



UNIVERSIDAD TÉCNICA DE COTOPAXI
FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS
CARRERA DE SISTEMAS DE INFORMACIÓN

**“SISTEMA WEB PARA LA GESTIÓN DE ACOPIO, CLASIFICACIÓN Y LIQUIDACIÓN
DE PRODUCTORES EN LA EMPRESA L.Y FLOWERS, CON EL FIN DE OPTIMIZAR
EL CONTROL Y SEGUIMIENTO DE LOS PROCESOS.”**

PROPUESTA TECNOLÓGICA PREVIA A LA OBTENCIÓN DEL
TÍTULO DE
INGENIEROS EN SISTEMAS DE INFORMACIÓN

AUTORES:

Evelin Paola Ganchala Yanchatipan

Leonardo David Quinaluisa Logaña

TUTOR:

Ing. Edison Patricio Bedón Salazar, Mg.

LATACUNGA, AGOSTO DE 2026

Latacunga, Agosto del 2026

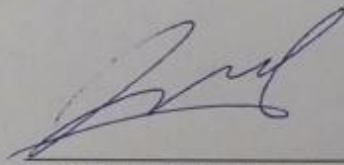
DECLARACIÓN DE AUTORÍA

Nosotros EVELIN PAOLA GANCHALA YANCHATIPAN, con cédula de ciudadanía No. 0503732463 y LEONARDO DAVID QUINALUISA LOGAÑA con cédula de ciudadanía No.1721266581 declaramos ser autores del proyecto de titulación **“SISTEMA WEB PARA LA GESTIÓN DE ACOPIO, CLASIFICACIÓN Y LIQUIDACIÓN DE PRODUCTORES EN LA EMPRESA L.Y FLOWERS, CON EL FIN DE OPTIMIZAR EL CONTROL Y SEGUIMIENTO DE LOS PROCESOS.”**, siendo el Ing. Edison Patricio Bedón Salazar tutor del presente trabajo de titulación; y eximo expresamente a la Universidad Técnica de Cotopaxi y a sus representantes legales de posibles reclamos o acciones legales.

Además, certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo de titulación, son de mi exclusiva responsabilidad.



Evelin Paola Ganchala Yanchatipan
CC.0503732463

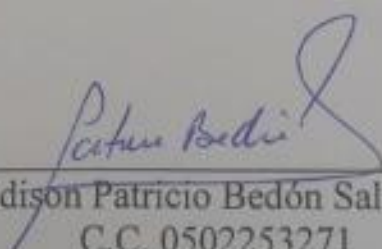


Leonardo David Quinaluisa Logaña
CC. 1721266581

Latacunga, 6 de Agosto del 2026

AVAL DEL TUTOR DE LA PROPUESTA TECNOLÓGICA

En calidad de Tutor de la **PROPUESTA TECNOLÓGICA** sobre el título: “**SISTEMA WEB PARA LA GESTIÓN DE ACOPIO, CLASIFICACIÓN Y LIQUIDACIÓN DE PRODUCTORES EN LA EMPRESA L.Y. FLOWERS, CON EL FIN DE OPTIMIZAR EL CONTROL Y SEGUIMIENTO DE LOS PROCESOS**”, propuesto por los estudiantes **GANCHALA YANCHATIPAN EVELIN PAOLA** y **QUINALUISA LOGAÑA LEONARDO** de la Carrera de Sistemas de información, considero que dicho proyecto de titulación cumple con los requerimientos metodológicos y aportes científico-técnicos suficientes para ser sometidos al tribunal de lectores.



Ing. Edison Patricio Bedón Salazar, Mg.
C.C. 0502253271
TUTOR

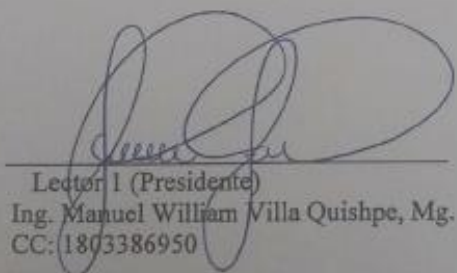
Latacunga, Agosto del 2026

AVAL DE APROBACIÓN DE LECTORES

Cumpliendo con el Reglamento de Titulación de la Universidad Técnica de Cotopaxi, en calidad de Lectores de Tribunal de Proyecto de Investigación con el Título “SISTEMA WEB PARA LA GESTIÓN DE ACOPIO, CLASIFICACIÓN Y LIQUIDACIÓN DE PRODUCTORES EN LA EMPRESA L.Y FLOWERS, CON EL FIN DE OPTIMIZAR EL CONTROL Y SEGUIMIENTO DE LOS PROCESOS.”, propuesto por los estudiantes Evelin Paola Ganchala Yanchatipan, con cédula de ciudadanía No. 0503732463 y Leonardo David Quinaluisa Logaña con cédula de ciudadanía No.1721266581 de la Carrera de Sistemas de Información, me permito indicar que el o la estudiante ha concluido todas las observaciones y realizado las correcciones señaladas por el Tribunal de Lectores, además de validar el funcionamiento de la propuesta tecnológica, por lo cual presentamos el Aval de aprobación del Proyecto de Titulación correspondiente a la modalidad Propuesta Tecnológica en virtud de lo cual el o la postulante puede presentarse a la Defensa de su Proyecto de Titulación.

Particular que pongo en su conocimiento para los fines legales pertinentes.

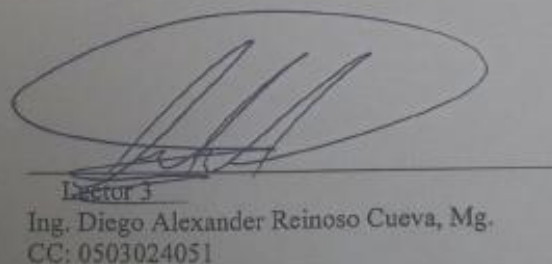
Atentamente,



Lector 1 (Presidente)
Ing. Manuel William Villa Quishpe, Mg.
CC: 1803386950



Lector 2
Ing. Diego Geovanny Falconí Punguil, Mg.
CC: 0550080774



Lector 3
Ing. Diego Alexander Reinoso Cueva, Mg.
CC: 0503024051



Latacunga, 7 de Agosto del 2026

AVAL DE IMPLEMENTACIÓN

Por medio del presente, yo, Luis Pedro Yanchatipan Añarumba, con cédula de ciudadanía N.º 0502729908, en calidad de propietario de la empresa L.Y Flowers, hago constar que los estudiantes Evelin Paola Ganchala Yanchatipan, con cédula N.º 0503732463, y Leonardo David Quinaluisa Logaña, con cédula N.º 1721266581, desarrollaron e implementaron en la empresa el trabajo de titulación denominado:

“SISTEMA WEB PARA LA GESTIÓN DE ACOPIO, CLASIFICACIÓN Y LIQUIDACIÓN DE PRODUCTORES EN LA EMPRESA L.YFLOWERS, CON EL FIN DE OPTIMIZAR EL CONTROL Y SEGUIMIENTO DE LOS PROCESOS.”

El sistema fue presentado, probado e implementado satisfactoriamente en el establecimiento, permitiendo apoyar la gestión de acopio, clasificación y liquidación de productores, así como optimizar el control y seguimiento de los procesos de la empresa.

El presente aval se expide a petición de los interesados para los fines académicos y legales que correspondan.

ESTUDIANTE	CORREO INSTITUCIONAL	TELÉFONO
Evelin Paola Ganchala Yanchatipan	evelin.ganchala2463@utc.edu.ec	0979111576
Leonardo David Quinaluisa Logaña	leonardo.quinaluisa6581@utc.edu.ec	0998105211

Luis Pedro Yanchatipan Añarumba
PROPIETARIO
 L.Y Flowers
 C.I. 0502729908

Latacunga Barrio Patután
 Teléfono: 032 270 086
 Celular: 0968100308
 Email: lyflowers@hotmail.com

AGRADECIMIENTO

En primer lugar, agradezco profundamente a Dios por guiarme durante este camino, por concederme salud, sabiduría y fortaleza para superar cada dificultad y permitirme culminar esta importante etapa de mi vida.

A mis padres, Carmen Yanchatipan y Geovany Ganchala, por su amor, apoyo incondicional, consejos y por confiar siempre en mí. Gracias por acompañarme en cada paso de este proceso, por motivarme a seguir adelante en los momentos difíciles y por enseñarme que con esfuerzo y perseverancia es posible alcanzar cada meta.

A mi hermana, Lisbeth Ganchala, por su cariño, apoyo y por hacer que este camino fuera más llevadero con sus risas y compañía. Gracias por estar presente en los momentos buenos y difíciles.

A mis abuelitos, tíos y primos, por su cariño, comprensión y palabras de aliento, que de una u otra manera me motivaron a continuar y alcanzar esta meta.

De manera especial, agradezco al Ing. Patricio Bedón, docente tutor de la propuesta tecnológica, por su orientación, conocimientos y acompañamiento durante el desarrollo de este trabajo. Sus recomendaciones fueron fundamentales para culminar este proceso.

A mi compañero de tesis y pareja, David Quinaluisa, por compartir conmigo esta etapa universitaria y el desarrollo de este trabajo. Gracias por tu paciencia, comprensión, cariño y apoyo constante, y por estar a mi lado en cada desafío.

Finalmente, agradezco a todas las personas que, de una u otra manera, contribuyeron a mi formación y estuvieron presentes durante este camino. Este logro es el resultado de esfuerzo, sacrificios y aprendizajes, pero también del apoyo y cariño de quienes me acompañaron hasta alcanzar esta meta.

Evelin Ganchala

DEDICATORIA

Dedico este trabajo, con profundo amor y gratitud, a Dios, por guiar cada uno de mis pasos, concederme salud y brindarme la sabiduría y fortaleza necesarias para superar las dificultades y culminar esta importante etapa de mi vida.

A mis padres, Carmen Yanchatipán y Geovany Ganchala, por ser pilares fundamentales de mi vida, por su amor incondicional, sus consejos, sacrificios y por creer siempre en mí. Gracias por acompañarme en cada paso de este camino y por darme la fuerza necesaria para alcanzar esta meta. Este logro también es suyo.

A mi hermana, Lisbeth Ganchala, por su cariño, apoyo y por hacer que este camino fuera más llevadero con sus risas y compañía.

A mis abuelitos, tíos y primos, por su cariño, comprensión y por estar presentes de una u otra manera durante este proceso. Gracias por creer en mí y motivarme a seguir adelante.

A mi compañero de tesis y pareja, David Quinaluisa, por compartir conmigo esta trayectoria universitaria y este importante logro. Gracias por tu cariño, paciencia, comprensión y por estar a mi lado en cada desafío. Tu compañía hizo de esta etapa una experiencia única y especial.

A mis amigos, por su amistad, compañía, risas y por todos los momentos inolvidables que hicieron de estos años una etapa más significativa.

Finalmente, dedico este logro a mis queridos animalitos, especialmente a Keyla y Esmeralda, por su cariño y compañía en los días buenos y difíciles; y a mi querido patito Lucas, quien, aunque ya no está físicamente conmigo, dejó recuerdos hermosos que siempre llevaré en mi corazón.

Con amor y profunda gratitud, dedico este trabajo a todos ustedes, quienes fueron parte importante de este camino y de la persona que hoy soy.

Evelin Ganchala

DEDICATORIA

Dedico este trabajo, con profundo amor y gratitud, a Dios, por acompañarme en cada paso de este camino, por concederme vida, salud y fortaleza, y por brindarme la sabiduría necesaria para superar las dificultades y alcanzar esta importante meta.

*A mis padres, **Rosa Logaña** y **Patricio Quinaluisa**, pilares fundamentales de mi vida y mi mayor ejemplo de esfuerzo, responsabilidad y perseverancia. Gracias por su amor, paciencia, confianza y apoyo incondicional, por creer siempre en mí y enseñarme que con esfuerzo y constancia es posible alcanzar cada meta. A mis hermanos, **Christian Quinaluisa** y **Ricardo Quinaluisa**, gracias por su apoyo, compañía y por todos los momentos compartidos; y a mi sobrina, **Sofía Quinaluisa Casa**, y a mi cuñada, **Tatiana Casa**, gracias por formar parte de mi familia y llenar de alegría tantos momentos. A mi querida abuelita **Rosa Aimacaña**, que en paz descansa, le dedico también este logro, pues su recuerdo y el cariño que siempre le tuve permanecen en mi corazón.*

*A mi pareja y compañera de tesis, **Evelin Ganchala**, con quien tuve la dicha de compartir no solo este trabajo, sino también una parte muy importante de mi trayectoria universitaria. Gracias por tu amor, paciencia, comprensión y apoyo constante; por acompañarme en los momentos difíciles, compartir nuestros aprendizajes y celebrar cada avance hasta alcanzar esta meta. Sin duda, vivir esta etapa contigo hizo que este camino fuera mucho más especial y significativo.*

*A todas las personas que estuvieron presentes durante este proceso y que, de una u otra manera, aportaron con una palabra, un consejo, una sonrisa o simplemente con su compañía. Hoy, al mirar atrás, recuerdo una frase de Andy Bernard en The Office: “**Ojalá hubiera una forma de saber que estás en los buenos viejos tiempos antes de que realmente te hayas ido de ellos**”. Tal vez durante estos años no siempre fui consciente de que estaba viviendo momentos que algún día extrañaría, pero hoy comprendo que cada clase, cada trabajo, cada dificultad, cada risa y cada momento compartido forman parte de una etapa que siempre recordaré.*

DAVID QUINALUISA

AGRADECIMIENTO

En primer lugar, agradezco profundamente a Dios por la vida, la salud, mi familia y la oportunidad de llegar hasta este momento. Gracias por brindarme fortaleza, sabiduría y paciencia para afrontar cada desafío y permitirme culminar una etapa tan importante de mi formación profesional.

Expreso mi sincero agradecimiento a la Universidad Técnica de Cotopaxi y a todos los docentes que formaron parte de mi trayectoria universitaria, por sus conocimientos y enseñanzas, que contribuyeron a mi formación académica y profesional. De manera especial, agradezco al Ing. Patricio Bedón, tutor de la propuesta tecnológica, por su orientación y acompañamiento durante el desarrollo de este trabajo. Sus recomendaciones fueron fundamentales para culminar satisfactoriamente esta propuesta.

*A mis padres, **Rosa Logaña** y **Patricio Quinaluisa**, mi más profundo agradecimiento, porque este logro también es de ustedes. Gracias por su amor, confianza y apoyo incondicional, por cada sacrificio y palabra de ánimo, y por enseñarme con su ejemplo el valor del esfuerzo y la perseverancia. A pesar de las dificultades, siempre estuvieron a mi lado, creyendo en mí y motivándome a seguir adelante. Todo lo que han hecho por mí fue fundamental para llegar hasta aquí, y este logro es también una forma de agradecerles por nunca dejar de creer en mí. A mis hermanos, **Christian Quinaluisa** y **Ricardo Quinaluisa**, gracias por su apoyo y compañía; y a mi sobrina, **Sofía Quinaluisa Casa**, gracias por formar parte de mi vida y aportar momentos de alegría durante esta etapa.*

*A mi pareja y compañera de tesis, **Evelin Ganchala**, gracias por compartir conmigo esta etapa tan importante. Gracias por tu amor, paciencia y apoyo constante, por estar a mi lado en los momentos difíciles y compartir cada avance y logro. Haber recorrido juntos este camino hizo que esta experiencia fuera mucho más significativa.*

Finalmente, agradezco a todas las personas que, de una u otra manera, fueron parte de este camino. Este logro representa el resultado de mi esfuerzo y dedicación, pero también refleja el apoyo, la confianza y el cariño de quienes estuvieron conmigo durante esta etapa. A todos ustedes, muchas gracias.

DAVID QUINALUISA

RESUMEN

El presente proyecto tuvo como objetivo desarrollar un sistema informático para optimizar los procesos de recepción, clasificación y liquidación de cuentas con los proveedores de L.Y. Flowers, empresa ubicada en la parroquia Eloy Alfaro, cantón Latacunga. Anteriormente, la organización utilizaba registros físicos y hojas de cálculo, lo que generaba duplicación de información, errores en el cálculo de pagos, pérdida de documentos, dificultades para consultar datos históricos y demoras en la elaboración de reportes.

Ante esta situación, se diseñó una plataforma web utilizando Python y Django como tecnologías principales, PostgreSQL como motor de base de datos y HTML, CSS, JavaScript y Bootstrap para el desarrollo de la interfaz de usuario. Finalmente, la aplicación fue alojada en Railway. El proceso de desarrollo siguió los lineamientos de la metodología Extreme Programming (XP), mediante historias de usuario e iteraciones sucesivas con retroalimentación constante.

Para el diagnóstico inicial, se combinó la investigación documental con el trabajo de campo, que incluyó la observación directa y una encuesta aplicada a las nueve personas involucradas en estos procesos. Debido al tamaño de la población, se realizó un censo completo.

Como resultado, se obtuvo un sistema integral que permite gestionar usuarios, personal, productores, fincas, variedades de rosas, recepciones, clasificaciones, precios, liquidaciones y niveles de inventario. Además, incorpora control de acceso, generación de reportes y auditoría, con el fin de garantizar la trazabilidad y la responsabilidad en cada operación.

La validación se realizó mediante pruebas unitarias, funcionales y de aceptación, las cuales confirmaron el funcionamiento correcto de los flujos críticos, los cálculos y las restricciones correspondientes a cada perfil de usuario. En conclusión, el sistema permitió centralizar y automatizar las operaciones principales de la empresa, mejorando el control, la organización y la disponibilidad de la información durante todo el proceso, desde la recepción de las flores hasta la liquidación final con los productores.

Palabras clave: Sistema web, recepción de flores, clasificación, pago a productores, trazabilidad, Django, PostgreSQL, metodología XP.

ABSTRACT

TECHNICAL UNIVERSITY OF COTOPAXI

FACULTY OF ENGINEERING AND APPLIED SCIENCES

Theme: “Web-Based System for the Management of Collection, Sorting and Payment of Producers at L.Y Flowers, with the aim of Optimising the Control and Monitoring of Processes.”

Authors:

Ganchala Yanchatipan Evelin Paola
Quinaluisa Logaña Leonardo David

ABSTRACT

This project aimed to create a computerised system designed to optimise the processes of receiving, sorting and settling accounts with suppliers at L.Y. Flowers, a company located in the parish of Eloy Alfaro, Latacunga canton. Previously, the organisation relied on physical records and spreadsheets, which led to duplication of information, errors in payment calculations, lost documents, difficulties in retrieving past data and delays in producing reports. In response to this situation, a web platform was designed using Python and Django as the core technologies, with PostgreSQL as the database engine, whilst the front-end was developed using HTML, CSS, JavaScript and Bootstrap, with the application ultimately hosted on Railway. The development process followed the guidelines of Extreme Programming (XP), structuring the work into user stories and successive iterations with constant feedback. For the initial assessment, desk research was combined with fieldwork, including direct observation and a survey conducted amongst all nine people involved in these processes, as a comprehensive census of the population was carried out. The end result was a comprehensive system covering the management of users, staff, growers, farms, rose varieties, receipts, classifications, pricing, settlements and stock levels, whilst also incorporating access control, report generation and auditing to ensure traceability and accountability in every operation. Validation was carried out through unit, functional and acceptance tests, which confirmed the correct functioning of critical workflows, calculations and restrictions by user profile. In summary, the system succeeded in centralising and automating the company’s key operations, improving control, organisation and the availability of information throughout the entire cycle, from the receipt of flowers to the final settlement with producers.

Keywords: Web System, Flower Collection, Sorting, Payment to Producers, Traceability, Django, PostgreSQL, XP Methodology.

ÍNDICE GENERAL

CONTENIDO

1. INFORMACIÓN GENERAL	1
2. INTRODUCCIÓN	3
2.1. SITUACIÓN PROBLEMÁTICA	4
2.2. FORMULACIÓN DEL PROBLEMA	5
2.3. OBJETO Y CAMPO DE ACCIÓN	5
2.3.1. Objeto de investigación	5
2.3.2. Campo de acción	5
2.4. BENEFICIARIOS	5
2.4.1. Directos	5
2.4.2. Indirectos	6
2.5. JUSTIFICACIÓN	7
2.6. OBJETIVOS	8
2.6.1. Objetivo general	8
2.6.2. Objetivos específicos	8
2.7. TAREAS POR OBJETIVO	8
3. MARCO TEÓRICO	9
3.1. GESTIÓN DEL ACOPIO, CLASIFICACIÓN Y LIQUIDACIÓN DE PROVEEDORES	11
3.1.1. Acopio y recepción de flores	11
3.1.2. Clasificación de flores	11
3.1.3. Liquidación de proveedores	12
3.1.4. Gestión administrativa	12
3.1.5. Gestión de inventarios	13
3.1.6. Control de inventarios	13
3.1.7. Importancia de control de inventarios	13
3.1.8. Automatización de procesos	13

3.1.9. Automatización de procesos empresariales	14
3.1.10. Beneficios de la automatización	14
3.2. SISTEMA WEB.....	14
3.2.1. Herramientas de desarrollo web.....	15
3.2.2. Arquitectura	24
3.3. METODOLOGÍAS DE DESARROLLO DE SOFTWARE	25
3.3.1. Metodologías ágiles frente a metodologías tradicionales	26
3.3.2. Comparación entre metodologías ágiles	27
3.3.1. Programación extrema (XP).....	28
4. MÉTODOS Y PROCEDIMIENTOS.....	31
4.1. DISEÑO METODOLÓGICO DE LA INVESTIGACIÓN	31
4.1.1. Enfoque de investigación	31
4.1.2. Tipos de investigación	32
4.1.3. Métodos de investigación.....	33
4.1.4. Población y muestra	33
4.2. TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE INFORMACIÓN	33
4.2.1. Revisión bibliográfica	33
4.2.2. Encuesta	34
4.3. METODOLOGÍA DE DESARROLLO DE SOFTWARE XP	34
4.3.1. Fase de planificación.....	34
4.3.2. Fase de diseño	36
4.3.3. Fase de codificación.....	37
4.3.4. Fase de pruebas	37
4.3.5. Fase de implementación y despliegue.....	38
5. ANÁLISIS Y DISCUSIÓN DE RESULTADOS.....	39
5.1. RESULTADOS DE LA ENTREVISTA	39
5.2. RESULTADOS DE LA ENCUESTA	41

5.2.1. Pregunta 1: ¿Con qué frecuencia registra o consulta información relacionada con recepción, clasificación o liquidación de flores?	42
5.2.2. Pregunta 2: ¿Qué medio se utilizaba principalmente para registrar la información antes del Sistema?	42
5.2.3. Pregunta 3: ¿Con qué frecuencia se presentaban dificultades para localizar registros anteriores?	43
5.2.4. Pregunta 4: ¿Cuál era el principal problema del proceso manual?	44
5.2.5. Pregunta 5: ¿Considera necesario relacionar cada recepción con productor, variedad y responsable?	45
5.2.6. Pregunta 6: ¿La automatización de liquidaciones reduciría tiempo y errores?.....	46
5.2.7. Pregunta 7: ¿Qué tan importante es disponer de reportes y trazabilidad por fecha y usuario?	47
5.2.8. Pregunta 8: ¿Considera que FlorLY mejorará el control y seguimiento de los procesos?	48
5.3. RESULTADOS DE LA APLICACIÓN DE LA METODOLOGÍA XP	49
5.3.1. Resultados de la fase de planificación	49
5.3.2. Resultados de la fase de diseño.....	56
5.3.3. Resultados de la fase de codificación	69
5.3.4. Resultados de la fase de pruebas.....	72
5.3.5. Resultados de la fase de implementación y despliegue	81
5.4. COSTO ESTIMADO DEL SOFTWARE.....	89
5.5. RESPUESTA A LA PREGUNTA DE INVESTIGACIÓN	91
6. CONCLUSIONES Y RECOMENDACIONES.....	92
6.1. CONCLUSIONES	92
6.2. RECOMENDACIONES	93
7. REFERENCIAS BIBLIOGRÁFICAS.....	93
8. ANEXOS	99
ROLES Y PERMISOS.....	146
Panel según el rol	146
Recuperar el acceso.....	148
Cerrar sesión	148

ELEMENTOS GENERALES DE LA INTERFAZ.....	148
PANEL PRINCIPAL	148
ADMINISTRACIÓN DE USUARIOS Y PERSONAL	149
Gestionar usuarios.....	149
Gestionar personal.....	149
REGISTRO DE RECEPCIONES	150
Consultar recepciones	150
Registrar una recepción.....	151
CLASIFICACIÓN DE FLORES	152
Consultar clasificaciones.....	152
Clasificar un lote	152
LIQUIDACIONES.....	153
INVENTARIO	154
Crear y consultar productos	154
Registrar movimientos	155
REPORTES.....	155
AUDITORÍA	156
RECOMENDACIONES DE USO Y SOLUCIÓN DE PROBLEMAS	157
Buenas prácticas.....	157
Problemas frecuentes	158
Soporte	158

ÍNDICE DE FIGURAS

Figura 3.1. Arquitectura cliente servidor [10].....	14
Figura 3.2. Lenguaje de programación Python [21]	16
Figura 3.3. Framework Django [27]	18
Figura 3.4. Sistema gestor de base de datos PostgreSQL [32]	20
Figura 3.5. Plataforma de despliegue Railway [46].....	24
Figura 3.6. Arquitectura Model View Template.....	25
Figura 3.7. Fases de la metodología XP [?]	29
Figura 3.8. Valores de Extreme Programming (XP).....	30
Figura 3.9. Principios de Extreme Programming (XP).....	31
Figura 5.1. Resultados de la pregunta 1	45
Figura 5.2. Resultados de la pregunta 2	46
Figura 5.3. Resultados de la pregunta 3	47
Figura 5.4. Resultados de la pregunta 4	48
Figura 5.5. Resultados de la pregunta 5	49
Figura 5.6. Resultados de la pregunta 6	50
Figura 5.7. Resultados de la pregunta 7	51
Figura 5.8. Resultados de la pregunta 8	52
Figura 5.9. Diagrama de casos de uso acceso al Sistema	60
Figura 5.10. Diagrama de caso de uso de la vista general del Sistema FlorLY.....	60
Figura 5.11. Diagrama de caso de uso de gestión de proveedores, fincas y variedades	61
Figura 5.12. Diagrama de caso de uso del proceso de recepción de flores.....	62
Figura 5.13. Diagrama de caso de uso de clasificación de flores	62
Figura 5.14. Diagrama de caso de uso de tarifarios y liquidaciones.....	63
Figura 5.15. Diagrama de caso de uso de gestión del personal	64
Figura 5.16. Diagrama de caso de uso de inventario de productos químicos	64
Figura 5.17. Diagrama de caso de uso de reportes, seguimiento y auditoría.....	65
Figura 5.18. Modelado de la base de datos	66
Figura 5.19. Arquitectura MVT del Sistema FlorLY.....	68
Figura 5.20. Diagrama BPMN - Postcosecha	69
Figura 5.21. Diagrama BPMN - Recepción de flor	70
Figura 5.22. Diagrama BPMN - Clasificación de flor	71
Figura 5.23. Diagrama BPMN - Proveedores, fincas y variedades	72
Figura 5.24. Diagrama BPMN - Tarifas por variedad	73

Figura 5.25. Diagrama BPMN - Liquidación de productores.....	74
Figura 5.26. Diagrama BPMN - Gestión del personal.....	75
Figura 5.27. Diagrama BPMN - Inventario de químicos.....	76
Figura 5.28. Diagrama BPMN - Reportes de producción.....	77
Figura 5.29. Diagrama BPMN - Trazabilidad y auditoría	78
Figura 5.30. interfaz de inicio de sesión	91
Figura 5.31. interfaz del panel principal	92
Figura 5.32. interfaz de gestión de usuarios.....	92
Figura 5.33. interfaz de gestión del personal	93
Figura 5.34. interfaz de recepciones de flor.....	94
Figura 5.35. interfaz de registro de recepción.....	94
Figura 5.36. interfaz de clasificaciones de flor	95
Figura 5.37. interfaz de registro de clasificación	96
Figura 5.38. interfaz de liquidaciones de productores	96
Figura 5.39. interfaz de inventario de productos químicos.....	97
Figura 5.40. interfaz del reporte de producción	98
Figura 5.41. configuración de la conexión con PostgreSQL	99
Figura 5.42. configuración del servicio mediante railway.json	100
Figura 5.43. publicación automática del Sistema en Railway	100

ÍNDICE DE TABLAS

Tabla 2.1. Beneficiarios Directos.....	5
Tabla 2.2. Beneficiarios Indirectos	5
Tabla 2.3. Planificación de actividades.....	8
Tabla 3.1. Comparación entre lenguajes de programación	15
Tabla 3.2. Comparación entre el Framework Django y otros Frameworks	17
Tabla 3.3. Comparación entre Sistemas gestores de bases de datos	18
Tabla 3.4. Tecnologías utilizadas en la interfaz Web	21
Tabla 3.5. Arquitectura en 3 capas.....	25
Tabla 3.6. Comparación entre metodologías tradicionales y ágiles.....	27
Tabla 3.7. Comparación entre metodologías ágiles	28
Tabla 3.8. Etapas de la metodología XP	29
Tabla 4.1. Diseño de las historias de usuario	36
Tabla 4.2. Diseño del Product Backlog.....	37
Tabla 4.3. Diseño del Iteration Backlog	37
Tabla 4.4. Diseño de los casos de prueba	39
Tabla 5.1. Matriz de análisis de contenido.....	41
Tabla 5.2. Resultados sintetizados de la entrevista	43
Tabla 5.3. Distribución de respuestas de la pregunta 1.....	44
Tabla 5.4. Distribución de respuestas de la pregunta 2.....	45
Tabla 5.5. Distribución de respuestas de la pregunta 3.....	46
Tabla 5.6. Distribución de respuestas de la pregunta 4.....	47
Tabla 5.7. Distribución de respuestas de la pregunta 5.....	48
Tabla 5.8. Distribución de respuestas de la pregunta 6.....	49
Tabla 5.9. Distribución de respuestas de la pregunta 7.....	50
Tabla 5.10. Distribución de respuestas de la pregunta 8.....	51
Tabla 5.11. HU001: Gestión de usuarios	53
Tabla 5.12. HU002: Asignación de roles y permisos	53
Tabla 5.13. HU003: Consulta del panel principal.....	53
Tabla 5.14. Roles de XP.....	54
Tabla 5.15. Product Backlog.....	55
Tabla 5.16. Iteración 1	58
Tabla 5.17. Iteración 2	58
Tabla 5.18. Iteración 3	59

Tabla 5.19. Codificación HU001	79
Tabla 5.20. Codificación HU002	80
Tabla 5.21. Caso de prueba HU001	81
Tabla 5.22. Caso de prueba HU008	82
Tabla 5.23. Caso de prueba HU011	83
Tabla 5.24. Caso de prueba HU016	84
Tabla 5.25. Caso de prueba HU023	85
Tabla 5.26. Caso de prueba HU025	86
Tabla 5.27. Caso de prueba HU020	87
Tabla 5.28. Caso de prueba HU029	88
Tabla 5.29. Herramientas de desarrollo e implementación.....	90
Tabla 5.30. Equipos de hardware.....	90
Tabla 5.31. Cálculo por hora del programador	101
Tabla 5.32. Puntos de los módulos	101
Tabla 5.33. Costo total del software	102
Tabla 5.34. Evidencias de respuesta a la pregunta de investigación.....	103
Tabla A.1. HU004: Registro de proveedores	109
Tabla A.2. HU005: Administración de proveedores.....	110
Tabla A.3. HU006: Gestión de fincas	110
Tabla A.4. HU007: Gestión de variedades	111
Tabla A.5. HU008: Registro de personal	112
Tabla A.6. HU009: Gestión de permisos y vacaciones.....	112
Tabla A.7. HU010: Consulta del historial laboral	113
Tabla A.8. HU011: Registro de recepción de flor de proveedor	113
Tabla A.9. HU012: Registro de recepción de flor propia	114
Tabla A.10. HU013: Registro de múltiples variedades por recepción.....	115
Tabla A.11. HU014: Generación del ticket de recepción	115
Tabla A.12. HU015: Consulta de recepciones pendientes.....	116
Tabla A.13. HU016: Registro de clasificación exportable	117
Tabla A.14. HU017: Registro de tallos adicionales.....	117
Tabla A.15. HU018: Registro de flor nacional o rechazada	118
Tabla A.16. HU019: Clasificación en fecha posterior	118
Tabla A.17. HU020: Reporte diario de postcosecha.....	119
Tabla A.18. HU021: Registro de tarifarios	120

Tabla A.19. HU022: Conservación del historial de tarifas	120
Tabla A.20. HU023: Generación automática de liquidaciones.....	121
Tabla A.21. HU024: Consulta del detalle de liquidación	122
Tabla A.22. HU025: Gestión de categorías y productos químicos.....	122
Tabla A.23. HU026: Registro de entradas de inventario	123
Tabla A.24. HU027: Registro de salidas y consumos parciales	124
Tabla A.25. HU028: Alertas de stock mínimo.....	124
Tabla A.26. HU029: Consulta de reportes y bitácora	125
Tabla B.1. Iteración 4.....	126
Tabla B.2. Iteración 5.....	126
Tabla B.3. Iteración 6.....	126
Tabla B.4. Iteración 7.....	127
Tabla B.5. Iteración 8.....	127
Tabla B.6. Iteración 9.....	127
Tabla B.7. Iteración 10.....	128
Tabla C.1. Codificación HU003	129
Tabla C.2. Codificación HU004	130
Tabla C.3. Codificación HU005	131
Tabla C.4. Codificación HU006	132
Tabla C.5. Codificación HU007	133
Tabla C.6. Codificación HU008	134
Tabla C.7. Codificación HU009	135
Tabla C.8. Codificación HU010	136
Tabla C.9. Codificación HU011	137
Tabla C.10. Codificación HU012	138
Tabla C.11. Codificación HU013	139
Tabla C.12. Codificación HU014	140
Tabla C.13. Codificación HU015	141
Tabla C.14. Codificación HU016	142
Tabla C.15. Codificación HU017	143
Tabla C.16. Codificación HU018	144
Tabla C.17. Codificación HU019	145
Tabla C.18. Codificación HU020	146
Tabla C.19. Codificación HU021	147

Tabla C.20. Codificación HU022	148
Tabla C.21. Codificación HU023	149
Tabla C.22. Codificación HU024	150
Tabla C.23. Codificación HU025	151
Tabla C.24. Codificación HU026	152
Tabla C.25. Codificación HU027	153
Tabla C.26. Codificación HU028	154
Tabla C.27. Codificación HU029	155

1. INFORMACIÓN GENERAL

Tema del proyecto:

SISTEMA WEB PARA LA GESTIÓN DE ACOPIO, CLASIFICACIÓN Y LIQUIDACIÓN DE PRODUCTORES EN LA EMPRESA L.YFLOWERS, CON EL FIN DE OPTIMIZAR EL CONTROL Y SEGUIMIENTO DE LOS PROCESOS.

Modalidad de Titulación:

Propuesta Tecnológica

Fecha de Inicio:

Abril 2026

Fecha de Finalización:

Agosto 2026

Lugar de Ejecución: Empresa L.Y

Flowers

Facultad:

Ciencias de la Ingeniería y Aplicadas

Carrera:

Sistemas de Información

Trabajo de Titulación Vinculado al Proyecto:

No aplica

Equipo de Trabajo del Trabajo de Titulación:

TUTOR:

Nombres: MSc. Edison Patricio Bedón Salazar

Cedula: 0502253271

Correo: edison.bedon@utc.edu.ec

Teléfono: 0984257262

Estudiantes:

Nombres: Evelin Paola Ganchala Yanchatipan

Cedula: 0503732463

Correo: evelin.ganchala2463@utc.edu.ec

Teléfono: 0979111576

Nombres: Leonardo David Quinaluisa Logaña

Cedula: 1721266581

Correo: leonardo.quinaluisa6581@utc.edu.ec

Teléfono: 0998105211

Área de Conocimiento: 330000 Ciencias Tecnológicas

La Tabla 1.1 presenta la clasificación del trabajo conforme a la Nomenclatura Internacional de la UNESCO para los campos de ciencia y tecnología, resaltando en negrita la línea específica a la que se alinea la propuesta.

Tabla 1.1. Áreas Conocimientos Unesco.

No se encuentran elementos de tabla de ilustraciones.	061 Información y Comunicación (TIC)	0611 El uso del ordenador
		0612 Base de datos, diseño y administración de redes
		0613 Software y desarrollo y análisis de aplicativos

Línea de investigación: Tecnología de la información y las comunicaciones, robótica, automatización y optimización de sistemas.

Sublíneas de investigación de la Carrera: Ciencias informáticas para la modelación de Sistemas de información a través del desarrollo de software.

2. INTRODUCCIÓN

El sector florícola constituye una de las principales actividades productivas y de exportación no tradicional en el Ecuador, debido a su importante aporte a la economía nacional, la generación de empleo y el fortalecimiento del comercio. En este ámbito la producción y comercialización de flores, especialmente de rosas, exige el cumplimiento de procesos organizados que garanticen la calidad del producto y la adecuada gestión de la información generada en cada etapa desde el acopio hasta la liquidación de los productores [1].

En los últimos años, la transformación digital ha impulsado la incorporación de Sistemas de información en empresas del sector agrícola y florícola con el propósito de optimizar sus procesos operativos y administrativos. Estas herramientas permiten registrar, consultar y controlar información de manera centralizada, mejorar la trazabilidad de las operaciones, reduce errores ocasionados por registros manuales y proporciona datos confiables para respaldar la toma de decisiones [2].

En la provincia de Cotopaxi, una de las principales zonas productoras de rosas del país, numerosas empresas del sector, trabajan diariamente con productores independientes que abastecen de materia prima para su posterior clasificación y venta [3], [4]. Debido al incremento de la producción y al volumen de información generado, resulta indispensable contar con herramientas tecnológicas que permitan administrar eficientemente los procesos de recepción, clasificación y liquidación de los productores, garantizando un adecuado control y seguimiento de cada operación realizada.

La empresa L.Y Flowers, ubicada en la parroquia Eloy Alfaro (San Felipe), cantón Latacunga, desarrolla actividades enfocadas en el acopio, clasificación y liquidación de productores de rosas. Sin embargo, actualmente gran parte de estos procesos se realiza mediante formularios físicos y hojas de cálculo, situación que dificulta la organización de la información y limita el seguimiento oportuno de las operaciones internas. Esta dependencia de métodos tradicionales incrementa la probabilidad de cometer errores en los registros de recepción y tarifarios, ocasiona retrasos constantes en la elaboración de liquidaciones y genera una alta probabilidad de incurrir en inconsistencias financieras en la empresa y que afecten la relación con los productores.

Con el propósito de resolver estas necesidades, el presente trabajo propone el desarrollo de un Sistema Web para la gestión de acopio, clasificación y liquidación de productores en la empresa L.Y FLOWERS, con el fin de optimizar control de seguimiento de sus procesos. Para ello se propone la construcción de la herramienta tecnológica utilizando el framework Django y el Sistema gestor de bases de datos PostgreSQL. Permitiendo que la herramienta gestione de forma centralizada la información de registro, control de productores, fincas, variedades de rosas recepciones,

clasificaciones y liquidaciones en un entorno seguro y accesible. La aplicación permitirá la administración del personal, el control de inventarios y un módulo de auditoría que fortalecerá la gestión administrativa de la empresa.

2.1. SITUACIÓN PROBLEMÁTICA

La transformación digital se ha convertido en un elemento estratégico para mejorar la competitividad de las organizaciones, permitiendo optimizar la gestión de la información y fortalecer el control de los procesos mediante el uso de Sistemas informáticos. En sectores productivos como el agrícola y florícola, la incorporación de tecnologías de la información facilita el registro, almacenamiento y consulta de datos, reduciendo errores ocasionados por procedimientos manuales y proporcionando información confiable para la toma de decisiones. No obstante, muchas micro, pequeñas y medianas empresas aún desarrollan sus actividades utilizando formularios físicos y hojas de cálculo, situación que incrementa la posibilidad de pérdida de información, duplicidad de registros, retrasos en las operaciones y dificultades para realizar el seguimiento de los procesos.

En el Ecuador, el sector florícola representa una de las principales actividades económicas orientadas a la exportación, siendo reconocido internacionalmente por la calidad de sus flores, especialmente de las rosas. La provincia de Cotopaxi constituye una de las principales zonas productoras del país, concentrando un importante número de empresas dedicadas al cultivo, comercialización y exportación de flores. El crecimiento de esta actividad ha incrementado el volumen de información generado durante los procesos de acopio, clasificación y liquidación de productores, haciendo necesaria la implementación de herramientas tecnológicas que permitan optimizar la gestión de estos procesos, mejorar el control de la información y fortalecer la trazabilidad de las operaciones desarrolladas dentro de las empresas florícolas.

En la empresa L.Y Flowers, ubicada en la parroquia Eloy Alfaro (San Felipe), cantón Latacunga, se identificó que los procesos de acopio, clasificación y liquidación de productores se realizan principalmente mediante registros manuales apoyados en formularios físicos y hojas de cálculo. Esta forma de trabajo dificulta la organización y disponibilidad de la información, ocasionando duplicidad de registros, inconsistencias en las liquidaciones, pérdida de documentos y retrasos en la generación de reportes administrativos. Asimismo, la consulta de información histórica y el seguimiento de las operaciones demandan mayor tiempo debido a que los registros se encuentran dispersos en diferentes documentos.

Frente a esta problemática, se evidencia la necesidad de desarrollar un Sistema Web que permita gestionar de manera centralizada los procesos de acopio, clasificación y liquidación de productores,

optimizando el control y seguimiento de la información generada en cada etapa del proceso. Como apoyo a esta gestión, el Sistema incorporará módulos para la administración del personal, el control de inventarios, la gestión de fincas y un módulo de auditoría que fortalecerá la trazabilidad de las operaciones, contribuyendo a mejorar la eficiencia administrativa y operativa de la empresa.

2.2. FORMULACIÓN DEL PROBLEMA

¿De qué manera se podría mejorar el registro y control de la información generada durante los procesos de acopio, clasificación y liquidación de productores en la empresa L.Y. Flowers?

2.3. OBJETO Y CAMPO DE ACCIÓN

2.3.1. Objeto de investigación

Gestión de los procesos de acopio, clasificación y liquidación de productores en la empresa L.Y. Flowers.

2.3.2. Campo de acción

1203.23 Desarrollo de Sistemas basados en tecnologías de información.

2.4. BENEFICIARIOS

Los beneficiarios directos del proyecto serán el propietario de la empresa L.Y Flowers, el personal administrativo y operativo, quienes dispondrán de una herramienta tecnológica para gestionar de manera eficiente los procesos de acopio, clasificación y liquidación de productores, así como la administración del personal, fincas e inventarios de insumos agrícolas. De igual manera, los productores se beneficiarán al contar con un proceso de liquidación más organizado, transparente y oportuno. Como beneficiarios indirectos se encuentran otras micro y pequeñas empresas del sector florícola, como muestra el registro de unidades de producción certificadas con Buenas Prácticas Agropecuarias de Agrocalidad, las cuales podrán tomar este proyecto como referencia para la implementación de soluciones tecnológicas orientadas a optimizar la gestión de sus procesos productivos y administrativos.

2.4.1. Directos

Los beneficiarios principales del trabajo de titulación son: En la Tabla 2.1 se observa beneficiarios Directos.

Tabla 2.1. Beneficiarios Directos

Beneficiarios Directos	Cargo	Descripción	N.º de Personas
Administrador	Propietario/Administrador	Administraré el Sistema con acceso completo para gestionar productores, fincas, personal, variedades de rosas, recepciones, clasificaciones, tarifarios, liquidaciones, inventarios, usuarios, auditoría y reportes.	1 persona
Secretaria	Secretaria Administrativa	Gestionará la información administrativa del personal, fincas, proveedores, documentos y liquidaciones, además de consultar reportes y registrar pagos según los permisos asignados.	1 persona
Personal de Postcosecha	Operarios de Postcosecha	Utilizarán el Sistema durante el proceso de recepción y clasificación de las flores, registrando la información del acopio y seleccionando al trabajador responsable de cada recepción para mantener la trazabilidad de los procesos.	7 personas
Subtotales beneficiarios directos			9 personas

2.4.2. Indirectos

Los beneficiarios secundarios del trabajo de titulación son: En la Tabla 2.2 se observa beneficiarios Indirectos.

Tabla 2.2. Beneficiarios Indirectos

Beneficiarios Indirectos	Cargo	Descripción	N.º de Personas
Productores	Proveedores de flores	Se beneficiarán de un proceso de recepción, clasificación y liquidación más organizado, transparente y oportuno, reduciendo errores en el registro de la información y en los cálculos de las liquidaciones.	20

Beneficiarios Indirectos	Cargo	Descripción	N.º de Personas
Empresas florícolas	Propietarios o administradores	Podrán tomar el Sistema desarrollado como modelo de referencia para implementar soluciones tecnológicas orientadas a optimizar el control y seguimiento de los procesos de acopio, clasificación y liquidación de productores.	8
Subtotal beneficiarios indirectos			28 personas

Se estableció un total de 28 beneficiarios indirectos. De ellos, 20 corresponden a productores que proveen flores a L.Y. Flowers y que se beneficiarán de una gestión más organizada y transparente de la recepción, clasificación y liquidación de sus productos. Los 8 beneficiarios restantes corresponden a propietarios o administradores de unidades de producción de rosa registradas en la provincia de Cotopaxi, quienes podrán utilizar el proyecto como referencia para implementar Sistemas de información similares. Esta cantidad se encuentra respaldada por el registro de unidades de producción certificadas con Buenas Prácticas Agropecuarias de Agrocalidad, en el cual se identificaron ocho registros cuyo [5].

2.5. JUSTIFICACIÓN

El desarrollo de este proyecto se plantea debido a la necesidad de fortalecer la gestión de los procesos de acopio, clasificación, liquidación de productores e inventario en L.Y Flowers. El empleo de registros manuales y archivos dispersos dificulta la consulta de información histórica, incrementa el riesgo de errores y limita el seguimiento oportuno de las operaciones. La implementación de una plataforma centralizada permitirá disponer de datos organizados, actualizados y accesibles para el personal autorizado.

El Sistema Web aportará beneficios operativos y administrativos al registrar productores, fincas, variedades, recepciones, clasificaciones, tarifarios, liquidaciones y movimientos de inventario dentro de una misma solución. Esta integración permitirá reducir la duplicación de datos, automatizar cálculos, conservar el historial de las operaciones, controlar los accesos según el rol del usuario y generar consultas y reportes para apoyar la toma de decisiones.

La trazabilidad constituye uno de los principales aportes del proyecto, debido a que la información de cada recepción podrá relacionarse con la finca, el productor, las variedades entregadas, la

clasificación obtenida, el tarifario vigente y la liquidación correspondiente. De esta forma, la empresa contará con evidencia verificable del recorrido de la información y podrá identificar con mayor rapidez inconsistencias o novedades.

El alcance del proyecto comprende el análisis, diseño, implementación, prueba y despliegue de un Sistema Web desarrollado con Django y PostgreSQL. La solución incluye autenticación, usuarios, grupos y permisos; administración de productores, fincas, personal y variedades; recepción y clasificación de flores; gestión de tarifarios y liquidaciones; control de inventarios; consultas, reportes y auditoría de operaciones.

El proyecto no contempla facturación electrónica, contabilidad general, gestión de exportaciones, integración con entidades bancarias, aplicación móvil ni automatización de procesos externos a L.Y Flowers. Estas funcionalidades podrán considerarse en trabajos futuros. La delimitación permite concentrar los recursos en los procesos prioritarios identificados por la empresa y validar una solución coherente con sus necesidades actuales.

2.6. OBJETIVOS

2.6.1. Objetivo general

Desarrollar un Sistema Web para la empresa L.Y. Flowers, aplicando la metodología Programación Extrema (XP), con el propósito de optimizar el control y seguimiento de los procesos de acopio, clasificación y liquidación de productores.

2.6.2. Objetivos específicos

- Fundamentar las bases teóricas y tecnológicas relacionadas gestión integral de procesos y el desarrollo de Sistemas Web, mediante la revisión de literatura científica, para sustentar el desarrollo de la propuesta tecnológica.
- Aplicar la metodología Programación Extrema (XP) durante el desarrollo del Sistema Web, mediante la planificación y ejecución de iteraciones, para construir las funcionalidades de acuerdo con los requerimientos establecidos por la empresa L.Y Flowers.
- Implementar el Sistema Web para automatizar el registro y la gestión integral de procesos, con el fin de optimizar el control y seguimiento de la información en la empresa L.Y Flowers.

2.7. TAREAS POR OBJETIVO

1. En la Tabla 2.3 se observa planificación de actividades.

Tabla 2.3. Planificación de actividades

Objetivos específicos	Actividades (tareas)	Resultados esperados	Técnicas, medios e instrumentos
Fundamentar las bases teóricas y tecnológicas relacionadas con la gestión de los procesos de acopio, clasificación y liquidación de productores y el desarrollo de Sistemas Web, mediante la revisión de literatura científica, para sustentar el desarrollo de la propuesta tecnológica.	• Buscar información bibliográfica. • Revisar documentación científica. • Analizar y organizar la información recopilada.	Fundamentación teórica y tecnológica del proyecto.	Técnica: Revisión bibliográfica. Medios: Internet, libros, artículos científicos y biblioteca digital UTC. Instrumento: Fichas bibliográficas.
Aplicar la metodología Programación Extrema (XP) durante el desarrollo del Sistema Web, mediante la planificación y ejecución de iteraciones, para construir las funcionalidades de acuerdo con los requerimientos establecidos por la empresa L.Y Flowers.	• Levantar requerimientos. • Elaborar historias de usuario. • Planificar las iteraciones. • Desarrollar las funcionalidades del Sistema.	Sistema Web desarrollado mediante la metodología XP.	Técnicas: Programación Extrema (XP), entrevista y observación. Medios: Visual Studio Code, Django, PostgreSQL, Git y GitHub. Instrumentos: Historias de usuario, guía de entrevista y cuestionario.
Implementar el Sistema Web para automatizar el registro y la gestión integral de procesos, con el fin de optimizar el control y seguimiento de la información en la empresa L.Y Flowers.	• Diseñar la arquitectura del Sistema y la base de datos. • Implementar los módulos del Sistema. • Integrar las funcionalidades. • Ejecutar pruebas funcionales.	Sistema Web implementado y validado.	Técnicas: Desarrollo de software y pruebas funcionales. Medios: Django, PostgreSQL, Python, Bootstrap y Visual Studio Code. Instrumentos: Casos de prueba, código fuente y manual técnico.

3. MARCO TEÓRICO

En la investigación denominada [6], se desarrolló un Sistema informático con el propósito de optimizar la gestión de la producción mediante el registro y control de las actividades realizadas durante el proceso productivo de rosas. El estudio evidenció que el manejo manual de la información ocasionaba retrasos, duplicidad de datos y dificultades en el seguimiento de la producción, incluso la pérdida de información. Como resultado, se implementó una solución de un Sistema informática que

mejoró la organización de la información y el control de los procesos y un mejor manejo. Este antecedente aporta al presente proyecto al demostrar la importancia que conlleva al automatizar los procesos florícolas mediante Sistemas de información. De manera similar, el Sistema Web propuesto para L.Y Flowers permitirá gestionar el acopio, la clasificación y la liquidación de proveedores de forma centralizada.

Por otro lado, en la investigación [7] se desarrolló un Sistema para mejorar el control de la producción y del inventario utilizando la metodología Programación Extrema (XP). Los resultados evidenciaron una reducción de errores en los registros y una mejor administración de la información. Este trabajo constituye un referente para el presente proyecto, ya que respalda el uso de la metodología XP y la integración de procesos como proveedores, clasificación, inventarios y liquidaciones dentro de una misma plataforma.

Asimismo, en el trabajo [8], se diseñó una aplicación Web para administrar el inventario de insumos y materiales, mejorando el control de existencias en la empresa, las entradas y salidas de productos y la generación de reportes administrativos. Este antecedente aporta al presente proyecto debido a que el Sistema desarrollado para L.Y Flowers también incorpora un módulo para la gestión de inventarios de productos e insumos agrícolas.

De igual manera, la investigación [9] estuvo orientada a mejorar la organización y el control de las actividades de postcosecha mediante la estandarización de procesos, logrando incrementar la eficiencia operativa de los procesos con las rosas. Este estudio se relaciona con el presente proyecto porque la clasificación de flores constituye una de las actividades principales del Sistema, permitiendo mantener la trazabilidad de las recepciones y clasificaciones realizadas en L.Y Flowers.

En la investigación [10] se implementó un Sistema orientado a optimizar las actividades de postcosecha, obteniendo mejoras en la productividad y un mayor control de los procesos. Este trabajo evidencia la importancia de fortalecer el control operativo mediante herramientas tecnológicas, aspecto que también se persigue con el desarrollo del Sistema Web para L.Y Flowers.

En conjunto los antecedentes analizados evidencian que la implementación de Sistemas de información contribuye a optimizar los procesos florícolas, mejorar el control de la información y reducir errores derivados de la gestión manual que lleva la empresa. Sin embargo, las investigaciones revisadas abordan principalmente la producción, los inventarios o la gestión de procesos de manera independiente. En contraste, el presente proyecto propone una solución integral que permitirá gestionar el acopio, la clasificación y la liquidación de proveedores, además de integrar módulos para

la administración de fincas, personal, variedades de rosas, inventarios de insumos agrícolas y auditoría, fortaleciendo así el control y seguimiento de los procesos desarrollados en L.Y Flowers.

3.1. GESTIÓN DEL ACOPIO, CLASIFICACIÓN Y LIQUIDACIÓN DE PROVEEDORES

La gestión del acopio, clasificación y liquidación de proveedores comprende el conjunto de actividades que permiten registrar, controlar y administrar la producción de flores entregada por los proveedores a una empresa florícola. Este proceso inicia con la recepción de la flor, continúa con la clasificación de los tallos de acuerdo con criterios de calidad y finaliza con el cálculo del valor económico correspondiente a cada proveedor según los tarifarios establecidos por la empresa.

Una gestión adecuada de estos procesos permite mantener un control permanente de la información, reducir errores derivados de registros manuales y disponer de datos confiables para la toma de decisiones administrativas. Asimismo, la integración de todas estas actividades dentro de un Sistema Web facilita la trazabilidad de cada entrega, optimiza los tiempos de procesamiento y mejora el seguimiento de las operaciones desarrolladas en la empresa.

3.1.1. Acopio y recepción de flores

El acopio de flores consiste en recibir y concentrar la producción proveniente de las diferentes fincas de la empresa y productores para su posterior procesamiento. Durante la recepción se registra información relacionada con el proveedor, la finca de origen de la empresa ya que esta cuenta con tres lugares Langualo, Mulaló, La Matriz en Patután, la variedad de la flor, la cantidad de mallas entregadas y el número de tallos recibidos.

Este proceso constituye el punto de partida para las etapas posteriores de clasificación y liquidación, ya que la información registrada durante la recepción sirve como referencia para verificar la cantidad real de flores procesadas de los productores y de las fincas. Un registro adecuado permite mantener la trazabilidad de cada entrega y facilita el control de la producción recibida por la empresa.

3.1.2. Clasificación de flores

La clasificación de flores corresponde al proceso mediante el cual se evalúa la calidad de los tallos recibidos durante el acopio el cual está en dos calidades estar y optimo. Esta actividad consiste en inspeccionar cada flor para determinar si cumple con los estándares de exportación establecidos por la empresa o si debe clasificarse como flor nacional si tiene alguna lesión o imperfección.

Durante este proceso también pueden identificarse diferencias entre la cantidad de tallos registrada en la recepción y la cantidad encontrada al abrir las mallas. Por esta razón, resulta necesario registrar

los tallos adicionales o las novedades detectadas para garantizar que la información utilizada en la liquidación refleje la producción realmente procesada para tener un registro claro y sobre todo detallado [11].

La clasificación constituye una de las actividades más importantes del área de postcosecha, debido a que determina la calidad del producto y sirve como base para el cálculo económico correspondiente a cada proveedor.

3.1.3. Liquidación de proveedores

La liquidación de proveedores es el procedimiento mediante el cual se calcula el valor económico que la empresa debe pagar por la producción entregada. Para realizar este cálculo se consideran factores como la variedad de la flor, la cantidad de tallos clasificados por calidad y el tarifario vigente para la fecha establecida de recepción, ya que está clasificado esta constara de dos tipos dependiendo de la calidad que son estándar y óptimo.

Debido a que los precios pueden variar con el tiempo, es importante mantener un historial de tarifarios que permita conservar la información correspondiente a cada período de vigencia. De esta manera, las liquidaciones reflejan el precio aplicable al momento en que la producción fue recibida, evitando inconsistencias y garantizando la confiabilidad de los cálculos realizados.

La automatización de este proceso permite reducir errores de cálculo, agilizar la generación de liquidaciones de proveedores y mejorar el control de los pagos mensuales efectuados a cada proveedor.

3.1.4. Gestión administrativa

La gestión administrativa comprende el conjunto de actividades destinadas a organizar y controlar la información necesaria para el funcionamiento de los procesos operativos de la empresa. Entre esta información se encuentran los registros de proveedores, fincas, variedades de flores, personal, tarifarios, inventario de insumos agrícolas y demás datos requeridos para el desarrollo de las actividades de recepción, clasificación y liquidación de los proveedores [12].

Mantener esta información centralizada dentro de un Sistema Web facilita la actualización de los registros, mejora la disponibilidad de los datos y fortalece el control de los procesos administrativos que lleva la empresa. Además, permite relacionar la información de los diferentes módulos del Sistema, favoreciendo la generación de reportes y apoyando en la toma de decisiones.

3.1.5. Gestión de inventarios

La gestión de inventarios comprende las actividades relacionadas con el control, registro y seguimiento de los insumos que ingresan y productos utilizados para cada finca dependiendo su necesidad. Su finalidad es mantener información actualizada sobre las existencias disponibles, las entradas y salidas de productos y el consumo generado durante las labores en el cultivo.

En las empresas florícolas, una adecuada gestión de inventarios permite administrar los productos químicos e insumos agrícolas empleados en las diferentes fincas, evitando pérdidas por desabastecimiento, exceso de existencias o registros incorrectos. La incorporación de un Sistema Web facilita el control permanente del inventario y mejora la disponibilidad de información para la planificación de compras y el seguimiento de los recursos utilizados.

3.1.6. Control de inventarios

El control de inventarios consiste en registrar de manera continua las entradas, salidas y existencias de los productos almacenados. Este proceso permite conocer la disponibilidad real de cada artículo y facilita la planificación del abastecimiento de acuerdo con las necesidades de la empresa.

Un control adecuado disminuye el riesgo de pérdidas, evita inconsistencias en los registros y contribuye al uso eficiente de los recursos disponibles. Además, permite generar reportes que apoyan el seguimiento de los movimientos realizados y fortalecen la gestión administrativa.

3.1.7. Importancia de control de inventarios

El control de inventarios constituye un elemento fundamental para garantizar la disponibilidad oportuna de los insumos agrícolas utilizados en las actividades productivas. Su correcta administración permite optimizar los recursos, reducir costos llevar un mejor registro operativo y prevenir los problemas derivados del desabastecimiento o del exceso de productos almacenados.

En el caso de L.Y Flowers, el control de inventarios facilita llevar seguimiento de los productos químicos utilizados en cada una de las fincas, permitiendo registrar entradas, salidas y existencias dentro de una misma plataforma informática, fortaleciendo así el control administrativo de la empresa.

3.1.8. Automatización de procesos

La automatización de procesos consiste en utilizar herramientas tecnológicas para ejecutar actividades de forma sistemática, reduciendo la intervención manual y mejorando la eficiencia

operativa. Mediante Sistemas informáticos es posible registrar información, procesar datos y generar resultados de manera más rápida, confiable y organizada [11].

En el sector florícola, la automatización permite integrar procesos como el acopio, la clasificación, la liquidación de proveedores y el control de inventarios dentro de una misma plataforma, disminuyendo errores en los registros y facilitando el seguimiento de las operaciones desarrolladas por la empresa.

3.1.9. Automatización de procesos empresariales

La automatización de procesos empresariales consiste en incorporar soluciones tecnológicas que permitan optimizar las actividades administrativas y operativas de una organización. Su aplicación facilita la integración de la información, reduce tareas repetitivas y mejora la productividad mediante la ejecución sistemática de los procesos [12].

En empresas florícolas, la automatización contribuye a mejorar el control de la producción, administrar la información de los proveedores y mantener un seguimiento continuo de las actividades realizadas desde la recepción de la flor hasta la liquidación correspondiente.

3.1.10. Beneficios de la automatización

La automatización de procesos proporciona diversos beneficios para las organizaciones al mejorar la eficiencia en la gestión de la información y optimizar el desarrollo de las actividades operativas laborables. Entre los principales beneficios se encuentran la reducción de errores derivados de registros manuales, cálculos al azar que llevan los empleados, la disminución de tiempos de procesamiento, la disponibilidad de información en tiempo real, consultas de reportes y el fortalecimiento del control administrativo.

la automatización permitirá centralizar la información de una manera clara y coherente correspondiente al acopio de flores, clasificación, liquidación de proveedores, inventarios de insumos agrícolas y la gestión de empleados, facilitando la generación de reportes en tiempo real, el seguimiento de todas las operaciones y la toma de decisiones dentro de la empresa.

3.2. SISTEMA WEB

Un Sistema Web es una aplicación informática que funciona mediante un navegador de Internet y permite a los usuarios acceder a sus funcionalidades desde diferentes dispositivos conectados a una red. A diferencia de las aplicaciones de escritorio, los Sistemas Web centralizan la información en un servidor, facilitando el acceso simultáneo de varios usuarios, el almacenamiento seguro de los datos y la actualización permanente de la información [13].

En la figura 3.1 se presenta la arquitectura cliente servidor.



Figura 3.1. Arquitectura cliente servidor [14].

Los Sistemas Web son ampliamente utilizados por organizaciones que requieren automatizar procesos administrativos y operativos, debido a que mejoran la disponibilidad de la información, optimizan los tiempos de trabajo y favorecen la toma de decisiones mediante el acceso oportuno a los datos.

3.2.1. Herramientas de desarrollo web

Es fundamental abordar el tema de los lenguajes en una investigación, son la base técnica sobre la que se construye e implementa una solución tecnológica. Al especificar el lenguaje que se utiliza, se pueden entender mejor las capacidades, limitaciones, compatibilidades y el alcance del Sistema propuesto. Además, esto aporta claridad y un respaldo académico al trabajo, demostrando que la elección tecnológica no fue al azar, sino que se basó en criterios como eficiencia, rendimiento, escalabilidad o adaptabilidad al entorno del proyecto. Incluir los lenguajes también ayuda a que el estudio sea reproducible y refuerza su validez técnica [15].

3.2.1.1. Lenguaje de programación

Los lenguajes de programación son herramientas que permiten dar instrucciones a una computadora para crear aplicaciones, páginas web, videojuegos y otros programas.

La Tabla 3.1 presenta la comparación entre los lenguajes de programación considerados.

Tabla 3.1. Comparación entre lenguajes de programación

Característica	Python[16]	Java[17]	PHP[18]
Facilidad de aprendizaje	Alta, sintaxis clara y sencilla	Media, sintaxis más extensa	Alta, relativamente sencillo
Desarrollo Web	Muy bueno, con Frameworks como Django y Flask	Muy bueno, con Spring Boot	Muy bueno, especialmente con Laravel
Seguridad	Alta, dependiendo del Framework utilizado	Alta, con herramientas robustas	Buena, dependiendo de la implementación
Rendimiento	Bueno para aplicaciones Web	Alto en aplicaciones de gran escala	Bueno para aplicaciones Web
Escalabilidad	Alta	Alta	Media-Alta
Mantenimiento	Fácil por su legibilidad	Bueno, aunque puede requerir más código	Bueno con una estructura adecuada
Código abierto	Sí	Sí	Sí
Comunidad	Muy amplia	Muy amplia	Muy amplia
Integración con bases de datos	Amplia compatibilidad	Amplia compatibilidad	Amplia compatibilidad
Uso en backend	Muy adecuado	Muy adecuado	Muy adecuado

3.2.1.2. Python

Python es un lenguaje de programación de alto nivel que se caracteriza por su sintaxis clara, sencilla y fácil de mantener. Su amplia comunidad de desarrolladores y la disponibilidad de numerosas bibliotecas lo convierten en uno de los lenguajes más utilizados para el desarrollo de aplicaciones Web, inteligencia artificial, automatización de procesos y análisis de datos. [19]

Una de sus principales ventajas es la facilidad para desarrollar aplicaciones robustas con menor cantidad de código, permitiendo mejorar la productividad durante el desarrollo del software.



Figura 3.2. Lenguaje de programación Python [20]

3.2.1.2.1. Características

- Fácil de aprender: Python tiene documentación en el cual detalla fácilmente como podemos implementar o usar su código [20].
- Aplicaciones: Python aloja miles de módulos de terceros para su biblioteca estándar, así como módulos de la comunidad de Python [20].
- Código abierto: Python está desarrollado con licencia de código abierto, lo cual lo hace fácilmente utilizable y distribuible [20].

3.2.1.3. Framework de desarrollo

Un Framework de desarrollo es un conjunto de herramientas y estructuras prediseñadas que facilita y acelera la creación de aplicaciones o páginas web.

La Tabla 3.2 presenta la comparación entre los Frameworks a comparar.

Tabla 3.2. Comparación entre el Framework Django y otros Frameworks

Criterio	Django [21]	Laravel [22]	Express.js [23]
Lenguaje	Python , lenguaje legible, versátil y con gran soporte para desarrollo Web y científico.	PHP , lenguaje orientado al desarrollo Web, ampliamente utilizado, aunque con una sintaxis menos moderna.	JavaScript , permite desarrollar el frontend y el backend con un mismo lenguaje, aunque su asincronía puede aumentar la complejidad del desarrollo.
Rendimiento	Alto rendimiento, estable y eficiente gracias a su arquitectura bien estructurada.	Rendimiento medio; puede requerir optimizaciones adicionales en proyectos de gran tamaño.	Muy alto rendimiento gracias al motor V8; sin embargo, la estabilidad depende de las buenas prácticas del desarrollador.

Criterio	Django [21]	Laravel [22]	Express.js [23]
Seguridad	Muy alta seguridad con protección integrada contra CSRF, XSS, SQL Injection y autenticación robusta. Es ideal para Sistemas críticos.	Alta seguridad, aunque algunas funcionalidades requieren configuraciones adicionales.	La seguridad depende de las librerías utilizadas y de la configuración implementada por el desarrollador.
Base de datos	Incorpora un ORM potente compatible con PostgreSQL, MySQL, Oracle, SQLite y otros gestores de bases de datos.	Incluye Eloquent ORM, intuitivo y fácil de usar, aunque puede presentar menor rendimiento en determinados escenarios.	No incorpora un ORM de forma nativa; es necesario instalar y configurar uno según las necesidades del proyecto.
Ventajas	Desarrollo rápido, alta seguridad, escalabilidad, panel administrativo automático y estructura adecuada para proyectos de gran tamaño.	Sintaxis elegante, buena documentación y una comunidad amplia y activa.	Alta flexibilidad y velocidad, especialmente para el desarrollo de APIs y microservicios.
Desventajas	Su estructura puede resultar rígida para proyectos pequeños y presenta una curva de aprendizaje media.	Puede consumir más recursos y disminuir su rendimiento en aplicaciones de gran escala.	Requiere instalar múltiples librerías, ofrece poca estructura por defecto y existe un mayor riesgo de errores y malas prácticas.

Django tiene incorporado ORM que ofrece un alto nivel de seguridad, estabilidad y rendimiento en la gestión de datos. Permite almacenar y organizar la información de manera confiable, manteniendo su integridad mediante el manejo de transacciones. Además, es de código abierto, por lo que no requiere costos de licenciamiento. Su escalabilidad y compatibilidad con diferentes tecnologías lo convierten en una alternativa sólida para aplicaciones Web que requieren un manejo seguro y eficiente de la información.

3.2.1.3.1. Framework Django

Django es un Framework de desarrollo Web de código abierto desarrollado en Python, diseñado para facilitar la creación de aplicaciones Web seguras, escalables y de alto rendimiento. Su arquitectura basada en el patrón Modelo–Vista–Plantilla (MVT) permite organizar el código de manera estructurada, separando la lógica del negocio, la presentación de la información y el acceso a los datos. Entre las principales características de Django se encuentran la administración integrada de

usuarios y permisos, el Sistema de autenticación, el manejo de formularios, la protección contra vulnerabilidades comunes y el uso de un ORM (Object Relational Mapping), que facilita la interacción con bases de datos relacionales sin necesidad de escribir consultas SQL complejas [24].



Figura 3.3. Framework Django [25]

3.2.1.4. Sistema gestor de bases de datos

La Tabla 3.3 presenta la comparación entre los GBD a comparar.

Tabla 3.3. Comparación entre Sistemas gestores de bases de datos

Criterio	MongoDB [26]	Oracle Database [27]	PostgreSQL [28]
Modelo de datos	No relacional (documentos JSON/BSON).	Relacional.	Relacional orientado a objetos.
Licencia	Community Edition gratuita y Enterprise de pago.	Licencia comercial con alto costo.	Software libre y de código abierto (Open Source).
Rendimiento	Muy alto para datos no estructurados y consultas simples.	Alto rendimiento en aplicaciones empresariales de gran escala.	Alto rendimiento para transacciones complejas y grandes volúmenes de datos.
Integridad de datos	Limitada; depende del diseño de la aplicación.	Muy alta mediante restricciones y transacciones ACID.	Muy alta mediante restricciones, transacciones ACID y control de concurrencia (MVCC).
Escalabilidad	Excelente escalabilidad horizontal.	Escalabilidad vertical y horizontal mediante herramientas propietarias.	Escalable tanto vertical como horizontal mediante extensiones y replicación.
Seguridad	Buena, requiere configuración adicional.	Muy alta, incorpora mecanismos avanzados de seguridad empresarial.	Alta seguridad con autenticación, cifrado, control de acceso por roles y auditoría.

Criterio	MongoDB [26]	Oracle Database [27]	PostgreSQL [28]
Compatibilidad	Ideal para aplicaciones NoSQL.	Compatible con múltiples plataformas y herramientas empresariales.	Compatible con Django, Laravel, Express.js, Python, Java, PHP, Node.js y otros lenguajes.
Facilidad de uso	Fácil para aplicaciones con datos flexibles.	Requiere mayor experiencia en administración.	Administración sencilla y amplia documentación.
Costo	Gratuito en su edición Community.	Alto costo de licenciamiento y mantenimiento.	Gratuito, sin costos de licenciamiento.
Ventajas	Flexibilidad en el almacenamiento de datos y escalabilidad horizontal.	Alta disponibilidad, robustez y soporte empresarial.	Alto rendimiento, estabilidad, seguridad, soporte para consultas complejas, cumplimiento ACID, amplia comunidad y excelente integración con Django.
Desventajas	Menor consistencia en relaciones complejas y menor capacidad para consultas SQL avanzadas.	Alto costo, complejidad de administración y dependencia del proveedor.	Requiere conocimientos de bases de datos relacionales para aprovechar todas sus funcionalidades.

PostgreSQL es un gestor de base de datos que ofrece un alto nivel de seguridad, estabilidad y rendimiento en el manejo de la información. Permite almacenar grandes cantidades de datos de manera organizada y confiable, garantizando su integridad mediante el cumplimiento de las propiedades ACID. Además, es de código abierto, por lo que no requiere el pago de licencias, y cuenta con una buena compatibilidad con Django mediante su ORM. También proporciona herramientas para el control de accesos y la gestión de transacciones, características importantes para mantener la información segura y consistente dentro del Sistema.

3.2.1.4.1. PostgreSQL

PostgreSQL es un Sistema gestor de bases de datos relacional de código abierto, reconocido por su estabilidad, seguridad y capacidad para administrar grandes volúmenes de información. Su arquitectura permite mantener la integridad de los datos mediante restricciones, claves primarias y foráneas, transacciones y mecanismos de recuperación ante fallos [29].

Además, PostgreSQL ofrece un excelente rendimiento en aplicaciones empresariales, permitiendo ejecutar múltiples operaciones de lectura y escritura de manera concurrente, sin comprometer la consistencia de la información.



Figura 3.4. Sistema gestor de base de datos PostgreSQL [29]

3.2.1.4.2. Características

- Tipos de datos variados: permite trabajar con datos numéricos, texto, booleanos, fechas, JSON y otros formatos [30].
- Integridad de datos: utiliza claves primarias, claves foráneas y restricciones como único y no nulo para asegurar la consistencia de la información [30].
- Indexación avanzada: mejora la velocidad de búsqueda y consulta dentro de la base de datos [30].
- Control de concurrencia (MVCC): permite que varios usuarios trabajen al mismo tiempo sin afectar el rendimiento [30].
- Transacciones seguras: garantiza que las operaciones con datos se realicen de forma correcta y completa [30].
- Seguridad: cuenta con Sistemas de autenticación y control de acceso para proteger la información [30].
- Replicación y recuperación: incluye mecanismos para respaldo y recuperación de datos ante fallos [30].

3.2.1.5. Tecnologías para el desarrollo de interfaces Web

Las interfaces Web constituyen el medio mediante el cual los usuarios interactúan con un Sistema informático, permitiendo así acceder de manera sencilla a las diferentes funcionalidades que ofrece la aplicación. Para su construcción se emplean tecnologías que permiten estructurar el contenido, definir el diseño visual e incorporar funcionalidades dinámicas que mejoran la experiencia del usuario. Una interfaz bien diseñada facilita la navegación, reduce la complejidad en el uso del Sistema y contribuye a que las tareas puedan realizarse de forma rápida, intuitiva y eficiente [31].

La Tabla 3.4 sintetiza la función de las tecnologías empleadas en la construcción de la interfaz.

Tabla 3.4. Tecnologías utilizadas en la interfaz Web

Tecnología	Tipo	Aplicación en el Sistema
HTML	Estructura	Organización de formularios, tablas, menús y contenido.
CSS	Presentación	Definición de colores, tipografía, espaciado y distribución.
JavaScript	Interactividad	Validaciones, confirmaciones y respuestas dinámicas.
Bootstrap	Framework visual	Componentes responsivos para computadoras, tabletas y móviles.
ThemeWagon	Plantilla	Base visual del panel administrativo, personalizada para L.Y Flowers.

3.2.1.5.1. HTML

HTML (HyperText Markup Language) es el lenguaje encargado de estructurar el contenido de las páginas Web. Permite organizar formularios, tablas, menús, botones y demás elementos que conforman la interfaz del Sistema [32].

3.2.1.5.2. CSS

CSS (Cascading Style Sheets) es el lenguaje utilizado para definir la apariencia visual de una página Web. Mediante hojas de estilo es posible controlar colores, tipografías, tamaños, espacios y distribución de los elementos que conforman la interfaz [33].

3.2.1.5.3. JavaScript

JavaScript es un lenguaje de programación que permite incorporar funcionalidades dinámicas e interactivas dentro de las aplicaciones Web. Mediante este lenguaje es posible validar información, responder a las acciones del usuario y actualizar contenido sin necesidad de recargar completamente la página [34].

3.2.1.5.4. Bootstrap

Bootstrap es un Framework de diseño que proporciona componentes visuales prediseñados para el desarrollo de interfaces Web adaptables a diferentes tamaños de pantalla. Facilita la construcción de

formularios, tablas, botones, menús y paneles administrativos manteniendo una apariencia uniforme [35].

3.2.1.5.5. ThemeWagon

ThemeWagon es una plataforma que ofrece plantillas Web prediseñadas para el desarrollo de paneles administrativos y aplicaciones empresariales. Estas plantillas proporcionan una estructura visual moderna adaptable a las necesidades específicas de cada proyecto. [36].

3.2.1.6. Herramientas de desarrollo y control de versiones

Durante el desarrollo del Sistema fue necesario utilizar herramientas que facilitaran la programación, organización del código fuente y administración de las diferentes versiones del proyecto

3.2.1.6.1. Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft que proporciona herramientas para escribir, depurar y organizar aplicaciones desarrolladas en diferentes lenguajes de programación. Su compatibilidad con extensiones para Python, Django y PostgreSQL facilitó el desarrollo del Sistema [37].

3.2.1.6.2. Git

Git es un Sistema distribuido de control de versiones que permite registrar los cambios realizados en el código fuente, facilitando la recuperación de versiones anteriores y el seguimiento de las modificaciones efectuadas durante el desarrollo del software [38].

3.2.1.6.3. GitHub

GitHub es una plataforma basada en Git que permite almacenar repositorios en la nube y mantener un historial completo del desarrollo de una aplicación. Además, facilita la colaboración entre desarrolladores y la administración de versiones del proyecto [39].

3.2.1.7. Herramienta de despliegue Railway

Railway es una plataforma en la nube que permite desplegar aplicaciones Web y servicios asociados de forma rápida y sencilla, proporcionando una infraestructura administrada para el alojamiento de aplicaciones, bases de datos y otros servicios necesarios para su funcionamiento. La plataforma automatiza tareas relacionadas con la configuración del entorno de ejecución, la administración de variables de entorno y la publicación de aplicaciones en Internet [44].

Una de las principales ventajas de Railway es la integración de múltiples servicios dentro de una misma plataforma, permitiendo alojar aplicaciones desarrolladas con diferentes tecnologías y gestionar bases de datos relacionales como PostgreSQL sin realizar configuraciones complejas de infraestructura. Además, ofrece características como monitoreo, escalabilidad, generación automática de dominios públicos y administración de recursos, facilitando el mantenimiento de aplicaciones en producción [40].



Figura 3.5. Plataforma de despliegue Railway [41]

3.2.2. Arquitectura

3.2.2.1. Arquitectura Model–View–Template

Django emplea el patrón arquitectónico Modelo–Vista–Plantilla (MVT), el cual separa las responsabilidades de la aplicación en tres componentes principales. El Modelo representa la estructura de los datos y gestiona su persistencia mediante el ORM de Django; la Vista recibe las solicitudes del usuario, procesa la lógica de la aplicación y coordina la respuesta; mientras que la Plantilla se encarga de presentar la información mediante interfaces desarrolladas con HTML, CSS, JavaScript y Bootstrap [42].

En la figura 3.3 se presenta la arquitectura MVT

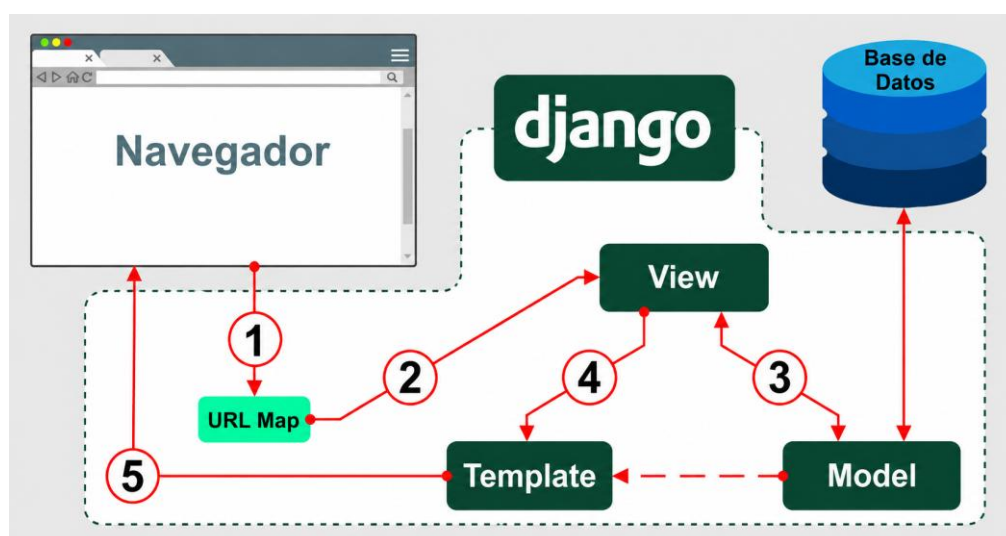


Figura 3.6. Arquitectura Model View Template [42]

En la Tabla 3.5 se describe la arquitectura en tres capas M-V-T.

Tabla 3.5. Arquitectura en 3 capas

Componente	Descripción	Aplicación del Sistema	Funcion principal
Modelo (Model)	Representa la estructura de la información y permite administrar los datos mediante el ORM de Django.	Modelos relacionados con usuarios, proveedores, variedades, recepciones, clasificaciones, inventarios y liquidaciones, almacenados en PostgreSQL.	Consultar, registrar, actualizar y eliminar información, además de establecer relaciones y restricciones entre los datos.
Vista (View)	Recibe las solicitudes enviadas por los usuarios, ejecuta la lógica correspondiente y genera una respuesta.	Vistas, formularios, validaciones, permisos y procesos desarrollados en Django para gestionar el acopio, la clasificación y la liquidación de productores.	Procesar las solicitudes, validar los datos, aplicar las reglas del Sistema y coordinar la comunicación entre los modelos y las plantillas.
Plantilla (Template)	Corresponde a la interfaz mediante la cual se presenta la información y se permite la interacción con el Sistema.	Plantillas desarrolladas con HTML, CSS, JavaScript y Bootstrap para los diferentes perfiles de usuario.	Mostrar la información procesada, capturar los datos ingresados mediante formularios y presentar los resultados al usuario.

La arquitectura MVT presenta ventajas como la escalabilidad, la mantenibilidad, la seguridad y la flexibilidad tecnológica. La separación de responsabilidades entre el modelo, la vista y la plantilla favorece el modularidad del Sistema, facilita la identificación y corrección de errores y permite modificar o ampliar sus funcionalidades sin afectar significativamente a los demás componentes. De esta manera, se favorece la evolución progresiva del software de acuerdo con las nuevas necesidades de la empresa.

3.3. METODOLOGÍAS DE DESARROLLO DE SOFTWARE

Las metodologías de desarrollo de software constituyen un conjunto de principios, prácticas y procedimientos que orientan la planificación, diseño, construcción, implementación y mantenimiento de aplicaciones informáticas. Su finalidad es proporcionar una forma organizada de desarrollar software, permitiendo administrar los recursos, controlar los tiempos de ejecución y garantizar que el producto final satisfaga los requerimientos del usuario [42].

A lo largo de la evolución del desarrollo de software han surgido diferentes metodologías, las cuales pueden clasificarse principalmente en metodologías tradicionales y metodologías ágiles. Mientras las metodologías tradicionales se caracterizan por seguir una planificación secuencial y una documentación extensa, las metodologías ágiles priorizan la colaboración con el cliente, la entrega continua de funcionalidades y la capacidad de adaptación frente a cambios en los requerimientos.

3.3.1. Metodologías ágiles frente a metodologías tradicionales

Las metodologías tradicionales de desarrollo de software, como el modelo en cascada, organizan el proyecto en fases secuenciales donde cada etapa debe finalizar antes de iniciar la siguiente. Este enfoque resulta adecuado cuando los requerimientos del Sistema se encuentran claramente definidos desde el inicio del proyecto y presentan pocas modificaciones durante su desarrollo. Sin embargo, cuando los requisitos cambian con frecuencia, este tipo de metodologías puede dificultar la incorporación de nuevas funcionalidades y aumentar el tiempo necesario para completar el software [43].

Por otra parte, las metodologías ágiles surgieron con el propósito de responder de manera más eficiente a los cambios que suelen presentarse durante el desarrollo de aplicaciones informáticas. Estas metodologías promueven la colaboración continua entre el equipo de desarrollo y los usuarios finales, favoreciendo la construcción del software mediante iteraciones cortas que permiten entregar funcionalidades de manera progresiva y realizar mejoras conforme evolucionan las necesidades del proyecto.

La Tabla 3.6 presenta las principales diferencias entre las metodologías tradicionales y las metodologías ágiles.

Tabla 3.6. Comparación entre metodologías tradicionales y ágiles

Criterio	Metodologías tradicionales[44]	Metodologías ágiles[45]
Enfoque de desarrollo	Desarrollo secuencial por fases (análisis, diseño, implementación y pruebas).	Desarrollo iterativo e incremental mediante entregas frecuentes.
Gestión de cambios	Los cambios son difíciles de incorporar una vez iniciado el proyecto.	Los cambios se aceptan y gestionan durante todo el desarrollo.
Participación del cliente	Limitada, principalmente al inicio y al finalizar el proyecto.	Continua, con retroalimentación constante durante el desarrollo.
Entregas	Se realiza una única entrega al finalizar el proyecto.	Se realizan entregas parciales y funcionales en cada iteración.

Criterio	Metodologías tradicionales[44]	Metodologías ágiles[45]
Documentación	Amplia y detallada; constituye la base del desarrollo.	Esencial y suficiente para apoyar el desarrollo del software.
Planificación	Se define al inicio y presenta pocos cambios durante el proyecto.	Es flexible y se ajusta conforme evolucionan los requerimientos.
Flexibilidad	Baja, debido a la rigidez del proceso de desarrollo.	Alta, permitiendo responder rápidamente a nuevas necesidades.
Tiempo de respuesta	Mayor, ya que los cambios requieren replantear fases del proyecto.	Menor, gracias a las iteraciones cortas y la retroalimentación continua.
Adecuada para	Proyectos con requisitos claramente definidos y estables.	Proyectos con requisitos cambiantes o en constante evolución.

3.3.2. Comparación entre metodologías ágiles

La Tabla 3.7 compara XP, Scrum y Kanban considerando su enfoque, adaptación al cambio, participación del cliente, entregas y calidad del software.

Tabla 3.7. Comparación entre metodologías ágiles

Criterio	XP [46]	Scrum [47]	Kanban [48]
Enfoque principal	Desarrollo de software mediante prácticas de ingeniería y mejora continua de la calidad.	Gestión ágil de proyectos mediante iteraciones llamadas Sprints.	Gestión visual del flujo de trabajo y optimización de procesos continuos.
Ciclos de trabajo	Iteraciones cortas de 1 a 2 semanas.	Sprints de duración fija entre 2 y 4 semanas.	Flujo continuo sin iteraciones definidas.
Adaptación al cambio	Muy alta, permite incorporar cambios durante el desarrollo.	Media, los cambios se realizan generalmente al finalizar un Sprint.	Alta, los cambios pueden incorporarse en cualquier momento.
Participación del cliente	Continua; el cliente participa activamente definiendo prioridades y validando funcionalidades.	El Product Owner representa al cliente y administra los requerimientos.	Variable, depende de las necesidades del proyecto.
Calidad del software	Muy alta, gracias a pruebas continuas, programación en parejas e integración continua.	Alta, basada en revisiones periódicas y retrospectivas del equipo.	Depende de las prácticas implementadas por el equipo.

Criterio	XP [46]	Scrum [47]	Kanban [48]
Prácticas principales	Programación en parejas, desarrollo guiado por pruebas (TDD), integración continua, refactorización y diseño simple.	Sprint Planning, Daily Scrum, Sprint Review y Sprint Retrospective.	Tablero Kanban, límites de trabajo en progreso (WIP) y mejora continua del flujo.

3.3.1. Programación extrema (XP)

Extreme Programming (XP) es una metodología ágil de desarrollo de software propuesta por Kent Beck con el objetivo de mejorar la calidad del software y responder rápidamente a los cambios en los requerimientos del cliente. XP promueve una comunicación constante entre los usuarios y el equipo de desarrollo, favoreciendo la construcción del Sistema mediante pequeñas iteraciones que permiten validar continuamente las funcionalidades implementadas [49].

Esta metodología se fundamenta en valores como la comunicación, la simplicidad, la retroalimentación, el respeto y el coraje, los cuales orientan el desarrollo colaborativo del software y facilitan la adaptación del proyecto a nuevas necesidades durante su ejecución. Asimismo, XP incorpora prácticas como las historias de usuario, el desarrollo incremental, las pruebas continuas y la integración frecuente del código

En la Figura 3.7 se presentan las fases de la metodología Extreme Programming (XP).



Figura 3.7. Fases de la metodología XP [50].

En la Tabla 3.8 se detallan las etapas de la metodología Extreme Programming (XP).

Tabla 3.8. Etapas de la metodología XP

Etapas	Descripción
Planificación	Se recopilan las historias de usuario y se establecen los criterios de aceptación. El trabajo se organiza y prioriza de acuerdo con las necesidades del cliente [50].
Diseño	Se elaboran prototipos y diagramas que orientan el desarrollo de la solución, además de definir la estructura y los componentes principales del Sistema [50].
Codificación	Se desarrollan las funcionalidades del Sistema de forma incremental, procurando obtener un código comprensible, organizado y fácil de mantener [50].
Pruebas	Se ejecutan pruebas unitarias, funcionales y de aceptación para identificar errores oportunamente y comprobar el cumplimiento de los requisitos [50].
Lanzamiento	Se publica el Sistema una vez implementadas y verificadas las historias de usuario planificadas, dejándolo disponible para su utilización en el entorno operativo [50].

3.3.1.1. Valores de XP

En la Figura 3.8 se detallan los valores de la metodología Extreme Programming (XP).



Figura 3.8. Valores de Extreme Programming (XP)

Fuente: elaboración propia.

Comunicación. Durante el desarrollo del proyecto se mantuvo una comunicación permanente con los responsables de la empresa para validar requerimientos y mejorar el Sistema en cada iteración [51].

Simplicidad. El Sistema fue diseñado considerando únicamente las funcionalidades necesarias para automatizar los procesos de la empresa [51].

Retroalimentación. Cada módulo fue presentado a los usuarios para recibir observaciones y realizar mejoras antes de continuar [51].

Respeto. Se consideraron las opiniones de los usuarios para construir una solución alineada con las necesidades reales [51].

Coraje. Fue posible incorporar cambios en los requerimientos sin afectar la estabilidad del Sistema [51].

3.3.1.2. Principios de XP

En la Figura 3.9 se detallan los principios de la metodología Extreme Programming (XP).

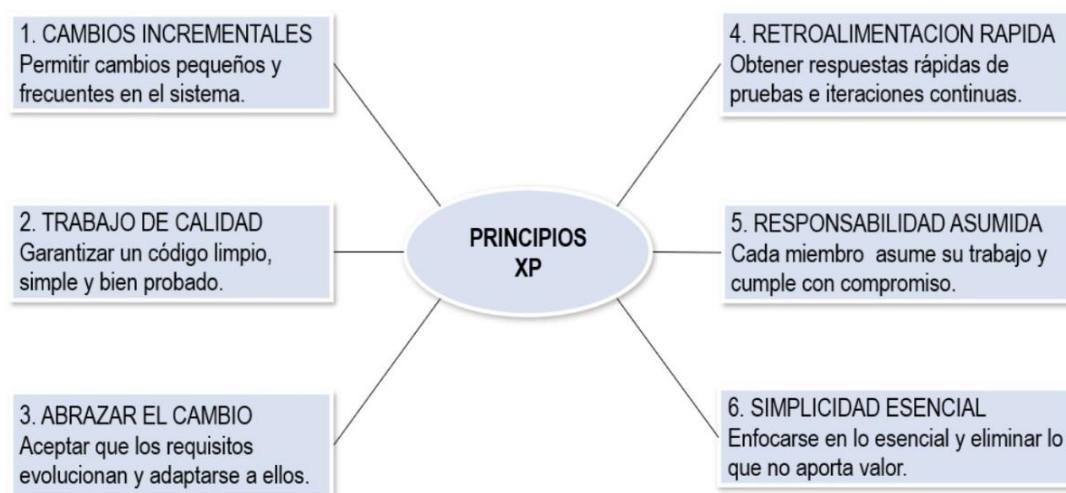


Figura 3.9. Principios de Extreme Programming (XP)

Fuente: elaboración propia.

Cambios incrementales. El Sistema fue construido mediante pequeñas iteraciones implementando progresivamente los módulos del proyecto [52].

Trabajo de calidad. Cada funcionalidad fue desarrollada siguiendo buenas prácticas de programación y pruebas funcionales [52].

Abrazar el cambio. Los nuevos requerimientos fueron incorporados sin afectar el resto de la aplicación [52].

Retroalimentación rápida. Cada versión fue revisada por los responsables de la empresa antes de la siguiente iteración [52].

Responsabilidad asumida. Las actividades fueron ejecutadas siguiendo la planificación mediante historias de usuario [52].

Simplicidad esencial. La arquitectura se mantuvo modular implementando únicamente las funcionalidades necesarias [52].

3.3.1.3. Ventajas de XP

La metodología Extreme Programming ofrece diversas ventajas para el desarrollo de Sistemas Web, especialmente en proyectos donde los requerimientos pueden modificarse durante el proceso de construcción del software. Su enfoque iterativo permite entregar funcionalidades de manera progresiva, facilitando la validación temprana del Sistema y reduciendo el riesgo de desarrollar características que no respondan a las necesidades reales del usuario [53].

Entre los principales beneficios de XP se encuentra la mejora continua de la calidad del software mediante pruebas frecuentes, la comunicación permanente con los usuarios, la rápida incorporación de cambios y la entrega incremental de nuevas funcionalidades. Estas características contribuyen a disminuir errores, optimizar los tiempos de desarrollo y mantener una mayor flexibilidad frente a cambios en los procesos de negocio.

4. MÉTODOS Y PROCEDIMIENTOS

4.1. DISEÑO METODOLÓGICO DE LA INVESTIGACIÓN

En este capítulo se describen el enfoque, los tipos y métodos de investigación, las técnicas e instrumentos de recolección de datos y el procedimiento tecnológico empleado para desarrollar FlorLY. También se detallan la aplicación de la metodología Extreme Programming (XP), los criterios de modelado del Sistema, las herramientas de implementación y el procedimiento de pruebas. Los productos obtenidos mediante estos procedimientos se presentan y analizan en el capítulo 5.

4.1.1. Enfoque de investigación

Para el desarrollo del Sistema, se realizó una investigación con enfoque mixto, integrando métodos cuantitativos y cualitativos, con la finalidad de examinar la situación actual de los procesos internos desarrollados en la empresa L.Y Flowers.

El enfoque cuantitativo se aplicó principalmente al personal administrativo y a los trabajadores de postcosecha, debido a que constituyen los beneficiarios directos. Se utilizaron encuestas para evaluar aspectos como la eficiencia de los procesos, el empleo de las herramientas actuales y el nivel de aceptación de un Sistema automatizado que permita optimizar estas actividades. En el enfoque cualitativo se realizaron entrevistas al gerente, a la secretaria y al trabajador con mayor antigüedad

en el área de postcosecha, con el propósito de conocer claramente el funcionamiento interno de los procesos empresariales. La combinación de ambos enfoques permitió comprender la problemática e identificar el nivel de aceptación de la implementación de un Sistema orientado a mejorar estas actividades.

4.1.2. Tipos de investigación

Para la ejecución de este proyecto se emplearon diversos tipos de investigación, para comprender la problemática desde varias perspectivas, y elegir la metodología que se ajuste al proyecto.

4.1.2.1. Investigación documental

Este tipo de investigación es primordial e importante para el desarrollo del presente proyecto, ya que mediante este enfoque se sustenta teóricamente el estudio y se respalda el desarrollo del Sistema Web. Además, permitió obtener información relevante de diversas fuentes bibliográficas, informes, documentos, artículos científicos, libros, tesis y sitios oficiales, con el propósito de establecer conceptos fundamentales relacionados con la gestión de procesos, las tecnologías utilizadas y la metodología XP. La información recopilada también facilitó la selección de herramientas adecuadas y eficientes para diseñar e implementar una solución acorde con las necesidades identificadas en L.Y Flowers.

4.1.2.2. Investigación de campo

Esta clase de investigación se realizó en la empresa L.Y Flowers, para conocer cómo se gestiona la información actualmente y conversar con los participantes. Además, se efectuó una encuesta mediante un cuestionario elaborado cuidadosamente y se reforzó con entrevistas.

4.1.2.3. Investigación tecnológica

La investigación tecnológica orientó la transformación de las necesidades identificadas en una solución informática verificable. El procedimiento comprendió el levantamiento de requerimientos, el diseño de la arquitectura y de la base de datos, el desarrollo incremental de los módulos, la ejecución de pruebas funcionales y el despliegue del Sistema Web FlorLY.

4.1.3. Métodos de investigación

4.1.3.1. Método descriptivo

Se aplicó este método para conocer la situación actual de los procesos de acopio, clasificación y liquidación de productores en la empresa L.Y Flowers. Con esto se observaron y describieron las actividades realizadas, las herramientas utilizadas para registrar la información y las principales dificultades relacionadas con el control y seguimiento de los procesos.

4.1.3.2. Método analítico

Mediante este método se analizaron los datos obtenidos en las encuestas y entrevistas, lo que permitió descomponer los procesos de acopio, clasificación y liquidación en sus diferentes actividades. Además, se examinaron fuentes bibliográficas relacionadas con el tema para establecer conceptos y criterios aplicables al diseño y desarrollo del Sistema Web.

4.1.3.3. Método deductivo

Se empleó este método con base en los resultados obtenidos del análisis teórico y de campo, lo que permitió definir un diseño acorde con los lineamientos identificados y desarrollar un Sistema Web apropiado, garantizando el cumplimiento de los requerimientos reales de la empresa.

4.1.4. Población y muestra

La población estuvo integrada por nueve beneficiarios directos: una administradora, una secretaria y siete trabajadores de postcosecha. Al ser una población accesible y reducida se aplicó un censo, por lo que no fue necesario calcular una muestra.

4.2. TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE INFORMACIÓN

4.2.1. Revisión bibliográfica

Esta etapa del proyecto fue fundamental para sustentar los conceptos relacionados con la gestión de postcosecha, los cuales sirvieron para desarrollar el marco teórico. Se consultaron bases de datos científicas y diferentes fuentes académicas, como artículos científicos, tesis, libros y documentación técnica relacionada con el Sistema Web.

4.2.1.1. Ficha bibliográfica

Registró autor, título, año y aporte de las fuentes utilizadas en la fundamentación científico-técnica.

4.2.2. Encuesta

Se aplicó una encuesta al personal administrativo y a los trabajadores del área de postcosecha de L.Y Flowers, quienes participan directamente en los procesos estudiados. El cuestionario permitió recopilar información estructurada sobre la frecuencia de uso de los registros, los medios empleados, las dificultades del proceso manual, las necesidades de trazabilidad y la aceptación de la automatización. Las respuestas se tabularon mediante frecuencias absolutas y porcentajes respecto de la población censada; las tablas, gráficas y su interpretación se presentan en el apartado 5.2.

4.2.1.1. Cuestionario de la encuesta

El cuestionario estuvo conformado por ocho preguntas cerradas, orientadas a conocer la percepción y experiencia de los participantes respecto al desarrollo de un Sistema para gestionar la información. Además, permitió comprobar si los beneficiarios directos reconocen sus posibles ventajas e identificar sus principales problemas y necesidades.

4.3. METODOLOGÍA DE DESARROLLO DE SOFTWARE XP

Para organizar el desarrollo del Sistema se empleó Extreme Programming (XP), debido a su orientación incremental, la comunicación frecuente con el cliente y la posibilidad de incorporar cambios durante iteraciones cortas. El procedimiento comenzó con la identificación y priorización de requerimientos mediante historias de usuario; después se distribuyeron las funcionalidades en iteraciones y tareas, se desarrollaron los módulos mediante programación en pareja y se integraron progresivamente en un repositorio compartido. GitHub se utilizó para el control de versiones y la trazabilidad de los cambios. Al finalizar cada iteración se ejecutaron pruebas vinculadas con los criterios de aceptación y se revisaron los módulos con el cliente. Los artefactos y evidencias resultantes de la aplicación de XP se presentan en el apartado 5.3.

Los artefactos metodológicos seleccionados fueron las historias de usuario, el Product Backlog y el Iteration Backlog. Sus formatos se definieron en este capítulo para establecer los campos, criterios de prioridad, estimaciones, responsables y estados que debían utilizarse de manera uniforme durante el proyecto.

4.3.1. Fase de planificación

La fase de planificación permitió transformar las necesidades identificadas mediante entrevistas y encuestas en requerimientos organizados. El procedimiento comprendió la redacción de historias de usuario, la identificación de roles, la priorización de funcionalidades, la estimación mediante puntos

de historia y la distribución del trabajo en el Product Backlog y el Iteration Backlog. Los artefactos completados como resultado de esta fase se presentan en el apartado 5.3.1.

4.3.1.1. Historias de usuario

El proyecto se organizó en historias de usuario específicas y manejables, para satisfacer los requerimientos funcionales del Sistema Web de gestión de postcosecha de L.Y Flowers. Las historias de usuario se definieron considerando las necesidades y expectativas de cada usuario. Para estimar su complejidad, se aplicó la secuencia de Fibonacci y se identificaron posteriormente los actores involucrados junto con las funcionalidades correspondientes dentro del Sistema.

En la Tabla 4.1 se presenta el formato utilizado para documentar las historias de usuario del proyecto.

Tabla 4.1. Diseño de las historias de usuario

HISTORIA DE USUARIO			
Número:		Usuario:	
Objetivo:			
Prioridad:	Estimación:		Iteración:
Responsables:			
Descripción:			

4.3.1.2. Roles XP

Los roles se determinaron de acuerdo con la participación de los involucrados en el proyecto. Se consideraron el cliente como fuente de necesidades y responsable de validar las funcionalidades; los programadores como responsables del diseño, desarrollo e integración; el tester como responsable de verificar los criterios de aceptación; el coach como orientador de las prácticas XP; y el tracker como encargado del seguimiento de las estimaciones y del avance. Los roles asignados se presentan en el apartado 5.3.1.2.

4.3.1.3. Product Backlog

Se utilizó el Product Backlog de XP para distribuir el trabajo en actividades específicas y asignar responsables y estados dentro de cada iteración. Este artefacto permitió organizar las tareas de manera clara y controlada, con el propósito de identificar al programador encargado y conocer el estado actual en el que se encuentra cada iteración.

En la Tabla 4.2 se presenta el formato utilizado para documentar el Product Backlog.

Tabla 4.2. Diseño del Product Backlog

Historia de usuario	Tarea técnica	Tipo de tarea	Responsable	Estado

4.3.1.4. Iteration Backlog

Uno de los artefactos empleados fue el Iteration Backlog, en el cual se organizan y priorizan las historias de usuario de acuerdo con los criterios del equipo de desarrollo. Posteriormente, a cada historia seleccionada se le asigna un nivel de complejidad y la iteración en la que será desarrollada.

En la Tabla 4.3 se presenta el formato utilizado para organizar el Iteration Backlog.

Tabla 4.3. Diseño del Iteration Backlog

Iteración	ID	Historia de usuario	Prioridad	Puntos

4.3.2. Fase de diseño

El diseño tecnológico se organizó mediante la arquitectura Modelo-Vista-Plantilla (MVT) de Django, con separación entre presentación, lógica de negocio y persistencia de datos. Para la implementación se estableció el uso de Python, Django, PostgreSQL, HTML, CSS, JavaScript y Bootstrap. El diseño se documentó mediante modelos de datos, diagramas UML de casos de uso y diagramas BPMN de los procesos operativos.

4.3.2.1. Diseño de casos de uso

Para representar la interacción entre los actores y las funcionalidades del Sistema se emplearon diagramas de casos de uso basados en UML. El procedimiento consistió en identificar actores, asociarles las funcionalidades definidas en las historias de usuario y verificar la correspondencia entre requerimientos y módulos. Los diagramas obtenidos se presentan en el apartado 5.3.2.1.

4.3.2.2. Modelado de la base de datos

Para el modelado de la base de datos se aplicó un enfoque relacional. Se identificaron las entidades, atributos, claves primarias y foráneas, relaciones y restricciones necesarias para mantener la

integridad y trazabilidad de la información. PostgreSQL se seleccionó como Sistema gestor y Django ORM como mecanismo de acceso a los datos. El modelo obtenido se presenta en el apartado 5.3.2.2.

4.3.2.3. Diseño de la arquitectura del sistema

La arquitectura se diseñó mediante el patrón Modelo-Vista-Plantilla (MVT) de Django. Se definieron la capa de presentación mediante plantillas, HTML, CSS, JavaScript y Bootstrap; la capa de lógica mediante vistas, formularios, validaciones y reglas de negocio en Python y Django; y la capa de datos mediante modelos, Django ORM y PostgreSQL. La arquitectura obtenida se presenta en el apartado 5.3.2.3.

4.3.2.4. Modelado de procesos BPMN

Los procesos de postcosecha se representaron mediante la notación BPMN. Para cada proceso se identificaron el evento inicial, las actividades, las decisiones, los responsables, las entradas y las salidas. Este procedimiento permitió documentar los flujos de recepción, clasificación, liquidación, inventario, reportes y auditoría. Los diagramas resultantes se presentan en el apartado 5.3.2.4.

4.3.3. Fase de codificación

La codificación se ejecutó de manera incremental según las historias asignadas a cada iteración. Para cada funcionalidad se implementaron los modelos, formularios, vistas, plantillas, validaciones y reglas de negocio requeridas. Se aplicó programación en pareja, control de versiones con GitHub, integración progresiva y refactorización para mejorar la organización, legibilidad y mantenimiento del código sin modificar el comportamiento esperado. La evidencia de las funcionalidades desarrolladas y de la refactorización se presenta en el apartado 5.3.3.

Para documentar esta fase se registraron la historia de usuario atendida, las pruebas realizadas durante la programación, el fragmento o evidencia del código implementado y la refactorización efectuada. Este registro permitió mantener la trazabilidad entre el requerimiento, su implementación y la verificación posterior.

4.3.4. Fase de pruebas

4.3.4.1. Diseño de casos de prueba

Las pruebas se definieron a partir de las historias de usuario y sus criterios de aceptación. Para cada caso se registraron el identificador, el escenario o funcionalidad evaluada, los datos o pasos de ejecución, el resultado esperado, el resultado obtenido y el estado de aceptación. Los casos se

ejecutaron al finalizar las iteraciones correspondientes; cuando se detectaron diferencias, se efectuaron correcciones antes de repetir la validación. Los resultados de las pruebas se presentan en el apartado 5.3.4.

En la Tabla 4.4 se presenta el formato utilizado para registrar los casos de prueba.

Tabla 4.4. Diseño de los casos de prueba

Iteración – Caso de prueba	
Historia de usuario	
Prueba ejecutada	
Resultados obtenidos	
Evidencia	

4.3.4.2. Procedimiento de ejecución

Los casos de prueba se ejecutaron al finalizar las iteraciones correspondientes. Se ingresaron los datos definidos, se realizaron los pasos establecidos y se comparó el comportamiento observado con el resultado esperado. Cuando se detectó una diferencia, se corrigió la funcionalidad y se repitió la validación antes de registrar el resultado definitivo.

4.3.4.3. Criterio de aceptación

Una prueba se consideró aceptada cuando el resultado obtenido coincidió con el resultado esperado y se cumplió el criterio de aceptación de la historia de usuario. La evidencia, el resultado obtenido y el estado final de los casos ejecutados se presentan en el apartado 5.3.4.

4.3.5. Fase de implementación y despliegue

La implementación comprendió la preparación del entorno de producción, la configuración de PostgreSQL, la definición de variables de entorno, la ejecución de migraciones, la recolección de archivos estáticos y el inicio de la aplicación mediante Unicorn. Railway se empleó como plataforma de despliegue y GitHub como origen del código publicado. Finalmente, se verificaron el acceso mediante HTTPS, la conexión con los servicios de datos, la carga de recursos estáticos y el funcionamiento de los módulos principales. Las interfaces y evidencias de publicación se presentan en el apartado 5.3.5.

5. ANÁLISIS Y DISCUSIÓN DE RESULTADOS

5.1. RESULTADOS DE LA ENTREVISTA

La entrevista se realizó el 28 de abril de 2026 en la ciudad de Latacunga, en las instalaciones de la empresa L.Y Flowers. Se entrevistó al gerente, a la secretaria y al trabajador con mayor experiencia en el área de postcosecha, lo que permitió identificar claramente las necesidades existentes y evidenciar la importancia de implementar un Sistema que responda a ellas. Posteriormente, se efectuó una entrevista grupal al personal de postcosecha, dirigida por el gerente. Esta segunda entrevista permitió contrastar y validar la información recopilada previamente, fortaleciendo la consistencia de los resultados mediante la coincidencia de criterios entre los participantes.

Uno de los principales problemas identificados fue la dependencia de registros manuales y hojas de cálculo, lo que dificulta el seguimiento de las recepciones, la clasificación de las flores y la trazabilidad de las operaciones realizadas. Esta situación puede ocasionar pérdida o duplicación de información, errores en el cálculo de las liquidaciones y retrasos en la generación de reportes. Además, limita la disponibilidad de información confiable para la toma oportuna de decisiones administrativas. Por esta razón, se consideró necesaria la implementación del Sistema Web FlorLY, con el propósito de mejorar el control de la gestión de postcosecha y optimizar el tiempo del personal involucrado.

La matriz de análisis de contenido evidenció categorías recurrentes relacionadas con registros manuales, dispersión de la información, dificultades de trazabilidad, demoras en la consolidación de datos y riesgo de errores en las liquidaciones. La coincidencia entre las entrevistas individuales y la entrevista grupal respaldó la priorización de estas necesidades para el desarrollo de FlorLY.

En la Tabla 5.1 se presenta la matriz de análisis de contenido utilizada para sintetizar e interpretar las entrevistas.

Tabla 5.1. Matriz de análisis de contenido

Pregunta	Categoría	Gerente	Secretaria	Personal de postcosecha	Interpretación
¿Cómo se gestionan actualmente las recepciones, clasificaciones y liquidaciones?	Gestión actual	Se emplean registros manuales y hojas de cálculo.	La información se organiza en archivos separados.	La clasificación se registra manualmente.	No existe un Sistema centralizado que integre los procesos.

Pregunta	Categoría	Gerente	Secretaria	Personal de postcosecha	Interpretación
¿Qué dificultades existen para dar seguimiento a las recepciones y clasificaciones?	Seguimiento y control	No se dispone de información actualizada en tiempo real.	Es necesario revisar diferentes registros.	Las clasificaciones pendientes se controlan manualmente.	Se requiere automatizar el seguimiento de las recepciones y clasificaciones.
¿Cómo se controla la procedencia y cantidad de flores recibidas?	Control de recepciones	Se revisan registros físicos y hojas de cálculo.	Se registran el proveedor, la finca y las variedades entregadas.	Se comprueba la cantidad real de tallos durante la clasificación.	El control manual dificulta relacionar la recepción con la clasificación.
¿El proceso actual es manual, digital o mixto?	Tipo de proceso	Es mixto, con predominio de actividades manuales.	Se utilizan documentos físicos y archivos de Excel.	La información se anota manualmente durante el trabajo.	Existe una alta dependencia de tareas manuales que puede ocasionar errores.
¿Qué problemas se presentan con mayor frecuencia?	Problemas de información	Falta de información centralizada y actualizada.	Duplicidad, inconsistencias y dificultad para localizar registros.	Diferencias entre los tallos registrados y los realmente clasificados.	La dispersión de los datos afecta el control y la trazabilidad.
¿Existen retrasos al generar liquidaciones y reportes?	Generación de resultados	Los reportes requieren recopilar información de varias fuentes.	Las liquidaciones necesitan cálculos y verificaciones manuales.	Los resultados diarios se consolidan al finalizar la jornada.	La automatización reduciría el tiempo de elaboración de liquidaciones y reportes.
¿La información histórica se encuentra organizada y accesible?	Información histórica	Existen registros almacenados en diferentes medios.	La búsqueda de información anterior requiere revisar varios archivos.	No se dispone de consultas directas sobre clasificaciones anteriores.	La falta de una base de datos dificulta las consultas históricas.
¿Qué funciones considera indispensables para el Sistema?	Funcionalidades	Reportes, indicadores, auditoría y control general.	Gestión de proveedores, recepciones, tarifarios y liquidaciones.	Registro y consulta de las clasificaciones realizadas.	FlorLY debe integrar las funciones operativas y administrativas de la empresa.

Pregunta	Categoría	Gerente	Secretaria	Personal de postcosecha	Interpretación
¿Es importante disponer de indicadores automáticos?	Toma de decisiones	Permitirían conocer la producción y el estado de los procesos.	Facilitarían la elaboración de reportes y liquidaciones.	Ayudarían a controlar los tallos procesados y pendientes.	Los indicadores permitirían tomar decisiones basadas en información actualizada.
¿Considera necesaria la implementación de FlorLY?	Necesidad del Sistema	Mejoraría el control y seguimiento de las operaciones.	Reduciría errores y tiempos de procesamiento.	Facilitaría el registro de las actividades de postcosecha.	Existe aceptación sobre la necesidad de implementar un Sistema centralizado.

Mientras que los resultados de entrevista aplicadas al director y al personal de postcosecha se presentan a continuación

5.2. RESULTADOS DE LA ENCUESTA

A continuación, se presentan los resultados obtenidos de los nueve beneficiarios directos que integraron el censo. Para cada una de las ocho preguntas se muestra la frecuencia absoluta, el porcentaje, la representación gráfica y la interpretación correspondiente.

Los resultados evidenciaron la dependencia de registros manuales y hojas de cálculo, la dispersión de la información y las dificultades para consultar datos históricos. También se identificaron posibles errores en el registro de recepciones, demoras en la consolidación de las clasificaciones y retrasos en la generación de liquidaciones y reportes. Además, los encuestados reconocieron la importancia de relacionar cada recepción con el productor, la finca, la variedad y el trabajador responsable.

Finalmente, se determinó un nivel favorable de aceptación hacia la implementación del Sistema Web FlorLY. Los participantes consideraron que la automatización permitirá centralizar la información, reducir errores, mejorar la trazabilidad de las operaciones y optimizar el control y seguimiento de la gestión de postcosecha. Estos resultados sustentaron la necesidad de desarrollar la solución y permitieron priorizar sus principales funcionalidades.

5.2.1. Pregunta 1: ¿Con qué frecuencia registra o consulta información relacionada con recepción, clasificación o liquidación de flores?

En la Tabla 5.3 se presenta la distribución de respuestas correspondiente a la pregunta 1 de la encuesta.

Tabla 5.3. Distribución de respuestas de la pregunta 1

Opción	Frecuencia	Porcentaje
Diariamente	7	77.8 %
Varias veces por semana	2	22.2 %
Ocasionalmente	0	0.0 %
Total	9	100.0 %

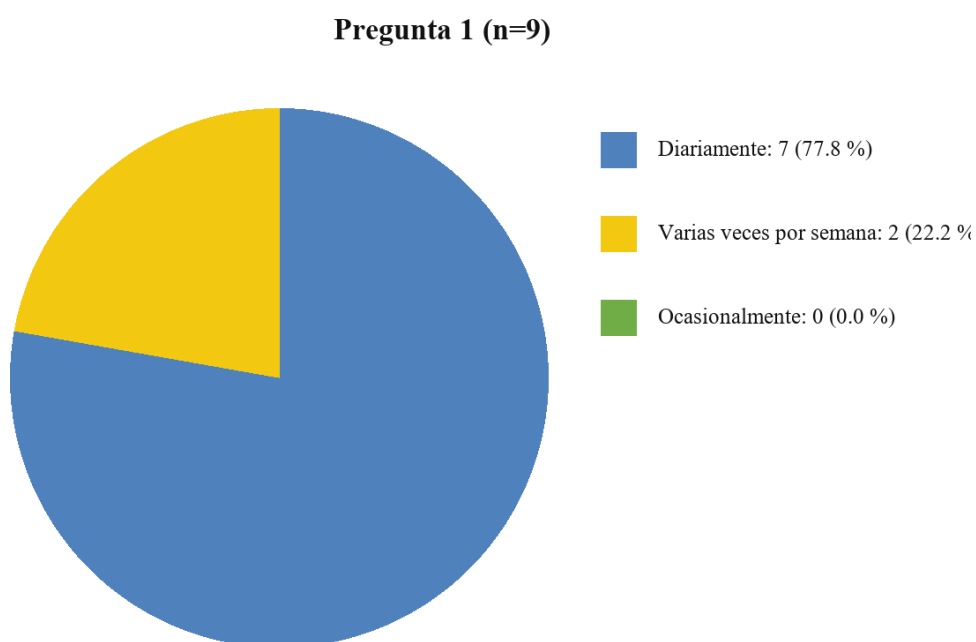


Figura 5.1. Resultados de la pregunta 1

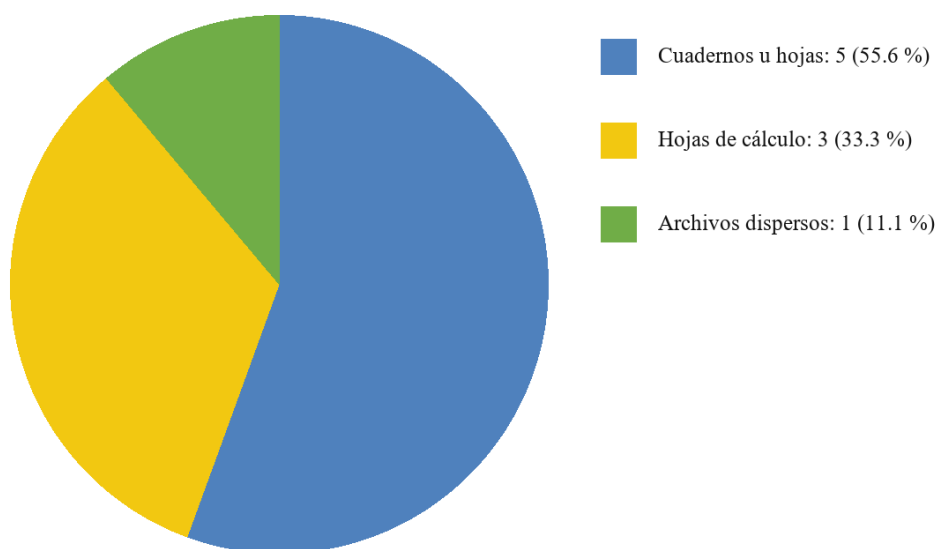
Análisis de resultados: La opción predominante fue «Diariamente», seleccionada por 7 participantes (77.8 %). Este resultado confirma la conveniencia de centralizar el proceso y priorizar controles que faciliten el registro, la consulta y la trazabilidad de la información.

5.2.2. Pregunta 2: ¿Qué medio se utilizaba principalmente para registrar la información antes del Sistema?

En la Tabla 5.4 se presenta la distribución de respuestas correspondiente a la pregunta 2 de la encuesta.

Tabla 5.4. Distribución de respuestas de la pregunta 2

Opción	Frecuencia	Porcentaje
Cuadernos u hojas	5	55.6 %
Hojas de cálculo	3	33.3 %
Archivos dispersos	1	11.1 %
Total	9	100.0 %

Pregunta 2 (n=9)*Figura 5.2. Resultados de la pregunta 2*

Análisis de resultados: La opción predominante fue «Cuadernos u hojas», seleccionada por 5 participantes (55.6 %). Este resultado confirma la conveniencia de centralizar el proceso y priorizar controles que faciliten el registro, la consulta y la trazabilidad de la información.

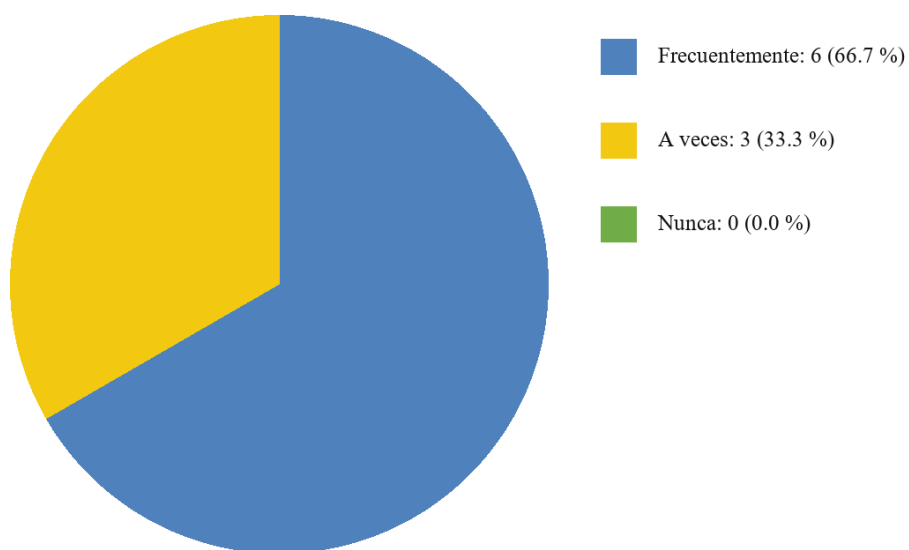
5.2.3. Pregunta 3: ¿Con qué frecuencia se presentaban dificultades para localizar registros anteriores?

En la Tabla 5.5 se presenta la distribución de respuestas correspondiente a la pregunta 3 de la encuesta.

Tabla 5.5. Distribución de respuestas de la pregunta 3

Opción	Frecuencia	Porcentaje
Frecuentemente	6	66.7 %

Opción	Frecuencia	Porcentaje
A veces	3	33.3 %
Nunca	0	0.0 %
Total	9	100.0 %

Pregunta 3 (n=9)*Figura 5.3. Resultados de la pregunta 3*

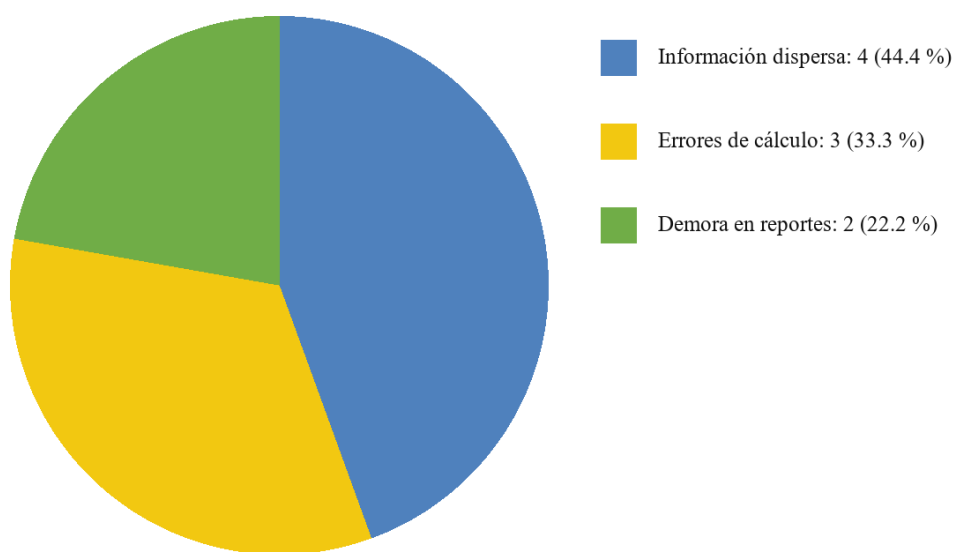
Análisis de resultados: La opción predominante fue «Frecuentemente», seleccionada por 6 participantes (66.7 %). Este resultado confirma la conveniencia de centralizar el proceso y priorizar controles que faciliten el registro, la consulta y la trazabilidad de la información.

5.2.4. Pregunta 4: ¿Cuál era el principal problema del proceso manual?

En la Tabla 5.6 se presenta la distribución de respuestas correspondiente a la pregunta 4 de la encuesta.

Tabla 5.6. Distribución de respuestas de la pregunta 4

Opción	Frecuencia	Porcentaje
Información dispersa	4	44.4 %
Errores de cálculo	3	33.3 %
Demora en reportes	2	22.2 %
Total	9	100.0 %

Pregunta 4 (n=9)*Figura 5.4. Resultados de la pregunta 4*

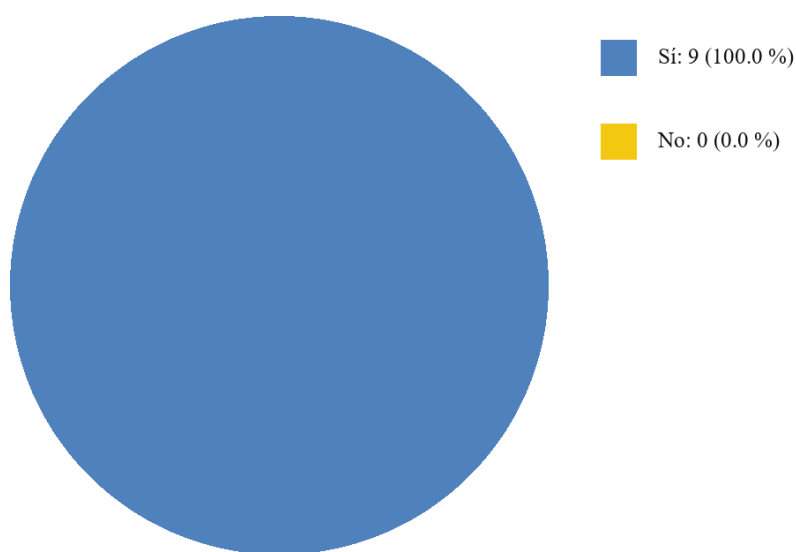
Análisis de resultados: La opción predominante fue «Información dispersa», seleccionada por 4 participantes (44.4 %). Este resultado confirma la conveniencia de centralizar el proceso y priorizar controles que faciliten el registro, la consulta y la trazabilidad de la información.

5.2.5. Pregunta 5: ¿Considera necesario relacionar cada recepción con productor, variedad y responsable?

En la Tabla 5.7 se presenta la distribución de respuestas correspondiente a la pregunta 5 de la encuesta.

Tabla 5.7. Distribución de respuestas de la pregunta 5

Opción	Frecuencia	Porcentaje
Sí	9	100.0 %
No	0	0.0 %
Total	9	100.0 %

Pregunta 5 (n=9)*Figura 5.5. Resultados de la pregunta 5*

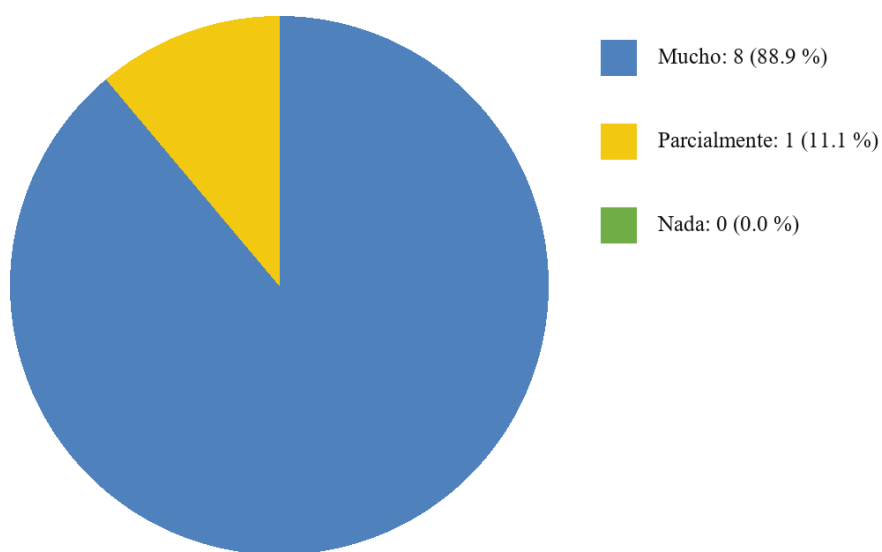
Análisis de resultados: La opción predominante fue «Sí», seleccionada por 9 participantes (100.0 %). Este resultado confirma la conveniencia de centralizar el proceso y priorizar controles que faciliten el registro, la consulta y la trazabilidad de la información.

5.2.6. Pregunta 6: ¿La automatización de liquidaciones reduciría tiempo y errores?

En la Tabla 5.8 se presenta la distribución de respuestas correspondiente a la pregunta 6 de la encuesta.

Tabla 5.8. Distribución de respuestas de la pregunta 6

Opción	Frecuencia	Porcentaje
Mucho	8	88.9 %
Parcialmente	1	11.1 %
Nada	0	0.0 %
Total	9	100.0 %

Pregunta 6 (n=9)*Figura 5.6. Resultados de la pregunta 6*

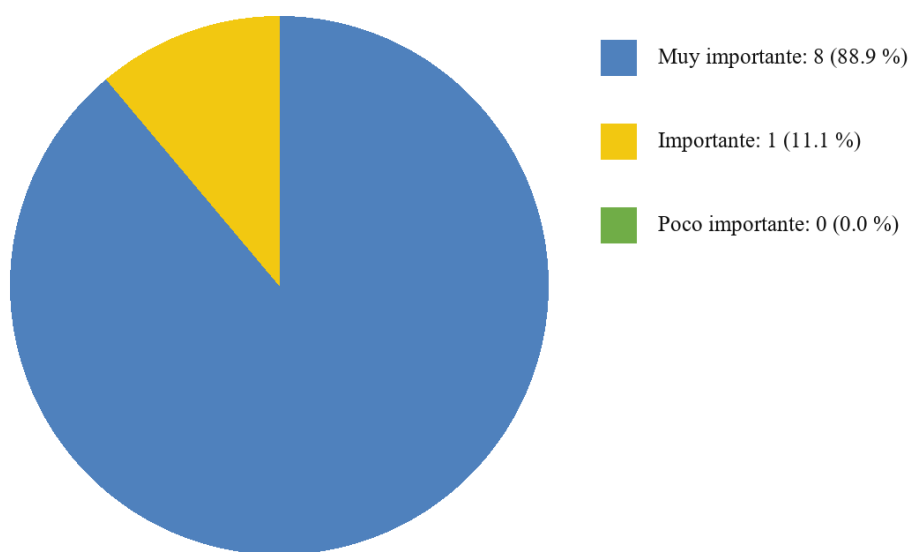
Análisis de resultados: La opción predominante fue «Mucho», seleccionada por 8 participantes (88.9 %). Este resultado confirma la conveniencia de centralizar el proceso y priorizar controles que faciliten el registro, la consulta y la trazabilidad de la información.

5.2.7. Pregunta 7: ¿Qué tan importante es disponer de reportes y trazabilidad por fecha y usuario?

En la Tabla 5.9 se presenta la distribución de respuestas correspondiente a la pregunta 7 de la encuesta.

Tabla 5.9. Distribución de respuestas de la pregunta 7

Opción	Frecuencia	Porcentaje
Muy importante	8	88.9 %
Importante	1	11.1 %
Poco importante	0	0.0 %
Total	9	100.0 %

Pregunta 7 (n=9)*Figura 5.7. Resultados de la pregunta 7*

Análisis de resultados: La opción predominante fue «Muy importante», seleccionada por 8 participantes (88.9 %). Este resultado confirma la conveniencia de centralizar el proceso y priorizar controles que faciliten el registro, la consulta y la trazabilidad de la información.

5.2.8. Pregunta 8: ¿Considera que FlorLY mejorará el control y seguimiento de los procesos?

En la Tabla 5.10 se presenta la distribución de respuestas correspondiente a la pregunta 8 de la encuesta.

Tabla 5.10. Distribución de respuestas de la pregunta 8

Opción	Frecuencia	Porcentaje
Sí	9	100.0 %
No	0	0.0 %
Total	9	100.0 %

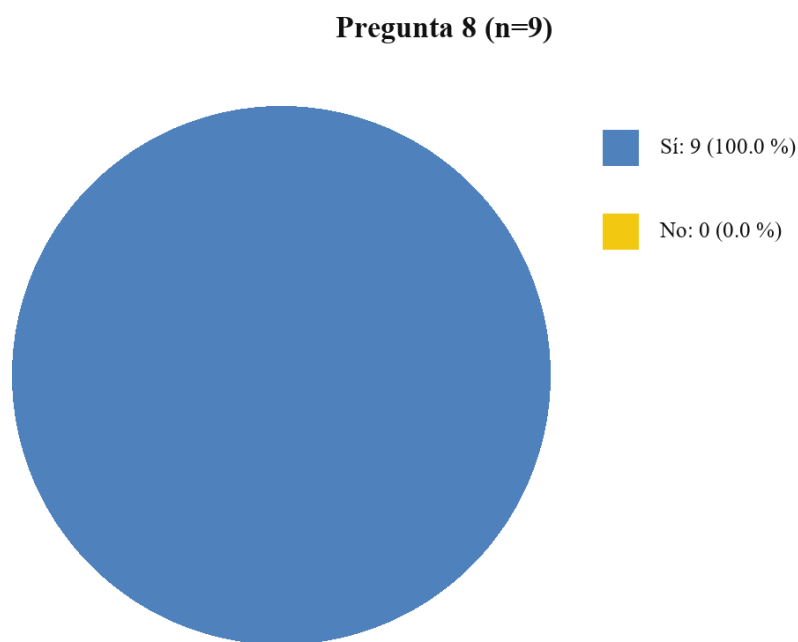


Figura 5.8. Resultados de la pregunta 8

Análisis de resultados: La opción predominante fue «Sí», seleccionada por 9 participantes (100.0 %). Este resultado confirma la conveniencia de centralizar el proceso y priorizar controles que faciliten el registro, la consulta y la trazabilidad de la información.

5.3. RESULTADOS DE LA APLICACIÓN DE LA METODOLOGÍA XP

5.3.1. Resultados de la fase de planificación

5.3.1.1. Historia de usuario

En las siguientes tablas se ilustran las historias de usuario, donde se presentan las necesidades y expectativas de los diferentes actores involucrados. Esto nos ayudará a saber la función de cada historia de usuario desglosada de manera más detallada.

Se muestran las historias más relevantes en las Tablas 5.11, 5.12 y 5.13. El resto de las tablas se encuentra en el Anexo A.

Tabla 5.11. HU001: Gestión de usuarios

Nº 1	HISTORIA DE USUARIO		
Número: HU001		Usuario: Administrador	
Objetivo: Gestión de usuarios			
Prioridad: Alta	Estimación: 5		Iteración: 1
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como Administrador, quiero registrar, actualizar y desactivar cuentas de usuario, para controlar quién puede acceder a FlorLY.			
Criterios de aceptación: Dado que el administrador está autenticado, cuando registra una cuenta con los datos obligatorios y un nombre de usuario único, entonces el Sistema debe guardar al usuario. Dado que el nombre de usuario ya existe, cuando intenta guardar otra cuenta con el mismo valor, entonces el Sistema debe impedir el registro e informar la duplicidad.			

Tabla 5.12. HU002: Asignación de roles y permisos

Nº 2	HISTORIA DE USUARIO		
Número: HU002		Usuario: Administrador	
Objetivo: Asignación de roles y permisos			
Prioridad: Alta	Estimación: 8		Iteración: 1
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como Administrador, quiero asignar roles y permisos a cada cuenta, para restringir el acceso a la información y a las operaciones del Sistema.			
Criterios de aceptación: Dado que existe un usuario registrado, cuando el administrador le asigna un rol, entonces el Sistema debe habilitar únicamente las opciones autorizadas. Dado que un usuario intenta acceder a una función no permitida, cuando solicita la operación, entonces el Sistema debe negar el acceso sin modificar información.			

Tabla 5.13. HU003: Consulta del panel principal

Nº 3	HISTORIA DE USUARIO	
Número: HU003	Usuario: Usuario	
Objetivo: Consulta del panel principal		

Nº 3	HISTORIA DE USUARIO	
Prioridad: Media	Estimación: 5	Iteración: 1
Responsables: Evelin Ganchala y Leonardo Quinaluisa		
Descripción: Yo como Usuario, quiero visualizar indicadores generales del Sistema, para conocer el estado de las operaciones de L.Y Flowers.		
Criterios de aceptación: Dado que existen datos registrados, cuando el administrador abre el panel, entonces el Sistema debe mostrar indicadores de recepciones, clasificaciones, liquidaciones e inventario. Dado que se selecciona un período, cuando se actualiza la consulta, entonces los indicadores deben corresponder a las fechas elegidas.		

5.3.1.2. Roles XP

Los roles facilitaron la organización eficiente del proyecto, garantizando el cumplimiento de cada fase, la calidad del Sistema y la retroalimentación permanente.

A continuación, en la tabla 5.14 se presentan los roles considerados para el desarrollo del Sistema.

Tabla 5.14. Roles de XP

ROL	DESCRIPCIÓN
Cliente (Customer/Usuario clave)	Está representado por el gerente, la secretaria y los empleados de postcosecha de L.Y Flowers, quienes definen las necesidades, revisan las funcionalidades y verifican el cumplimiento de las historias de usuario.
Programadores (Equipo de Desarrollo XP)	Está conformado por los autores del proyecto, responsables de analizar, diseñar y desarrollar el Sistema FlorLY mediante programación en parejas y de acuerdo con las historias de usuario.
Tester/QA	Es el responsable de ejecutar las pruebas unitarias, de integración y aceptación, verificando que las funcionalidades desarrolladas cumplan los criterios establecidos.

ROL	DESCRIPCIÓN
Coach XP	Es el tutor del proyecto, quien orienta al equipo durante las fases de XP y promueve la aplicación adecuada del desarrollo iterativo e incremental.
Tracker	Es el encargado de supervisar el avance de las iteraciones, controlar el cumplimiento de las historias de usuario y dar seguimiento a los tiempos establecidos para el desarrollo.

5.3.1.3. Product Backlog

A continuación, se muestra el Product Backlog, en el que se detallan las tareas, el programador asignado y el estado correspondiente.

En la tabla 5.15 se muestra el Product Backlog

Tabla 5.15. Product Backlog

Historia de usuario	Tarea técnica	Tipo de tarea	Responsable	Estado
HU001	Diseñar e implementar la gestión de usuarios, roles y validaciones de datos duplicados.	Desarrollo/Pruebas	Leonardo Quinaluisa	Terminado
HU002	Configurar los permisos de acceso para Administrador, Secretaria y Empleado.	Desarrollo/Seguridad	Evelin Ganchala	Terminado
HU003	Diseñar el panel principal con indicadores adaptados al rol del usuario.	Diseño/Desarrollo	Leonardo Quinaluisa	Terminado
HU004	Implementar el registro de proveedores con código, identificación y datos de contacto.	Desarrollo/BD	Evelin Ganchala	Terminado
HU005	Desarrollar las funciones de consulta, actualización y desactivación de proveedores.	Desarrollo/Pruebas	Leonardo Quinaluisa	Terminado
HU006	Implementar la gestión de fincas con código, ubicación, estado y validación de unicidad.	Desarrollo/BD	Evelin Ganchala	Terminado

Historia de usuario	Tarea técnica	Tipo de tarea	Responsable	Estado
HU007	Desarrollar el catálogo de variedades de rosas con códigos y nombres únicos.	Desarrollo/BD	Leonardo Quinaluisa	Terminado
HU008	Implementar el registro del personal con información laboral, área y finca asignada.	Desarrollo/BD	Evelin Ganchala	Terminado
HU009	Desarrollar la gestión de permisos y vacaciones con validación de fechas.	Desarrollo/Pruebas	Leonardo Quinaluisa	Terminado
HU010	Diseñar la consulta cronológica del historial laboral de los empleados.	Diseño/Desarrollo	Evelin Ganchala	Terminado
HU011	Implementar el registro de recepciones de flor entregada por proveedores externos.	Desarrollo/BD	Leonardo Quinaluisa	Terminado
HU012	Implementar el registro de recepciones procedentes de las fincas de L.Y Flowers.	Desarrollo/BD	Evelin Ganchala	Terminado
HU013	Permitir el registro de varias variedades, mallas, largos y tallos dentro de una recepción.	Desarrollo/Pruebas	Leonardo Quinaluisa	Terminado
HU014	Diseñar y generar el ticket de recepción con detalles, totales y espacios para firmas.	Diseño/Desarrollo	Evelin Ganchala	Terminado
HU015	Implementar la consulta de recepciones y variedades pendientes de clasificación.	Desarrollo/Consultas	Leonardo Quinaluisa	Terminado
HU016	Desarrollar el registro de tallos exportables con calidad Óptimo o Estándar.	Desarrollo/BD	Evelin Ganchala	Terminado
HU017	Implementar el cálculo y registro de tallos adicionales detectados durante la clasificación.	Desarrollo/Pruebas	Leonardo Quinaluisa	Terminado
HU018	Desarrollar el registro de flor nacional o rechazada con cantidades y observaciones.	Desarrollo/BD	Evelin Ganchala	Terminado

Historia de usuario	Tarea técnica	Tipo de tarea	Responsable	Estado
HU019	Permitir que una recepción sea clasificada en una fecha posterior a su registro.	Desarrollo/Pruebas	Leonardo Quinaluisa	Terminado
HU020	Diseñar el reporte diario de tallos procesados, bunches y resultados por variedad.	Reportes/Desarrollo	Evelin Ganchala	Terminado
HU021	Implementar tarifarios por variedad, calidad, precio y período de vigencia.	Desarrollo/BD	Leonardo Quinaluisa	Terminado
HU022	Desarrollar el cierre de tarifas anteriores y conservar el historial de precios.	Desarrollo/Pruebas	Evelin Ganchala	Terminado
HU023	Implementar el cálculo automático de liquidaciones mediante los tallos exportables y la tarifa vigente.	Desarrollo/Pruebas	Leonardo Quinaluisa	Terminado
HU024	Diseñar la consulta detallada de liquidaciones con cantidades, tarifas, subtotales y total.	Diseño/Reportes	Evelin Ganchala	Terminado
HU025	Implementar la gestión de categorías y productos químicos con unidad y stock mínimo.	Desarrollo/BD	Leonardo Quinaluisa	Terminado
HU026	Desarrollar el registro de entradas de inventario y la actualización automática del stock.	Desarrollo/Pruebas	Evelin Ganchala	Terminado
HU027	Implementar salidas y consumos parciales, validando la disponibilidad del producto.	Desarrollo/Pruebas	Leonardo Quinaluisa	Terminado
HU028	Diseñar alertas para productos cuyo stock sea igual o inferior al mínimo establecido.	Desarrollo/Alertas	Evelin Ganchala	Terminado
HU029	Implementar consultas, exportación de reportes y registro de acciones en la bitácora.	Reportes/Auditoría	Leonardo Quinaluisa	Terminado
General	Configurar Django, PostgreSQL y las variables del entorno de producción en Railway.	Infraestructura	Evelin Ganchala	Terminado

Historia de usuario	Tarea técnica	Tipo de tarea	Responsable	Estado
General	Ejecutar pruebas unitarias, funcionales, de integración y regresión.	Pruebas	Leonardo Quinaluisa	Terminado
General	Elaborar el manual de usuario, el manual técnico y la documentación del despliegue.	Documentación	Ambos	Terminado

5.3.1.4. Iteración Backlog

A continuación, se presentan las iteraciones definidas en el Iteration Backlog, donde se muestra la planificación y ejecución gradual de las historias de usuario priorizadas. Como se presentan en las tablas siguientes.

En la Tabla 5.16 se presentan las historias de usuario planificadas para la Iteración 1.

Tabla 5.16. Iteración 1

ITERACIÓN	ID HU	HISTORIA DE USUARIO	PRIORIDAD	PUNTOS
Iteración 1	HU001	Gestión de usuarios.	Alta	5
	HU002	Asignación de roles y permisos.	Alta	8
	HU003	Consulta del panel principal.	Media	5
Total de puntos Iteración				18

En la Tabla 5.17 se presentan las historias de usuario planificadas para la Iteración 2.

Tabla 5.17. Iteración 2

ITERACIÓN	ID HU	HISTORIA DE USUARIO	PRIORIDAD	PUNTOS
Iteración 2	HU004	Registro de proveedores.	Alta	5
	HU005	Administración de proveedores.	Media	3

ITERACIÓN	ID HU	HISTORIA DE USUARIO	PRIORIDAD	PUNTOS
	HU00 6	Gestión de fincas.	Alta	5
Total de puntos Iteración				13

En la Tabla 5.18 se presentan las historias de usuario planificadas para la Iteración 3; las iteraciones restantes se incluyen en el Anexo B.

Tabla 5.18. Iteración 3

ITERACIÓN	ID HU	HISTORIA DE USUARIO	PRIORIDAD	PUNTOS
Iteración 3	HU00 7	Gestión de variedades.	Alta	5
	HU00 8	Registro de personal.	Media	5
	HU00 9	Gestión de permisos y vacaciones.	Media	5
Total de puntos Iteración				15

5.3.2. Resultados de la fase de diseño

5.3.2.1. Caso de uso

En este apartado se muestran los diagramas de casos de uso, mediante los cuales se representan los requerimientos funcionales y se facilita la comprensión de las actividades que cada usuario puede realizar dentro del Sistema. La metodología Extreme Programming (XP) establece el levantamiento de requerimientos mediante historias de usuario; no obstante, la inclusión de casos de uso en este proyecto surge de la necesidad de disponer de una representación estructurada que complemente el análisis funcional. Por esta razón, los diagramas de casos de uso permitieron organizar y representar las funcionalidades establecidas previamente en las historias de usuario, manteniendo coherencia entre el análisis y el diseño del Sistema.

En la Figura 5.9 se presenta el diagrama general de casos de uso al acceso del Sistema FlorLY, en el cual se representa la interacción de los actores Administrador, secretaria y Empleado con las funcionalidades asignadas a cada perfil.

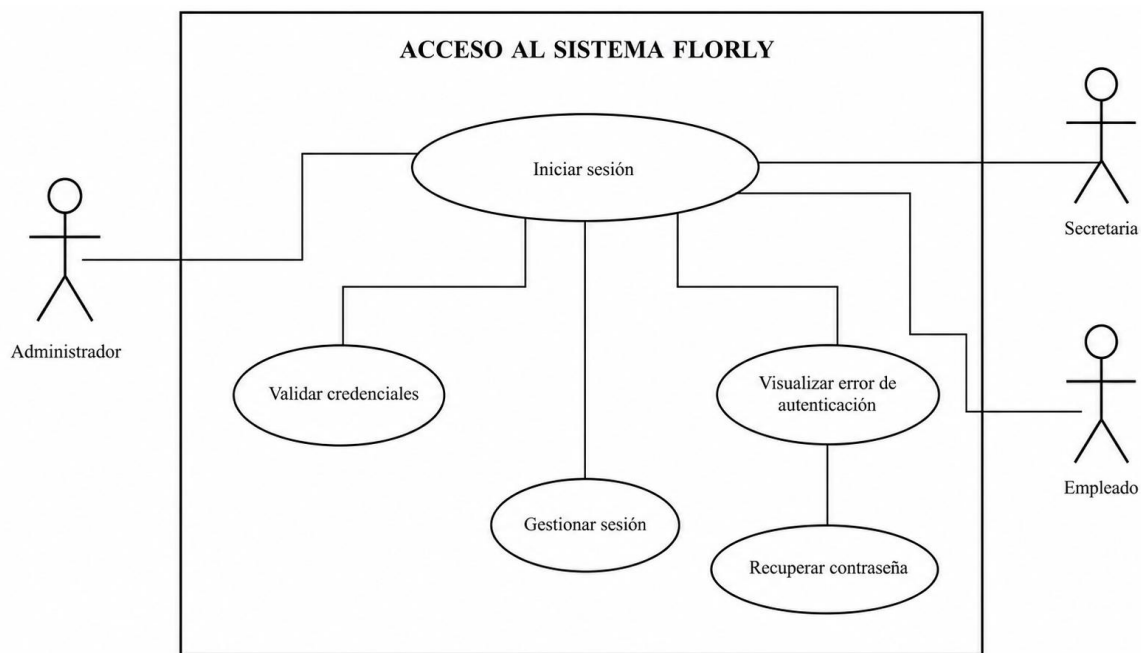


Figura 5.9. Diagrama de casos de uso acceso al Sistema

En la Figura 5.10 se presenta el caso de uso correspondiente a la vista general de FlorLY, donde se representan las principales funcionalidades disponibles para los usuarios de acuerdo con su rol, permitiendo visualizar de manera integral los módulos que conforman el Sistema.

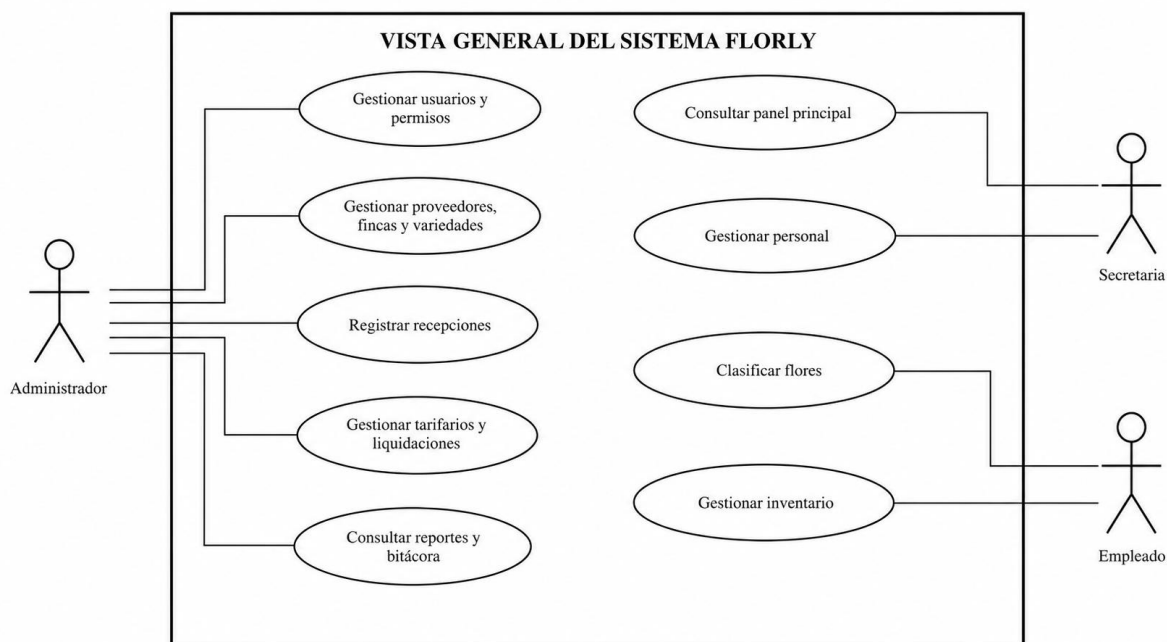


Figura 5.10. Diagrama de caso de uso de la vista general del Sistema FlorLY

En la Figura 5.11 se representa el caso de uso relacionado con la gestión de proveedores, fincas y variedades, donde se describe la interacción del Administrador y la secretaria con las funciones de registro, consulta, actualización y control de estado de los catálogos utilizados por FlorLY.

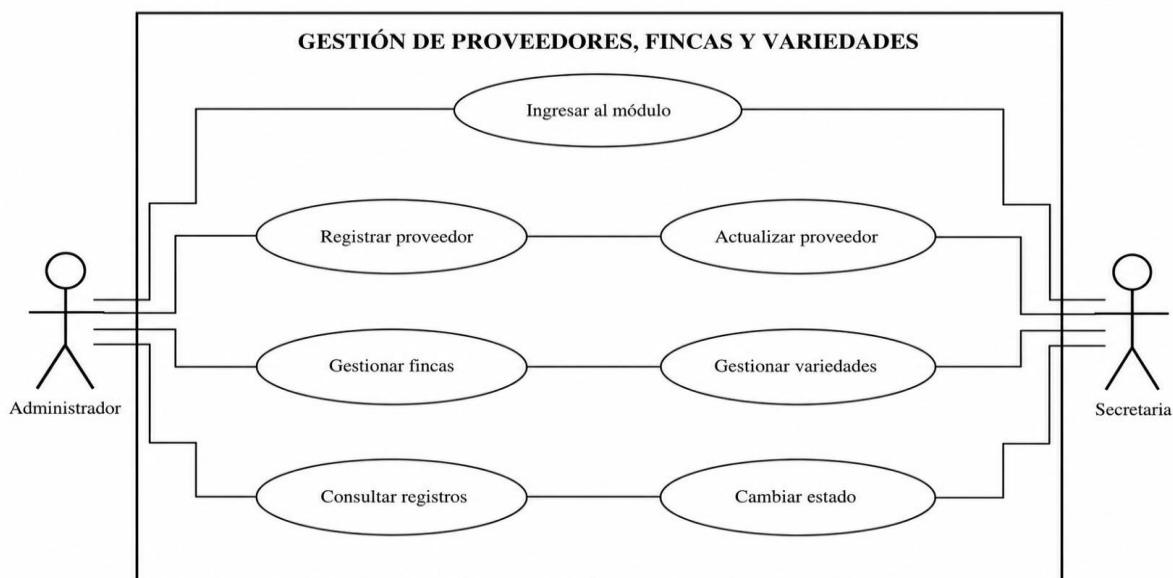


Figura 5.11. Diagrama de caso de uso de gestión de proveedores, fincas y variedades

En la Figura 5.12 se presenta el caso de uso correspondiente al proceso de recepción de flores, donde la secretaria y el Empleado registran el origen de la entrega, las variedades, las mallas, el largo y la cantidad de tallos, además de generar el ticket de recepción.

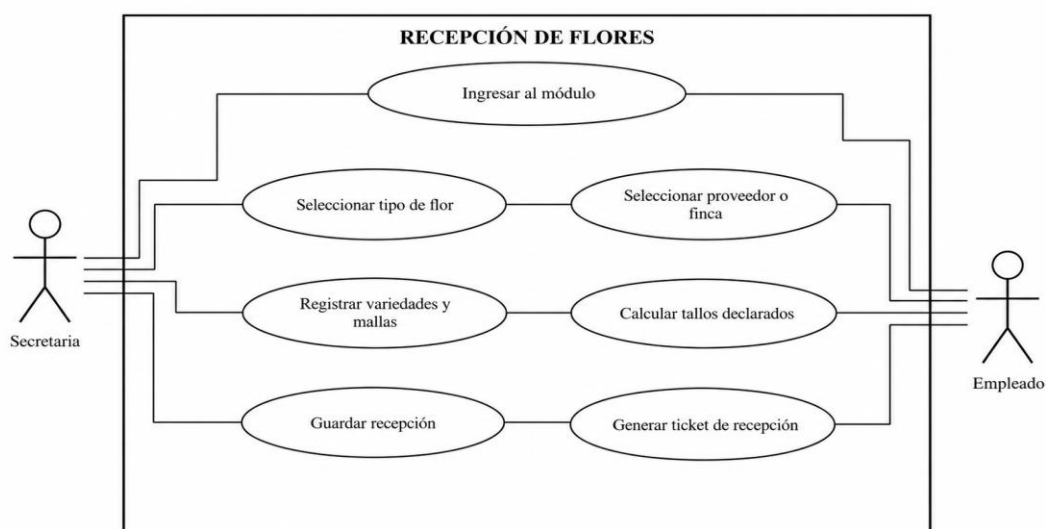


Figura 5.12. Diagrama de caso de uso del proceso de recepción de flores

En la Figura 5.13 se representa el caso de uso asociado con la clasificación de flores, mediante el cual el Empleado consulta las recepciones pendientes, registra los tallos exportables y rechazados, identifica tallos adicionales y finaliza el proceso de clasificación.

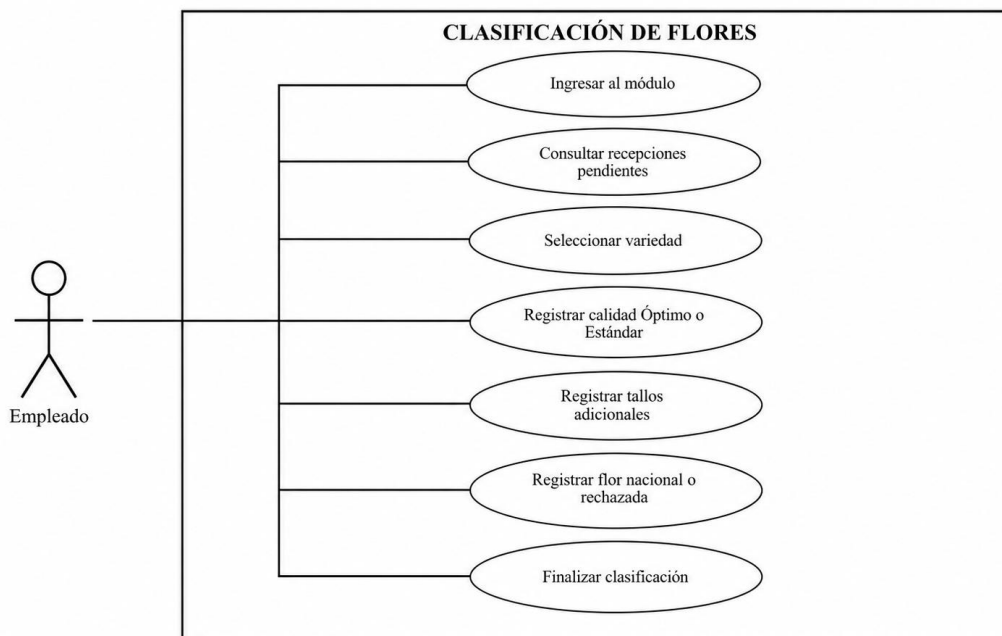


Figura 5.13. Diagrama de caso de uso de clasificación de flores

En la Figura 5.14 se muestra el caso de uso correspondiente a la gestión de tarifarios y liquidaciones, donde el Administrador y la secretaria registran precios por vigencia, conservan su historial y generan automáticamente el valor que debe pagarse por los tallos exportables.

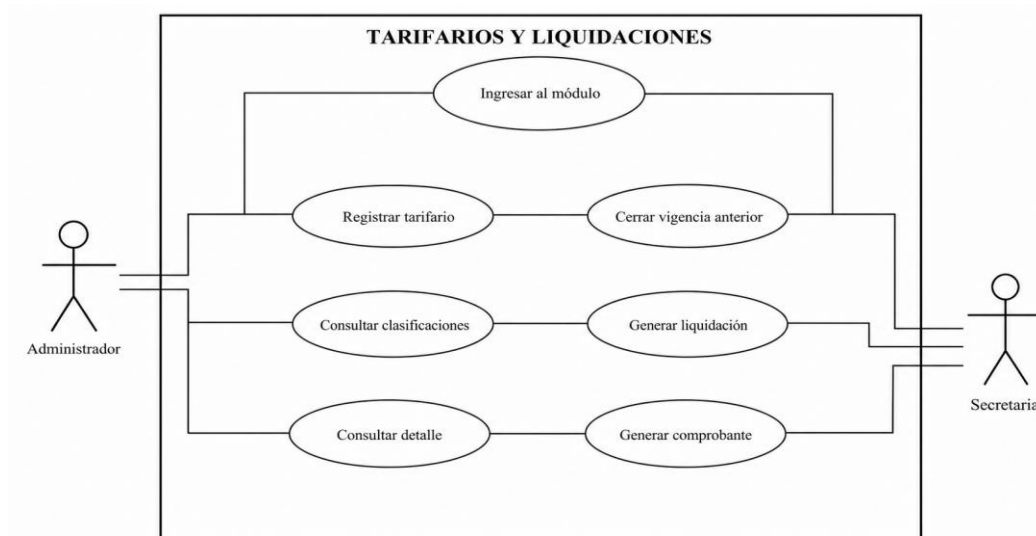


Figura 5.14. Diagrama de caso de uso de tarifarios y liquidaciones

En la Figura 5.15 se presenta el caso de uso relacionado con la gestión del personal, donde el Administrador y la secretaria registran trabajadores, asignan áreas y fincas, controlan permisos y vacaciones, y consultan el historial laboral.

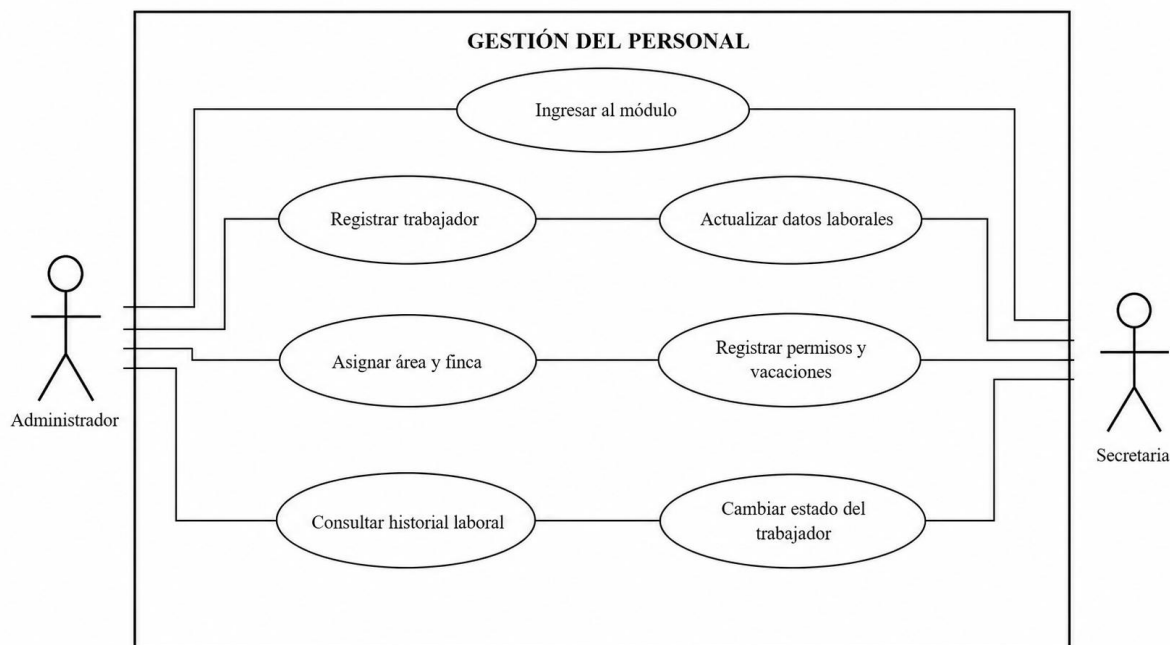


Figura 5.15. Diagrama de caso de uso de gestión del personal

En la Figura 5.16 se representa el caso de uso correspondiente al control de inventario de productos químicos, donde el Administrador y el Empleado gestionan categorías y productos, registran entradas y salidas, controlan consumos parciales y consultan alertas de stock mínimo.

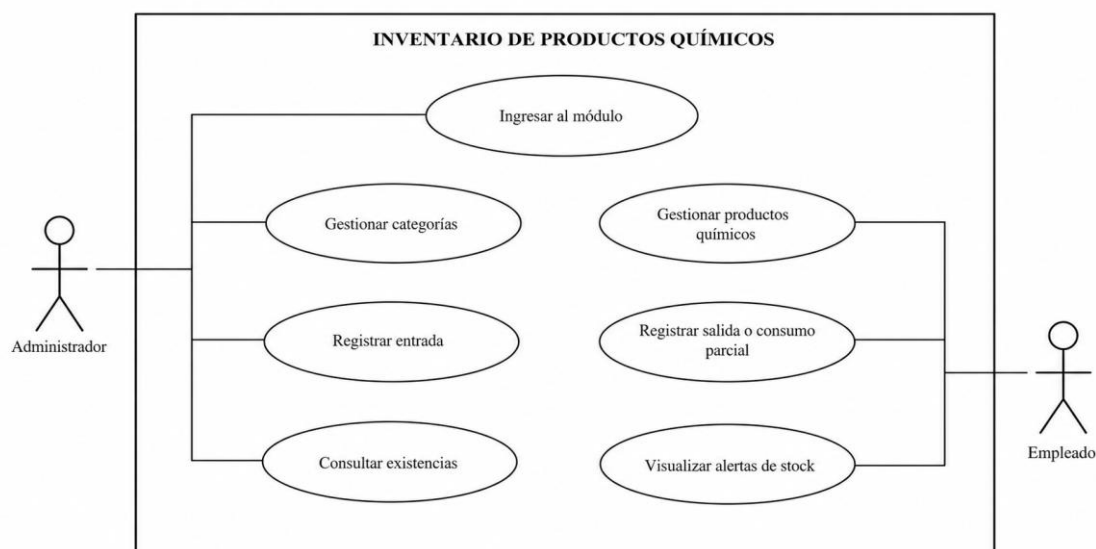


Figura 5.16. Diagrama de caso de uso de inventario de productos químicos

En la Figura 5.17 se presenta el caso de uso asociado con la generación de reportes y el seguimiento del Sistema, permitiendo al Administrador y a la secretaria consultar indicadores, aplicar filtros, visualizar información histórica y exportar los resultados de las operaciones realizadas.

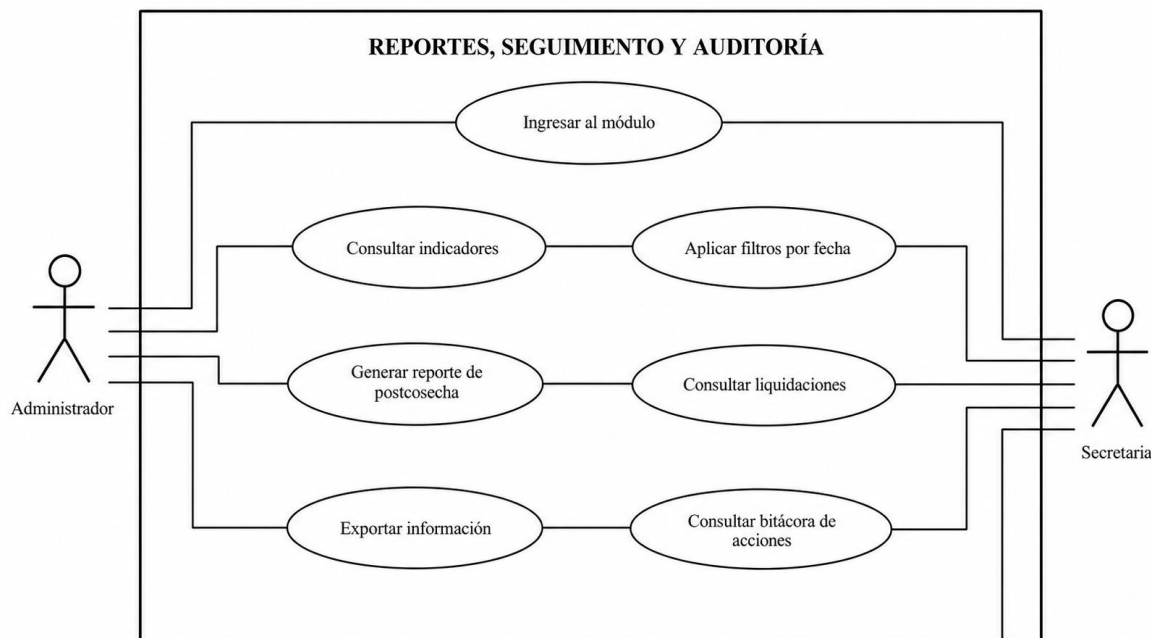


Figura 5.17. Diagrama de caso de uso de reportes, seguimiento y auditoría

5.3.2.2. Modelado de la base de datos

El modelo relacional vincula usuarios, roles, proveedores, fincas, variedades, personal, recepciones, detalles de clasificación, tarifarios, liquidaciones, pagos, productos, movimientos y bitácora. Las claves foráneas mantienen la trazabilidad y las restricciones evitan registros huérfanos o inconsistentes.

En la figura 5.18 se observa el modelo de la base de datos.

5.3.2.3. Arquitectura del Sistema

El Sistema FlorLY implementa la arquitectura Modelo-Vista-Plantilla (MVT) proporcionada por Django, la cual permite separar la presentación de la información, la lógica de procesamiento y la gestión de los datos. Esta organización facilita el mantenimiento, la seguridad y la ampliación de los diferentes módulos del Sistema.

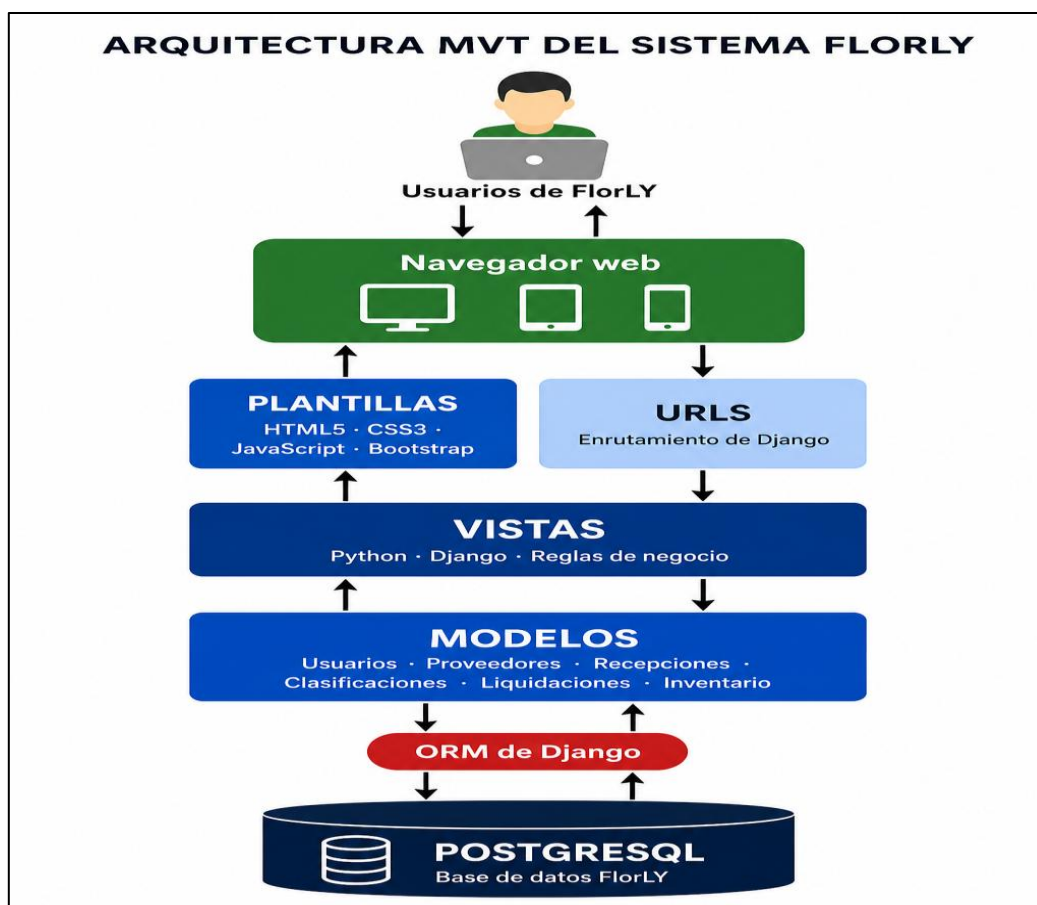


Figura 5.19. Arquitectura MVT del Sistema FlorLY

- Capa de presentación (Template/Frontend): Desarrollada mediante plantillas de Django, HTML5, CSS3, JavaScript y Bootstrap. Para la estructura visual se utilizó una plantilla de ThemeWagon, adaptada a la identidad de L.Y Flowers. La interfaz es responsiva y permite el acceso desde computadoras, tabletas y dispositivos móviles.
- Capa de lógica del Sistema (View): Desarrollada en Python mediante el framework Django. Se encarga de recibir las solicitudes de los usuarios, validar formularios, aplicar permisos y ejecutar las reglas de negocio. Esta capa controla procesos como el registro de recepciones, la clasificación de flores, la detección de tallos adicionales, la gestión de tarifarios, el cálculo automático de liquidaciones, los movimientos de inventario y la generación de reportes.

- Capa de datos (Model): Está conformada por los modelos de Django y el Sistema gestor de bases de datos PostgreSQL. El ORM de Django actúa como intermediario entre la lógica del Sistema y la base de datos, permitiendo realizar operaciones de registro, consulta, actualización y eliminación sin ejecutar directamente instrucciones SQL. También administra las relaciones, restricciones e integridad de la información correspondiente a usuarios, proveedores, fincas, variedades, personal, recepciones, clasificaciones, tarifarios, liquidaciones, inventario y bitácora.

5.3.2.4. Diagramas BPMN

Este modelado permite representar de manera gráfica, clara y organizada los flujos de trabajo, los roles y las decisiones que intervienen en el Sistema. Mediante estos diagramas se modelaron los procesos internos de gestión de postcosecha realizados en la empresa L.Y Flowers.

5.3.2.4.1. Postcosecha

La Figura 5.20 presenta el diagrama de procesos (BPMN) correspondiente a la gestión de Postcosecha. El flujo integra la recepción de flor, la identificación de su procedencia, la validación de los datos y el envío del registro al proceso de clasificación, garantizando la trazabilidad de cada ingreso.

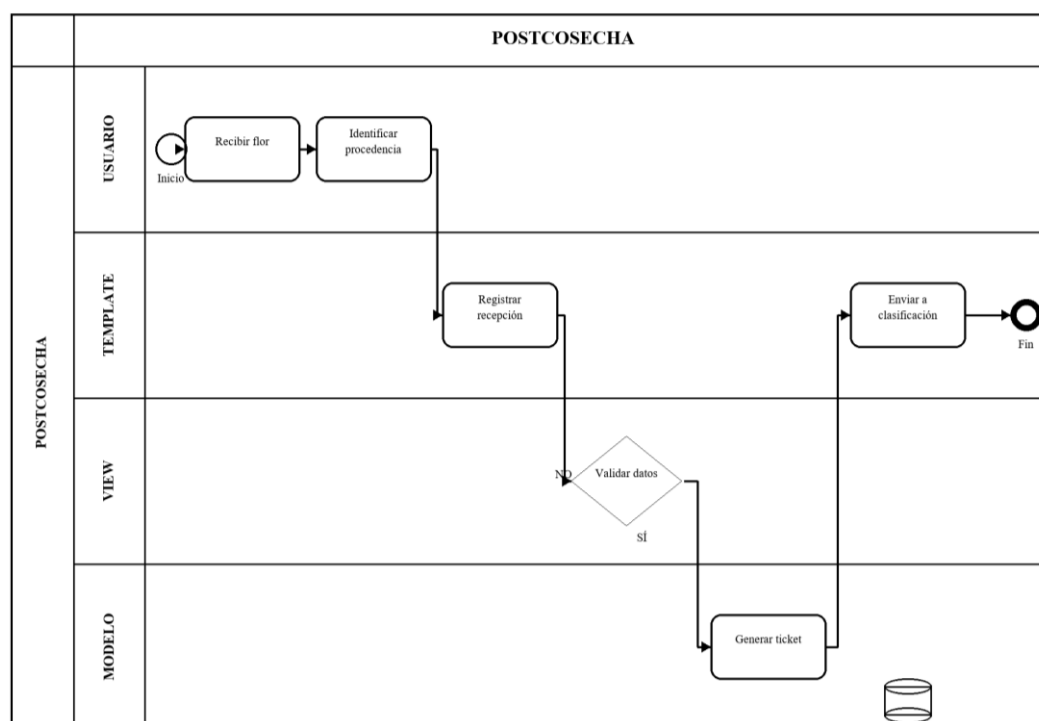


Figura 5.20. Diagrama BPMN - Postcosecha

5.3.2.4.2. Recepción de flor

En la Figura 5.21 se detalla el proceso de recepción de flor propia y de proveedores. El diagrama contempla la selección del productor o finca, el registro de variedades, mallas, longitudes y tallos, así como la validación de la información antes de generar el comprobante de recepción.

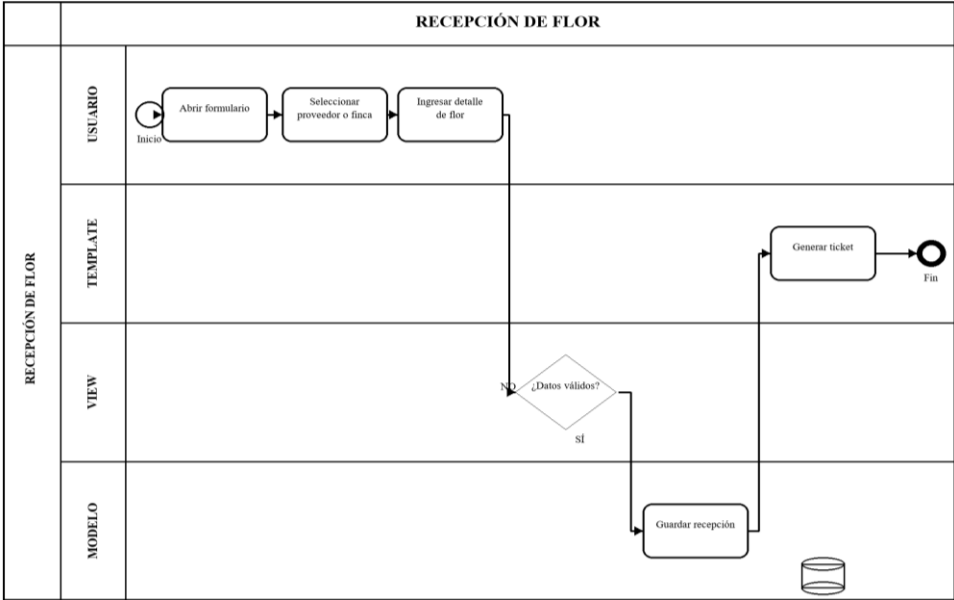


Figura 5.21. Diagrama BPMN - Recepción de flor

5.3.2.4.3. Clasificación de flor

La Figura 5.22 representa el proceso de clasificación de la flor recibida. Se registran los tallos óptimos, estándar y rechazados por variedad; además, el Sistema permite identificar tallos adicionales y consolidar los resultados para el control de la producción.

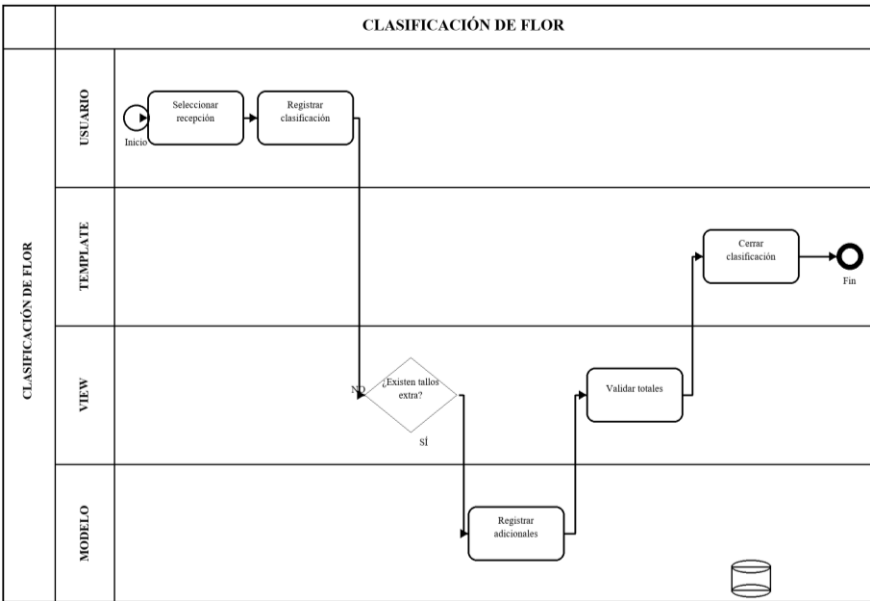


Figura 5.22. Diagrama BPMN - Clasificación de flor

5.3.2.4.4. Proveedores, fincas y variedades

La Figura 5.23 muestra el proceso de administración de proveedores, fincas y variedades. El usuario registra o actualiza la información, el Sistema valida los campos obligatorios y conserva los datos necesarios para utilizarlos posteriormente en la recepción y clasificación de la flor.

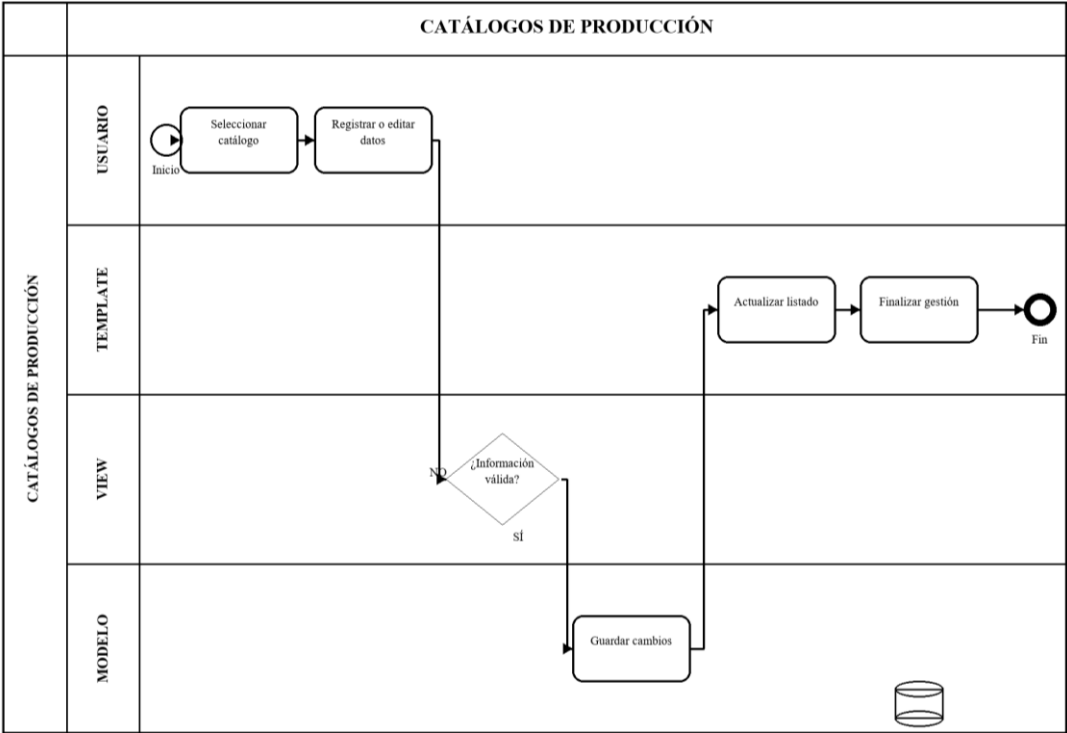


Figura 5.23. Diagrama BPMN - Proveedores, fincas y variedades

5.3.2.4.5. Tarifas por variedad

La Figura 5.24 expone el flujo para gestionar las tarifas por variedad, calidad y período de vigencia. El Sistema verifica que los valores sean válidos y conserva el historial de precios, evitando modificar las tarifas utilizadas en liquidaciones anteriores.

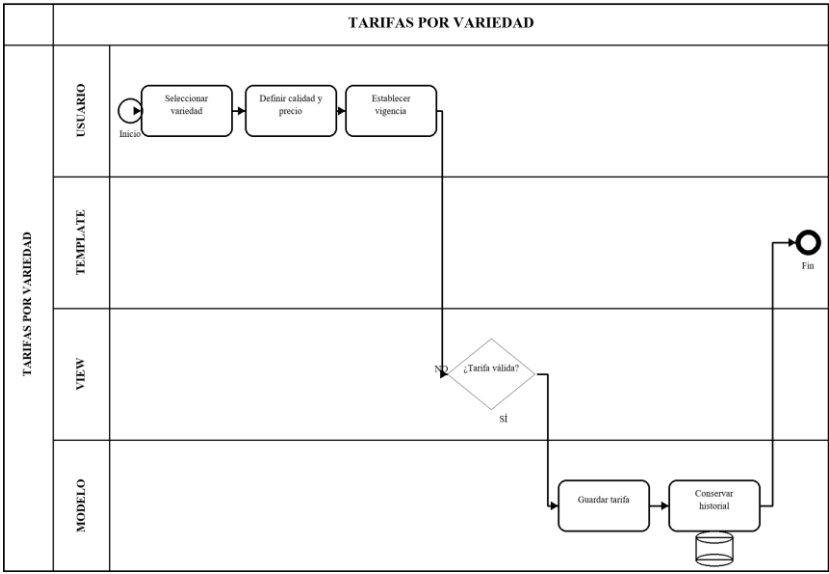


Figura 5.24. Diagrama BPMN - Tarifas por variedad

5.3.2.4.6. Liquidación de productores

En la Figura 5.25 se representa el proceso de liquidación de productores. El Sistema obtiene los tallos exportables clasificados, consulta la tarifa vigente en la fecha de recepción, calcula el valor correspondiente y genera la liquidación para su revisión y emisión.

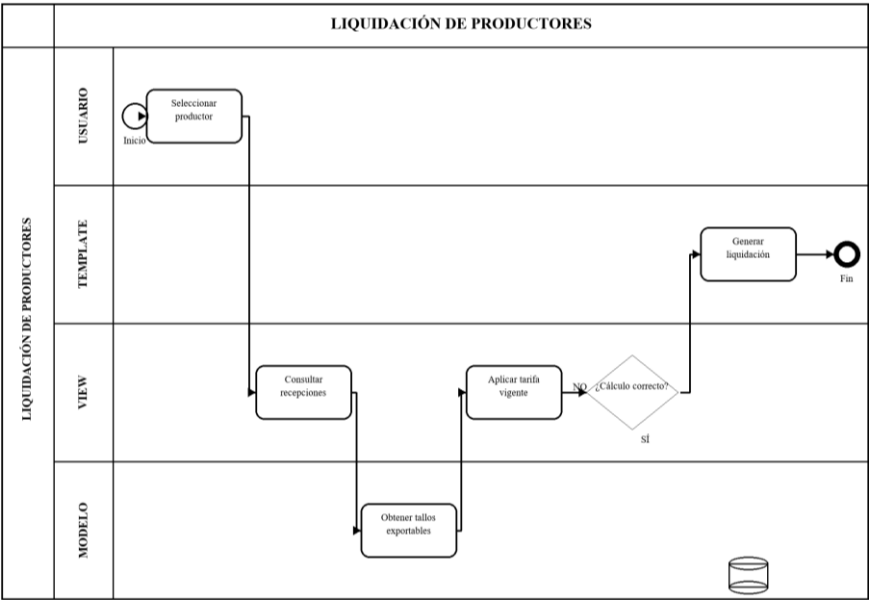


Figura 5.25. Diagrama BPMN - Liquidación de productores

5.3.2.4.7. Gestión del personal

La Figura 5.26 describe el proceso para administrar la información del personal de L.Y Flowers. Incluye el registro de empleados, la actualización de sus datos y el control de permisos, vacaciones e historial, con las validaciones necesarias antes de almacenar cada novedad.

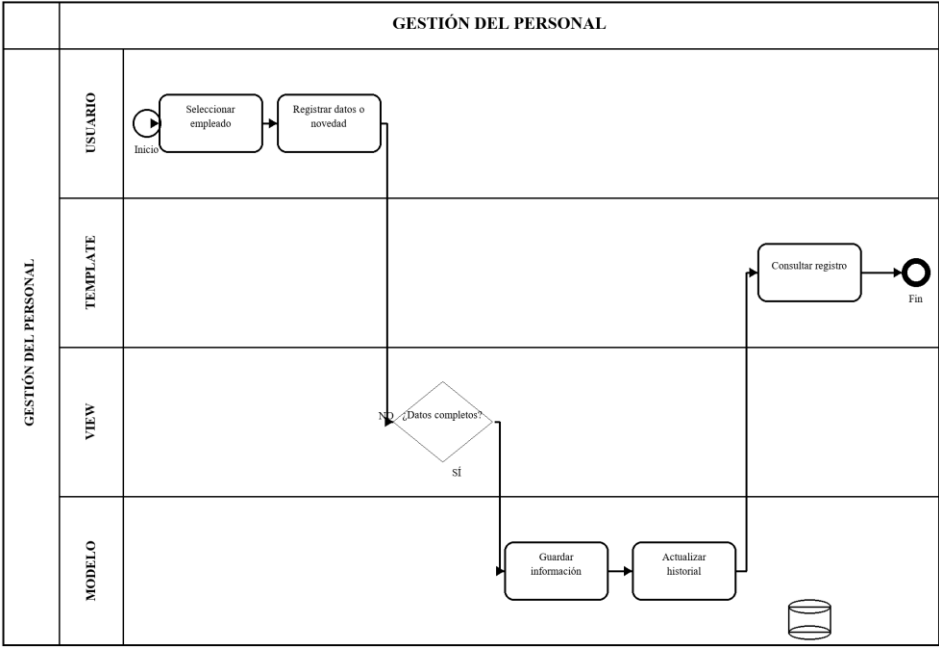


Figura 5.26. Diagrama BPMN - Gestión del personal

5.3.2.4.8. Inventario Químico

La Figura 5.27 presenta el proceso de control del inventario de químicos. El flujo comprende el registro de productos y categorías, las entradas, salidas y consumos parciales, la actualización automática de existencias y la generación de alertas cuando el stock alcanza el nivel mínimo.

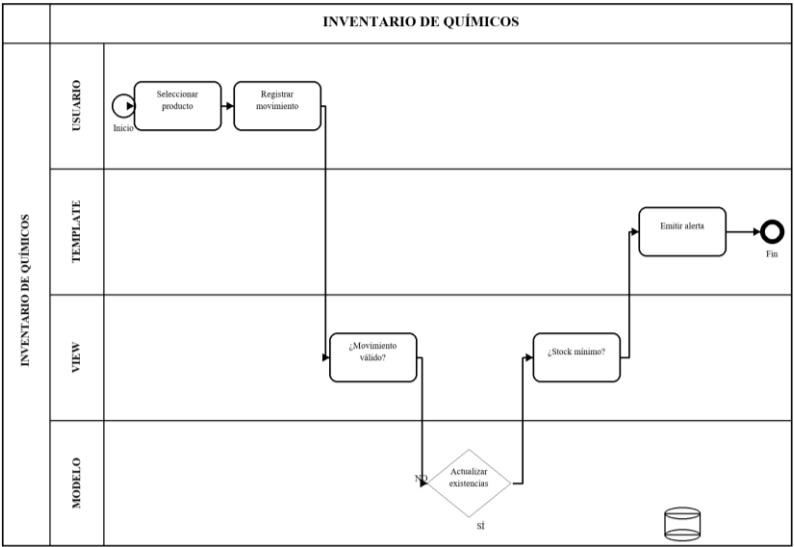


Figura 5.27. Diagrama BPMN - Inventario de químicos

5.3.2.4.9. Reportes de producción

La Figura 5.28 muestra el proceso de generación de reportes de producción. El usuario define los filtros requeridos, el Sistema consolida la información de recepciones y clasificaciones, presenta los resultados por fecha, variedad o procedencia y permite obtener el reporte correspondiente.

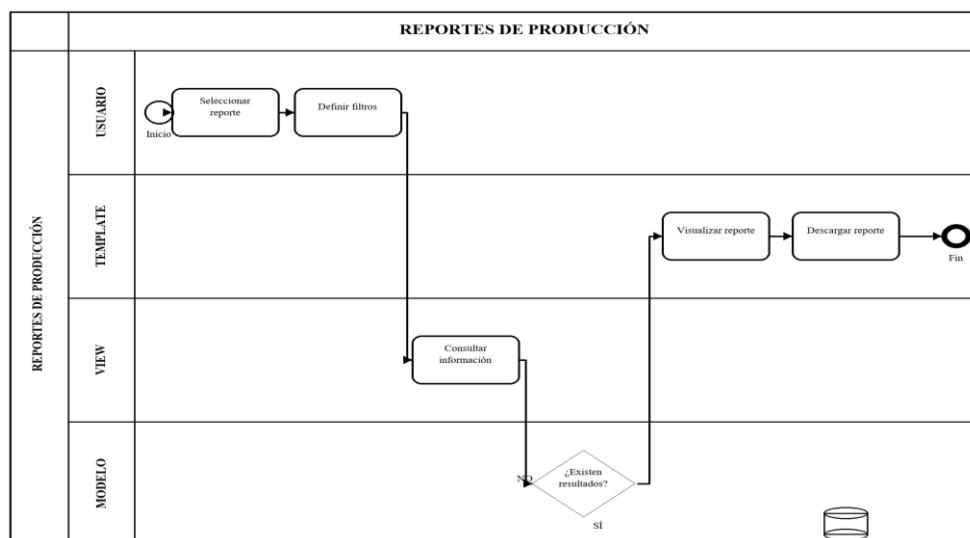


Figura 5.28. Diagrama BPMN - Reportes de producción

5.3.2.4.10. Trazabilidad y auditoría

La Figura 5.29 expone el proceso de trazabilidad y auditoría del Sistema FlorLY. Cada operación relevante queda asociada al usuario, la fecha y la acción realizada; posteriormente, el administrador puede aplicar filtros, revisar el historial y utilizar esta información para el seguimiento de los procesos.

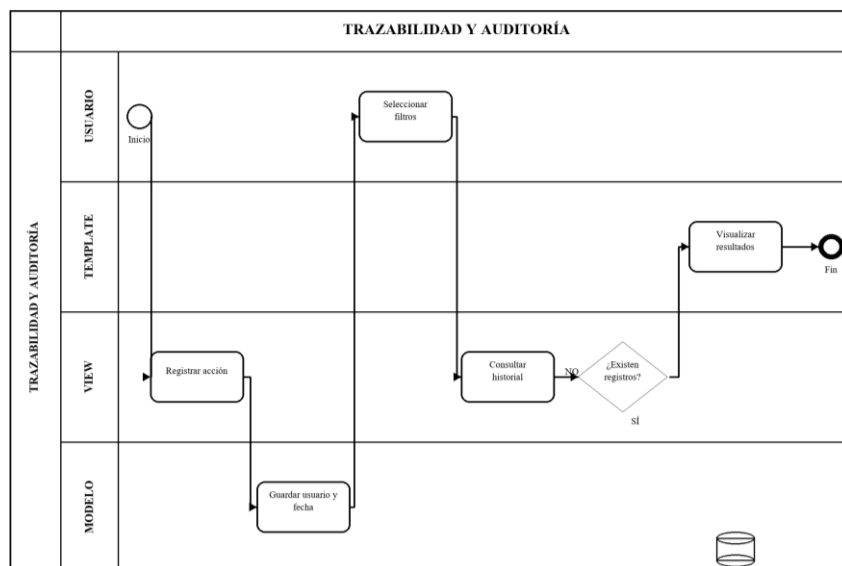


Figura 5.29. Diagrama BPMN - Trazabilidad y auditoría

5.3.3. Resultados de la fase de codificación

Esta fase se desarrolló bajo la metodología Extreme Programming (XP), mediante historias de usuario organizadas por iteraciones. Para la construcción del Sistema se utilizó Django con Python, aplicando la arquitectura Modelo–Vista–Plantilla (MVT), junto con PostgreSQL para la gestión de los datos.

Durante este proceso se implementaron progresivamente los módulos destinados a optimizar la gestión interna de postcosecha de la empresa L.Y Flowers.

La codificación del Sistema se realizó de manera incremental mediante iteraciones, conforme a la planificación establecida en la metodología XP y aplicando pruebas continuas para verificar el correcto funcionamiento de cada módulo. En las Tablas 5.19 y 5.20 se presenta la codificación correspondiente a las historias de usuario HU001 y HU002 de la Iteración 1; las historias restantes se detallan en el Anexo C.

Tabla 5.19. Codificación HU001

ITERACIÓN 1 – HU001		
HISTORIA USUARIO	DE	HU001 – Gestión de usuarios.
PRUEBAS REALIZADAS		Prueba unitaria de validación: Se verificó que los datos requeridos para gestión de usuarios sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO		<pre> 178 def guardarUsuario(request): 179 username = request.POST.get('username', '').strip() 180 password = request.POST.get('password', '') 181 rol = request.POST.get('rol', '') 182 if rol not in ROLES_SISTEMA or User.objects.filter(username__iexact 183 messages.error(request, 'El usuario ya existe o el rol no es vá 184 return redirect('nuevoUsuario') 185 if password != request.POST.get('password_confirmacion', ''): 186 messages.error(request, 'Las contraseñas no coinciden') 187 return redirect('nuevoUsuario') 188 try: 189 validate_password(password) 190 except ValidationError as error: 191 messages.error(request, ' '.join(error.messages)) 192 return redirect('nuevoUsuario') 193 grupo, _ = Group.objects.get_or_create(name=rol) 194 usuario = User.objects.create_user(195 username=username, </pre>

ITERACIÓN 1 – HU001	
REFACTORIZACIÓN	Se reorganizó la implementación de gestión de usuarios, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla 5.20. Codificación HU002

ITERACIÓN 1 – HU002	
HISTORIA DE USUARIO	HU002 – Asignación de roles y permisos.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para asignación de roles y permisos sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 215 def procesarUsuario(request): 216 usuario = get_object_or_404(User, pk=request.POST['id']) 217 rol = request.POST.get('rol', '') 218 if rol not in ROLES_SISTEMA: 219 messages.error(request, 'Rol no válido') 220 return redirect('editarUsuario', id=usuario.pk) 221 activo = request.POST.get('is_active') == 'on' 222 if usuario == request.user and (not activo or rol != 'Administrador'): 223 messages.error(request, 'No puede desactivar ni retirar su prop 224 return redirect('editarUsuario', id=usuario.pk) 225 if usuario.is_superuser and (not activo or rol != 'Administrador'): 226 messages.error(request, 'Una cuenta superadministradora no pued 227 return redirect('editarUsuario', id=usuario.pk) 228 grupo, _ = Group.objects.get_or_create(name=rol) 229 usuario.first_name = request.POST.get('nombre', '').strip() 230 usuario.last_name = request.POST.get('apellido', '').strip() 231 usuario.is_active = activo 232 usuario.groups.set([grupo]) </pre>

ITERACIÓN 1 – HU002	
REFACTORIZACIÓN	Se reorganizó la implementación de asignación de roles y permisos, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

5.3.4. Resultados de la fase de pruebas

El proceso de pruebas se desarrolló bajo la metodología Extreme Programming (XP). Cada funcionalidad fue validada mediante casos de prueba definidos a partir de los criterios de aceptación de las historias de usuario. Posteriormente, se implementó el código necesario y se ejecutaron nuevamente las pruebas para comprobar el correcto funcionamiento de los módulos, asegurando que el Sistema responda a las necesidades de la empresa L.Y Flowers.

Para esta fase se consideraron las historias de usuario más relevantes relacionadas con la gestión de postcosecha, para los casos de prueba.

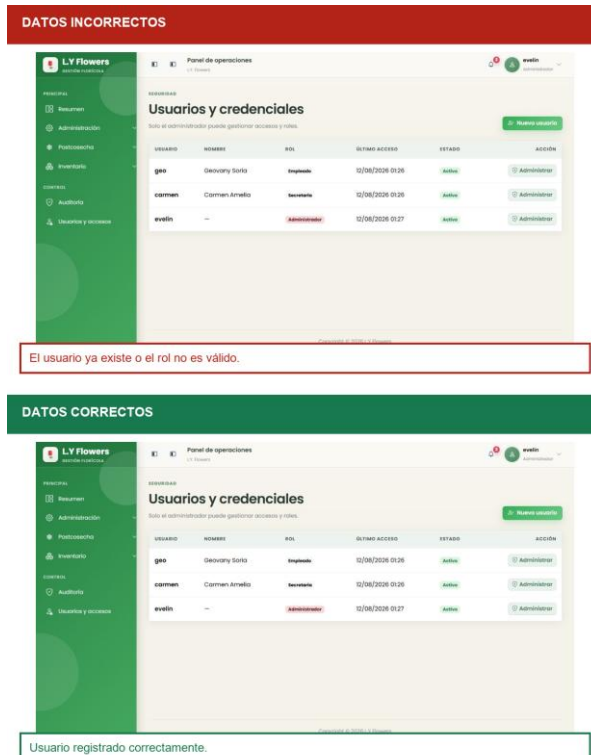
En la Tabla 5.21 se presenta la ejecución y los resultados del caso de prueba correspondiente a la historia de usuario HU001.

Tabla 5.21. Caso de prueba HU001

ITERACIÓN 1 – Caso de Prueba 1	
HISTORIA USUARIO	DE HU001 – Gestión de usuarios.
PRUEBA EJECUTADA	Ingresar al módulo Gestión de usuarios. Seleccionar la opción Registrar usuario. Ingresar datos válidos y asignar un rol. Guardar y comprobar que la cuenta aparezca en el listado.
RESULTADOS OBTENIDOS	El Sistema registró la cuenta y asignó el rol seleccionado. Al ingresar información incompleta o duplicada, impidió el registro y presentó el mensaje de validación.

ITERACIÓN 1 – Caso de Prueba 1

EVIDENCIA



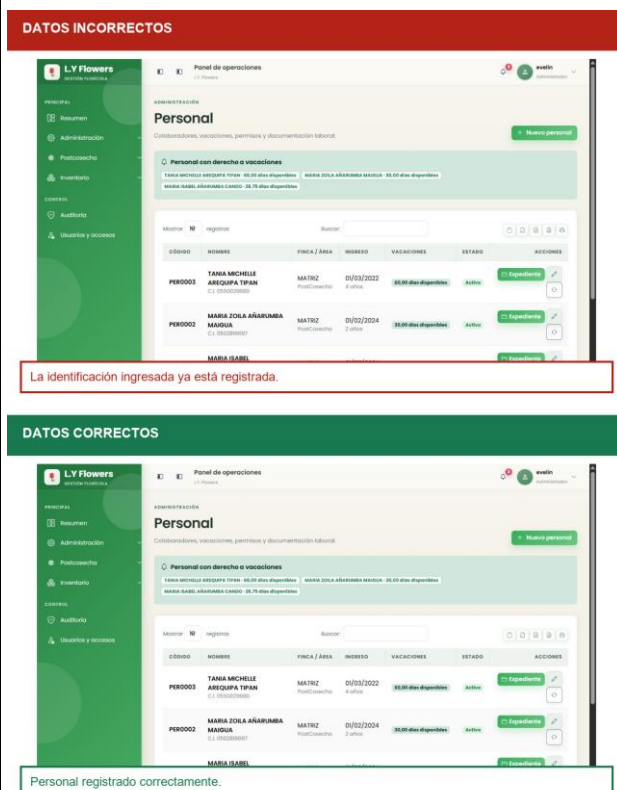
En la Tabla 5.22 se presenta la ejecución y los resultados del caso de prueba correspondiente a la historia de usuario HU008.

Tabla 5.22. Caso de prueba HU008

ITERACIÓN 3 – Caso de Prueba 2	
HISTORIA DE USUARIO	HU008 – Registro de personal.
PRUEBA EJECUTADA	Ingresar al módulo Gestión del personal. Completar la ficha laboral del empleado. Guardar el registro. Verificar la información en el listado de personal.
RESULTADOS OBTENIDOS	La ficha del empleado se almacenó correctamente. Cuando se ingresó una identificación repetida o se dejaron campos obligatorios vacíos, el Sistema rechazó la operación.

ITERACIÓN 3 – Caso de Prueba 2

EVIDENCIA



En la Tabla 5.23 se presenta la ejecución y los resultados del caso de prueba correspondiente a la historia de usuario HU011.

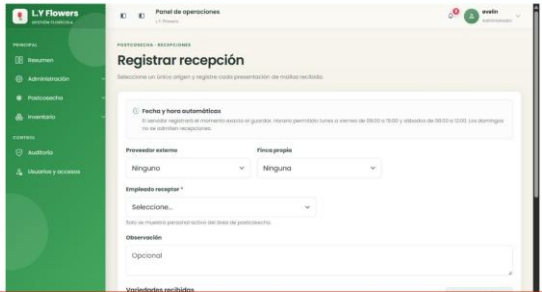
Tabla 5.23. Caso de prueba HU011

ITERACIÓN 4 – Caso de Prueba 3		
HISTORIA USUARIO	DE	HU011 – Registro de recepción de flor de proveedor.
PRUEBA EJECUTADA		Ingresa al módulo Recepciones. Seleccionar un proveedor activo. Registrar variedad, mallas y tallos por malla. Guardar la recepción y verificar el ticket generado.
RESULTADOS OBTENIDOS		La recepción fue almacenada con su proveedor y detalle de variedades. El Sistema evitó guardar cuando faltó el origen o se ingresaron cantidades iguales o menores que cero.

ITERACIÓN 4 – Caso de Prueba 3

EVIDENCIA

DATOS INCORRECTOS



Seleccione un proveedor y registre cantidades mayores que cero.

DATOS CORRECTOS



Recepción guardada exitosamente.

En la Tabla 5.24 se presenta la ejecución y los resultados del caso de prueba correspondiente a la historia de usuario HU016.

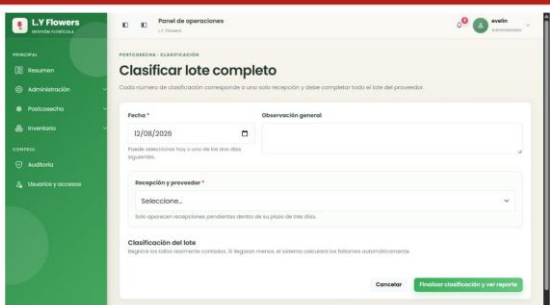
Tabla 5.24. Caso de prueba HU016

ITERACIÓN 6 – Caso de Prueba 4	
HISTORIA USUARIO	DE HU016 – Registro de clasificación exportable.
PRUEBA EJECUTADA	Ingresar al módulo Clasificaciones. Seleccionar una recepción pendiente. Registrar tallos óptimos, estándar y nacionales. Guardar y comprobar el cierre de la clasificación.
RESULTADOS OBTENIDOS	La clasificación se registró con las cantidades correspondientes. Ante valores negativos, filas incompletas o totales inconsistentes, el Sistema no almacenó la información.

ITERACIÓN 6 – Caso de Prueba 4

EVIDENCIA

DATOS INCORRECTOS



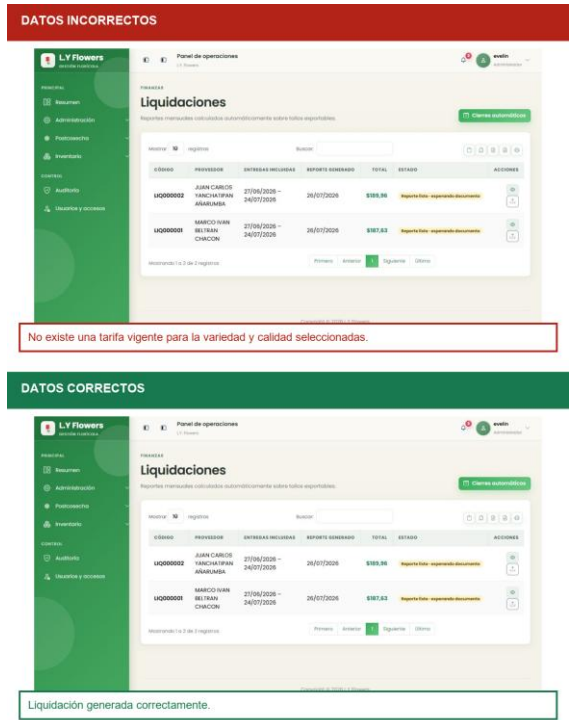
DATOS CORRECTOS



En la Tabla 5.25 se presenta la ejecución y los resultados del caso de prueba correspondiente a la historia de usuario HU023.

Tabla 5.25. Caso de prueba HU023

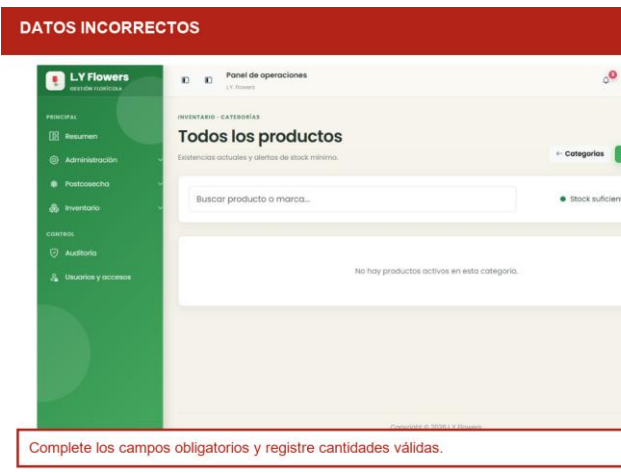
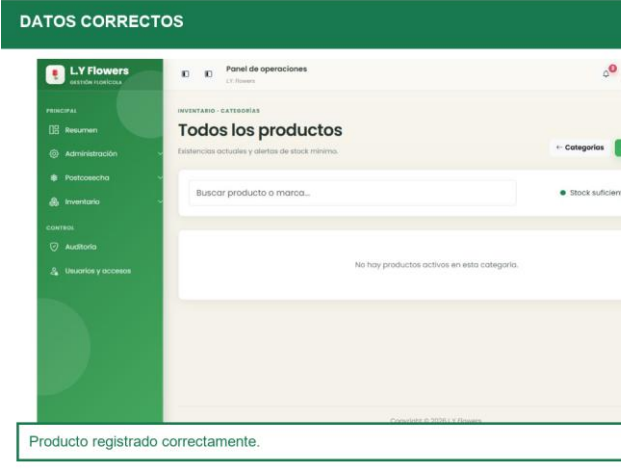
ITERACIÓN 8 – Caso de Prueba 5		
HISTORIA USUARIO	DE	HU023 – Generación automática de liquidaciones.
PRUEBA EJECUTADA		Ingresar al módulo Liquidaciones. Seleccionar el período y el productor. Generar la liquidación. Verificar tallos exportables, tarifa aplicada, subtotales y total.

ITERACIÓN 8 – Caso de Prueba 5	
RESULTADOS OBTENIDOS	El Sistema calculó el pago utilizando únicamente tallos exportables y la tarifa vigente en la fecha de recepción. Cuando no existió una tarifa aplicable, informó el problema y no generó valores incorrectos.
EVIDENCIA	 The evidence consists of two screenshots of the 'Liquidaciones' (Liquidations) module in the 'LY flowers' system. The top screenshot, titled 'DATOS INCORRECTOS', shows a table with two rows of data. The first row, for 'JUAN CARLOS VASCONI' (ID: UQ000002), has a total of \$109.56. The second row, for 'MARCO IVAN CHACON' (ID: UQ000008), has a total of \$187.82. A red box highlights a message: 'No existe una tarifa vigente para la variedad y calidad seleccionadas.' The bottom screenshot, titled 'DATOS CORRECTOS', shows the same table with the same data, but a green box highlights a message: 'Liquidación generada correctamente.'

En la Tabla 5.26 se presenta la ejecución y los resultados del caso de prueba correspondiente a la historia de usuario HU025.

Tabla 5.26. Caso de prueba HU025

ITERACIÓN 9 – Caso de Prueba 6	
HISTORIA USUARIO	DE HU025 – Gestión de categorías y productos químicos.
PRUEBA EJECUTADA	Ingresar al módulo Inventario. Seleccionar la opción Registrar producto. Completar categoría, unidad, stock y stock mínimo. Guardar y consultar el producto registrado.

ITERACIÓN 9 – Caso de Prueba 6	
RESULTADOS OBTENIDOS	El producto quedó registrado con su categoría y unidad de medida. Los datos incompletos, códigos duplicados o cantidades inválidas fueron rechazados.
EVIDENCIA	DATOS INCORRECTOS  Complete los campos obligatorios y registre cantidades válidas. DATOS CORRECTOS  Producto registrado correctamente.

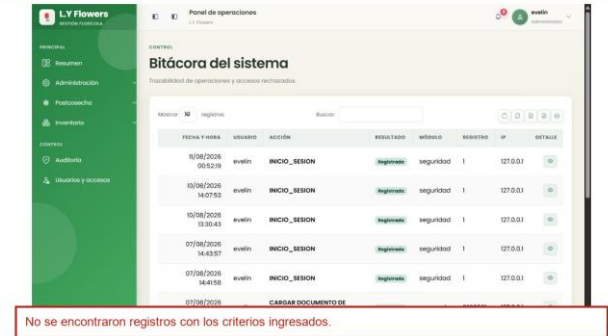
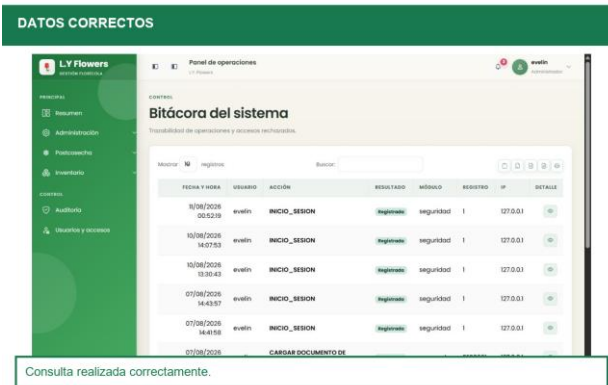
En la Tabla 5.27 se presenta la ejecución y los resultados del caso de prueba correspondiente a la historia de usuario HU020.

Tabla 5.27. Caso de prueba HU020

ITERACIÓN 7 – Caso de Prueba 7		
HISTORIA USUARIO	DE	HU020 – Reporte diario de postcosecha.
PRUEBA EJECUTADA		Ingresar al módulo Reportes. Seleccionar el reporte diario de postcosecha. Definir una fecha válida. Generar y revisar los totales por variedad.
RESULTADOS OBTENIDOS		El reporte mostró los tallos procesados y los totales por variedad para la fecha seleccionada. Cuando el rango fue inválido o no existieron registros, el Sistema presentó la advertencia correspondiente.
EVIDENCIA		DATOS INCORRECTOS No existen datos para los filtros seleccionados. DATOS CORRECTOS Reporte generado correctamente.

En la Tabla 5.28 se presenta la ejecución y los resultados del caso de prueba correspondiente a la historia de usuario HU029.

Tabla 5.28. Caso de prueba HU029

ITERACIÓN 10 – Caso de Prueba 8		
HISTORIA DE USUARIO	DE	HU029 – Consulta de reportes y bitácora.
PRUEBA EJECUTADA		Ingresar al módulo Auditoría. Aplicar filtros por usuario, módulo o fecha. Ejecutar la consulta. Verificar usuario, acción, fecha y registro afectado.
RESULTADOS OBTENIDOS		La bitácora presentó las acciones coincidentes con los filtros. Cuando se utilizaron fechas inconsistentes o criterios sin coincidencias, no se mostraron registros ajenos a la búsqueda.
EVIDENCIA		DATOS INCORRECTOS  DATOS CORRECTOS 

5.3.5. Resultados de la fase de implementación y despliegue

5.3.5.1. Recursos empleados en la implementación

El desarrollo del Sistema se realizó siguiendo estándares modernos de diseño Web, orientados específicamente a satisfacer las necesidades operativas y las características de los procesos internos de postcosecha de la empresa L.Y Flowers.

5.3.5.1.1. Herramientas de desarrollo

En la Tabla 5.29 se detallan las herramientas utilizadas para el desarrollo y la implementación del Sistema.

Tabla 5.29. Herramientas de desarrollo e implementación

Herramienta	Uso
Visual Studio Code	Edición y depuración del proyecto.
Python / Django	Desarrollo del backend y arquitectura MVT.
PostgreSQL	Base de datos en desarrollo y producción.
Git / GitHub	Versionamiento y repositorio del código.
Railway	Alojamiento de la aplicación y base de datos.

5.3.5.1.2. Equipos de hardware

Para desarrollar el Sistema fue necesario contar con una infraestructura de hardware apropiada que garantizara un rendimiento eficiente. Los equipos utilizados permitieron realizar el desarrollo, las pruebas y el despliegue del Sistema. En la Tabla 5.30 se detallan las características técnicas de los equipos de trabajo empleados por los programadores durante este proceso.

Tabla 5.30. Equipos de hardware

Equipo	Características mínimas	Uso
Computador de desarrollo	Procesador de 4 núcleos, 8 GB RAM, 20 GB disponibles	Codificación, pruebas y documentación.
Equipo cliente	Navegador actualizado y conexión a Internet	Acceso a FlorLY.
Servidor Railway	Recursos administrados según el plan contratado	Ejecución de Django y PostgreSQL.

5.3.5.2. Resultado del desarrollo del Sistema Web

A continuación, se presentan los resultados del desarrollo del Sistema Web mediante sus diferentes módulos y funcionalidades, en los cuales se evidencia la aplicación de la programación y la lógica de negocio.

En la Figura 5.30 se presenta la interfaz de inicio de sesión del Sistema FlorLY, mediante la cual los usuarios ingresan sus credenciales para acceder a las funcionalidades habilitadas de acuerdo con el rol asignado.

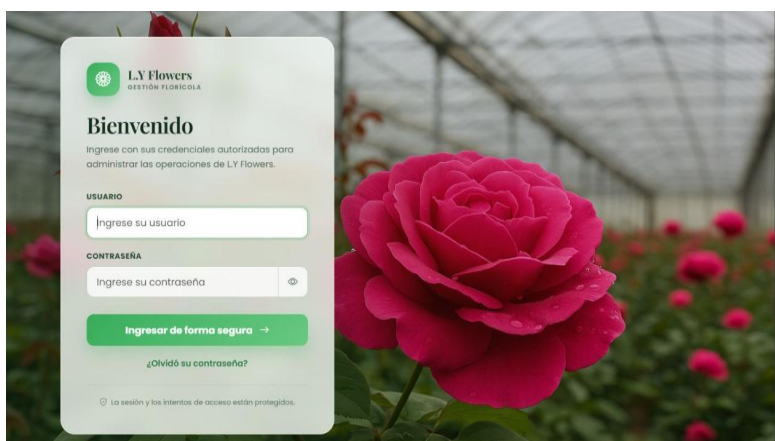


Figura 5.30. Interfaz de inicio de sesión

En la Figura 5.31 se observa el panel principal del administrador, donde se muestran los indicadores generales y los accesos a los módulos de administración, postcosecha, inventario, liquidaciones y auditoría.

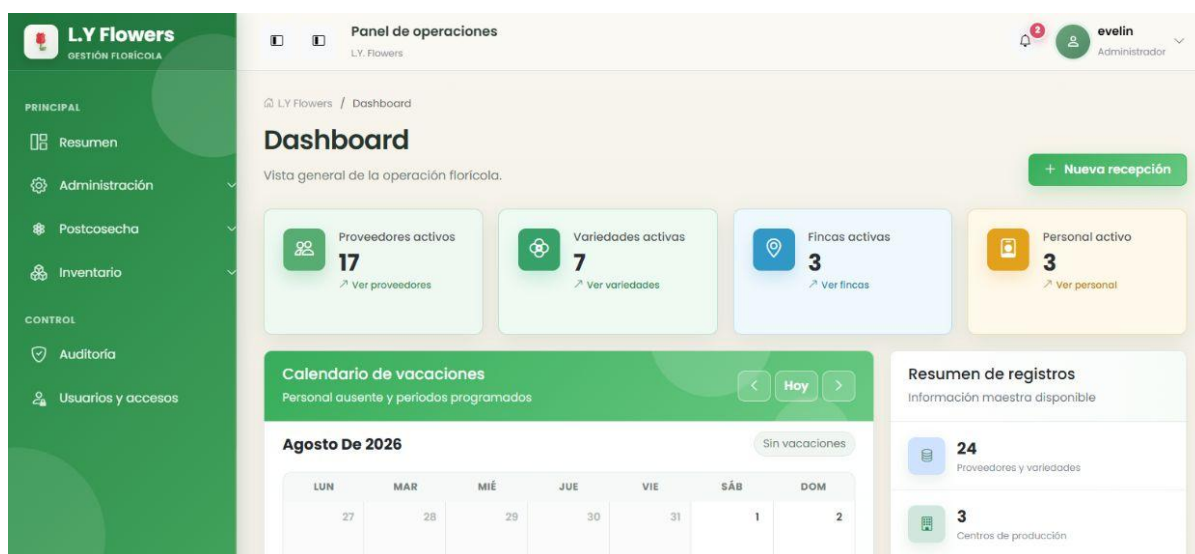


Figura 5.31. Interfaz del panel principal

En la Figura 5.32 se presenta el módulo de gestión de usuarios, que permite registrar, consultar, actualizar y controlar las cuentas de acceso, junto con los roles asignados dentro del Sistema.

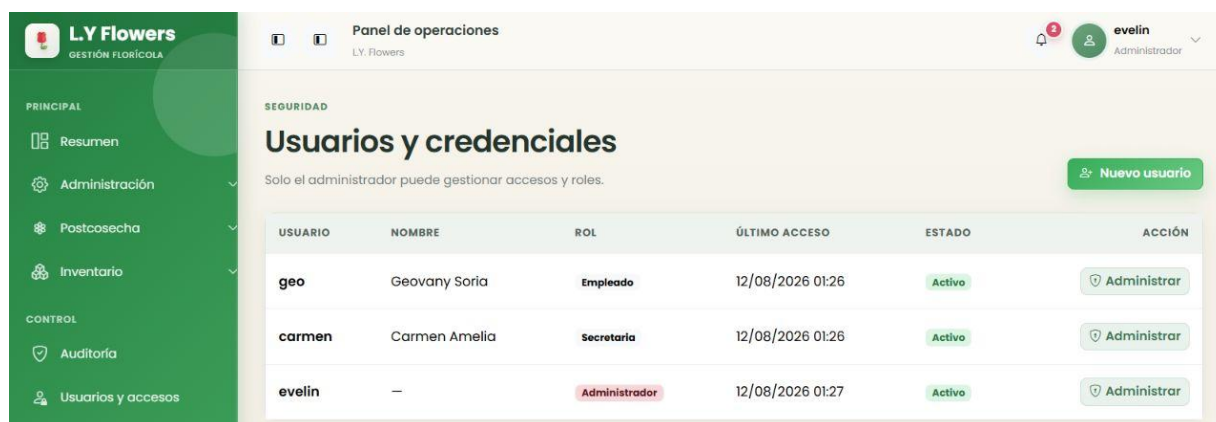


Figura 5.32. Interfaz de gestión de usuarios

En la Figura 5.33 se visualiza el módulo de gestión del personal de L.Y Flowers, donde se administra la información laboral de los empleados, así como sus permisos, vacaciones, documentos e historial.

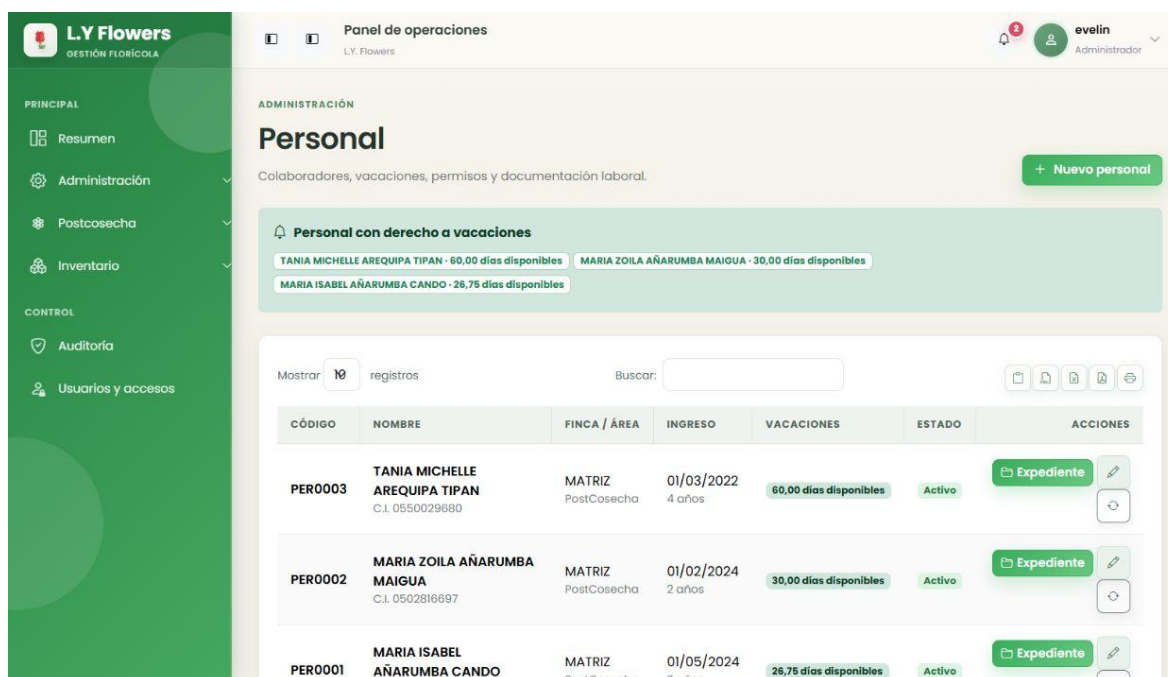


Figura 5.33. Interfaz de gestión del personal

En la Figura 5.34 se muestra el listado de recepciones de flor registradas, permitiendo consultar la fecha, la procedencia, el empleado receptor y el estado de cada ingreso realizado en postcosecha.

L.Y Flowers
GESTIÓN FLORÍCOLA

Panel de operaciones
L.Y. Flowers

POSTCOSECHA

Recepciones de flores

Ingreso de mallas desde proveedores o fincas propias.

+ Nueva recepción

Mostrar 10 registros

Buscar:

NÚMERO	FECHA	ORIGEN	VARIETADES	TOTAL TALLOS	OBSERVACIÓN	ACCIÓN
N.º 9 REC000009	25/07/2026 10:00	MARCO IVAN BELTRAN CHACON	FREEDOM - 40M x 30	1200	RECEPCION CREADA PARA PRUEBAS DE CLASIF...	Ver detalle
N.º 8 REC000008	24/07/2026 10:00	JUAN CARLOS YANCHATIPAN AÑARUMBA	EXPLORER - 6M x 25 MONDIAL - 7M x 25	325	PRUEBA AUTOMÁTICA LIQUIDACIÓN 2026- 07-26	Ver detalle
N.º 7 REC000007	14/07/2026 10:00	JUAN CARLOS YANCHATIPAN AÑARUMBA	FREEDOM - 5M x 25 EXPLORER - 6M x 25	275	PRUEBA AUTOMÁTICA LIQUIDACIÓN 2026- 07-26	Ver detalle
N.º 6 REC000006	05/07/2026 10:00	JUAN CARLOS YANCHATIPAN AÑARUMBA	MONDIAL - 4M x 25 FREEDOM - 5M x 25	225	PRUEBA AUTOMÁTICA LIQUIDACIÓN 2026- 07-26	Ver detalle

Figura 5.34. Interfaz de recepciones de flor

En la Figura 5.35 se observa el formulario para registrar una recepción de flor propia o proveniente de un proveedor externo. La interfaz permite seleccionar el origen, el empleado receptor y detallar las variedades, mallas y tallos recibidos.

L.Y Flowers
GESTIÓN FLORÍCOLA

Panel de operaciones
L.Y. Flowers

POSTCOSECHA · RECEPCIONES

Registrar recepción

Seleccione un único origen y registre cada presentación de mallas recibida.

Fecha y hora automáticas
El servidor registrará el momento exacto al guardar. Horario permitido: lunes a viernes de 08:00 a 15:00 y sábados de 08:00 a 12:00. Los domingos no se admiten recepciones.

Proveedor externo **Finca propia**

Ninguno Ninguna

Empleado receptor *

Seleccione...

Solo se muestra personal activo del área de postcosecha.

Observación

Opcional

Figura 5.35. Interfaz de registro de recepción

En la Figura 5.36 se presenta el módulo de clasificaciones, en el cual se consultan los registros procesados y las recepciones pendientes de clasificación, conservando la trazabilidad entre ambos procesos.

Panel de operaciones
L.Y. Flowers

POSTCOSECHA
Clasificaciones por proveedor

Cada código representa un lote completo de una sola recepción.

Mostrar 10 registros | Buscar: | Reporte diario general | Nueva clasificación

CLASIFICACIÓN	FECHA	PROVEEDOR / LOTE	VARIETADES RECIBIDAS	ACCIÓN
CLA000008	25/07/2026	JUAN CARLOS YANCHATIPAN AÑARUMBA Recepción N.º 8	EXPLORER - 6 mallas MONDIAL - 7 mallas	Reporte Ver
CLA000007	15/07/2026	JUAN CARLOS YANCHATIPAN AÑARUMBA Recepción N.º 7	FREEDOM - 5 mallas EXPLORER - 6 mallas	Reporte Ver
CLA000006	06/07/2026	JUAN CARLOS YANCHATIPAN AÑARUMBA Recepción N.º 6	MONDIAL - 4 mallas FREEDOM - 5 mallas	Reporte Ver
CLA000005	28/06/2026	JUAN CARLOS YANCHATIPAN AÑARUMBA Recepción N.º 5	EXPLORER - 3 mallas MONDIAL - 4 mallas	Reporte Ver
CLA000004	25/07/2026	MARCO IVAN BELTRAN CHACON Recepción N.º 4	FREEDOM - 6 mallas EXPLORER - 7 mallas	Reporte Ver

Figura 5.36. Interfaz de clasificaciones de flor

En la Figura 5.37 se visualiza el formulario de clasificación de flor, donde el empleado registra los tallos óptimos, estándar, nacionales y sobrantes correspondientes a cada variedad recibida.

Panel de operaciones
L.Y. Flowers

POSTCOSECHA - CLASIFICACIÓN
Clasificar lote completo

Cada número de clasificación corresponde a una sola recepción y debe completar todo el lote del proveedor.

Fecha * 12/08/2026 | Observación general

Puede seleccionar hoy o uno de los dos días siguientes.

Recepción y proveedor * Seleccione...
Solo aparecen recepciones pendientes dentro de su plazo de tres días.

Clasificación del lote
Registre los tallos realmente contados. Si llegaron menos, el sistema calculará los faltantes automáticamente.

Cancelar Finalizar clasificación y ver reporte

Figura 5.37. Interfaz de registro de clasificación

En la Figura 5.38 se muestra el módulo de liquidaciones, que permite consultar los pagos generados para los productores con base en los tallos exportables y las tarifas vigentes en la fecha de recepción.

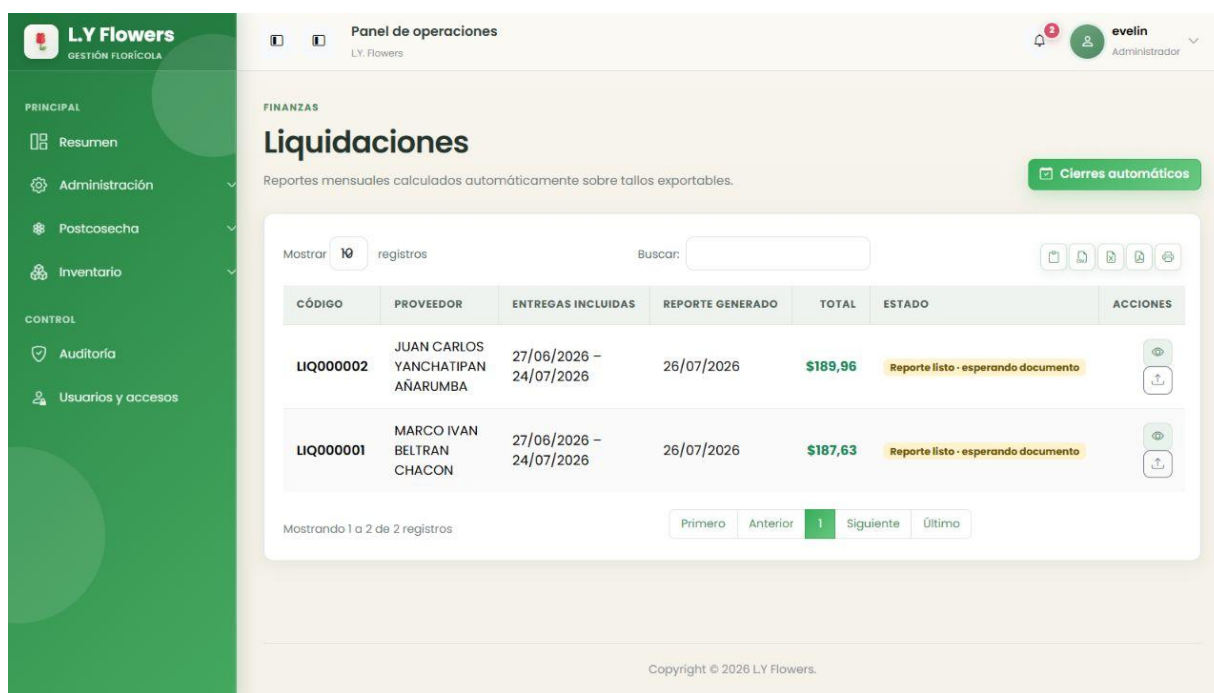


Figura 5.38. Interfaz de liquidaciones de productores

En la Figura 5.39 se presenta el módulo de inventario de productos químicos, donde se controlan las categorías, productos, unidades de medida, entradas, salidas, existencias disponibles y alertas de stock mínimo.



Figura 5.39. Interfaz de inventario de productos químicos

En la Figura 5.40 se observa el reporte de producción de postcosecha, el cual permite consultar los tallos procesados y los resultados consolidados por fecha, variedad y procedencia para apoyar el control operativo.



Figura 5.40. Interfaz del reporte de producción

Como parte de la documentación de apoyo para la operación del Sistema FlorLY, se elaboró el manual de usuario, el cual se presenta en el Anexo D.

5.3.5.3. Configuración del servidor del despliegue

5.3.5.3.1. Hosting y servidor de aplicaciones

Railway fue seleccionado como plataforma de despliegue. El servicio construye el proyecto desde el repositorio, instala las dependencias declaradas y ejecuta la aplicación mediante un servidor WSGI. La base PostgreSQL se mantiene como servicio asociado dentro del mismo proyecto.

5.3.5.3.2. Gestión de Dominio y DNS

El acceso inicial puede realizarse mediante el dominio asignado por Railway. Si la empresa incorpora un dominio propio, se debe registrar el dominio en Railway y crear en el proveedor DNS los registros indicados por la plataforma. No se incluyen valores inventados; deben documentarse los datos definitivos del dominio contratado.

5.3.5.3.3. Despliegue

El proceso de puesta en producción del Sistema FlorLY se realizó en la plataforma Railway, siguiendo una configuración orientada a garantizar la ejecución de la lógica de negocio, la persistencia de los datos, la seguridad de las credenciales y la disponibilidad de los archivos almacenados.

5.3.5.3.3.1. Preparación de la base de datos

La conexión con la base de datos se configuró mediante la variable de entorno DATABASE_URL proporcionada por Railway. Django interpreta esta dirección para establecer dinámicamente el nombre de la base, el usuario, la contraseña, el servidor y el puerto, evitando almacenar credenciales sensibles en el código fuente.

Configuración de PostgreSQL mediante DATABASE_URL

```

01 DATABASE_URL = os.getenv('DATABASE_URL', '').strip()
02 if DATABASE_URL:
03     parsed_database = urlparse(DATABASE_URL)
04     if parsed_database.scheme not in ('postgres', 'postgresql'):
05         raise RuntimeError('DATABASE_URL debe utilizar el esquema postgres:// o postgres
06 ql://')
07     database_options = {
08         key: values[-1]
09         for key, values in parse_qs(parsed_database.query).items()
10         if key in {'sslmode', 'target_session_attrs'}
11     }
12     DATABASES = {
13         'default': {
14             'ENGINE': 'django.db.backends.postgresql',
15             'NAME': unquote(parsed_database.path.lstrip('/')),
16             'USER': unquote(parsed_database.username or ''),
17             'PASSWORD': unquote(parsed_database.password or ''),
18             'HOST': parsed_database.hostname or '',
19             'PORT': str(parsed_database.port or 5432),
20             'CONN_MAX_AGE': 60,
21             'CONN_HEALTH_CHECKS': True,
22             'OPTIONS': database_options,
23         }
24     }
25 elif DEBUG:

```

Figura 5.41. Configuración de la conexión con PostgreSQL

5.3.5.3.3.2. Ajuste de la conexión

La configuración lee DATABASE_URL y credenciales desde variables de entorno, sin incorporarlas al repositorio.

5.3.5.3.3.3. Configuración del servicio Railway

El archivo railway.json define las instrucciones utilizadas por la plataforma durante la construcción y puesta en producción. Se configuró la recolección de archivos estáticos, la ejecución automática de migraciones, el inicio mediante Gunicorn, la ruta de comprobación del servicio y la política de reinicio ante fallos.

railway.json: construcción, migraciones y ejecución

```

01  {
02    "$schema": "https://railway.com/railway.schema.json",
03    "build": {
04      "builder": "RAILPACK",
05      "buildCommand": "python manage.py collectstatic --noinput"
06    },
07    "deploy": {
08      "preDeployCommand": [
09        "python manage.py migrate --noinput"
10      ],
11      "startCommand": "gunicorn FlorLY.wsgi:application --bind 0.0.0.0:$PORT --workers 2 -
12 -threads 2 --timeout 120 --access-logfile - --error-logfile -",
13      "healthcheckPath": "/health/",
14      "healthcheckTimeout": 120,
15      "restartPolicyType": "ON_FAILURE",
16      "restartPolicyMaxRetries": 3
17    }
18  }

```

*Figura 5.42. Configuración del servicio mediante railway.json***5.3.5.3.3.4. Publicación**

La publicación se realizó vinculando el repositorio de GitHub con el servicio de Railway. Cada actualización autorizada inicia la construcción, ejecuta las migraciones y levanta la aplicación con Gunicorn. Finalmente, se verificó el acceso mediante HTTPS, la carga de archivos estáticos, la conexión con los servicios de datos y el funcionamiento de la interfaz de inicio de sesión.

Flujo automático de publicación en Railway

```

01  Repositorio GitHub conectado al servicio de aplicación
02  Build: python manage.py collectstatic --noinput
03  Pre-deploy: python manage.py migrate --noinput
04  Start: gunicorn FlorLY.wsgi:application --bind 0.0.0.0:$PORT
05  Healthcheck: /health/
06  Política de reinicio: ON_FAILURE
07  Dominio público y HTTPS administrados por Railway

```

*Figura 5.43. Publicación automática del Sistema en Railway***5.4. COSTO ESTIMADO DEL SOFTWARE**

Para estimar el presupuesto del Sistema Web FlorLY se utilizó la técnica Planning Poker, propia de la metodología Extreme Programming (XP), mediante la cual se asignaron puntos de historia de acuerdo con la complejidad de cada funcionalidad desarrollada.

Como resultado de este proceso se establecieron 182 puntos de historia para el proyecto. Para calcular el presupuesto se consideró que cada punto equivale a 2 horas de desarrollo y se fijó un costo estimado de \$9 por hora de trabajo, incluyendo los recursos utilizados durante la construcción del Sistema.

En la Tabla 5.31 se presenta el cálculo del costo por hora de trabajo del programador.

Tabla 5.31. Cálculo por hora del programador

CÁLCULO POR HORA DEL PROGRAMADOR	
Salario promedio del programador junior	\$800
Horas laborales mensuales	160 horas
Costo estimado por hora	\$9

En la Tabla 5.32 se presenta la estimación de puntos de historia asignados a los módulos del Sistema.

Tabla 5.32. Puntos de los módulos

MÓDULO	DESCRIPCIÓN	PUNTOS DE HISTORIAS DE USUARIO
Usuarios, roles y panel principal	Administración de cuentas, permisos e indicadores generales	18
Proveedores, fincas y variedades	Gestión de productores, fincas propias y variedades de rosas	18
Gestión del personal	Registro de empleados, permisos, vacaciones e historial laboral	13
Recepción de flor	Registro de flor propia y externa, variedades, mallas y tickets	29
Clasificación de flor	Control de tallos exportables, nacionales, adicionales y fechas	28
Reporte diario de postcosecha	Consolidación de tallos procesados y resultados por variedad	8
Tarifarios y liquidaciones	Historial de precios, cálculo y consulta de pagos a productores	34
Inventario de químicos	Productos, categorías, entradas, salidas y alertas de existencias	26

MÓDULO	DESCRIPCIÓN	PUNTOS DE HISTORIAS DE USUARIO
Reportes y auditoría	Consulta de información operativa y trazabilidad de acciones	8
Total		182

Con estos valores, el presupuesto estimado se calculó de la siguiente manera:

$$\text{Presupuesto} = \text{Total de puntos de historia} \times \text{Horas por punto} \times \text{Costo por hora}$$

$$\text{Presupuesto} = 182 \times 2 \times 9 = 3.276 \text{ dólares}$$

En la Tabla 5.33 se presenta el costo total estimado para el desarrollo del Sistema Web FlorLY.

Tabla 5.33. Costo total del software

ESTIMACIÓN DEL COSTO TOTAL	
Gastos	Valor total
Costo estimado del desarrollo del Sistema Web FlorLY	\$1.200

5.5. RESPUESTA A LA PREGUNTA DE INVESTIGACIÓN

Con la información recopilada mediante las entrevistas y encuestas, el desarrollo del Sistema Web y la ejecución de las pruebas funcionales, se respondió a la problemática planteada: ¿Cómo optimizar la gestión de los procesos de acopio, clasificación y liquidación de productores en la empresa L.Y Flowers mediante un Sistema Web?

Los resultados del diagnóstico evidenciaron que el uso de registros manuales y hojas de cálculo dificultaba la organización de la información, incrementaba el riesgo de errores y duplicidad de datos, y retrasaba la elaboración de reportes y liquidaciones. Las entrevistas realizadas al gerente, a la secretaria y al representante de los empleados permitieron comprender el flujo de trabajo y establecer las necesidades principales. Asimismo, la encuesta aplicada a nueve beneficiarios directos —una administradora, una secretaria y siete empleados— confirmó la necesidad de centralizar y automatizar estos procesos.

Como respuesta, se desarrolló FlorLY, un Sistema Web que integra el registro de proveedores, fincas y variedades; la recepción y clasificación de flor; la gestión del personal; el control de tarifarios; la

generación de liquidaciones; el inventario de químicos; y la consulta de reportes. La implementación de estos módulos permitió mantener la información centralizada, mejorar la trazabilidad de las operaciones y disponer de datos actualizados para apoyar el control administrativo y la toma de decisiones.

En la Tabla 5.34 se relacionan las evidencias obtenidas con la respuesta a la pregunta de investigación.

Tabla 5.34. Evidencias de respuesta a la pregunta de investigación

EVIDENCIA	RESULTADO PRINCIPAL	APORTE AL PROYECTO
Entrevistas semiestructuradas	Se identificaron procesos manuales, información dispersa y demoras en el seguimiento de las operaciones.	Permitieron definir los módulos, actores e historias de usuario del Sistema.
Encuesta a beneficiarios (n = 9)	Los participantes reconocieron dificultades en las herramientas actuales y la necesidad de automatizar la gestión.	Respaldó la aceptación de la propuesta y la priorización de funcionalidades.
Pruebas funcionales	Los casos de prueba verificaron el registro, validación, consulta y procesamiento correcto de la información.	Comprobaron el cumplimiento de los criterios de aceptación de las historias de usuario.

Con base en estas evidencias, se concluye que el Sistema Web FlorLY respondió a la pregunta de investigación al Sistematizar los procesos de postcosecha y reducir la dependencia de registros manuales. Su funcionamiento fue comprobado mediante casos de prueba vinculados con los criterios de aceptación, sin aplicar la escala System Usability Scale (SUS), debido a que esta técnica no formó parte de los instrumentos definidos para la investigación. Por tanto, la valoración de la solución se sustenta en los resultados del levantamiento de información y en la verificación funcional del Sistema.

6. CONCLUSIONES Y RECOMENDACIONES

6.1. CONCLUSIONES

- La fundamentación teórica y tecnológica permitió seleccionar Python, Django, PostgreSQL y una arquitectura MVT adecuada para centralizar los procesos operativos y administrativos de L.Y Flowers.
- La aplicación de XP organizó el trabajo mediante historias de usuario e iteraciones cortas, lo que facilitó priorizar el flujo crítico de recepción, clasificación y liquidación, además de incorporar retroalimentación de los usuarios.

- La implementación del sistema web FlorLY permitió centralizar la información de productores, fincas, variedades, recepciones y clasificaciones, automatizar el cálculo de liquidaciones y mejorar la trazabilidad de los procesos operativos y administrativos de L.Y Flowers.

6.2. RECOMENDACIONES

- Implementar una aplicación móvil con funcionamiento sin conexión para registrar recepciones y clasificaciones desde el área de postcosecha, sincronizando la información con el Sistema cuando se restablezca el acceso a Internet.
- Incorporar códigos QR en los tickets de recepción para identificar cada entrega y consultar rápidamente su recorrido desde la recepción y clasificación hasta la liquidación del productor.
- Integrar FlorLY con un Sistema contable o de facturación electrónica para transferir automáticamente los valores de las liquidaciones y pagos, evitando que la información tenga que registrarse nuevamente de forma manual.

7. REFERENCIAS BIBLIOGRÁFICAS

[1] B. P. Castro-Acosta, E. P. Mendoza-De La Cruz y J. E. Arias-Montero, «Estudio de benchmarking como herramienta de optimización estratégica de las exportaciones de rosas ecuatorianas», *Economía y Negocios*, vol. 15, n.º 1, pp. 124-143, ene. 2024, doi: 10.29019/eyn.v15i1.1270.

[2] F. J. B. Ordóñez, «Impacto y desafíos del sector florícola en Ecuador: entre certificaciones de responsabilidad social y realidades laborales», *Revista InveCom*, vol. 5, n.º 2, 2025, doi: 10.5281/zenodo.13381972.

[3] J. N. Y. Zavala y S. Z. F. Tenezaca, «ANÁLISIS Y BENEFICIOS DE LA TENDENCIA CRECIENTE DE LAS EXPORTACIONES DE FLORES ECUATORIANAS.».

[4] W. Veloz y D. Guerrero, «Situación actual de la floricultura en la provincia de Cotopaxi», *Medwave*, supl. 1, sep. 2023, doi: 10.5867/medwave.2023.S1.UTA357.

[5] Agencia de Regulación y Control Fito y Zoosanitario (AGROCALIDAD), Unidades de Producción Certificadas con Buenas Prácticas Agropecuarias, corte: 30 de junio de 2024, Ecuador, 2024.

- [6] S. A. Rodríguez Enríquez, «Sistema informático para la gestión de producción en la florícola León Roses del cantón Bolívar», trabajo de titulación, Universidad Politécnica Estatal del Carchi, Tulcán, Ecuador, 2025.
- [7] J. E. Iguamba Túquerres y D. M. Pullupaxi Cando, «Sistema informático de gestión de información para los procesos de producción e inventarios en la florícola Rosadex Cía. Ltda.», propuesta tecnológica, Universidad Técnica de Cotopaxi, Latacunga, Ecuador, 2020.
- [8] D. F. Túquerres Lara, «Desarrollo de un sistema web de inventarios para una florícola», trabajo de titulación, Universidad de Las Américas, Quito, Ecuador, 2019.
- [9] A. R. Méndez Díaz, «Sistema de gestión por procesos en el área de post cosecha de la empresa florícola Pontón ForeverFlor Cía. Ltda.», trabajo de titulación, Universidad Técnica de Ambato, Ambato, Ecuador, 2022.
- [10] J. P. Gualavisi Chimarro, «Implementación de un sistema de gestión por procesos para el mejoramiento de la productividad en el área de postcosecha de la empresa ABAX Industrias y Flores Cía. Ltda., en la ciudad de Tabacundo», trabajo de grado, Universidad Técnica del Norte, Ibarra, Ecuador, 2016.
- [11] C. D. Vélez Andrade, «Optimización del proceso de poscosecha en una empresa de rosas de exportación ubicada en la parroquia Nono», trabajo de titulación, Universidad de Las Américas, Quito, Ecuador, 2015.
- [12] B. E. Cachipueno Ruiz, «Modelo de gestión administrativa para el fortalecimiento de la competitividad de la empresa florícola Edén Roses, Cayambe», trabajo de titulación, Universidad de Otavalo, Otavalo, Ecuador, 2024.
- [13] International Organization for Standardization, ISO/IEC 25010:2023 Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE)—Product quality model. Ginebra, Suiza: ISO, 2023.
- [14] J. P. Acevedo Caguante, “Manufactura esbelta para la optimización de los procesos postcosecha en una empresa florícola,” tesis de Ingeniería Industrial, Facultad de Ingeniería en Sistemas, Electrónica e Industrial, Universidad Técnica de Ambato, Ambato, Ecuador, sep. 2022. Accedido: 8 de marzo de 2026. [En línea]. Disponible en: <https://repositorio.uta.edu.ec/handle/123456789/36622>.

- [15] S. Monsalve, K. Quiñónez, D. García Arango y C. Henao Villa, «Identificación del desarrollo de aplicaciones web en la actualidad», 2020, pp. 272-282.
- [16] C. D. Guerra Ruiz, “Estudio de lenguajes de programación orientados a GPU para la creación de un prototipo de procesamiento de imágenes de actividades agrícolas,” trabajo de grado, Carrera de Software, Facultad de Ingeniería en Ciencias Aplicadas, Universidad Técnica del Norte, Ibarra, Ecuador, 2022. Accedido: 8 de marzo de 2026. [En línea]. Disponible en: [Repositorio Digital UTN](#).
- [17] A. A. Sayed and M. O. S. Patel, “The influence of JavaScript on website development and user experience,” *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 11, no. 11, pp. 87–89, nov. 2023, doi: [10.22214/ijraset.2023.56284](#).
- [18] N. E. Zuleta Sevilla, “Estudio de la integración del lenguaje de programación PHP con la base de datos NoSQL Apache CouchDB: Sistema de gestión de expedientes de los estudiantes de la carrera de Ingeniería en Sistemas Computacionales,” tesis de pregrado, Carrera de Ingeniería en Sistemas Computacionales, Facultad de Ingeniería en Ciencias Aplicadas, Universidad Técnica del Norte, Ibarra, Ecuador, 2018. Accedido: 20 de julio de 2026. [En línea]. Disponible en: <https://repositorio.utn.edu.ec/handle/123456789/8261>.
- [19] Python Software Foundation, «Welcome to Python.org», Python.org, 2026. Accedido: 20 de julio de 2026. [En línea]. Disponible en: <https://www.python.org/about/>
- [20] «Welcome to Python.org», Python.org. Accedido: 6 de junio de 2026. [En línea]. Disponible en: <https://www.python.org/about/>
- [21] Django Software Foundation, «How to deploy with WSGI», Django Documentation, 2026. Accedido: 8 de julio de 2026. [En línea]. Disponible en: <https://docs.djangoproject.com/en/6.0/howto/deployment/wsgi/>
- [22] «Installation | Laravel 12.x - The clean stack for Artisans and agents». Accedido: 8 de julio de 2026. [En línea]. Disponible en: <https://laravel.com/docs/12.x/installation>
- [23] «Express - Node.js Marco de aplicación web». Accedido: 8 de junio de 2026. [En línea]. Disponible en: <https://expressjs.com/es/>

- [24] «Django», Django Project. Accedido: 6 de enero de 2026. [En línea]. Disponible en: <https://www.djangoproject.com/>
- [25] Django Software Foundation, «How to deploy with WSGI», Django Documentation, 2026. Accedido: 8 de marzo de 2026. [En línea]. Disponible en: <https://docs.djangoproject.com/en/6.0/howto/deployment/wsgi/>
- [26] D. Y. Chango Gavilánez y J. J. Jácome Salas, “Bases de datos no relacionales: Utilización de MongoDB como base de datos no relacional empleando formato GeoJSON,” trabajo de titulación, Carrera de Ingeniería de Sistemas, Universidad Politécnica Salesiana, sede Quito, Quito, Ecuador, jul. 2016. [En línea]. Disponible: <https://dspace.ups.edu.ec/handle/123456789/17241>. [Accedido: 17-ago-2026].
- [27] O. L. Fierro Montenegro y J. C. García Pinchao, “SABDO: Sistema de auditoría para bases de datos Oracle,” tesis de pregrado, Escuela de Ingeniería en Sistemas Computacionales, Facultad de Ingeniería en Ciencias Aplicadas, Universidad Técnica del Norte, Ibarra, Ecuador, abr. 2009. [En línea]. Disponible: <https://repositorio.utn.edu.ec/handle/123456789/598>. [Accedido: 17-ago-2026].
- [28] M. S. Gines Colana, “Análisis comparativo del desempeño de los sistemas gestores de bases de datos: PostgreSQL y MySQL con los datos académicos de la Universidad Nacional Jorge Basadre Grohmann–2019,” tesis de pregrado, Escuela Profesional de Ingeniería en Informática y Sistemas, Universidad Nacional Jorge Basadre Grohmann, Tacna, Perú, 2022. [En línea]. Disponible: <https://repositorio.unjbg.edu.pe/handle/20.500.12510/3889>. [Accedido: 17-ago-2026].
- [29] «PostgreSQL: Acerca de». Accedido: 8 de junio de 2026. [En línea]. Disponible en: <https://www.postgresql.org/about/>
- [30] Wardy, “Rendimiento maestro de PostgreSQL: prácticas de WeWard para bases de datos eficientes,” *Blog de WeWard*, 15-ene-2025. Accedido: 6 de junio de 2026. [En línea]. Disponible en: <https://www.wewardapp.com/es/blog/master-postgresql-performance---wewards-practices-for-efficient-databases>.
- [31] S. Monsalve, K. Quinónez, D. García Arango, y C. Henao Villa, «Identificación del desarrollo de aplicaciones web en la actualidad», 2020, pp. 272-282.

- [32] M. Á. Durán, "Lenguaje HTML", LenguajeHTML.com. Accedido: 1 de agosto de 2026. [En línea]. Disponible en: <https://lenguajehtml.com/>
- [33] M. E. C. Alonso, *CSS: Hojas de estilo en cascada*. Google Books, 2023. [En línea]. Disponible en: <https://books.google.com.ec/books?id=0jnCEAAAQBAJ>
- [34] J. Flanagan, *JavaScript: La guía definitiva*. 7.^a ed., O'Reilly Media, 2023. [En línea]. Disponible en: <https://books.google.com.ec/books?id=ezPoEAAAQBAJ>
- [35] Bootstrap Team, «Introduction», Bootstrap Documentation, 2026. Accedido: 20 de julio de 2026. [En línea]. Disponible en: <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
- [36] ThemeWagon, «Free and premium Bootstrap templates», ThemeWagon, 2026. Accedido: 20 de julio de 2026. [En línea]. Disponible en: <https://themewagon.com/>
- [37] Microsoft, «Why did we build Visual Studio Code?», Visual Studio Code Documentation, 2026. Accedido: 20 de julio de 2026. [En línea]. Disponible en: <https://code.visualstudio.com/docs/editor/whyvscode>
- [38] A. Pérez y J. Sánchez, «Git: sistema de control de versiones distribuido», *Revista Ecosistemas*, vol. 31, n.º 2, 2022. [En línea]. Disponible en: <https://revistaecosistemas.net/index.php/ecosistemas/article/view/2332>
- [39] GitHub, «Acerca de GitHub y Git», Documentación de GitHub, 2026. Accedido: 8 de mayo de 2026. [En línea]. Disponible en: <https://docs.github.com/es/get-started/start-your-journey/about-github-and-git>
- [40] Railway Corp., «Railway», 2026. Accedido: 30 de julio de 2026. [En línea]. Disponible en: <https://railway.com/>
- [41] Railway Corp., «Deploy a Django App», Railway Docs, 2026. Accedido: 30 de julio de 2026. [En línea]. Disponible en: <https://docs.railway.com/guides/django>
- [42] EspiFreelancer, "Qué es el patrón MTV (Model Template View)," *EspiFreelancer*, 3 ago. 2019, actualizado el 8 ago. 2019. [En línea]. Disponible: <https://espifreelancer.com/mtv-django.html>. [Accedido: 17-ago-2026].

- [43] M. A. Tabares, «Metodologías de desarrollo de software», Universidad Católica Argentina, Buenos Aires, Argentina. [En línea]. Disponible en: <https://repositorio.uca.edu.ar/handle/123456789/522>
- [44] M. E. Gabriel y J. Pacienza, «Metodologías de desarrollo de software,» UCA, Argentina, 2015.
- [45] D. M. Correa y J. A. Gómez, «Metodologías tradicionales y metodologías ágiles para el desarrollo de software», *Revista CINTEX*, vol. 24, n.º 1, 2019. [En línea]. Disponible en: <https://revistas.pascualbravo.edu.co/index.php/cintex/article/view/334/312>
- [46] «Extreme Programming (XP): Values, Principles, and Practices», AltexSoft. Accedido: 5 de enero de 2026. [En línea]. Disponible en: <https://www.altexsoft.com/blog/extremeprogramming-values-principles-and-practices/>
- [47] A. Akhtar, B. Bakhtawar, y S. Akhtar, «EXTREME PROGRAMMING VS SCRUM: A COMPARISON OF AGILE MODELS», *Int. J. Technol. Innov. Manag. IJTIM*, vol. 2, n.o 2, oct. 2022, doi: 10.54489/ijtim.v2i2.77.
- [48] L. Boiser, «Kanban Zone», Kanban Zone: The Visual Management Platform that Solves your Workflow Challenges. Accedido: 5 de enero de 2026. [En línea]. Disponible en: <https://kanbanzone.com/2019/kanban-vs-scrum/>
- [49] «Extreme Programming (XP): Values, Principles, and Practices», AltexSoft. Accedido: 5 de enero de 2026. [En línea]. Disponible en: <https://www.altexsoft.com/blog/extremeprogramming-values-principles-and-practices/>
- [50] R. Fojtik, «Extreme Programming in development of specific software», *Procedia Computer Science*, vol. 3, pp. 1464-1468, 2011, doi: 10.1016/j.procs.2011.01.032
- [51] «Extreme Programming (XP): Values, Principles, and Practices», AltexSoft. Accedido: 5 de junio de 2026. [En línea]. Disponible en: <https://www.altexsoft.com/blog/extremeprogramming-values-principles-and-practices/>
- [52] Teach Computer Science, «Extreme Programming (XP) | Values, Principles, Advantages», *Teach Computer Science*. [En línea]. Disponible en: <https://teachcomputerscience.com/extreme-programming-xp/>. [Accedido: 4-ago-2026]

[53] J. L. Castillo, «Beneficios de XP para el desarrollo de sistemas web», *Revista Veritas*, vol. 1, n.º 2, 2023. [En línea]. Disponible en: <https://revistaveritas.org/index.php/veritas/article/view/314>

8. ANEXOS

Anexo A: Historias de usuario

Tabla A.1. HU004: Registro de proveedores

Nº 4	HISTORIA DE USUARIO		
Número: HU004		Usuario: Administrador y Secretaria	
Objetivo: Registro de proveedores			
Prioridad: Alta	Estimación: 5		Iteración: 1
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como Administrador y Secretaria, quiero registrar los datos de los productores externos, para relacionarlos con sus entregas y liquidaciones.			
Criterios de aceptación:			
Dado que la secretaria está autenticada, cuando ingresa código único, nombres, apellidos, cédula o RUC y teléfono válidos, entonces el Sistema debe registrar al proveedor.			
Dado que el código o la identificación ya existe, cuando intenta guardar el proveedor, entonces el Sistema debe impedir la duplicidad y mostrar una advertencia.			

Tabla A.2. HU005: Administración de proveedores

Nº 5	HISTORIA DE USUARIO		
Número: HU005		Usuario: Administrador y secretaria	
Objetivo: Administración de proveedores			
Prioridad: Media	Estimación: 3		Iteración: 1
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como Administrador y secretaria, quiero consultar, actualizar y cambiar el estado de los proveedores, para mantener vigente la información utilizada en las operaciones.			
Criterios de aceptación:			
Dado que un proveedor está registrado, cuando la secretaria modifica datos permitidos, entonces el Sistema debe actualizar la información.			
Dado que un proveedor posee operaciones históricas, cuando se lo desactiva, entonces el Sistema debe conservar sus registros anteriores y excluirlo de nuevas recepciones.			

Tabla A.3. HU006: Gestión de fincas

Nº 6	HISTORIA DE USUARIO		
Número: HU006		Usuario: Administrador y secretaria	
Objetivo: Gestión de fincas			
Prioridad: Alta	Estimación: 5		Iteración: 1
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como Administrador y secretaria, quiero registrar y administrar las fincas de la empresa, para identificar el origen de la flor propia sin limitar la cantidad de fincas.			
Criterios de aceptación:			

N° 6	HISTORIA DE USUARIO
Dado que se requiere una nueva finca, cuando se ingresan código único, nombre, ubicación y estado, entonces el Sistema debe guardarla. Dado que una finca está inactiva, cuando se registra una nueva recepción, entonces no debe aparecer entre los orígenes disponibles.	

Tabla A.4. HU007: Gestión de variedades

Nº 7	HISTORIA DE USUARIO		
Número: HU007		Usuario: Administrador y secretaria	
Objetivo: Gestión de variedades			
Prioridad: Alta	Estimación: 5		Iteración: 1
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como Administrador y Secretaria, quiero registrar y administrar las variedades de rosas, para utilizarlas en recepciones, clasificaciones, tarifarios y reportes.			
Criterios de aceptación: Dado que se registra una variedad con código y nombre únicos, cuando se confirma el formulario, entonces el Sistema debe almacenarla. Dado que el código o nombre ya está registrado, cuando se intenta guardar la variedad, entonces el Sistema debe mostrar un error y evitar la duplicidad.			

Tabla A.5. HU008: Registro de personal

Nº 8	HISTORIA DE USUARIO		
Número: HU008		Usuario: Administrador y secretaria	
Objetivo: Registro de personal			
Prioridad: Media	Estimación: 5		Iteración: 2
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como Administrador y secretaria, quiero registrar los datos laborales de los trabajadores, para mantener actualizado el personal de la empresa.			
Criterios de aceptación:			
Dado que se incorpora un trabajador, cuando se ingresan nombres, apellidos, cédula, teléfono, fecha de ingreso, área y finca, entonces el Sistema debe guardar su ficha.			
Dado que la cédula ya pertenece a otro trabajador, cuando se intenta registrar nuevamente, entonces el Sistema debe impedir la duplicidad.			

Tabla A.6. HU009: Gestión de permisos y vacaciones

Nº 9	HISTORIA DE USUARIO		
Número: HU009		Usuario: Administrador y Secretaria	
Objetivo: Gestión de permisos y vacaciones			
Prioridad: Media	Estimación: 5		Iteración: 2
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como Administrador y secretaria, quiero registrar permisos y vacaciones del personal, para controlar sus ausencias y conservar el historial laboral.			
Criterios de aceptación:			

N° 9	HISTORIA DE USUARIO
Dado que un trabajador está activo, cuando se registra un permiso o vacaciones con fechas válidas, entonces el Sistema debe almacenar la novedad. Dado que la fecha final es anterior a la inicial, cuando se intenta guardar la solicitud, entonces el Sistema debe rechazarla y explicar la validación.	

Tabla A.7. HU010: Consulta del historial laboral

N° 10	HISTORIA DE USUARIO		
Número: HU010		Usuario: Administrador y secretaria	
Objetivo: Consulta del historial laboral			
Prioridad: Baja	Estimación: 3		Iteración: 2
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como Administrador y secretaria, quiero consultar el historial de un trabajador, para revisar asignaciones, permisos, vacaciones y cambios de estado.			
Criterios de aceptación: Dado que el trabajador tiene novedades registradas, cuando se abre su historial, entonces el Sistema debe mostrarlas ordenadas cronológicamente. Dado que se aplican filtros por tipo o fecha, cuando se ejecuta la búsqueda, entonces deben mostrarse únicamente los registros coincidentes.			

Tabla A.8. HU011: Registro de recepción de flor de proveedor

N° 11	HISTORIA DE USUARIO		
Número: HU011		Usuario: Secretaria y Empleado	
Objetivo: Registro de recepción de flor de proveedor			
Prioridad: Alta	Estimación: 8		Iteración: 2
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como secretaria y Empleado, quiero registrar una entrega de flor externa, para identificar al proveedor, la fecha y el detalle de las variedades recibidas.			
Criterios de aceptación: Dado que el proveedor está activo, cuando se registra la fecha y al menos una variedad con mallas, largo y tallos por malla, entonces el Sistema debe guardar la recepción y calcular los tallos declarados. Dado que falta el proveedor o el detalle de variedades, cuando se intenta guardar, entonces el Sistema debe mostrar los campos requeridos y no crear la recepción.			

Tabla A.9. HU012: Registro de recepción de flor propia

N° 12	HISTORIA DE USUARIO		
Número: HU012		Usuario: Secretaria y Empleado	
Objetivo: Registro de recepción de flor propia			
Prioridad: Alta	Estimación: 8		Iteración: 2
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como secretaria y Empleado, quiero registrar una entrega proveniente de una finca de L.Y Flowers, para controlar la producción propia de forma separada.			

N° 12	HISTORIA DE USUARIO
Criterios de aceptación: Dado que la finca está activa, cuando se selecciona flor propia y se registra el detalle de variedades, entonces el Sistema debe asociar la recepción con la finca. Dado que se selecciona flor propia, cuando se intenta registrar un proveedor externo como origen, entonces el Sistema debe impedir la combinación inconsistente.	

Tabla A.10. HU013: Registro de múltiples variedades por recepción

N° 13	HISTORIA DE USUARIO		
Número: HU013		Usuario: Secretaria y Empleado	
Objetivo: Registro de múltiples variedades por recepción			
Prioridad: Alta	Estimación: 8		Iteración: 2
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como secretaria y Empleado, quiero agregar varias variedades con cantidades diferentes dentro de una recepción, para representar correctamente cada entrega.			
Criterios de aceptación: Dado que una entrega contiene varias variedades, cuando se añade cada detalle con sus mallas, largo y tallos por malla, entonces el Sistema debe conservar los valores de forma independiente. Dado que cada detalle es válido, cuando se guarda la recepción, entonces el total declarado debe ser la suma de los tallos calculados en todos los detalles.			

Tabla A.11. HU014: Generación del ticket de recepción

N° 14	HISTORIA DE USUARIO		
Número: HU014		Usuario: Secretaria y Empleado	
Objetivo: Generación del ticket de recepción			
Prioridad: Alta	Estimación: 5		Iteración: 2
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como secretaria y Empleado, quiero generar un ticket de la entrega registrada, para disponer de un comprobante con el detalle y los espacios de firmas.			
Criterios de aceptación:			
Dado que la recepción fue guardada, cuando se solicita el ticket, entonces el Sistema debe mostrar origen, fecha, variedades, mallas, tallos declarados y total recibido.			
Dado que el ticket corresponde a flor de proveedor, cuando se genera el documento, entonces debe incluir espacios para la firma del proveedor y de la empresa.			

Tabla A.12. HU015: Consulta de recepciones pendientes

N° 15	HISTORIA DE USUARIO		
Número: HU015		Usuario: Empleado	
Objetivo: Consulta de recepciones pendientes			
Prioridad: Alta	Estimación: 5		Iteración: 3
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como Empleado, quiero consultar las variedades pendientes de clasificación, para organizar el trabajo según la planificación diaria.			
Criterios de aceptación:			

N° 15	HISTORIA DE USUARIO
Dado que existen detalles de recepción sin clasificar, cuando el trabajador consulta los pendientes, entonces el Sistema debe mostrarlos por fecha, origen y variedad. Dado que una recepción fue clasificada parcialmente, cuando se consulta nuevamente, entonces solo debe aparecer la cantidad o los detalles todavía pendientes.	

Tabla A.13. HU016: Registro de clasificación exportable

N° 16	HISTORIA DE USUARIO		
Número: HU016		Usuario: Empleado	
Objetivo: Registro de clasificación exportable			
Prioridad: Alta	Estimación: 8		Iteración: 3
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como Empleado, quiero registrar los tallos exportables como Óptimo o Estándar, para conservar el resultado real de la clasificación.			
Criterios de aceptación: Dado que existe una recepción pendiente, cuando se ingresan variedad, calidad de largo y cantidad de tallos exportables, entonces el Sistema debe guardar la clasificación. Dado que se selecciona una calidad distinta de Óptimo o Estándar, cuando se intenta guardar, entonces el Sistema debe rechazar el valor.			

Tabla A.14. HU017: Registro de tallos adicionales

N° 17	HISTORIA DE USUARIO		
Número: HU017		Usuario: Empleado	
Objetivo: Registro de tallos adicionales			
Prioridad: Alta	Estimación: 5		Iteración: 3
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como Empleado, quiero registrar los tallos extras detectados durante la clasificación, para reflejar la cantidad realmente recibida.			
Criterios de aceptación:			
Dado que el conteo real supera los tallos declarados, cuando se registra la clasificación, entonces el Sistema debe calcular y almacenar la diferencia como tallos adicionales.			
Dado que no existe diferencia positiva, cuando se completa la clasificación, entonces el Sistema debe registrar cero tallos adicionales.			

Tabla A.15. HU018: Registro de flor nacional o rechazada

N° 18	HISTORIA DE USUARIO		
Número: HU018		Usuario: Empleado	
Objetivo: Registro de flor nacional o rechazada			
Prioridad: Alta	Estimación: 5		Iteración: 3
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como Empleado, quiero registrar los tallos que no cumplen los requisitos de exportación, para diferenciarlos de la flor que genera pago.			
Criterios de aceptación:			

N° 18	HISTORIA DE USUARIO
Dado que existen tallos no exportables, cuando se registra su cantidad y observación, entonces el Sistema debe clasificarlos como flor nacional o rechazada. Dado que se calcula una liquidación, cuando existen tallos rechazados, entonces estos no deben incluirse en el valor pagado al proveedor.	

Tabla A.16. HU019: Clasificación en fecha posterior

N° 19	HISTORIA DE USUARIO		
Número: HU019		Usuario: Empleado	
Objetivo: Clasificación en fecha posterior			
Prioridad: Alta	Estimación: 5		Iteración: 3
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como Empleado, quiero clasificar una recepción en una fecha diferente a la entrega, para respetar la planificación real del trabajo.			
Criterios de aceptación: Dado que una recepción permanece pendiente, cuando se procesa en una jornada posterior, entonces el Sistema debe permitir registrar la fecha real de clasificación. Dado que se consulta la trazabilidad, cuando la fecha de clasificación difiere de la recepción, entonces ambas fechas deben conservarse sin reemplazarse.			

Tabla A.17. HU020: Reporte diario de postcosecha

N° 20	HISTORIA DE USUARIO		
Número: HU020		Usuario: Administrador y Secretaria	
Objetivo: Reporte diario de postcosecha			
Prioridad: Alta	Estimación: 8		Iteración: 3
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como Administrador y Secretaria, quiero consultar el total de tallos procesados, bunches y cantidades por variedad, para controlar la producción diaria.			
Criterios de aceptación:			
Dado que existen clasificaciones en una fecha, cuando se genera el reporte diario, entonces el Sistema debe totalizar tallos procesados, bunches y resultados por variedad.			
Dado que el reporte es de control productivo, cuando se visualiza, entonces no debe calcular ni mostrar el valor económico de la liquidación.			

Tabla A.18. HU021: Registro de tarifarios

N° 21	HISTORIA DE USUARIO		
Número: HU021		Usuario: Administrador y Secretaria	
Objetivo: Registro de tarifarios			
Prioridad: Alta	Estimación: 8		Iteración: 4
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como Administrador y Secretaria, quiero registrar precios por tallo según variedad, calidad y período de vigencia, para calcular correctamente las liquidaciones.			
Criterios de aceptación:			

N° 21	HISTORIA DE USUARIO
Dado que se define un precio, cuando se ingresan variedad, calidad, valor por tallo y fechas válidas, entonces el Sistema debe guardar el tarifario. Dado que existe un tarifario vigente para la misma variedad y calidad con fechas superpuestas, cuando se intenta guardar otro, entonces el Sistema debe impedir el solapamiento.	

Tabla A.19. HU022: Conservación del historial de tarifas

N° 22	HISTORIA DE USUARIO		
Número: HU022		Usuario: Administrador y Secretaria	
Objetivo: Conservación del historial de tarifas			
Prioridad: Alta	Estimación: 8		Iteración: 4
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como Administrador y Secretaria, quiero cerrar la vigencia de una tarifa y crear una nueva, para mantener inalterables los precios utilizados anteriormente.			
Criterios de aceptación:			
Dado que cambia el precio del mercado, cuando el administrador actualiza el tarifario, entonces el Sistema debe cerrar la vigencia anterior y crear un nuevo registro.			
Dado que una tarifa fue utilizada en una liquidación, cuando se consulta el historial, entonces el valor original debe conservarse sin modificación.			

Tabla A.20. HU023: Generación automática de liquidaciones

N° 23	HISTORIA DE USUARIO		
Número: HU023		Usuario: Administrador y Secretaria	
Objetivo: Generación automática de liquidaciones			
Prioridad: Alta	Estimación: 13		Iteración: 4
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como Administrador y Secretaria, quiero generar la liquidación de un proveedor, para calcular el pago con base en los tallos exportables y la tarifa vigente.			
Criterios de aceptación: Dado que existen clasificaciones exportables de un proveedor, cuando se genera la liquidación, entonces el Sistema debe localizar la tarifa vigente en la fecha de recepción y multiplicarla por los tallos exportables. Dado que existen tallos adicionales exportables, cuando se calcula la liquidación, entonces deben incluirse en el pago; las mallas y los tallos rechazados no deben generar valores.			

Tabla A.21. HU024: Consulta del detalle de liquidación

N° 24	HISTORIA DE USUARIO		
Número: HU024		Usuario: Administrador y Secretaria	
Objetivo: Consulta del detalle de liquidación			
Prioridad: Alta	Estimación: 5		Iteración: 4
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como Administrador y Secretaria, quiero consultar el desglose de una liquidación, para verificar variedades, calidades, cantidades, tarifas y subtotales.			

N° 24	HISTORIA DE USUARIO
Criterios de aceptación: Dado que una liquidación fue generada, cuando se abre su detalle, entonces el Sistema debe mostrar las clasificaciones incluidas, el precio aplicado, los subtotales y el total. Dado que se consulta una liquidación histórica, cuando el tarifario actual cambió, entonces el documento debe conservar los valores usados originalmente.	

Tabla A.22. HU025: Gestión de categorías y productos químicos

N° 25	HISTORIA DE USUARIO		
Número: HU025		Usuario: Administrador y Empleado	
Objetivo: Gestión de categorías y productos químicos			
Prioridad: Media	Estimación: 8		Iteración: 5
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como Administrador y Empleado, quiero registrar categorías y productos químicos, para mantener organizado el inventario independiente del proceso de flores.			
Criterios de aceptación:			
Dado que se registra un producto, cuando se ingresan categoría, nombre, marca, unidad, proveedor, stock y mínimo, entonces el Sistema debe guardarlo.			
Dado que un producto utiliza una unidad definida, cuando se actualiza su información, entonces el Sistema debe mantener esa unidad y no efectuar conversiones automáticas.			

Tabla A.23. HU026: Registro de entradas de inventario

N° 26	HISTORIA DE USUARIO		
Número: HU026		Usuario: Administrador y Empleado	
Objetivo: Registro de entradas de inventario			
Prioridad: Alta	Estimación: 5		Iteración: 5
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como Administrador y Empleado, quiero registrar el ingreso de productos químicos, para incrementar las existencias disponibles.			
Criterios de aceptación:			
Dado que un producto está activo, cuando se registra fecha, proveedor y cantidad positiva en su misma unidad, entonces el Sistema debe aumentar el stock.			
Dado que la cantidad es cero, negativa o utiliza otra unidad, cuando se intenta guardar, entonces el Sistema debe rechazar el movimiento.			

Tabla A.24. HU027: Registro de salidas y consumos parciales

N° 27	HISTORIA DE USUARIO		
Número: HU027		Usuario: Administrador y Empleado	
Objetivo: Registro de salidas y consumos parciales			
Prioridad: Alta	Estimación: 8		Iteración: 5
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como Administrador y Empleado, quiero registrar salidas parciales de productos químicos, para controlar su entrega a una finca o bloque.			
Criterios de aceptación:			

N° 27	HISTORIA DE USUARIO
Dado que existe stock suficiente, cuando se registra fecha, cantidad, destino y motivo, entonces el Sistema debe descontar la cantidad entregada. Dado que se solicita una cantidad superior a la existencia, cuando se intenta guardar la salida, entonces el Sistema debe impedir el movimiento y conservar el stock.	

Tabla A.25. HU028: Alertas de stock mínimo

N° 28	HISTORIA DE USUARIO		
Número: HU028		Usuario: Administrador y Empleado	
Objetivo: Alertas de stock mínimo			
Prioridad: Media	Estimación: 5		Iteración: 5
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como Administrador y Empleado, quiero visualizar alertas de productos con existencias bajas, para gestionar oportunamente su reposición.			
Criterios de aceptación:			
Dado que el stock actual es igual o inferior al mínimo configurado, cuando se consulta el inventario o el panel, entonces el Sistema debe mostrar una alerta.			
Dado que una entrada eleva el stock por encima del mínimo, cuando se actualiza la consulta, entonces la alerta debe dejar de mostrarse.			

Tabla A.26. HU029: Consulta de reportes y bitácora

N° 29	HISTORIA DE USUARIO		
Número: HU029		Usuario: Administrador	
Objetivo: Consulta de reportes y bitácora			
Prioridad: Alta	Estimación: 8		Iteración: 5
Responsables: Evelin Ganchala y Leonardo Quinaluisa			
Descripción: Yo como Administrador, quiero consultar y exportar información operativa junto con la bitácora de acciones, para apoyar el control y la toma de decisiones.			
Criterios de aceptación:			
Dado que existen datos en FlorLY, cuando el administrador aplica filtros de fecha, módulo o usuario, entonces el Sistema debe mostrar los resultados coincidentes y permitir su exportación.			
Dado que se ejecuta una acción crítica, cuando la operación finaliza, entonces la bitácora debe registrar usuario, fecha, módulo, acción e identificación del registro afectado.			

ANEXO B:

Tabla B.1. Iteración 4

ITERACIÓN	ID HU	HISTORIA DE USUARIO	PRIORIDAD	PUNTOS
Iteración 4	HU010	Consulta del historial laboral.	Baja	3
	HU011	Registro de recepción de flor de proveedor.	Alta	8
	HU012	Registro de recepción de flor propia.	Alta	8
Total de puntos Iteración				19

Tabla B.2. Iteración 5

ITERACIÓN	ID HU	HISTORIA DE USUARIO	PRIORIDAD	PUNTOS
Iteración 5	HU013	Registro de múltiples variedades por recepción.	Alta	8
	HU014	Generación del ticket de recepción.	Alta	5
	HU015	Consulta de recepciones pendientes.	Alta	5
Total de puntos Iteración				18

Tabla B.3. Iteración 6

ITERACIÓN	ID HU	HISTORIA DE USUARIO	PRIORIDAD	PUNTOS
Iteración 6	HU016	Registro de clasificación exportable.	Alta	8
	HU017	Registro de tallos adicionales.	Alta	5
	HU018	Registro de flor nacional o rechazada.	Alta	5
Total de puntos Iteración				18

Tabla B.4. Iteración 7

ITERACIÓN	ID HU	HISTORIA DE USUARIO	PRIORIDAD	PUNTOS
Iteración 7	HU019	Clasificación en una fecha posterior.	Alta	5
	HU020	Reporte diario de postcosecha.	Alta	8
	HU021	Registro de tarifarios.	Alta	8
Total de puntos Iteración				21

Tabla B.5. Iteración 8

ITERACIÓN	ID HU	HISTORIA DE USUARIO	PRIORIDAD	PUNTOS
Iteración 8	HU02 2	Conservación del historial de tarifas.	Alta	8
	HU02 3	Generación automática de liquidaciones.	Alta	13
	HU02 4	Consulta del detalle de liquidación.	Alta	5
Total de puntos Iteración				26

Tabla B.6. Iteración 9

ITERACIÓN	ID HU	HISTORIA DE USUARIO	PRIORIDAD	PUNTOS
Iteración 9	HU02 5	Gestión de categorías y productos químicos.	Media	8
	HU02 6	Registro de entradas de inventario.	Alta	5
	HU02 7	Registro de salidas y consumos parciales.	Alta	8
Total de puntos Iteración				21

Tabla B.7. Iteración 10

ITERACIÓN	ID HU	HISTORIA DE USUARIO	PRIORIDAD	PUNTOS
Iteración 10	HU02 8	Alertas de stock mínimo.	Media	5
	HU02 9	Consulta de reportes y bitácora.	Alta	8
Total de puntos Iteración				13

ANEXO C:

Tabla C.1. Codificación HU003

ITERACIÓN 1 – HU003	
HISTORIA USUARIO	DE HU003 – Consulta del panel principal.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para consulta del panel principal sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 249 def inicioSistema(request): 250 ahora = timezone.localtime() 251 es_empleado = request.user.groups.filter(name='Empleado').exists() 252 if not es_empleado and ahora.hour >= 15: 253 from Aplicaciones.liquidaciones.services import generar_liquidaciones_vencidas 254 generadas, errores = generar_liquidaciones_vencidas(ahora.date()) 255 if generadas: 256 messages.success(request, f'{len(generadas)} reporte(s) mensual(es) se generaron automáticamente') 257 for error in errores: 258 messages.error(request, f'Liquidación pendiente: {error}') 259 contexto = { 260 'total_proveedores': Proveedor.objects.filter(estado=True).count(), 261 'total_variedades': Variedad.objects.filter(estado=True).count(), 262 'total_fincas': Finca.objects.filter(estado=True).count(), 263 'total_personal': Personal.objects.filter(estado=True).count(), 264 } 265 vacaciones = VacacionPersonal.objects.filter(266 personal_estado=True, </pre>
REFACTORIZACIÓN	Se reorganizó la implementación de consulta del panel principal, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.2. Codificación HU004

ITERACIÓN 2 – HU004	
HISTORIA DE USUARIO	HU004 – Registro de proveedores.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para registro de proveedores sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 401 def guardarProveedor(request): 402 cedula_ruc = request.POST.get('cedula_ruc', '').strip() 403 telefono = request.POST.get('telefono', '').strip() 404 tipo_documento = 'RUC' if len(cedula_ruc) == 13 else 'CEDULA' 405 if len(cedula_ruc) not in (10, 13) or not _documento_identidad_vali 406 messages.error(request, 'Ingrese una cédula de 10 dígitos o un 407 return redirect('nuevoProveedor') 408 if Proveedor.objects.filter(cedula_ruc=cedula_ruc).exists(): 409 messages.error(request, 'Ya existe un proveedor registrado con 410 return redirect('nuevoProveedor') 411 if not telefono_ecuatoriano_valido(telefono): 412 messages.error(request, 'Ingrese un celular nacional 09 de 10 d 413 return redirect('nuevoProveedor') 414 Proveedor.objects.create(415 codigo_proveedor=_generar_codigo(Proveedor, 'codigo_proveedor', 416 nombres=request.POST['nombres'], 417 apellidos=request.POST['apellidos'], 418 cedula_ruc=cedula_ruc, </pre>
REFACTORIZACIÓN	Se reorganizó la implementación de registro de proveedores, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.3. Codificación HU005

ITERACIÓN 2 – HU005	
HISTORIA USUARIO	DE HU005 – Administración de proveedores.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para administración de proveedores sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 456 def cambiarEstadoProveedor(request, codigo): 457 proveedor = get_object_or_404(Proveedor, pk=codigo) 458 proveedor.estado = not proveedor.estado 459 proveedor.save() 460 messages.success(request, 'Estado del proveedor actualizado exitoso') 461 return redirect('/proveedores') 462 463 464 # Variedades 465 def inicioVariedad(request): 466 return render(request, 'variedades/inicioVariedad.html', { 467 'variedades': Variedad.objects.all().order_by('-fecha_creacion') 468 }) 469 470 471 def nuevaVariedad(request): 472 return render(request, 'variedades/nuevaVariedad.html') 473 </pre>
REFACTORIZACIÓN	Se reorganizó la implementación de administración de proveedores, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.4. Codificación HU006

ITERACIÓN 2 – HU006	
HISTORIA DE USUARIO	HU006 – Gestión de fincas.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para gestión de fincas sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 558 def guardarFinca(request): 559 Finca.objects.create(560 codigo_finca=_generar_codigo(Finca, 'codigo_finca', 'FIN'), 561 nombre=request.POST['nombre'], 562 ubicacion=request.POST['ubicacion'].strip(), 563) 564 messages.success(request, 'Finca guardada exitosamente') 565 return redirect('/fincas') 566 567 568 def editarFinca(request, codigo): 569 return render(request, 'fincas/editarFinca.html', { 570 'fincaEditar': get_object_or_404(Finca, pk=codigo) 571 }) 572 573 574 @require_POST 575 def procesarEdicionFinca(request): </pre>
REFACTORIZACIÓN	Se reorganizó la implementación de gestión de fincas, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.5. Codificación HU007

ITERACIÓN 3 – HU007	
HISTORIA DE USUARIO	HU007 – Gestión de variedades.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para gestión de variedades sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 476 def guardarVariedad(request): 477 imagen = request.FILES.get('imagen') 478 if not imagen: 479 messages.error(request, 'Seleccione una imagen JPG o PNG para l 480 return redirect('nuevaVariedad') 481 try: 482 validate_and_secure_image(imagen) 483 except ValidationError as error: 484 messages.error(request, ' '.join(error.messages)) 485 return redirect('nuevaVariedad') 486 Variedad.objects.create(487 codigo_variedad=_generar_codigo(Variedad, 'codigo_variedad', 'V 488 nombre=request.POST['nombre'], 489 imagen=imagen, 490) 491 messages.success(request, 'Variedad guardada exitosamente') 492 return redirect('inicioVariedad') 493 </pre>
REFACTORIZACIÓN	Se reorganizó la implementación de gestión de variedades, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.6. Codificación HU008

ITERACIÓN 3 – HU008	
HISTORIA DE USUARIO	HU008 – Registro de personal.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para registro de personal sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 609 def guardarPersonal(request): 610 finca = get_object_or_404(Finca, pk=request.POST['finca']) 611 cedula = request.POST.get('cedula', '').strip() 612 telefono = request.POST.get('telefono', '').strip() 613 if not _documento_identidad_valido(cedula): 614 messages.error(request, 'La cédula ingresada no cumple la estruc 615 return redirect('nuevoPersonal') 616 if Personal.objects.filter(cedula=cedula).exists(): 617 messages.error(request, 'Ya existe un empleado registrado con e 618 return redirect('nuevoPersonal') 619 if not telefono_ecuatoriano_valido(telefono): 620 messages.error(request, 'Ingrese un celular nacional 09 de 10 d 621 return redirect('nuevoPersonal') 622 Personal.objects.create(623 codigo_personal=_generar_codigo(Personal, 'codigo_personal', 'P 624 nombres=request.POST['nombres'], 625 apellidos=request.POST['apellidos'], 626 cedula=cedula,</pre>
REFACTORIZACIÓN	Se reorganizó la implementación de registro de personal, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.7. Codificación HU009

ITERACIÓN 3 – HU009	
HISTORIA USUARIO	DE HU009 – Gestión de permisos y vacaciones.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para gestión de permisos y vacaciones sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 693 def guardarPermisoPersonal(request, codigo): 694 persona = get_object_or_404(Personal, pk=codigo) 695 try: 696 fecha_desde = datetime.strptime(request.POST['fecha_desde'], '%Y-%m-%d') 697 fecha_hasta = datetime.strptime(request.POST['fecha_hasta'], '%Y-%m-%d') 698 hoy = timezone.localdate() 699 if fecha_desde < hoy or fecha_hasta < hoy: 700 raise ValueError('La ausencia no puede registrarse en una fecha anterior a hoy') 701 if fecha_hasta < fecha_desde: 702 raise ValueError('La fecha final no puede ser anterior a la fecha de inicio') 703 hora_salida_texto = request.POST.get('hora_salida', '') 704 hora_retorno_texto = request.POST.get('hora_retorno', '') 705 hora_salida = datetime.strptime(hora_salida_texto, '%H:%M').time() 706 hora_retorno = datetime.strptime(hora_retorno_texto, '%H:%M').time() 707 if bool(hora_salida) != bool(hora_retorno): 708 raise ValueError('Debe ingresar tanto la hora de salida como la hora de retorno') 709 if hora_salida and hora_retorno: 710 inicio_jornada = datetime.strptime('08:00', '%H:%M').time() </pre>
REFACTORIZACIÓN	Se reorganizó la implementación de gestión de permisos y vacaciones, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.8. Codificación HU010

ITERACIÓN 4 – HU010	
HISTORIA USUARIO	DE HU010 – Consulta del historial laboral.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para consulta del historial laboral sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 678 def expedientePersonal(request, codigo): 679 persona = get_object_or_404(Personal.objects.select_related('finca' 680 hoy = timezone.localtime() 681 return render(request, 'personal/expedientePersonal.html', { 682 'persona': persona, 683 'permisos': persona.permisos.all().order_by('-fecha_registro', 684 'vacaciones': persona.vacaciones.all().order_by('-fecha_registro', 685 'documentos': persona.documentos.all().order_by('-fecha_registro', 686 'motivos_permiso': PermisoPersonal.MOTIVOS, 687 'fecha_actual': hoy.isoformat(), 688 'fecha_minima_vacaciones': _fecha_minima_inicio_vacaciones(hoy) 689 }) 690 691 692 @require_POST 693 def guardarPermisoPersonal(request, codigo): 694 persona = get_object_or_404(Personal, pk=codigo) 695 try: </pre>
REFACTORIZACIÓN	Se reorganizó la implementación de consulta del historial laboral, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.9. Codificación HU011

ITERACIÓN 4 – HU011	
HISTORIA DE USUARIO	HU011 – Registro de recepción de flor de proveedor.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para registro de recepción de flor de proveedor sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 79 def guardarRecepcion(request): 80 if request.method != 'POST': 81 return redirect('nuevaRecepcion') 82 momento_recepcion = timezone.localtime(timezone.now()) 83 if not _horario_recepcion_valido(momento_recepcion): 84 messages.error(85 request, 86 'Recepción fuera del horario permitido: lunes a viernes de 87 'y sábados de 08:00 a 12:00. Los domingos no se registran r 88) 89 return redirect('nuevaRecepcion') 90 proveedor_codigo = request.POST.get('proveedor') 91 finca_codigo = request.POST.get('finca') 92 empleado_codigo = request.POST.get('empleado_receptor') 93 if bool(proveedor_codigo) == bool(finca_codigo): 94 messages.error(request, 'Seleccione únicamente un proveedor o u 95 return redirect('/recepciones/nueva') 96 empleado = get_object_or_404(</pre>
REFACTORIZACIÓN	Se reorganizó la implementación de registro de recepción de flor de proveedor, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.10. Codificación HU012

ITERACIÓN 4 – HU012	
HISTORIA DE USUARIO	HU012 – Registro de recepción de flor propia.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para registro de recepción de flor propia sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 90 proveedor_codigo = request.POST.get('proveedor') 91 finca_codigo = request.POST.get('finca') 92 empleado_codigo = request.POST.get('empleado_receptor') 93 if bool(proveedor_codigo) == bool(finca_codigo): 94 messages.error(request, 'Seleccione únicamente un proveedor o u 95 return redirect('/recepciones/nueva') 96 empleado = get_object_or_404(97 Personal, 98 pk=empleado_codigo, 99 estado=True, 100 area__icontains='postcosecha', 101) 102 103 variedades = request.POST.getlist('variedad') 104 cantidades = request.POST.getlist('cantidad_mallas') 105 tallos_por_malla = request.POST.getlist('tallos_por_malla') 106 if not variedades or not (len(variedades) == len(cantidades) == len 107 messages.error(request, 'La recepción debe contener al menos un </pre>
REFACTORIZACIÓN	Se reorganizó la implementación de registro de recepción de flor propia, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.11. Codificación HU013

ITERACIÓN 5 – HU013		
HISTORIA USUARIO	DE	HU013 – Registro de múltiples variedades por recepción.
PRUEBAS REALIZADAS		Prueba unitaria de validación: Se verificó que los datos requeridos para registro de múltiples variedades por recepción sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO		<pre> 103 variedades = request.POST.getlist('variedad') 104 cantidades = request.POST.getlist('cantidad_mallas') 105 tallos_por_malla = request.POST.getlist('tallos_por_malla') 106 if not variedades or not (len(variedades) == len(cantidades) == len 107 messages.error(request, 'La recepción debe contener al menos un 108 return redirect('nuevaRecepcion') 109 110 try: 111 cantidades = [int(cantidad) for cantidad in cantidades] 112 tallos_por_malla = [int(tallos) for tallos in tallos_por_malla] 113 if any(valor < 1 for valor in cantidades + tallos_por_malla): 114 raise ValueError 115 except (TypeError, ValueError): 116 messages.error(request, 'Las mallas y los tallos por malla debe 117 return redirect('nuevaRecepcion') 118 119 ultimo_numero = Recepcion.objects.order_by('-numero_recepcion').val 120 recepcion = Recepcion.objects.create(</pre>
REFACTORIZACIÓN		Se reorganizó la implementación de registro de múltiples variedades por recepción, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.12. Codificación HU014

ITERACIÓN 5 – HU014	
HISTORIA USUARIO	DE HU014 – Generación del ticket de recepción.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para generación del ticket de recepción sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 140 def ticketRecepcion(request, codigo): 141 recepcion = get_object_or_404(142 Recepcion.objects.select_related('proveedor', 'finca').prefetch 143 pk=codigo, 144) 145 return render(request, 'recepciones/ticketRecepcion.html', {'recep 146 147 148 def detalleRecepcion(request, codigo): 149 recepcion = get_object_or_404(150 Recepcion.objects.select_related('proveedor', 'finca').prefetch 151 pk=codigo, 152) 153 return render(request, 'recepciones/detalleRecepcion.html', {'recep 154 155 156 def editarRecepcion(request, codigo): 157 return render(request, 'recepciones/editarRecepcion.html', { </pre>
REFACTORIZACIÓN	Se reorganizó la implementación de generación del ticket de recepción, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.13. Codificación HU015

ITERACIÓN 5 – HU015	
HISTORIA USUARIO	DE HU015 – Consulta de recepciones pendientes.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para consulta de recepciones pendientes sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 209 def inicioClasificacion(request): 210 clasificaciones = Clasificacion.objects.select_related(211 'repcion_proveedor', 'repcion_finca' 212).prefetch_related(213 'repcion_detalle_repcion_set_variedad', 214 'detalleclasificacion_set_detalle_repcion_variedad', 215).order_by('-fecha_creacion', '-pk') 216 return render(request, 'clasificaciones/inicioClasificacion.html', 217 218 219 def reporteDiarioClasificacion(request): 220 fecha = request.GET.get('fecha') or timezone.localtime().isoformat() 221 try: 222 fecha_consulta = datetime.strptime(fecha, '%Y-%m-%d').date() 223 except ValueError: 224 fecha_consulta = timezone.localtime() 225 226 resultados = DetalleClasificacion.objects.filter(</pre>
REFACTORIZACIÓN	Se reorganizó la implementación de consulta de recepciones pendientes, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.14. Codificación HU016

ITERACIÓN 6 – HU016	
HISTORIA DE USUARIO	HU016 – Registro de clasificación exportable.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para registro de clasificación exportable sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 302 def guardarClasificacion(request): 303 detalles_ids = request.POST.getlist('detalle_recepcion') 304 mallas = request.POST.getlist('cantidad_mallas_procesadas') 305 exportables_optimos = request.POST.getlist('tallos_exportables_opti 306 exportables_estandar = request.POST.getlist('tallos_exportables_est 307 nacionales = request.POST.getlist('tallos_nacionales') 308 sobrantes = request.POST.getlist('tallos_sobrantes') 309 if not detalles_ids: 310 messages.error(request, 'La clasificación debe contener al meno 311 return redirect('/clasificaciones/nueva') 312 313 detalles_seleccionados = list(DetalleRecepcion.objects.filter(314 pk__in=detalles_ids 315).select_related('recepcion', 'variedad')) 316 if len(detalles_seleccionados) != len(set(detalles_ids)): 317 messages.error(request, 'El lote contiene materiales inválidos 318 return redirect('nuevaClasificacion') 319 recepcion_ids = {detalle.recepcion_id for detalle in detalles_selec </pre>
REFACTORIZACIÓN	Se reorganizó la implementación de registro de clasificación exportable, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.15. Codificación HU017

ITERACIÓN 6 – HU017	
HISTORIA DE USUARIO	HU017 – Registro de tallos adicionales.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para registro de tallos adicionales sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 308 sobrantes = request.POST.getlist('tallos_sobrantes') 309 if not detalles_ids: 310 messages.error(request, 'La clasificación debe contener al meno 311 return redirect('/clasificaciones/nueva') 312 313 detalles_seleccionados = list(DetalleRecepcion.objects.filter(314 pk__in=detalles_ids 315).select_related('recepcion', 'variedad')) 316 if len(detalles_seleccionados) != len(set(detalles_ids)): 317 messages.error(request, 'El lote contiene materiales inválidos 318 return redirect('nuevaClasificacion') 319 recepcion_ids = {detalle.recepcion_id for detalle in detalles_selec 320 if len(recepcion_ids) != 1: 321 messages.error(request, 'Cada clasificación debe pertenecer a u 322 return redirect('nuevaClasificacion') 323 recepcion = detalles_seleccionados[0].recepcion 324 try: 325 fecha_clasificacion = datetime.strptime(</pre>
REFACTORIZACIÓN	Se reorganizó la implementación de registro de tallos adicionales, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.16. Codificación HU018

ITERACIÓN 6 – HU018	
HISTORIA DE USUARIO	HU018 – Registro de flor nacional o rechazada.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para registro de flor nacional o rechazada sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 307 nacionales = request.POST.getlist('tallos_nacionales') 308 sobrantes = request.POST.getlist('tallos_sobrantes') 309 if not detalles_ids: 310 messages.error(request, 'La clasificación debe contener al meno 311 return redirect('/clasificaciones/nueva') 312 313 detalles_seleccionados = list(DetalleRecepcion.objects.filter(314 pk__in=detalles_ids 315).select_related('recepcion', 'variedad')) 316 if len(detalles_seleccionados) != len(set(detalles_ids)): 317 messages.error(request, 'El lote contiene materiales inválidos 318 return redirect('nuevaClasificacion') 319 recepcion_ids = {detalle.recepcion_id for detalle in detalles_selec 320 if len(recepcion_ids) != 1: 321 messages.error(request, 'Cada clasificación debe pertenecer a u 322 return redirect('nuevaClasificacion') 323 recepcion = detalles_seleccionados[0].recepcion 324 try: </pre>
REFACTORIZACIÓN	Se reorganizó la implementación de registro de flor nacional o rechazada, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.17. Codificación HU019

ITERACIÓN 7 – HU019	
HISTORIA USUARIO	DE HU019 – Clasificación en fecha posterior.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para clasificación en fecha posterior sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 50 def _fecha_clasificacion_valida(fecha, hoy, fecha_recepcion): 51 """Permite hoy y los dos días siguientes, sin superar el plazo del 52 return (53 hoy <= fecha <= hoy + timedelta(days=2) 54 and fecha_recepcion <= fecha <= fecha_recepcion + timedelta(day 55) 56 57 58 # Recepciones 59 def inicioRecepcion(request): 60 recepciones = Recepcion.objects.select_related('proveedor', 'finca' 61 'detallerecepcion_set__variedad' 62).order_by('-fecha_creacion', '-pk') 63 return render(request, 'recepciones/inicioRecepcion.html', {'recepc 64 65 66 def nuevaRecepcion(request): 67 return render(request, 'recepciones/nuevaRecepcion.html', { </pre>
REFACTORIZACIÓN	Se reorganizó la implementación de clasificación en fecha posterior, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.18. Codificación HU020

ITERACIÓN 7 – HU020	
HISTORIA USUARIO	DE HU020 – Reporte diario de postcosecha.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para reporte diario de postcosecha sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 219 def reporteDiarioClasificacion(request): 220 fecha = request.GET.get('fecha') or timezone.localtime().isoformat(221 try: 222 fecha_consulta = datetime.strptime(fecha, '%Y-%m-%d').date() 223 except ValueError: 224 fecha_consulta = timezone.localtime() 225 226 resultados = DetalleClasificacion.objects.filter(227 clasificacion_fecha_clasificacion=fecha_consulta, 228 clasificacion_recepcion_isnull=False, 229).values(230 'clasificacion_id', 231 'clasificacion_recepcion_proveedor_id', 232 'clasificacion_recepcion_finca_id', 233 'detalle_recepcion_variedad_id', 234 'largo', 235).annotate(total_tallos=Sum('tallos_exportables')) 236 </pre>
REFACTORIZACIÓN	Se reorganizó la implementación de reporte diario de postcosecha, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.19. Codificación HU021

ITERACIÓN 7 – HU021	
HISTORIA DE USUARIO	HU021 – Registro de tarifarios.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para registro de tarifarios sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 579 def guardarTarifario(request): 580 variedad = get_object_or_404(Variedad, pk=request.POST.get('variedad')) 581 try: 582 fecha_inicio = datetime.strptime(request.POST.get('fecha_inicio'), '%Y-%m-%d') 583 precio_optimo = Decimal(request.POST.get('precio_optimo')) or '0' 584 precio_estandar = Decimal(request.POST.get('precio_estandar')) or '0' 585 if precio_optimo <= 0 or precio_estandar <= 0: 586 raise ValueError 587 except (InvalidOperation, ValueError): 588 messages.error(request, 'Ingrese una fecha válida y precios mayores a 0') 589 return redirect('nuevoTarifario') 590 591 tarifario_activo = Tarifario.objects.filter(variedad=variedad, esta_activo=True) 592 if tarifario_activo and fecha_inicio <= tarifario_activo.fecha_inicio: 593 messages.error(594 request, 595 f'La nueva vigencia debe iniciar después del {tarifario_activo.fecha_inicio.strftime("%Y-%m-%d")}. ' 596 f'Si el tarifario actual está equivocado y no se ha utilizado, por favor elimínelo.' 597) 598 return redirect('nuevoTarifario') 599 600 tarifario = Tarifario(variedad=variedad, fecha_inicio=fecha_inicio, precio_optimo=precio_optimo, precio_estandar=precio_estandar) 601 tarifario.save() 602 messages.success(request, 'Tarifario registrado exitosamente.') 603 return redirect('nuevoTarifario') </pre>
REFACTORIZACIÓN	Se reorganizó la implementación de registro de tarifarios, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.20. Codificación HU022

ITERACIÓN 8 – HU022	
HISTORIA USUARIO	DE HU022 – Conservación del historial de tarifas.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para conservación del historial de tarifas sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 667 def cerrarTarifario(request, codigo): 668 tarifario = get_object_or_404(Tarifario, pk=codigo) 669 if not tarifario.estado: 670 messages.error(request, 'El tarifario ya se encuentra cerrado') 671 return redirect('inicioTarifario') 672 try: 673 fecha_fin = datetime.strptime(request.POST.get('fecha_fin', ''), 674 '%Y-%m-%d') 675 if fecha_fin < tarifario.fecha_inicio: 676 raise ValueError 677 except ValueError: 678 messages.error(request, 'La fecha de cierre no puede ser anterior 679 a la fecha de inicio') 680 return redirect('inicioTarifario') 681 tarifario.fecha_fin = fecha_fin 682 tarifario.estado = False 683 tarifario.save() 684 messages.success(request, 'Vigencia del tarifario cerrada exitosamente') 685 return redirect('/tarifarios') </pre>
REFACTORIZACIÓN	Se reorganizó la implementación de conservación del historial de tarifas, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.21. Codificación HU023

ITERACIÓN 8 – HU023	
HISTORIA DE USUARIO	HU023 – Generación automática de liquidaciones.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para generación automática de liquidaciones sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 46 def generar_liquidacion_proveedor(proveedor, fecha_generacion, fecha_co 47 existente = Liquidacion.objects.filter(48 proveedor=proveedor, 49 fecha_liquidacion=fecha_generacion, 50).first() 51 if existente: 52 return existente, False 53 54 ids_elegibles = Clasificacion.objects.filter(55 recepcion__proveedor=proveedor, 56 recepcion__fecha_recepcion__date__lte=fecha_corte, 57 detalleclasificacion__isnull=False, 58 liquidaciones__isnull=True, 59).values_list('pk', flat=True).distinct() 60 clasificaciones = list(61 Clasificacion.objects.select_for_update(of='self',)).filter(62 pk__in=ids_elegibles, 63).select_related('recepcion').prefetch_related(</pre>
REFACTORIZACIÓN	Se reorganizó la implementación de generación automática de liquidaciones, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.22. Codificación HU024

ITERACIÓN 8 – HU024	
HISTORIA USUARIO	DE HU024 – Consulta del detalle de liquidación.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para consulta del detalle de liquidación sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 40 def verliquidacion(request, codigo): 41 liquidacion = get_object_or_404(Liquidacion.objects.select_related(42 materiales = DetalleRecepcion.objects.filter(43 recepcion__clasificacion__liquidaciones=liquidacion, 44).select_related('recepcion', 'variedad').prefetch_related('detalle 45 filas = defaultdict(lambda: { 46 'mallas': 0, 'tallos_recibidos': 0, 'optimos': 0, 'estandar': 0 47 'nacionales': 0, 'sobrantes': 0, 48 }) 49 for material in materiales: 50 fecha = material.recepcion.fecha_recepcion.date() 51 fila = filas[(fecha, material.variedad_id)] 52 fila['fecha'] = fecha 53 fila['variedad'] = material.variedad.nombre 54 fila['mallas'] += material.cantidad_mallas 55 fila['tallos_recibidos'] += material.total_tallos 56 for resultado in material.detalleclasificacion_set.all(): 57 calidad = 'optimos' if resultado.largo == 'OPTIMO' else 'es </pre>
REFACTORIZACIÓN	Se reorganizó la implementación de consulta del detalle de liquidación, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.23. Codificación HU025

ITERACIÓN 9 – HU025		
HISTORIA USUARIO	DE	HU025 – Gestión de categorías y productos químicos.
PRUEBAS REALIZADAS		Prueba unitaria de validación: Se verificó que los datos requeridos para gestión de categorías y productos químicos sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO		<pre> 259 def guardarProducto(request): 260 Producto.objects.create(261 codigo_producto=generar_codigo(Producto, 'codigo_producto', 'P 262 categoria=get_object_or_404(Categoria, pk=request.POST['categor 263 unidad_medida=get_object_or_404(UnidadMedida, pk=request.POST[' 264 nombre=request.POST['nombre'].strip(), 265 marca=request.POST.get('marca', '').strip(), 266 stock_minimo=Decimal(request.POST['stock_minimo'].replace(',', '' 267) 268 messages.success(request, 'Producto guardado exitosamente') 269 return redirect('/productos') 270 271 272 def editarProducto(request, codigo): 273 return render(request, 'productos/editarProducto.html', { 274 'productoEditar': get_object_or_404(Producto, pk=codigo), 275 'categorias': Categoria.objects.filter(estado=True).order_by('n 276 'unidades': UnidadMedida.objects.filter(estado=True).order_by(' </pre>
REFACTORIZACIÓN		Se reorganizó la implementación de gestión de categorías y productos químicos, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.24. Codificación HU026

ITERACIÓN 9 – HU026	
HISTORIA DE USUARIO	HU026 – Registro de entradas de inventario.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para registro de entradas de inventario sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 319 def guardarEntrada(request): 320 productos_ids = request.POST.getlist('producto') 321 presentaciones = request.POST.getlist('tipo_presentacion') 322 contenidos = request.POST.getlist('contenido_presentacion') 323 envases = request.POST.getlist('cantidad_envases') 324 if not productos_ids: 325 messages.error(request, 'La entrada debe contener al menos un p 326 return redirect('/entradas/nueva') 327 328 entrada = Entrada.objects.create(329 codigo_entrada=_generar_codigo(Entrada, 'codigo_entrada', 'ENT' 330 proveedor_insumo=get_object_or_404(ProveedorInsumo, pk=request. 331 fecha_entrada=request.POST['fecha_entrada'], 332 numero_factura=request.POST.get('numero_factura', '').strip(), 333 observacion=request.POST.get('observacion', '').strip(), 334) 335 for indice, producto_id in enumerate(productos_ids): 336 producto = Producto.objects.select_for_update().get(pk=producto </pre>
REFACTORIZACIÓN	Se reorganizó la implementación de registro de entradas de inventario, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.25. Codificación HU027

ITERACIÓN 9 – HU027	
HISTORIA USUARIO	DE HU027 – Registro de salidas y consumos parciales.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para registro de salidas y consumos parciales sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 378 def guardarSalida(request): 379 productos_ids = request.POST.getlist('producto') 380 cantidades = request.POST.getlist('cantidad_salida') 381 if not productos_ids: 382 messages.error(request, 'La salida debe contener al menos un pr 383 return redirect('/salidas/nueva') 384 385 salida = Salida.objects.create(386 codigo_salida=generar_codigo(Salida, 'codigo_salida', 'SAL', 6 387 finca=get_object_or_404(Finca, pk=request.POST['finca']), 388 fecha_salida=request.POST['fecha_salida'], 389 destino=request.POST.get('destino', '').strip(), 390 motivo=request.POST['motivo'].strip(), 391 observacion=request.POST.get('observacion', '').strip(), 392) 393 for indice, producto_id in enumerate(productos_ids): 394 producto = Producto.objects.select_for_update().get(pk=producto 395 cantidad = Decimal(cantidades[indice].replace(',','.')) </pre>
REFACTORIZACIÓN	Se reorganizó la implementación de registro de salidas y consumos parciales, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.26. Codificación HU028

ITERACIÓN 10 – HU028	
HISTORIA USUARIO	DE HU028 – Alertas de stock mínimo.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para alertas de stock mínimo sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 266 stock_minimo=Decimal(request.POST['stock_minimo'].replace(',','')) 267) 268 messages.success(request, 'Producto guardado exitosamente') 269 return redirect('/productos') 270 271 272 def editarProducto(request, codigo): 273 return render(request, 'productos/editarProducto.html', { 274 'productoEditar': get_object_or_404(Producto, pk=codigo), 275 'categorias': Categoria.objects.filter(estado=True).order_by('n'), 276 'unidades': UnidadMedida.objects.filter(estado=True).order_by('n') 277 }) 278 279 280 @require_POST 281 def procesarEdicionProducto(request): 282 producto = get_object_or_404(Producto, pk=request.POST['codigo_producto']) 283 producto.categoria = get_object_or_404(Categoria, pk=request.POST['categoria']) </pre>
REFACTORIZACIÓN	Se reorganizó la implementación de alertas de stock mínimo, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

Tabla C.27. Codificación HU029

ITERACIÓN 10 – HU029	
HISTORIA USUARIO	DE HU029 – Consulta de reportes y bitácora.
PRUEBAS REALIZADAS	Prueba unitaria de validación: Se verificó que los datos requeridos para consulta de reportes y bitácora sean válidos antes de ejecutar la operación. Prueba de persistencia y respuesta: Se comprobó que la información se guarde correctamente y que el Sistema muestre el resultado esperado.
CÓDIGO IMPLEMENTADO	<pre> 6 def inicioBitacora(request): 7 registros = Bitacora.objects.select_related('usuario').order_by('-f 8 return render(request, 'auditoria/inicioBitacora.html', {'registros 9 10 11 def verRegistroBitacora(request, id): 12 return render(request, 'auditoria/verRegistroBitacora.html', { 13 'registro': get_object_or_404(Bitacora.objects.select_related(' 14 }) </pre>
REFACTORIZACIÓN	Se reorganizó la implementación de consulta de reportes y bitácora, separando las validaciones de la operación principal y reutilizando las funciones del módulo. Además, se controlaron los errores y los campos obligatorios para mejorar la estabilidad, la legibilidad y el mantenimiento del código.

ANEXO D

MANUAL DE USUARIO

SISTEMA WEB FLORLY

Guía para Administrador, Secretaria y Empleado

ROLES Y PERMISOS

Rol	Funciones principales	Restricciones
Administrador	Acceso general; administra usuarios, catálogos, personal, postcosecha, liquidaciones, inventario, reportes y auditoría.	No debe retirar su propio acceso administrativo.
Secretaria	Gestiona personal, fincas, expedientes y liquidaciones; consulta reportes administrativos.	No consulta la bitácora ni opera los módulos no autorizados.
Empleado	Consulta proveedores y variedades; registra recepciones, clasificaciones y reportes productivos.	No accede a usuarios, personal, liquidaciones, inventario ni auditoría.

PANEL SEGÚN EL ROL

Al iniciar sesión, FlorLY identifica el rol y presenta únicamente los módulos y acciones permitidos. Esta separación protege la información y evita cambios fuera de las responsabilidades asignadas.

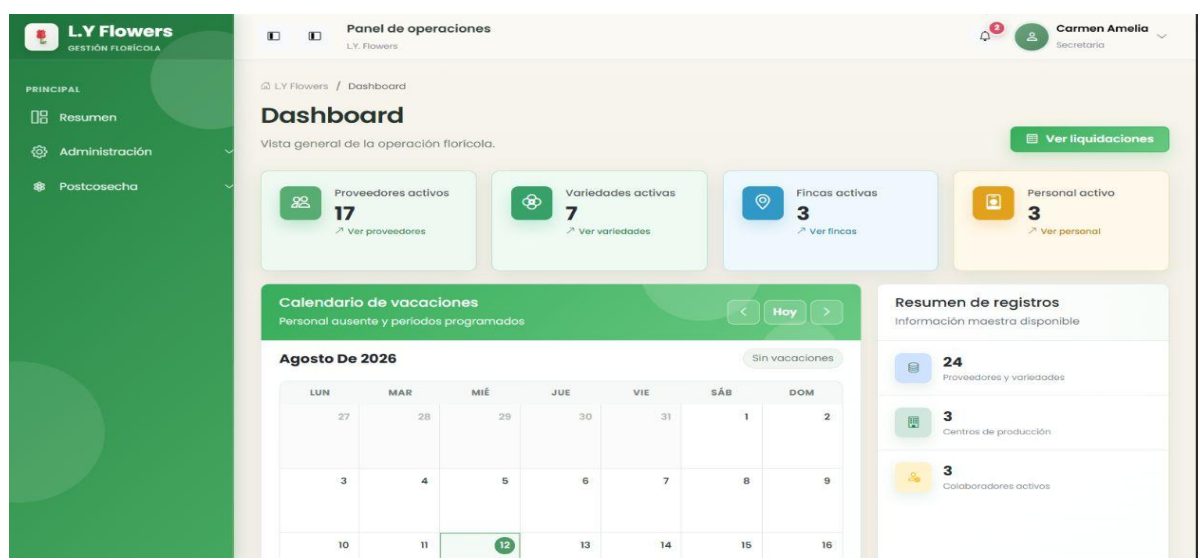


Figura 1. Panel principal mostrado a la Secretaria.

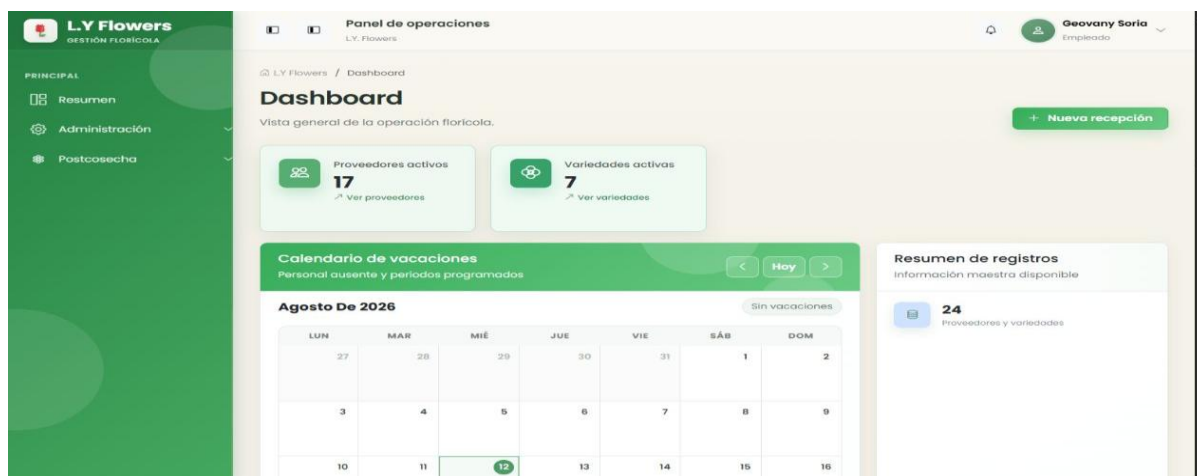


Figura 2. Panel principal mostrado al Empleado.

ACCESO Y CIERRE DE SESIÓN

Iniciar sesión

1. Abra la dirección web proporcionada para FlorLY.
2. Escriba el nombre de usuario en el campo Usuario.
3. Ingrese la contraseña en el campo Contraseña.
4. Presione Ingresar de forma segura.
5. Verifique que aparezca el panel principal correspondiente a su rol.

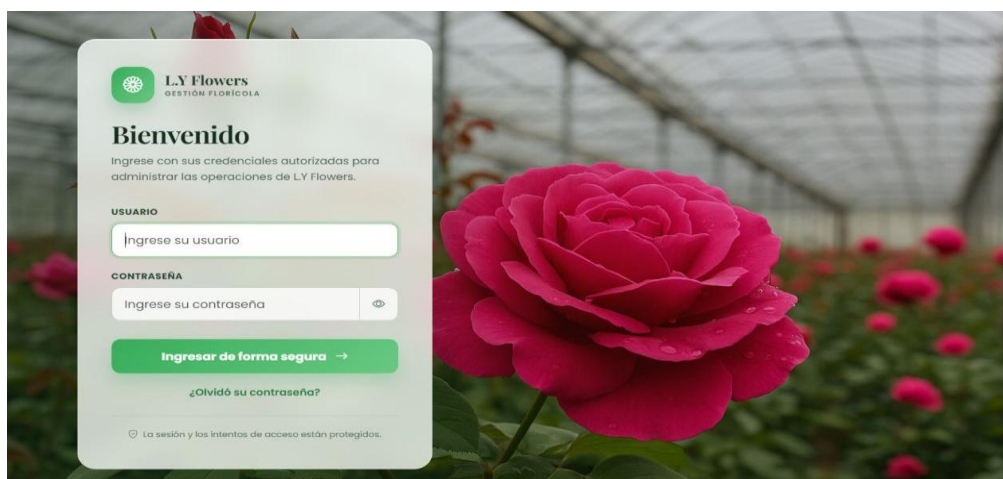


Figura 3. Pantalla de inicio de sesión de FlorLY.

Nota: Después de varios intentos fallidos el acceso puede bloquearse temporalmente. Compruebe las credenciales antes de volver a intentarlo.

RECUPERAR EL ACCESO

Seleccione ¿Olvidó su contraseña? y siga las indicaciones enviadas al correo autorizado. El enlace de recuperación tiene una vigencia limitada.

CERRAR SESIÓN

Abra el menú del usuario ubicado en la esquina superior derecha y seleccione Cerrar sesión. No cierre solamente la pestaña cuando trabaje en un equipo compartido.

ELEMENTOS GENERALES DE LA INTERFAZ

- Menú lateral: agrupa los módulos disponibles según el rol.
- Ruta de navegación: indica el módulo y la pantalla actual.
- Botones de acción: permiten crear registros, generar reportes, editar o consultar detalles.
- Buscador: filtra la información visible en los listados.
- Paginación: permite recorrer listados extensos.
- Menú del usuario: muestra la cuenta activa y permite cerrar la sesión.

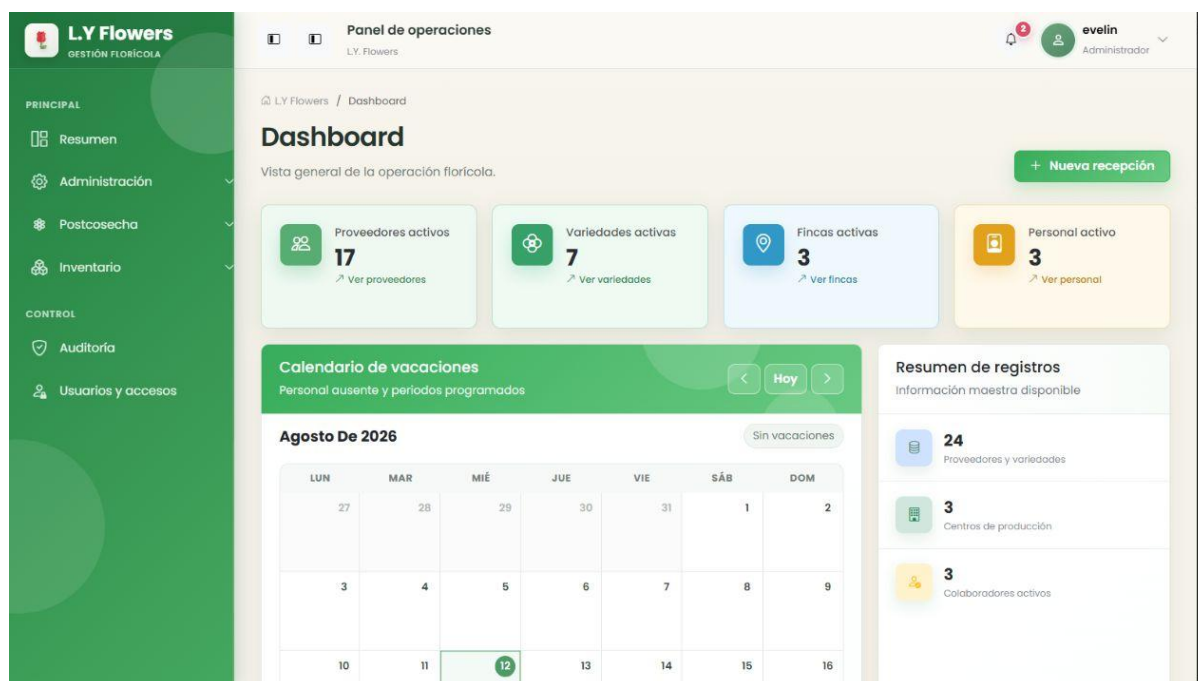


Figura 4. Elementos generales del panel del Administrador.

PANEL PRINCIPAL

El panel resume la operación florícola mediante contadores, calendario de vacaciones, registros recientes e indicadores. Los accesos rápidos cambian de acuerdo con el rol.

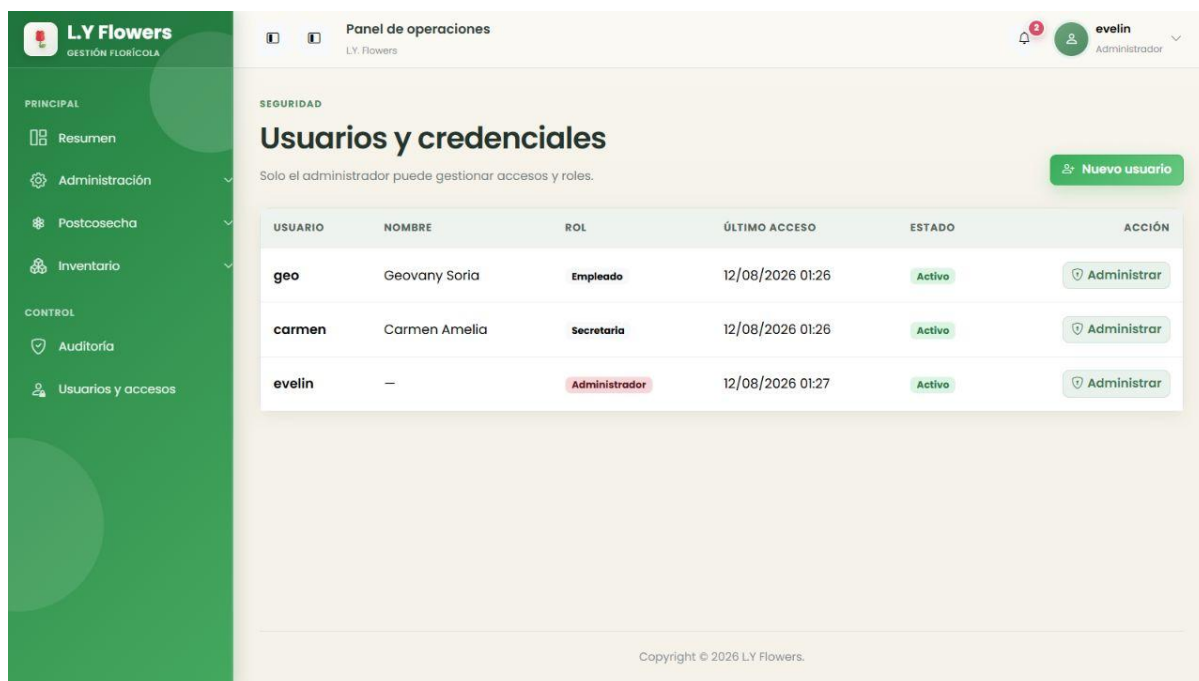
1. Revise los indicadores superiores para conocer proveedores, variedades, fincas y personal activos.
2. Consulte el calendario para identificar vacaciones registradas.
3. Use los accesos rápidos, como Nueva recepción o Ver liquidaciones, cuando estén disponibles.

ADMINISTRACIÓN DE USUARIOS Y PERSONAL

GESTIONAR USUARIOS

Esta función corresponde al Administrador y permite crear credenciales, asignar roles y administrar el estado de las cuentas.

1. Abra Administración y seleccione Usuarios y accesos.
2. Presione Nuevo usuario para registrar una cuenta.
3. Complete el usuario, la contraseña inicial, el rol y los datos solicitados.
4. Guarde la cuenta y verifique que aparezca activa en el listado.



USUARIO	NOMBRE	ROL	ÚLTIMO ACCESO	ESTADO	ACCIÓN
geo	Geovany Soria	Empleado	12/08/2026 01:26	Activo	Administrar
carmen	Carmen Amelia	Secretaria	12/08/2026 01:26	Activo	Administrar
evelin	—	Administrador	12/08/2026 01:27	Activo	Administrar

Figura 5. Listado de usuarios y credenciales.

GESTIONAR PERSONAL

1. Abra Administración y seleccione Personal.
2. Presione Nuevo personal y complete los datos obligatorios.
3. Guarde el registro y utilice Expediente para administrar permisos, vacaciones y documentos.

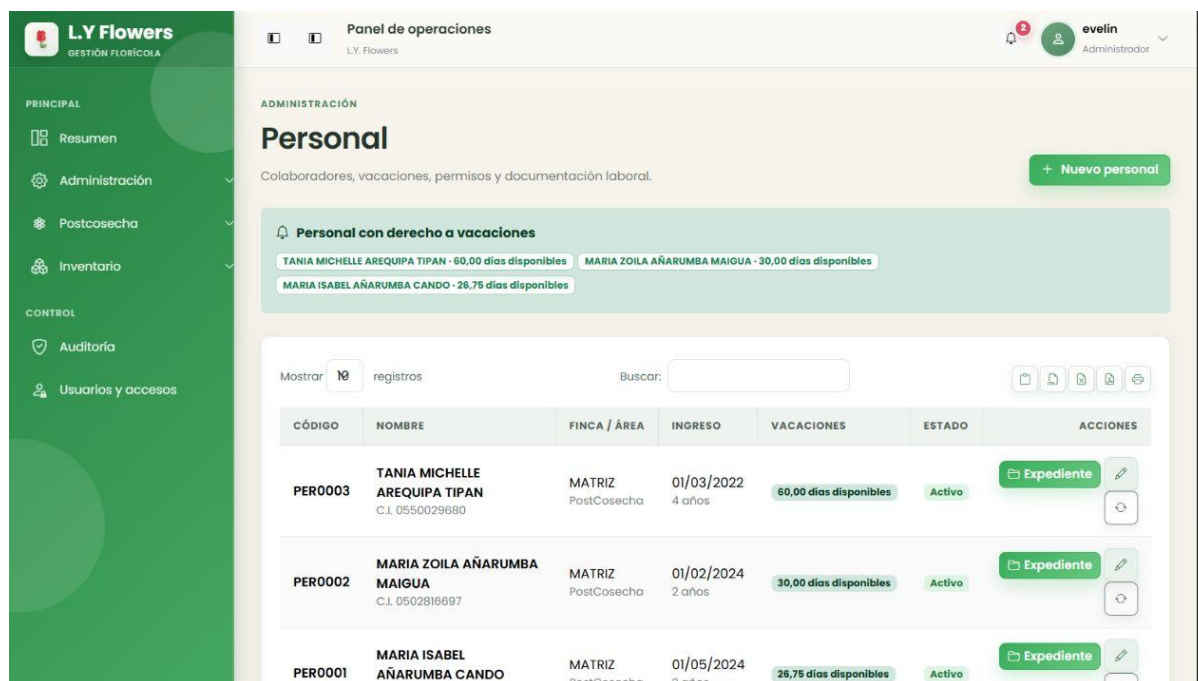


Figura 6. Gestión de personal y acceso a expedientes.

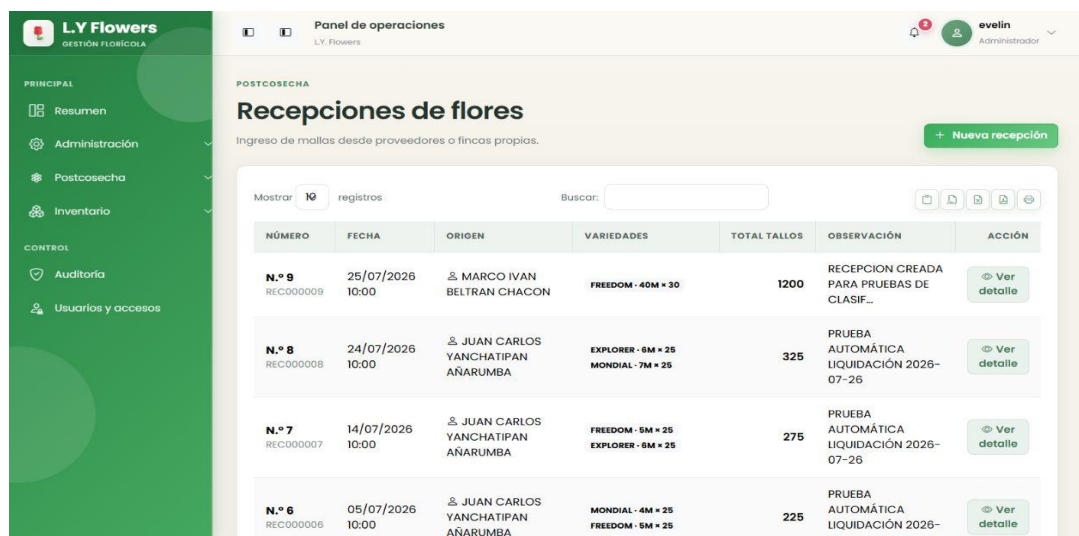
Nota: Antes de crear un registro nuevo, utilice el buscador para evitar duplicar personas, identificaciones o credenciales.

REGISTRO DE RECEPCIONES

La recepción registra el ingreso de flores a postcosecha y relaciona el proveedor, la finca, el empleado responsable, las variedades, las mallas y los tallos recibidos.

CONSULTAR RECEPCIONES

1. Abra Postcosecha y seleccione Recepciones.
2. Utilice el buscador para localizar una recepción por código, fecha, finca o proveedor.
3. Seleccione Ver detalle para revisar la información y el ticket relacionado.



Recepciones de flores
Ingreso de mallas desde proveedores o fincas propias.

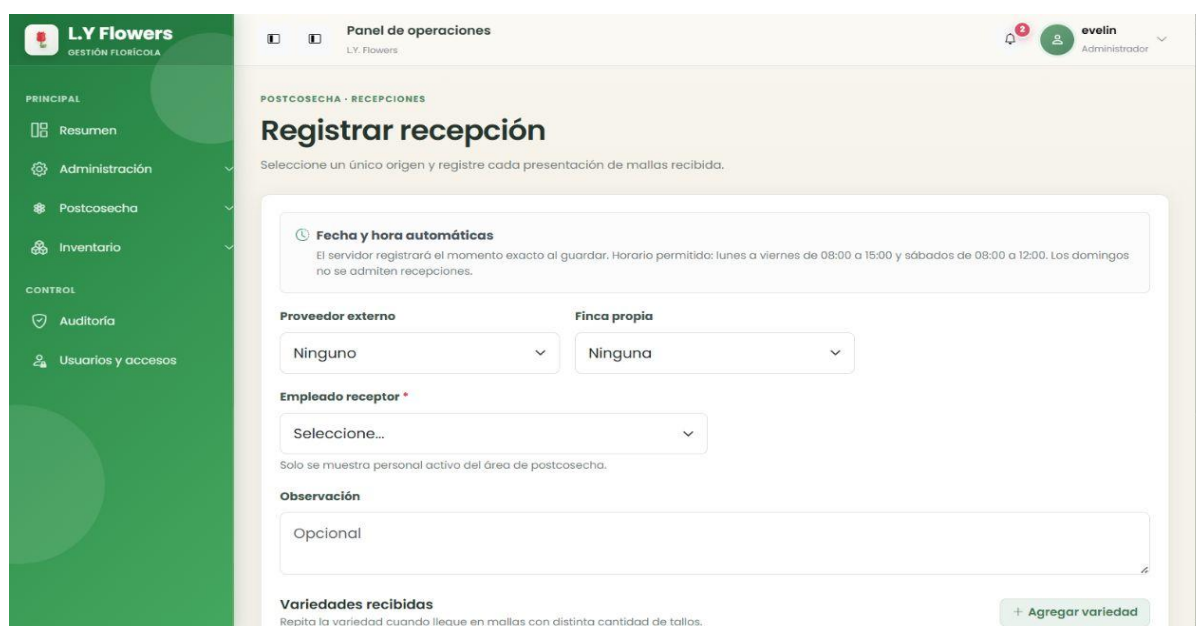
Mostrar 10 registros

NÚMERO	FECHA	ORIGEN	VARIETADES	TOTAL TALLOS	OBSERVACIÓN	ACCIÓN
N.º 9 REC000009	25/07/2026 10:00	MARCO IVAN BELTRAN CHACON	FREEDOM - 40M x 30	1200	RECEPCION CREADA PARA PRUEBAS DE CLASIF...	Ver detalle
N.º 8 REC000008	24/07/2026 10:00	JUAN CARLOS YANCHATIPAN AÑARUMBA	EXPLORER - 6M x 25 MONDIAL - 7M x 25	325	PRUEBA AUTOMÁTICA LIQUIDACIÓN 2026-07-26	Ver detalle
N.º 7 REC000007	14/07/2026 10:00	JUAN CARLOS YANCHATIPAN AÑARUMBA	FREEDOM - 5M x 25 EXPLORER - 6M x 25	275	PRUEBA AUTOMÁTICA LIQUIDACIÓN 2026-07-26	Ver detalle
N.º 6 REC000006	05/07/2026 10:00	JUAN CARLOS YANCHATIPAN AÑARUMBA	MONDIAL - 4M x 25 FREEDOM - 5M x 25	225	PRUEBA AUTOMÁTICA LIQUIDACIÓN 2026-07-26	Ver detalle

Figura 7. Listado de recepciones registradas.

REGISTRAR UNA RECEPCIÓN

1. Presione Nueva recepción.
2. Seleccione el proveedor externo o la finca propia.
3. Seleccione al empleado que recibe la flor.
4. Agregue una observación cuando sea necesaria.
5. Añada las variedades recibidas e indique mallas y tallos por malla.
6. Revise los totales y guarde la recepción.



Registrar recepción
Seleccione un único origen y registre cada presentación de mallas recibida.

Fecha y hora automáticas
El servidor registrará el momento exacto al guardar. Horario permitida: lunes a viernes de 08:00 a 15:00 y sábados de 08:00 a 12:00. Los domingos no se admiten recepciones.

Proveedor externo **Finca propia**

Ninguno Ninguna

Empleado receptor *

Seleccione...

Solo se muestra personal activo del área de postcosecha.

Observación

Opcional

Variedades recibidas
Repita la variedad cuando llegue en mallas con distinta cantidad de tallos.

+ Agregar variedad

Figura 8. Formulario para registrar una recepción.

Nota: La fecha y la hora se generan automáticamente. Verifique cuidadosamente el origen y las cantidades antes de guardar.

CLASIFICACIÓN DE FLORES

La clasificación registra el resultado real del procesamiento de una recepción y distribuye los tallos según las calidades definidas por L.Y Flowers.

CONSULTAR CLASIFICACIONES

1. Abra Postcosecha y seleccione Clasificaciones.
2. Localice el registro mediante el buscador.
3. Presione Reporte para consultar o imprimir el resultado.

CLASIFICACIÓN	FECHA	PROVEEDOR / LOTE	VARIEDADES RECIBIDAS	ACCIÓN
CLA000008	25/07/2026	JUAN CARLOS YANCHATIPAN AÑARUMBA Recepción N.º 8	EXPLORER - 6 mallas MONDIAL - 7 mallas	Reporte Ver
CLA000007	15/07/2026	JUAN CARLOS YANCHATIPAN AÑARUMBA Recepción N.º 7	FREEDOM - 5 mallas EXPLORER - 6 mallas	Reporte Ver
CLA000006	06/07/2026	JUAN CARLOS YANCHATIPAN AÑARUMBA Recepción N.º 6	MONDIAL - 4 mallas FREEDOM - 5 mallas	Reporte Ver
CLA000005	28/06/2026	JUAN CARLOS YANCHATIPAN AÑARUMBA Recepción N.º 5	EXPLORER - 3 mallas MONDIAL - 4 mallas	Reporte Ver
CLA000004	25/07/2026	MARCO IVAN BELTRAN CHACON Recepción N.º 4	FREEDOM - 6 mallas EXPLORER - 7 mallas	Reporte Ver

Figura 9. Listado de clasificaciones por proveedor.

CLASIFICAR UN LOTE

1. Presione Nueva clasificación.
2. Seleccione la fecha y la recepción pendiente de clasificación.
3. Registre las cantidades correspondientes a calidad óptima, estándar, nacional y sobrante.
4. Compruebe los tallos contabilizados y los faltantes.
5. Presione Finalizar clasificación y ver reporte.

L.Y Flowers
GESTIÓN FLORICOLA

Panel de operaciones
L.Y. Flowers

PRINCIPAL

- Resumen
- Administración
- Postcosecha
- Inventario

CONTROL

- Auditoría
- Usuarios y accesos

POSTCOSECHA - CLASIFICACIÓN

Clasificar lote completo

Cada número de clasificación corresponde a una sola recepción y debe completar todo el lote del proveedor.

Fecha *
12/08/2026

Observación general

Puede seleccionar hoy o uno de los dos días siguientes.

Recepción y proveedor *
Seleccione...

Solo aparecen recepciones pendientes dentro de su plazo de tres días.

Clasificación del lote
Registre los tallos realmente contados. Si llegaron menos, el sistema calculará los faltantes automáticamente.

Cancelar Finalizar clasificación y ver reporte

Figura 10. Formulario para clasificar un lote completo.

Nota: El sistema no permite que la suma clasificada sea superior a los tallos recibidos. Corrija las cantidades si aparece una validación.

LIQUIDACIONES

El módulo calcula y registra los valores correspondientes a los productores mediante las clasificaciones y tarifas vigentes.

1. Abra Liquidaciones desde el menú lateral.
2. Revise el estado de cada registro y utilice el buscador cuando sea necesario.
3. Presione Generar liquidación para seleccionar productor y periodo.
4. Compruebe las clasificaciones consideradas, tarifas, subtotales y total.
5. Guarde la liquidación y adjunte los documentos requeridos.
6. Cuando corresponda, registre el pago según los permisos asignados.

LY Flowers
GESTIÓN FLORÍCOLA

Panel de operaciones
LY Flowers

FINANZAS
Liquidaciones
Reportes mensuales calculados automáticamente sobre tallos exportables.

Mostrar 10 registros Buscar:

CÓDIGO	PROVEEDOR	ENTREGAS INCLUIDAS	REPORTE GENERADO	TOTAL	ESTADO	ACCIONES
LIQ000002	JUAN CARLOS YANCHATIPAN AÑARUMBA	27/06/2026 – 24/07/2026	26/07/2026	\$189,96	Reporte listo - esperando documento	
LIQ000001	MARCO IVAN BELTRAN CHACON	27/06/2026 – 24/07/2026	26/07/2026	\$187,63	Reporte listo - esperando documento	

Mostrando 1 a 2 de 2 registros [Primero](#) [Anterior](#) [1](#) [Siguiente](#) [Último](#)

Copyright © 2026 LY Flowers.

Figura 11. Listado y estado de las liquidaciones.

Nota: Revise el periodo y las tarifas antes de confirmar. Una liquidación depende de clasificaciones registradas y válidas.

INVENTARIO

El inventario permite administrar proveedores de insumos, categorías, unidades de medida, productos, entradas, salidas e historial de movimientos.

CREAR Y CONSULTAR PRODUCTOS

1. Abra Inventario y seleccione Productos.
2. Presione Nuevo producto.
3. Complete nombre, categoría, unidad, existencia inicial y stock mínimo.
4. Guarde el registro y confirme que aparezca en el listado.

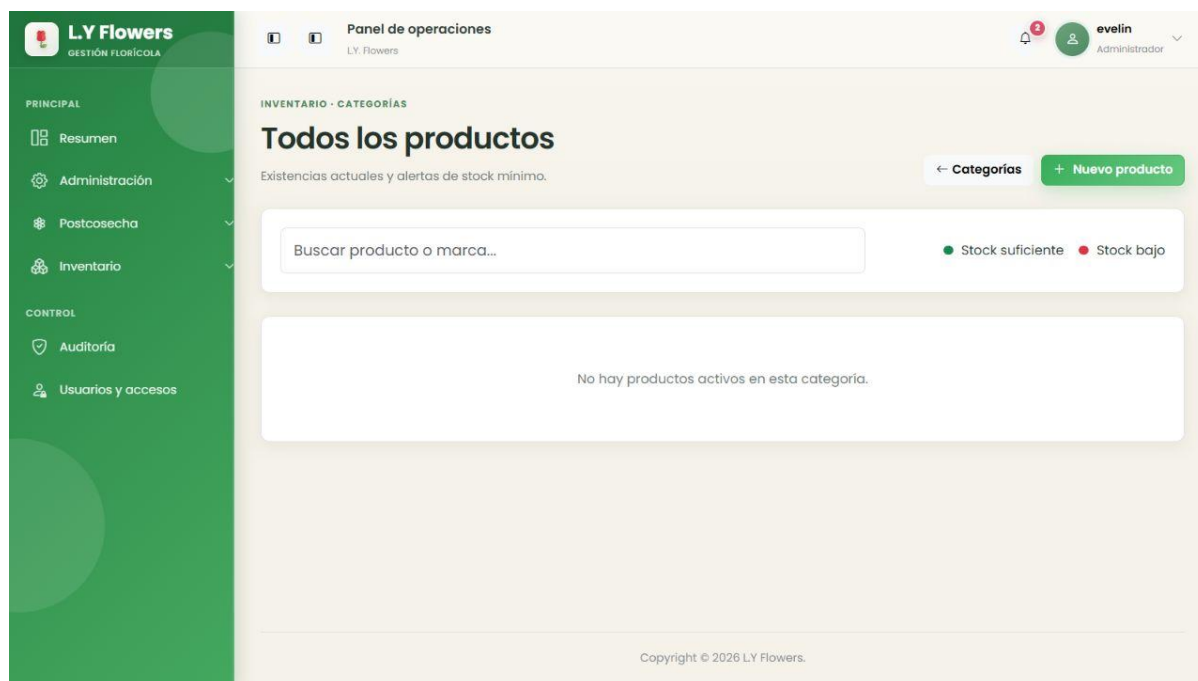


Figura 12. Catálogo de productos del inventario.

REGISTRAR MOVIMIENTOS

- Entrada: incrementa las existencias a partir de una compra o ingreso.
- Salida: descuenta existencias y registra la finca o destino correspondiente.
- Historial: permite consultar fecha, usuario, producto, cantidad y tipo de movimiento.

Nota: El sistema impide saldos negativos y muestra alertas cuando un producto llega al nivel mínimo configurado.

REPORTES

FlorLY ofrece reportes de clasificación y producción para consultar resultados, comprobar la trazabilidad y apoyar la toma de decisiones.

1. Abra el reporte requerido desde Postcosecha o desde el Panel y reportes.
2. Seleccione la fecha, periodo, proveedor u otro filtro disponible.
3. Presione Generar reporte.
4. Revise los resultados y utilice Imprimir o la exportación autorizada.



Figura 13. Reporte diario general de clasificación.

Nota: La ausencia de resultados puede indicar que no existen clasificaciones para la fecha seleccionada; no necesariamente representa un error del sistema.

AUDITORÍA

La bitácora conserva la trazabilidad de los accesos y operaciones realizadas. Su consulta está reservada al Administrador.

1. Abra Auditoría desde el menú lateral.
2. Filtre los eventos mediante fecha, usuario, acción o módulo.
3. Seleccione Ver detalle para consultar la información disponible del evento.
4. Utilice el registro para verificar quién realizó la operación y cuándo ocurrió.

Panel de operaciones
L.Y. Flowers

Bitácora del sistema
Trazabilidad de operaciones y accesos rechazados.

Mostrar 10 registros. Buscar:

FECHA Y HORA	USUARIO	ACCIÓN	RESULTADO	MÓDULO	REGISTRO	IP	DETALLE
11/08/2026 00:52:19	evelin	INICIO_SESION	Registrada	seguridad	1	127.0.0.1	
10/08/2026 14:07:53	evelin	INICIO_SESION	Registrada	seguridad	1	127.0.0.1	
10/08/2026 13:30:43	evelin	INICIO_SESION	Registrada	seguridad	1	127.0.0.1	
07/08/2026 14:43:57	evelin	INICIO_SESION	Registrada	seguridad	1	127.0.0.1	
07/08/2026 14:41:58	evelin	INICIO_SESION	Registrada	seguridad	1	127.0.0.1	
07/08/2026 04:08:10	evelin	CARGAR DOCUMENTO DE PERSONAL	Procesada	personal	PER0001	127.0.0.1	

Figura 14. Bitácora de operaciones y accesos del sistema.

Nota: Las contraseñas y archivos sensibles se protegen y no deben mostrarse en texto plano dentro de la bitácora.

RECOMENDACIONES DE USO Y SOLUCIÓN DE PROBLEMAS

BUENAS PRÁCTICAS

- Verifique la información antes de guardarla, especialmente cantidades, fechas, productores y tarifas.
- Use los buscadores para evitar registros duplicados.
- No comparta credenciales ni deje una sesión abierta sin supervisión.
- Mantenga actualizados el navegador y los manuales de operación.
- Realice copias de seguridad periódicas y compruebe su recuperación.

PROBLEMAS FRECUENTES

Situación	Causa posible	Acción recomendada
No puede iniciar sesión	Credenciales incorrectas, cuenta inactiva o bloqueo temporal.	Revise usuario y contraseña; espere el tiempo de bloqueo o solicite ayuda al Administrador.
No aparece un módulo	El rol no posee el permiso correspondiente.	Confirme sus responsabilidades y solicite al Administrador revisar el rol.
No se guarda un formulario	Faltan campos obligatorios o existe una validación.	Revise los mensajes, corrija los campos y vuelva a guardar.
No aparecen datos en un reporte	No existen registros para los filtros seleccionados.	Cambie la fecha o periodo y confirme que existan recepciones y clasificaciones.
No se puede registrar una salida	La existencia disponible es insuficiente.	Revise el historial y registre primero la entrada correspondiente.

SOPORTE

Cuando el problema continúe, informe al responsable técnico el usuario afectado, módulo, fecha y hora aproximada, acción realizada y mensaje mostrado. No envíe contraseñas por correo o mensajería.