada, titulo, descripcion);

// Refrescar los recordatorios de la fecha seleccionada en la tabla
cargarRecordatoriosPorFecha(fechaSeleccionada);

// Limpiar los campos después de guardar
Titulo.setText("");
Descripcion.setText("");

}

private void EliminarButtonActionPerformed(java.awt.event.ActionEvent evt) {
    int selectedRow = TablaRecordatorios.getSelectedRow();
    if (selectedRow == -1) { // Verifica que se haya seleccionado una fila
        JOptionPane.showMessageDialog(this, "Por favor selecciona un recordatorio para
eliminar.", "Advertencia", JOptionPane.WARNING_MESSAGE);
        return;
    }

    // Obtener datos de la fila seleccionada
    String titulo = (String) TablaRecordatorios.getValueAt(selectedRow, 0);
    Date fechaSeleccionada = jCalendar1.getDate(); // Fecha seleccionada en el calendario
    int usuarioId = obtenerUsuarioIdActual();

    try (Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/agroleaf_db", "root", "")) {
        // Consulta SQL para eliminar el recordatorio

        String sql = "DELETE FROM recordatorios WHERE user_id = ? AND fecha = ?
AND titulo = ?";

```

```

        PreparedStatement pst = conn.prepareStatement(sql);

        pst.setInt(1, usuarioId);

        pst.setDate(2, new java.sql.Date(fechaSeleccionada.getTime()));

        pst.setString(3, titulo);


        int filasAfectadas = pst.executeUpdate();

        if (filasAfectadas > 0) { // Verifica si se eliminó algún registro

            JOptionPane.showMessageDialog(this, "Recordatorio eliminado correctamente.");

            cargarRecordatoriosPorFecha(fechaSeleccionada); // Refresca la tabla

        } else {

            JOptionPane.showMessageDialog(this, "No se pudo eliminar el recordatorio.
            Verifica la información.", "Error", JOptionPane.ERROR_MESSAGE);

        }

    } catch (SQLException e) {

        e.printStackTrace();

        JOptionPane.showMessageDialog(this, "Error al eliminar el recordatorio: " +
        e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);

    }

}

// Variables declaration - do not modify
private javax.swing.JButton ButtonGuardar;
private javax.swing.JTextArea Descripcion;
private javax.swing.JButton EliminarButton;
private javax.swing.JTable TablaRecordatorios;
private javax.swing.JTextField Titulo;
private com.toedter.calendar.JCalendar jCalendar1;
private javax.swing.JLabel jLabel1;

```

```

private javax.swing.JLabel jLabel2;
private com.toedter.components.JLocaleChooser jLocaleChooser1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
// End of variables declaration
}

```

Vista Perfil Cliente

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
 this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JPanel.java to edit this
 template
 */
package Views;

import Login.LoginForm;
import Login.LoginForm.SesionUsuario;
import java.awt.Window;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.JOptionPane;
import javax.swing.SwingUtilities;

```

```

/**
 *
 * @author IGNITER
 */
public class VistaPerfil extends javax.swing.JPanel {

    private int clienteId; // Variable para almacenar el ID del cliente
    private boolean editando = false; // Variable para saber si está en modo edición


    /**
     * Creates new form VistaPrincipal
     */
    public VistaPerfil() {
        initComponents();
        cargarDatosPerfil(); // Cargar datos al inicializar la vista
        bloquearCampos(); // Bloquear campos inicialmente
        configurarEventos(); // Llama al método para configurar eventos
    }

    // Método para configurar los eventos
    private void configurarEventos() {

        // Evento para el botón de cerrar sesión
        BotonCerrar.addActionListener(evt -> cerrarSesion());


        // Evento para el botón de editar perfil
        EditarPerfil.addActionListener(evt -> {
            if (editando) {

```

```

        guardarCambiosPerfil(); // Guarda cambios si ya está en modo edición
    } else {
        editando = true;
        desbloquearCampos(); // Permite la edición
        EditarPerfil.setText("Guardar Cambios"); // Cambia el texto del botón
    }
});
}

```

```

// Método para cargar los datos del perfil del usuario actual
private void cargarDatosPerfil() {
    int usuarioId = SesionUsuario.obtenerUsuarioId();
    if (usuarioId == 0) {
        JOptionPane.showMessageDialog(this, "No se ha encontrado el usuario en la sesión.", "Error", JOptionPane.ERROR_MESSAGE);
        return;
    }

    try (Connection conn =
        DriverManager.getConnection("jdbc:mysql://localhost:3306/agroleaf_db", "root", "")) {
        String sql = "SELECT username, fullname, email FROM users WHERE id = ?";
        PreparedStatement pst = conn.prepareStatement(sql);
        pst.setInt(1, usuarioId);
        ResultSet rs = pst.executeQuery();

        if (rs.next()) {
            NombrePerfil.setText(rs.getString("username"));
            NombreUser.setText(rs.getString("fullname"));
            CorreoField.setText(rs.getString("email"));
        }
    }
}

```

```

        } else {

            JOptionPane.showMessageDialog(this, "No se encontraron datos para el
usuario.", "Información", JOptionPane.INFORMATION_MESSAGE);

        }

    } catch (Exception e) {

        e.printStackTrace();

        JOptionPane.showMessageDialog(this, "Error al cargar los datos del perfil: " +
e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);

    }

}

// Método para bloquear los campos de edición
private void bloquearCampos() {

    NombrePerfil.setEditable(false);

    NombreUser.setEditable(false);

    CorreoField.setEditable(false);

}

// Método para desbloquear los campos de edición
private void desbloquearCampos() {

    NombrePerfil.setEditable(true);

    NombreUser.setEditable(true);

    CorreoField.setEditable(true);

}

// Método para guardar los cambios en el perfil del usuario
private void guardarCambiosPerfil() {

    int usuarioId = SesionUsuario.obtenerUsuarioId();

```

```

String nuevoNombre = NombrePerfil.getText().trim();

String nuevoNombreuser = NombreUser.getText().trim();

String nuevoCorreo = CorreoField.getText().trim();


try (Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/agroleaf_db", "root", "")) {

    String sql = "UPDATE users SET username = ?, fullname = ?, email = ? WHERE id
= ?";

    PreparedStatement pst = conn.prepareStatement(sql);

    pst.setString(1, nuevoNombre);

    pst.setString(2, nuevoNombreuser);

    pst.setString(3, nuevoCorreo);

    pst.setInt(4, usuarioId);


    int filasAfectadas = pst.executeUpdate();

    if (filasAfectadas > 0) {

        JOptionPane.showMessageDialog(this, "Perfil actualizado correctamente.");

        bloquearCampos(); // Bloquea los campos después de guardar

        editando = false; // Cambia el estado de edición

    }

} catch (Exception e) {

    e.printStackTrace();

    JOptionPane.showMessageDialog(this, "Error al actualizar el perfil: " +
e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);

}

}

private void cerrarSesion() {

    // Cierra la sesión del usuario

```

```

        SesionUsuario.cerrarSesion();

        // Muestra un mensaje de confirmación
        JOptionPane.showMessageDialog(this, "Sesión cerrada.");

        // Abre la ventana de LoginForm
        LoginForm loginForm = new LoginForm();
        loginForm.setVisible(true);

        // Cierra la ventana actual
        // Primero verificamos si el contenedor es un JFrame
        javax.swing.SwingUtilities.getWindowAncestor(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() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();

```

```
BotonCerrar = new javax.swing.JButton();
Imagenperfillbl = new javax.swing.JLabel();
CorreoField = new javax.swing.JTextField();
NombreUser = new javax.swing.JTextField();
NombrePerfil = new javax.swing.JTextField();
EditarPerfil = new javax.swing.JButton();
```

```
setBackground(new java.awt.Color(255, 255, 255));
setPreferredSize(new java.awt.Dimension(830, 563));
```

```
jLabel1.setFont(new java.awt.Font("Arial", 1, 36)); // NOI18N
jLabel1.setText("Perfil:");
```

```
jLabel2.setFont(new java.awt.Font("Arial", 0, 18)); // NOI18N
jLabel2.setText("Nombre:");
```

```
jLabel3.setFont(new java.awt.Font("Arial", 0, 18)); // NOI18N
jLabel3.setText("Correo:");
```

```
BotonCerrar.setFont(new java.awt.Font("Arial", 1, 18)); // NOI18N
BotonCerrar.setText("Cerrar Sesion");
BotonCerrar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BotonCerrarActionPerformed(evt);
    }
});
```

```
Imagenperfillbl.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/iconoperfil (1).png"))); // NOI18N
```

```
Imagenperfilbl.setPreferredSize(new java.awt.Dimension(35, 35));
Imagenperfilbl.setRequestFocusEnabled(false);
```

```
CorreoField.setFont(new java.awt.Font("Arial", 0, 18)); // NOI18N
CorreoField.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        CorreoFieldActionPerformed(evt);
    }
});
```

```
NombreUser.setFont(new java.awt.Font("Arial", 0, 18)); // NOI18N
NombreUser.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        NombreUserActionPerformed(evt);
    }
});
```

```
NombrePerfil.setFont(new java.awt.Font("Arial", 0, 36)); // NOI18N
NombrePerfil.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        NombrePerfilActionPerformed(evt);
    }
});
```

```
EditarPerfil.setFont(new java.awt.Font("Arial", 1, 18)); // NOI18N
EditarPerfil.setText("Editar Perfil");
EditarPerfil.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```

        EditarPerfilActionPerformed(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()
            .addGap(19, 19, 19)
            .addComponent(Imagenperfillbl, javax.swing.GroupLayout.PREFERRED_SIZE,
231, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE,
107, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

            .addComponent(NombrePerfil,
javax.swing.GroupLayout.PREFERRED_SIZE, 289,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(layout.createSequentialGroup()
                .addComponent(jLabel3)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

            .addComponent(CorreoField,
javax.swing.GroupLayout.PREFERRED_SIZE, 301,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 164,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addGroup(layout.createSequentialGroup())

        .addGap(2, 2, 2)

        .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE,
86, javax.swing.GroupLayout.PREFERRED_SIZE)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(NombreUser,
javax.swing.GroupLayout.PREFERRED_SIZE, 297,
javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addContainerGap(276, Short.MAX_VALUE))

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup())

        .addGap(30, 30, 30)

        .addComponent(EditarPerfil, javax.swing.GroupLayout.PREFERRED_SIZE,
139, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(BotonCerrar)

        .addGap(49, 49, 49))

    );

    layout.setVerticalGroup(

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                .addGroup(layout.createSequentialGroup()

                    .addGap(30, 30, 30)

                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                        .addComponent(NombrePerfil,
javax.swing.GroupLayout.PREFERRED_SIZE,

```

```

javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE,
47, javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE,
47, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(33, 33, 33)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel2)

        .addComponent(NombreUser,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(36, 36, 36)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(CorreoField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel3)))

        .addComponent(Imagenperfilbl,
javax.swing.GroupLayout.PREFERRED_SIZE, 217,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
206, Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(EditarPerfil, javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(BotonCerrar, javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE))

```

```

        .addGap(32, 32, 32))

    );
} // </editor-fold>

private void EditarPerfilActionPerformed(java.awt.event.ActionEvent evt) {

}

private void BotonCerrarActionPerformed(java.awt.event.ActionEvent evt) {
    JOptionPane.showMessageDialog(this, "Sesión cerrada.");
}

private void NombrePerfilActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void NombreUserActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void CorreoFieldActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

// Variables declaration - do not modify
private javax.swing.JButton BotonCerrar;
private javax.swing.JTextField CorreoField;

```

```

private javax.swing.JButton EditarPerfil;
private javax.swing.JLabel Imagenperfilbl;
private javax.swing.JTextField NombrePerfil;
private javax.swing.JTextField NombreUser;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
// End of variables declaration
}

```

Vista Reportes Cliente

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
 this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JPanel.java to edit this
 template
 */

```

```

package Views;

```

```

import javax.swing.table.DefaultTableModel;
import Login.LoginForm.SesionUsuario;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

```

```

import java.sql.SQLException;
import java.util.Date;
import javax.swing.JOptionPane;
/**
 *
 * @author IGNITER
 */
public class VistaReportes extends javax.swing.JPanel {

    /**
     * Creates new form VistaPrincipal
     */
    public VistaReportes() {
        initComponents();
        configurarTabla(); // Configura la tabla
        cargarRecordatorios(); // Carga los recordatorios
    }
    private void configurarTabla() {
        DefaultTableModel model = new DefaultTableModel(
            new Object[]{"Titulo", "Descripcion", "Fecha"}, 0 // Define las columnas
        ) {
            @Override
            public boolean isCellEditable(int row, int column) {
                return false; // Evita que las celdas sean editables
            }
        };
        TablaRecordatorios.setModel(model);
        TablaRecordatorios.setRowHeight(30); // Ajusta la altura de las filas
    }

```

```

}

private int obtenerUsuarioIdActual() {
    return SesionUsuario.obtenerUsuarioId();
}

private void cargarRecordatorios() {
    DefaultTableModel model = (DefaultTableModel) TablaRecordatorios.getModel();
    model.setRowCount(0); // Limpia la tabla antes de cargar nuevos datos

    int usuarioId = obtenerUsuarioIdActual(); // Obtén el ID del usuario actual
    if (usuarioId == 0) {
        JOptionPane.showMessageDialog(this, "No se ha encontrado el usuario en la sesión.",
            "Error", JOptionPane.ERROR_MESSAGE);
        return;
    }

    try (Connection conn =
        DriverManager.getConnection("jdbc:mysql://localhost:3306/agroleaf_db", "root", "")) {
        // Nueva consulta: incluir el día actual y los próximos 3 días
        String sql = "SELECT titulo, descripcion, fecha " +
            "FROM recordatorios " +
            "WHERE user_id = ? AND fecha >= CURDATE() AND fecha <= " +
            "DATE_ADD(CURDATE(), INTERVAL 3 DAY) " +
            "ORDER BY fecha ASC";

        PreparedStatement pst = conn.prepareStatement(sql);
        pst.setInt(1, usuarioId);

        ResultSet rs = pst.executeQuery();
        boolean hayResultados = false;
    }
}

```

```

while (rs.next()) {
    hayResultados = true;

    String titulo = rs.getString("titulo");
    String descripcion = rs.getString("descripcion");
    Date fecha = rs.getDate("fecha");

    // Agregar los datos al modelo de la tabla
    model.addRow(new Object[]{titulo, descripcion, fecha});
}

if (!hayResultados) {
    System.out.println("No se encontraron recordatorios para el usuario y fechas
seleccionadas.");
}
} catch (SQLException e) {
    e.printStackTrace();

    JOptionPane.showMessageDialog(this, "Error al cargar los recordatorios: " +
e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
}
}

/**
 * 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() {

    jScrollPane1 = new javax.swing.JScrollPane();
    TablaRecordatorios = new javax.swing.JTable();
    ActualizarButton = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();

    setBackground(new java.awt.Color(255, 255, 255));
    setPreferredSize(new java.awt.Dimension(830, 563));
    setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    TablaRecordatorios.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
    TablaRecordatorios.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {
            {null, null, null},
            {null, null, null},
            {null, null, null},
            {null, null, null}
        },
        new String [] {
            "Fecha", "Titulo", "Recordatorio"
        }
    ) {
        boolean[] canEdit = new boolean [] {
            false, false, false
        };
    };
```

```

        public boolean isCellEditable(int rowIndex, int columnIndex) {
            return canEdit [columnIndex];
        }
    });

jScrollPane1.setViewportViewView(TablaRecordatorios);
if (TablaRecordatorios.getColumnModel().getColumnCount() > 0) {
    TablaRecordatorios.getColumnModel().getColumn(0).setResizable(false);
    TablaRecordatorios.getColumnModel().getColumn(1).setResizable(false);
    TablaRecordatorios.getColumnModel().getColumn(2).setResizable(false);
}

add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(64, 41, 716,
471));

ActualizarButton.setText("Actualizar");
ActualizarButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ActualizarButtonActionPerformed(evt);
    }
});

add(ActualizarButton, new org.netbeans.lib.awtextra.AbsoluteConstraints(312, 518,
174, -1));

jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/pasto.png"))); //
NOI18N

add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 840, 570));
} // </editor-fold>

private void ActualizarButtonActionPerformed(java.awt.event.ActionEvent evt) {

```

```

        // Llama al método cargarRecordatorios para refrescar la tabla
        cargarRecordatorios();

        JOptionPane.showMessageDialog(this, "Datos actualizados correctamente.",
        "Información", JOptionPane.INFORMATION_MESSAGE);
    }

    // Variables declaration - do not modify
    private javax.swing.JButton ActualizarButton;
    private javax.swing.JTable TablaRecordatorios;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JScrollPane jScrollPane1;
    // End of variables declaration
}

```

Vista Clima Cliente

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
 this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JPanel.java to edit this
 template
 */

```

```
package Views;
```

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import org.json.JSONObject;
import javax.swing.GroupLayout;
```

```
/**
```

```
 *
```

```
 * @author IGNITER
```

```
 */
```

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

```
    /**
```

```
     * Creates new form VistaPrincipal
```

```
    */
```

```
    public VistaClima() {
```

```
        initComponents();
```

```
        // Configura el botón para que llame a buscarClima cuando se haga clic
```

```
        jButton1.addActionListener(new java.awt.event.ActionListener() {
```

```
            public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
                buscarClima();
```

```
            }
```

```
        });
```

```
    }
```

```
    public class WeatherService {
```

```
private final String API_KEY = "e599ac340e9bd1285d33a1e62fc00473"; // Coloca tu
API Key de OpenWeatherMap aquí
```

```
public String obtenerClima(String ciudad) {

    String urlString = "http://api.openweathermap.org/data/2.5/weather?q=" + ciudad +
"&appid=" + API_KEY + "&units=metric&lang=es";

    try {

        URL url = new URL(urlString);

        HttpURLConnection connection = (HttpURLConnection) url.openConnection();

        connection.setRequestMethod("GET");


        BufferedReader in = new BufferedReader(new
InputStreamReader(connection.getInputStream()));

        String inputLine;

        StringBuilder content = new StringBuilder();

        while ((inputLine = in.readLine()) != null) {

            content.append(inputLine);

        }

        in.close();

        connection.disconnect();

        JSONObject json = new JSONObject(content.toString());

        // Datos meteorológicos

        String weatherDescription =
json.getJSONArray("weather").getJSONObject(0).getString("description");

        double temp = json.getJSONObject("main").getDouble("temp");

        double feelsLike = json.getJSONObject("main").getDouble("feels_like");
```

```

double tempMin = json.getJSONObject("main").getDouble("temp_min");
double tempMax = json.getJSONObject("main").getDouble("temp_max");
int humidity = json.getJSONObject("main").getInt("humidity");
int pressure = json.getJSONObject("main").getInt("pressure");

// Datos del viento
double windSpeed = json.getJSONObject("wind").getDouble("speed");
int windDeg = json.getJSONObject("wind").has("deg") ?
json.getJSONObject("wind").getInt("deg") : 0;

// Visibilidad y nubosidad
int visibility = json.has("visibility") ? json.getInt("visibility") : -1;
int cloudiness = json.getJSONObject("clouds").getInt("all");

// Amanecer y atardecer
long sunrise = json.getJSONObject("sys").getLong("sunrise") * 1000; //
Convertir a milisegundos
long sunset = json.getJSONObject("sys").getLong("sunset") * 1000; // Convertir
a milisegundos

return "Descripción: " + weatherDescription +
    "\nTemperatura: " + temp + "°C" +
    "\nSensación Térmica: " + feelsLike + "°C" +
    "\nTemperatura Mínima: " + tempMin + "°C" +
    "\nTemperatura Máxima: " + tempMax + "°C" +
    "\nHumedad: " + humidity + "%" +
    "\nPresión: " + pressure + " hPa" +
    "\nVelocidad del Viento: " + windSpeed + " m/s" +
    "\nDirección del Viento: " + windDeg + "°" +

```

```

        "\nVisibilidad: " + (visibility != -1 ? visibility + " m" : "N/A") +
        "\nNubosidad: " + cloudiness + "%" +
        "\nAmanecer: " + new
java.text.SimpleDateFormat("HH:mm:ss").format(new java.util.Date(sunrise)) +
        "\nAtardecer: " + new java.text.SimpleDateFormat("HH:mm:ss").format(new
java.util.Date(sunset));
    } catch (Exception e) {
        e.printStackTrace();
        return "Error al obtener el clima";
    }
}
}
}

```

// Método que se llama desde el botón para realizar la búsqueda

```

private void buscarClima() {
    String ciudad = jTextField1.getText().trim(); // Obtiene el texto del campo de ciudad
    if (!ciudad.isEmpty()) {
        WeatherService weatherService = new WeatherService();
        String clima = weatherService.obtenerClima(ciudad); // Llama al método
obtenerClima
        MostrarDatos.setText(clima); // Muestra el resultado en el JTextArea
    } else {
        MostrarDatos.setText("Por favor, ingresa una ciudad.");
    }
}
}

```

/**

```

* 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() {

    jScrollPane1 = new javax.swing.JScrollPane();
    MostrarDatos = new javax.swing.JTextArea();
    jButton1 = new javax.swing.JButton();
    jTextField1 = new javax.swing.JTextField();
    jLabel1 = new javax.swing.JLabel();

    setBackground(new java.awt.Color(255, 255, 255));
    setPreferredSize(new java.awt.Dimension(830, 563));

    MostrarDatos.setEditable(false);
    MostrarDatos.setBackground(new java.awt.Color(255, 255, 255));
    MostrarDatos.setColumns(20);
    MostrarDatos.setFont(new java.awt.Font("Arial", 0, 18)); // NOI18N
    MostrarDatos.setRows(5);
    MostrarDatos.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));
    jScrollPane1.setViewportView(MostrarDatos);

    jButton1.setFont(new java.awt.Font("Arial", 0, 18)); // NOI18N
    jButton1.setText("Buscar");
    jButton1.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));

```

```
jButton1.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        jButton1ActionPerformed(evt);  
    }  
});
```

```
jTextField1.setFont(new java.awt.Font("Arial", 0, 18)); // NOI18N  
jTextField1.setBorder(javax.swing.BorderFactory.createLineBorder(new  
java.awt.Color(0, 0, 0)));  
jTextField1.setCursor(new java.awt.Cursor(java.awt.Cursor.TEXT_CURSOR));  
jTextField1.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        jTextField1ActionPerformed(evt);  
    }  
});
```

```
jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/imagentiempo  
(1).png"))); // NOI18N
```

```
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);  
this.setLayout(layout);  
layout.setHorizontalGroup(  
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addGroup(layout.createSequentialGroup()  
            .addGap(253, 253, 253)  
            .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,  
361, javax.swing.GroupLayout.PREFERRED_SIZE)  
            .addGroup(layout.createSequentialGroup()  
                .addGap(380, 380, 380)
```

```

        .addComponent(jButton1))
        .addGroup(layout.createSequentialGroup()
            .addGap(253, 253, 253)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
361, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(jLabel1)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(21, 21, 21)
            .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, 37,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(12, 12, 12)
            .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(12, 12, 12)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
316, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 560,
javax.swing.GroupLayout.PREFERRED_SIZE)
    );
} // </editor-fold>

```

```

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
jButton1.addActionListener(new java.awt.event.ActionListener() {

```

```
public void actionPerformed(java.awt.event.ActionEvent evt) {  
    }  
});  
}
```

```
// Variables declaration - do not modify  
private javax.swing.JTextArea MostrarDatos;  
private javax.swing.JButton jButton1;  
private javax.swing.JLabel jLabel1;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JTextField jTextField1;  
// End of variables declaration  
}
```

Vista de Hortalizas

/*

* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JPanel.java to edit this template

*/

package Views;

import java.awt.Image;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import javax.swing.ImageIcon;

import javax.swing.JOptionPane;

import javax.swing.SwingUtilities;

/**

*

* @author IGNITER

*/

public class HortalizaDetallesDialog extends javax.swing.JPanel {

/**

```

* Creates new form VistaPrincipal
*/
public HortalizaDetallesDialog() {
    initComponents();
}

// Método para cargar los detalles de la hortaliza desde la base de datos
// Método para cargar los detalles de la hortaliza desde la base de datos
public void cargarDetalles(int id) {
    try (Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/agroleaf_db", "root", "")) {
        // Asegúrate de incluir todas las columnas que necesitas en la consulta
        String sql = "SELECT id, nombre, estado_region, temporada_siembra,
temperatura_optima, "
            + "requerimientos_riego, tipo_suelo, enfermedades_comunes,
frecuencia_fertilizacion, "
            + "espacio_plantacion, metodo_cosecha, adaptabilidad_climatica, imagen "
            + "FROM hortalizas WHERE id = ?";

        PreparedStatement pst = conn.prepareStatement(sql);
        pst.setInt(1, id);
        ResultSet rs = pst.executeQuery();

        if (rs.next()) {
            // Asignar los valores obtenidos de la base de datos a los JTextAreas
            ID.setText(String.valueOf(rs.getInt("id")));
            NombredeHortaliza.setText(rs.getString("nombre"));
            EstadooRegion.setText(rs.getString("estado_region"));
            Temporadadesiembra.setText(rs.getString("temporada_siembra"));
            jTextArea4.setText(rs.getString("temperatura_optima")); // Asignar temperatura
            óptima

```

```

Requerimientosderiego.setText(rs.getString("requerimientos_riego"));
TipoSuelo.setText(rs.getString("tipo_suelo"));
Enfermedades.setText(rs.getString("enfermedades_comunes"));
jTextArea9.setText(rs.getString("frecuencia_fertilizacion")); // Asignar frecuencia
de fertilización
Espacioplantacion.setText(rs.getString("espacio_plantacion"));
Metododecosecha.setText(rs.getString("metodo_cosecha"));
Adaptabilidadclimatica.setText(rs.getString("adaptabilidad_climatica"));

// Cargar la imagen desde la base de datos
// Cargar la imagen desde la base de datos
// Cargar y mostrar la imagen
byte[] imgBytes = rs.getBytes("imagen");
if (imgBytes != null) {
    SwingUtilities.invokeLater(() -> {
        int width = lblImagenHortaliza.getWidth();
        int height = lblImagenHortaliza.getHeight();

        if (width > 0 && height > 0) {
            ImageIcon icon = new ImageIcon(new
            ImageIcon(imgBytes).getImage().getScaledInstance(
                width, height, Image.SCALE_SMOOTH));
            lblImagenHortaliza.setIcon(icon);
        } else {
            System.out.println("El JLabel lblImagenHortaliza aún no tiene dimensiones
válidas.");
        }
    });
} else {

```

```

        lblImagenHortaliza.setIcon(null); // Limpia el JLabel si no hay imagen
    }

    } else {

        JOptionPane.showMessageDialog(this, "No se encontraron datos para el ID: " + id,
        "Advertencia", JOptionPane.WARNING_MESSAGE);

    }


    rs.close();

    pst.close();

} catch (SQLException e) {

    e.printStackTrace();

    JOptionPane.showMessageDialog(this, "Error al cargar los detalles: " +
    e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);

}

}

```

```

/**

```

```

 * 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();
jScrollPane1 = new javax.swing.JScrollPane();
ID = new javax.swing.JTextArea();
jLabel3 = new javax.swing.JLabel();
jScrollPane2 = new javax.swing.JScrollPane();
NombrededeHortaliza = new javax.swing.JTextArea();
jLabel4 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
jScrollPane3 = new javax.swing.JScrollPane();
Temporadadesiembra = new javax.swing.JTextArea();
Temperaturaoptima = new javax.swing.JScrollPane();
jTextArea4 = new javax.swing.JTextArea();
jLabel6 = new javax.swing.JLabel();
jScrollPane5 = new javax.swing.JScrollPane();
Requerimientosderiego = new javax.swing.JTextArea();
jLabel7 = new javax.swing.JLabel();
jScrollPane6 = new javax.swing.JScrollPane();
TipoSuelo = new javax.swing.JTextArea();
jLabel8 = new javax.swing.JLabel();
jScrollPane7 = new javax.swing.JScrollPane();
EstadooRegion = new javax.swing.JTextArea();
jLabel9 = new javax.swing.JLabel();
jScrollPane8 = new javax.swing.JScrollPane();
Enfermedades = new javax.swing.JTextArea();
Frecuencia = new javax.swing.JLabel();
jScrollPane9 = new javax.swing.JScrollPane();
jTextArea9 = new javax.swing.JTextArea();
jLabel11 = new javax.swing.JLabel();
```

```
jScrollPane10 = new javax.swing.JScrollPane();
Espacioplantacion = new javax.swing.JTextArea();
jLabel12 = new javax.swing.JLabel();
jScrollPane11 = new javax.swing.JScrollPane();
Metododecosecha = new javax.swing.JTextArea();
jLabel13 = new javax.swing.JLabel();
jScrollPane12 = new javax.swing.JScrollPane();
Adaptabilidadclimatica = new javax.swing.JTextArea();
lblImagenHortaliza = new javax.swing.JLabel();
jLabel1 = new javax.swing.JLabel();

setBackground(new java.awt.Color(255, 255, 255));
setPreferredSize(new java.awt.Dimension(884, 563));
setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel2.setFont(new java.awt.Font("Arial", 1, 18)); // NOI18N
jLabel2.setText("ID");
add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(120, 20, 20, -1));

ID.setEditable(false);
ID.setBackground(new java.awt.Color(255, 255, 255));
ID.setColumns(20);
ID.setFont(new java.awt.Font("Arial", 0, 14));
ID.setRows(5);
ID.setAutoscrolls(false);
ID.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
jScrollPane1.setViewportViewView(ID);

// Configura el ajuste de línea para evitar el desplazamiento horizontal
```

```

ID.setLineWrap(true); // Activa el ajuste de línea
ID.setWrapStyleWord(true); // Ajusta la línea en palabras completas

// Configura el JScrollPane para desactivar la barra de desplazamiento horizontal
jScrollPane1 = new javax.swing.JScrollPane(ID);

jScrollPane1.setHorizontalScrollBarPolicy(javax.swing.ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER); // Desactiva la barra horizontal

jScrollPane1.setVerticalScrollBarPolicy(javax.swing.ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED); // Solo muestra la barra vertical si es necesario

add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 40, 240, 30));

jLabel3.setFont(new java.awt.Font("Arial", 1, 18)); // NOI18N
jLabel3.setText("Nombre de Hortaliza");
add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(810, 220, -1, 30));

NombredeHortaliza.setEditable(false);
NombredeHortaliza.setBackground(new java.awt.Color(255, 255, 255));
NombredeHortaliza.setColumns(20);
NombredeHortaliza.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
NombredeHortaliza.setLineWrap(true);
NombredeHortaliza.setRows(5);
NombredeHortaliza.setWrapStyleWord(true);
NombredeHortaliza.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
jScrollPane2.setViewportViewView(NombredeHortaliza);

```

```
add(jScrollPane2, new org.netbeans.lib.awtextra.AbsoluteConstraints(810, 250, 180, 30));
```

```
jLabel4.setFont(new java.awt.Font("Arial", 1, 18)); // NOI18N
```

```
jLabel4.setText("Temporada de siembra");
```

```
add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 80, -1, -1));
```

```
jLabel5.setFont(new java.awt.Font("Arial", 1, 18)); // NOI18N
```

```
jLabel5.setText("Temperatura optima");
```

```
add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(300, 80, -1, 30));
```

```
Temporadadesiembra.setEditable(false);
```

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

```
Temporadadesiembra.setColumns(20);
```

```
Temporadadesiembra.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
```

```
Temporadadesiembra.setLineWrap(true);
```

```
Temporadadesiembra.setRows(5);
```

```
Temporadadesiembra.setWrapStyleWord(true);
```

```
jScrollPane3.setViewportViewView(Temporadadesiembra);
```

```
add(jScrollPane3, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 120, 240, 80));
```

```
jTextArea4.setEditable(false);
```

```
jTextArea4.setBackground(new java.awt.Color(255, 255, 255));
```

```
jTextArea4.setColumns(20);
```

```
jTextArea4.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
```

```
jTextArea4.setLineWrap(true);
```

```
jTextArea4.setRows(5);
```

```
jTextArea4.setWrapStyleWord(true);  
Temperaturaoptima.setViewportView(jTextArea4);
```

```
add(Temperaturaoptima, new org.netbeans.lib.awtextra.AbsoluteConstraints(270, 120,  
240, 80));
```

```
jLabel6.setFont(new java.awt.Font("Arial", 1, 18)); // NOI18N  
jLabel6.setText("Requerimientos de riego");  
add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 210, -1, -1));
```

```
Requerimientosderiego.setEditable(false);  
Requerimientosderiego.setBackground(new java.awt.Color(255, 255, 255));  
Requerimientosderiego.setColumns(20);  
Requerimientosderiego.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N  
Requerimientosderiego.setLineWrap(true);  
Requerimientosderiego.setRows(5);  
Requerimientosderiego.setWrapStyleWord(true);  
jScrollPane5.setViewportView(Requerimientosderiego);
```

```
add(jScrollPane5, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 240, 250,  
120));
```

```
jLabel7.setFont(new java.awt.Font("Arial", 1, 18)); // NOI18N  
jLabel7.setText("Tipo de suelo ");  
add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(320, 210, -1, -1));
```

```
TipoSuelo.setEditable(false);  
TipoSuelo.setBackground(new java.awt.Color(255, 255, 255));  
TipoSuelo.setColumns(20);
```

```
TipoSuelo.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
TipoSuelo.setLineWrap(true);
TipoSuelo.setRows(5);
TipoSuelo.setWrapStyleWord(true);
jScrollPane6.setViewportViewView(TipoSuelo);

add(jScrollPane6, new org.netbeans.lib.awtextra.AbsoluteConstraints(280, 240, -1,
120));

jLabel8.setFont(new java.awt.Font("Arial", 1, 18)); // NOI18N
jLabel8.setText("Estado o Region");
add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(320, 20, -1, -1));

EstadooRegion.setEditable(false);
EstadooRegion.setBackground(new java.awt.Color(255, 255, 255));
EstadooRegion.setColumns(20);
EstadooRegion.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
EstadooRegion.setLineWrap(true);
EstadooRegion.setRows(5);
EstadooRegion.setWrapStyleWord(true);
EstadooRegion.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));
jScrollPane7.setViewportViewView(EstadooRegion);

add(jScrollPane7, new org.netbeans.lib.awtextra.AbsoluteConstraints(270, 40, 240,
30));

jLabel9.setFont(new java.awt.Font("Arial", 1, 18)); // NOI18N
jLabel9.setText("Enfermedades comunes ");
```

```
add(jLabel9, new org.netbeans.lib.awtextra.AbsoluteConstraints(560, 90, -1, -1));
```

```
Enfermedades.setEditable(false);
```

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

```
Enfermedades.setColumns(20);
```

```
Enfermedades.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
```

```
Enfermedades.setLineWrap(true);
```

```
Enfermedades.setRows(5);
```

```
Enfermedades.setWrapStyleWord(true);
```

```
jScrollPane8.setViewportViewView(Enfermedades);
```

```
add(jScrollPane8, new org.netbeans.lib.awtextra.AbsoluteConstraints(550, 120, -1, 80));
```

```
Frecuencia.setFont(new java.awt.Font("Arial", 1, 18)); // NOI18N
```

```
Frecuencia.setText("Frecuencia de fertilizacion");
```

```
add(Frecuencia, new org.netbeans.lib.awtextra.AbsoluteConstraints(550, 210, -1, -1));
```

```
jTextArea9.setEditable(false);
```

```
jTextArea9.setBackground(new java.awt.Color(255, 255, 255));
```

```
jTextArea9.setColumns(20);
```

```
jTextArea9.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
```

```
jTextArea9.setLineWrap(true);
```

```
jTextArea9.setRows(5);
```

```
jTextArea9.setWrapStyleWord(true);
```

```
jScrollPane9.setViewportViewView(jTextArea9);
```

```
add(jScrollPane9, new org.netbeans.lib.awtextra.AbsoluteConstraints(550, 240, 240, 120));
```

```
jLabel11.setFont(new java.awt.Font("Arial", 1, 18)); // NOI18N
jLabel11.setText("Espacio plantacion");
add(jLabel11, new org.netbeans.lib.awtextra.AbsoluteConstraints(580, 390, -1, -1));
```

```
Espacioplantacion.setEditable(false);
Espacioplantacion.setBackground(new java.awt.Color(255, 255, 255));
Espacioplantacion.setColumns(20);
Espacioplantacion.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
Espacioplantacion.setLineWrap(true);
Espacioplantacion.setRows(5);
Espacioplantacion.setWrapStyleWord(true);
jScrollPane10.setViewportView(Espacioplantacion);
```

```
add(jScrollPane10, new org.netbeans.lib.awtextra.AbsoluteConstraints(550, 420, 240,
150));
```

```
jLabel12.setFont(new java.awt.Font("Arial", 1, 18)); // NOI18N
jLabel12.setText("Metodo de cosecha");
add(jLabel12, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 390, -1, -1));
```

```
Metododecosecha.setEditable(false);
Metododecosecha.setBackground(new java.awt.Color(255, 255, 255));
Metododecosecha.setColumns(20);
Metododecosecha.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
Metododecosecha.setLineWrap(true);
Metododecosecha.setRows(5);
Metododecosecha.setWrapStyleWord(true);
jScrollPane11.setViewportView(Metododecosecha);
```

```
add(jScrollPane11, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 420, 250, 150));
```

```
jLabel13.setFont(new java.awt.Font("Arial", 1, 18)); // NOI18N
jLabel13.setText("Adaptabilidad climatica");
add(jLabel13, new org.netbeans.lib.awtextra.AbsoluteConstraints(290, 390, -1, -1));
```

```
Adaptabilidadclimatica.setEditable(false);
Adaptabilidadclimatica.setBackground(new java.awt.Color(255, 255, 255));
Adaptabilidadclimatica.setColumns(20);
Adaptabilidadclimatica.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N
Adaptabilidadclimatica.setLineWrap(true);
Adaptabilidadclimatica.setRows(5);
Adaptabilidadclimatica.setWrapStyleWord(true);
jScrollPane12.setViewportViewView(Adaptabilidadclimatica);
```

```
add(jScrollPane12, new org.netbeans.lib.awtextra.AbsoluteConstraints(280, 420, 240, 150));
```

```
add(lblImagenHortaliza, new org.netbeans.lib.awtextra.AbsoluteConstraints(800, 20, 190, 190));
```

```
jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Fondo.png")));
// NOI18N
```

```
add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 1000, 610));
} // </editor-fold>
```

```
// Variables declaration - do not modify
```

```
private javax.swing.JTextArea Adaptabilidadclimatica;
```

```
private javax.swing.JTextArea Enfermedades;
private javax.swing.JTextArea Espacioplantacion;
private javax.swing.JTextArea EstadooRegion;
private javax.swing.JLabel Frecuencia;
private javax.swing.JTextArea ID;
private javax.swing.JTextArea Metododecosecha;
private javax.swing.JTextArea NombredeHortaliza;
private javax.swing.JTextArea Requerimientosderiego;
private javax.swing.JScrollPane Temperaturaoptima;
private javax.swing.JTextArea Temporadadesiembra;
private javax.swing.JTextArea TipoSuelo;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
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.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane10;
private javax.swing.JScrollPane jScrollPane11;
private javax.swing.JScrollPane jScrollPane12;
private javax.swing.JScrollPane jScrollPane2;
```

```
private javax.swing.JScrollPane jScrollPane3;  
private javax.swing.JScrollPane jScrollPane5;  
private javax.swing.JScrollPane jScrollPane6;  
private javax.swing.JScrollPane jScrollPane7;  
private javax.swing.JScrollPane jScrollPane8;  
private javax.swing.JScrollPane jScrollPane9;  
private javax.swing.JTextArea jTextArea4;  
private javax.swing.JTextArea jTextArea9;  
private javax.swing.JLabel lblImagenHortaliza;  
// End of variables declaration  
}
```