UNIVERSIDAD TÉCNICA DE COTOPAXI

FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

CARRERA DE SISTEMAS DE INFORMACIÓN

PROYECTO DE PROPUESTA TECNOLÓGICA

DESARROLLO DE UNA PLATAFORMA INFORMÁTICA PARA EL MONITOREO Y CONTROL LABORAL EN LOS PROCESOS DE CLASIFICACIÓN Y EMBONCHE DE ROSAS EN LA COMERCIALIZADORA COMEXIGER, EN EL BARRIO JOSEGUANGO BAJO, PARROQUIA DE MULALÓ

Propuesta tecnológica presentado previo a la obtención del título de ingenieros en Sistemas de Información

AUTORES:

Karen Anahí Barcia Chimba

Anthony Steeven Monta Chiliquinga

TUTOR:

ING. Diego Geovanny Falconí Punguil

LATACUNGA, MARZO,2026

DECLARACIÓN DE AUTORÍA

Latacunga, Marzo del 2026

DECLARACIÓN DE AUTORÍA

Nosotros BARCIA CHIMBA KAREN ANAHÍ, con cédula de ciudadanía No. 0504808759, Monta Chiliquinga Anthony Steeven, con cédula de ciudadanía No.050470221, declaramos ser autores del proyecto de titulación "**DESARROLLO DE UNA PLATAFORMA INFORMÁTICA PARA EL MONITOREO Y CONTROL LABORAL EN LOS PROCESOS DE CLASIFICACIÓN Y EMBONCHE DE ROSAS EN LA COMERCIALIZADORA COMEXIGER, EN EL BARRIO JOSEGUANGO BAJO, PARROQUIA DE MULALÓ**", siendo el Ing. Diego Geovanny Falconi Punguil 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.

Karen Anahí Barcia Chimba
CC:0504808759

Anthony Steeven Monta Chiliquinga
CC: 050470221

AVAL DEL TUTOR DE PROPUESTA TECNOLÓGICA

UNIVERSIDAD TÉCNICA DE COTOPAXI – CARRERA DE SISTEMAS DE INFORMACIÓN

AVAL DEL TUTOR DE LA PROPUESTA TECNOLÓGICA

En calidad de Tutor de la Propuesta Tecnológica sobre el título: **" DESARROLLO DE UNA PLATAFORMA INFORMÁTICA PARA EL MONITOREO Y CONTROL LABORAL EN LOS PROCESOS DE CLASIFICACIÓN Y EMBONCHE DE ROSAS EN LA COMERCIALIZADORA COMEXIGER, EN EL BARRIO JOSEGUANGO BAJO, PARROQUIA DE MULALÓ"** propuesto por Barcia Chimba Karen Anahí y Monta Chiliquinga Anthony Steeven, 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.

Latacunga, Marzo del 2026

Ing. Diego Geovanny Falconi Punguil, Mg.
C.C. 0550080774
TUTOR

Latacunga, Marzo 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 "**DESARROLLO DE UNA PLATAFORMA INFORMÁTICA PARA EL MONITOREO Y CONTROL LABORAL EN LOS PROCESOS DE CLASIFICACIÓN Y EMBONCHE DE ROSAS EN LA COMERCIALIZADORA COMEXIGER, EN EL BARRIO JOSEGUANGO BAJO, PARROQUIA DE MULALÓ**", propuesto por los estudiantes Barcia Chimba Karen Anahí y Monta Chiliquinga Anthony Steeven 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 presencial en virtud de lo cual los postulantes pueden presentarse a la Defensa de su Proyecto de Titulación.

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

Atentamente,

Lector 1 (Presidente)	Lector 2	Lector 3
Ing. Luis René Quisaguano Collaguazo	Ing. Miryam Dorila Iza Carate	Ing.Jorge Bladimir Rubio Peñaherrera
CC: 1721895181	CC: 0501957617	CC:0502222291

AVAL DE IMPLEMENTACIÓN

AVAL DE IMPLEMENTACIÓN

En calidad de dueños de la comercializadora "Comexiger" con tema de propuesta tecnológica " **DESARROLLO DE UNA PLATAFORMA INFORMÁTICA PARA EL MONITOREO Y CONTROL LABORAL EN LOS PROCESOS DE CLASIFICACIÓN Y EMBONCHE DE ROSAS EN LA COMERCIALIZADORA COMEXIGER, EN EL BARRIO JOSEGUANGO BAJO, PARROQUIA DE MULALÓ**" de los señores estudiantes Karen Anahí Barcia Chimba con C.I No: 0504808759 y Anthony Steeven Monta Chiliquinga con C.I No: 0504702291, pertenecientes a la carrera de SISTEMAS DE INFORMACIÓN.

CERTIFICO QUE:

Una vez revisada la implementación, considero que dicho trabajo investigativo cumple con los requerimientos metodológicos y aportes científicos-tecnológicos necesarios para ser sometidos a la Evaluación del Tribunal de Validación De Proyectos de Investigación, que el Consejo Directivo de la Facultad de Ingeniería y Aplicadas de la Universidad Técnica De Cotopaxi designe para su respectiva calificación.

Evelyn Franco
CC: 1204755993

Rafael Vergara
CC: 0501611891

AVAL DE TRADUCCIÓN

UNIVERSIDAD TÉCNICA DE COTOPAXI

CENTRO DE IDIOMAS

AVAL DE TRADUCCIÓN

En calidad de Docente del Idioma Inglés del Centro de Idiomas de la Universidad Técnica de Cotopaxi; en forma legal **CERTIFICO** que:

La traducción del resumen al idioma Inglés del proyecto de investigación cuyo título versa: **"DESARROLLO DE UNA PLATAFORMA INFORMÁTICA PARA EL MONITOREO Y CONTROL LABORAL EN LOS PROCESOS DE CLASIFICACIÓN Y EMBONCHE DE ROSAS EN LA COMERCIALIZADORA COMEXIGER, EN EL BARRIO JOSEGUANGO BAJO, PARROQUIA DE MULALÓ"** presentado por: **Karen Anahí Barcia Chimba y Anthony Steeven Monta Chiliquinga**, egresados de la Carrera de Sistemas de Información, perteneciente a la **Facultad de Ciencias de la Ingeniería y Aplicadas,** lo realizamos bajo mi supervisión y cumple con una correcta estructura gramatical del Idioma.

Es todo cuanto puedo certificar en honor a la verdad y autorizo al peticionario hacer uso del presente aval para los fines académicos legales.

Latacunga, Marzo del 2026

Atentamente,

Mr. Santiago Gabriel Ramón Amores
DOCENTE CENTRO DE IDIOMAS-UTC
CI: 0503568826

AGRADECIMIENTO

Ante todo, agradezco a Dios, quien ha sido el mayor promotor de este camino. Sin su guía y fortaleza, nada de esto hubiera sido posible. Fueron muchos los momentos en que me sentí incapaz de continuar, pero siempre encontré en Su palabra el sustento necesario para seguir adelante. Este logro es, en primer lugar, para Él.

A mis padres, les agradezco infinitamente por su amor, dedicación y sacrificio. A mi madre, Nancy Chimba, por no dejarme rendir nunca; sus palabras y abrazos llegaron siempre en el momento exacto para darme la fuerza que necesitaba. Gracias por trabajar tanto y con tanto amor para hacer posible este triunfo. A mi padre, Edgar Barcia, por su paciencia y compañía incondicional, por esperarme tantas noches sin importar el frío. Su cariño y apoyo han sido pilares en este camino. Este logro es de ustedes y de toda nuestra familia.

Al Ingeniero Diego Falconí, mi tutor, le agradezco por su guía, su disposición y la generosidad con la que compartió sus conocimientos. Sus enseñanzas fueron clave para culminar este proyecto.

Agradezco a Monta Anthony por su apoyo y la paciencia que ha tenido, gracias por ayudarme durante todo el transcurso de la carrera ha sido una parte importante con momentos increíbles que gracias a Dios pude tener.

Finalmente, agradezco a los amigos que la vida me dio durante este recorrido, personas que llegaron en el momento indicado y que, con su compañía y apoyo, hicieron de esta etapa una experiencia inolvidable.

Karen Barcia

DEDICATORIA

El presente proyecto de propuesta tecnológica se lo dedico a mis padres de los cuales me siento orgullosa de las personas que son y de la persona que forjaron en mí. A mis hermanos Anthony, Verónica y Lucero, por su apoyo incondicional, sus consejos oportunos y por estar presentes en cada momento importante de este camino. Su compañía ha sido un refugio y una motivación constante.

Con especial emoción, dedico este logro a la memoria de mi tío Fernando Chimba, quien siempre creyó en mí y esperó con ilusión este momento. Aunque hoy no pueda estar físicamente presente para celebrarlo, sé que desde donde se encuentra comparte esta alegría. Este título es también tuyo, tío.

Karen Barcia

AGRADECIMIENTO

Quiero expresar mi más profundo agradecimiento a mis padres, quienes con su amor, esfuerzo y apoyo incondicional han sido el pilar fundamental en cada etapa de mi vida. Gracias por creer en mí, por motivarme a seguir adelante incluso en los momentos difíciles y por enseñarme el valor del esfuerzo, la responsabilidad y la perseverancia.

A mis hermanos, por su compañía, por las palabras de ánimo y por estar presentes a lo largo de este camino. Su apoyo y cariño han sido muy importantes durante todo este proceso.

De manera especial, agradezco a mi tutor, Diego Falconí, por su guía, paciencia y apoyo durante el desarrollo de este trabajo. Gracias a sus conocimientos, orientación y acompañamiento, hemos podido cumplir con éxito la realización de esta investigación.

También deseo expresar un agradecimiento muy especial a mi pareja, Karen Barcia, con quien he compartido este camino. A pesar de las dificultades, los desacuerdos y los retos que se presentaron, hemos permanecido firmes en nuestros sueños. Gracias por caminar a mi lado, por el apoyo, la paciencia y por seguir adelante juntos hasta alcanzar esta meta.

También deseo dedicar un recuerdo lleno de cariño a mi querido gato Mordelón, quien aunque ya no está físicamente conmigo, fue un compañero que me brindó momentos de alegría y tranquilidad. Su recuerdo permanece siempre en mi corazón.

Este logro está lleno de amor, gratitud y dedicación hacia todas las personas que, de una u otra manera, han formado parte de este camino y han contribuido a que hoy pueda alcanzar esta meta.

"¿Acaso existe algo más valioso que tus sueños?"

— Monkey D. Luffy

Anthony Monta

DEDICATORIA

Dedico este trabajo, con todo mi amor y gratitud, a mis padres Carlos Monta y Barbarita Chiliquinga, quienes han sido el pilar fundamental de mi vida. Gracias por su esfuerzo, sacrificio y apoyo incondicional en cada etapa de mi camino. Este logro también es suyo, porque sin su amor, consejos y confianza en mí, nada de esto habría sido posible.

A mis queridos hermanos Jhonny, Carlos, Sandy, Jacky, Vanessa, Lis y Majo, por su compañía, su apoyo y por estar presentes a lo largo de este camino. Cada palabra de ánimo y cada momento compartido han sido una motivación para seguir adelante.

Y con un cariño muy especial, a mi querido gato Mordelón, quien aunque ya no está conmigo, siempre ocupará un lugar en mi corazón por los momentos de alegría y compañía que me brindó.

A todos ustedes, con amor, respeto y profunda gratitud, dedico este logro que representa el esfuerzo y los sueños cumplidos en este camino.

Anthony Monta

UNIVERSIDAD TÉCNICA DE COTOPAXI

FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

TÍTULO: "DESARROLLO DE UNA PLATAFORMA INFORMÁTICA PARA EL MONITOREO Y CONTROL LABORAL EN LOS PROCESOS DE CLASIFICACIÓN Y EMBONCHE DE ROSAS EN LA COMERCIALIZADORA COMEXIGER, EN EL BARRIO JOSEGUANGO BAJO, PARROQUIA DE MULALÓ"

Autores:

Karen Anahí Barcia Chimba

Anthony Steeven Monta Chiliquinga

RESUMEN

Este trabajo consistió en desarrollar una plataforma informática para optimizar el monitoreo y control del rendimiento laboral en los procesos de clasificación y embonche de rosas en la comercializadora Comexiger, ubicada en la parroquia Mulaló. La empresa realizaba estos registros de manera totalmente manual, utilizando pinzas de colores para identificar las mesas de trabajo y anotaciones en pizarras que luego se transcribían a hojas de Excel. Este método generaba errores frecuentes, pérdida de información y retrasos en el cálculo de la remuneración de los trabajadores. Para abordar esta situación, se construyó una solución compuesta por dos aplicaciones: un backend con API REST desarrollado en Python con Django, y dos frontends. El primero es una aplicación web para administradores que permite gestionar usuarios, mesas y visualizar reportes de rendimiento y disponibilidad. El segundo es una aplicación móvil creada con Flutter para los trabajadores, que les permite iniciar/finalizar jornada, imprimir etiquetas con códigos QR y consultar su rendimiento en tiempo real. El desarrollo se guió por la metodología ágil XP, lo que permitió retroalimentación constante con los usuarios. Para la recolección de requisitos se emplearon entrevistas con los administradores y encuestas al personal operativo, abarcando a toda la población involucrada (12 personas). Como resultado, se obtuvo un sistema integral y funcional. Los trabajadores ahora generan e imprimen etiquetas QR desde la app móvil. Al llegar al área de control, el personal escanea estas etiquetas con dos escáneres diferenciados: el primero suma el bonche al rendimiento del trabajador y a la disponibilidad de stock; la segunda resta el bonche de la disponibilidad al ser enviado. Toda la información se sincroniza en tiempo real mediante WebSockets, permitiendo a los administradores visualizar dashboards actualizados al instante desde la aplicación web y generar informes, mientras que los trabajadores pueden ver su propio progreso desde sus dispositivos móviles. En conclusión, la plataforma informática cumplió con el objetivo de automatizar y optimizar los procesos de control laboral. Se eliminaron los errores y demoras del registro manual, se agilizó el cálculo de remuneraciones, se proporcionó visibilidad en tiempo real de la productividad y el stock, mejorando la transparencia y eficiencia operativa de la comercializadora Comexiger.

Palabras clave: plataforma informática, monitoreo laboral, rendimiento, floricultura, Django, Flutter, código QR, metodología XP, post-cosecha.

TECHNICAL UNIVERSITY OF COTOPAXI

ENGINEERING SCIENCES AND APPLIED FACULTY

THEME: "DEVELOPMENT OF A SOFTWARE PLATFORM FOR MONITORING AND LABOR CONTROL IN THE ROSE SORTING AND BUNDLING PROCESSES AT THE COMEXIGER TRADING COMPANY, LOCATED IN JOSEGUANGO BAJO NEIGHBORHOOD, MULALÓ PARISH"

Authors:

Karen Anahí Barcia Chimba

Anthony Steeven Monta Chiliquinga

ABSTRACT

This work consisted of developing a software platform to optimize the monitoring and control of labor performance in the processes of sorting and bundling roses at the Comexiger trading company, located in Mulaló parish. The company previously carried out these records completely manually, using colored clips to identify work tables and notes on whiteboards that were later transcribed into Excel sheets. This method frequently caused errors, loss of information, and delays in calculating workers' compensation. To address this situation, a solution composed of two applications was built: a backend with a REST API developed in Python with Django, and two frontends. The first is a web application for administrators that allows user management, table management, and visualization of performance and availability reports. The second is a mobile application created with Flutter for workers, enabling them to start/end shifts, print labels with QR codes, and check their performance in real time. The development followed the agile XP methodology, which allowed for constant feedback from users. For requirements gathering, interviews were conducted with administrators and surveys with operational personnel, covering the entire population involved (12 people). As a result, a comprehensive and functional system was obtained. Workers now generate and print QR labels from the mobile app. Upon arriving at the control area, staff scan these labels with two differentiated scanners: the first adds the bundle to the worker's performance and to stock availability; the second subtracts the bundle from availability when it is shipped. All information is synchronized in real time via WebSockets, allowing administrators to instantly view updated dashboards from the web application and generate reports, while workers can see their own progress from their mobile devices. In conclusion, the software platform achieved the goal of automating and optimizing labor control processes. Manual recording errors and delays were eliminated, payroll calculation was streamlined, real-time visibility of productivity and stock was provided, and transparency and operational efficiency at Comexiger were improved.

Keywords: IT platform, labor monitoring, performance, floriculture, Django, Flutter, QR code, XP methodology, post-harvest.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1 INFORMACIÓN GENERAL

Tema del proyecto: Desarrollo de una plataforma informática para el monitoreo y control del rendimiento laboral en los procesos de clasificación y embonche de rosas en la comercializadora Comexiger del barrio Joseguango Bajo Parroquia de Mulaló.

Modalidad de Titulación:

MODALIDAD DE TITULACIÓN	HOMOLOGACIONES PARA INFORME FINAL DE TITULACIÓN	SELECCIÓN
Propuesta tecnológica	Informe de propuesta tecnológica	**X**
	Patente, Modelo de utilidad, Certificado de propiedad intelectual.	
	Artículo científico	
Proyecto de investigación	Informe de Proyecto de investigación	
	Artículo científico	
	Patente, Modelo de utilidad, Certificado de propiedad intelectual.	
Examen de indicadores de RDA		

Trabajo de Titulación Vinculado al Proyecto:

No vinculado

Equipo de Trabajo del Trabajo de Titulación:

Karen Anahí Barcia Chimba

Anthony Steeven Monta Chiliquinga

Tutor de Titulación:

ING. Diego Geovanny Falconi Punguil

Área de Conocimiento: 330000 CIENCIAS TECNOLÓGICAS

06 Información y Comunicación (TIC)	**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: Desarrollo y optimización de sistemas productivos agroindustriales.

Sublíneas de investigación de la Carrera: Aplicaciones tecnológicas para la gestión de procesos agroindustriales

2 INTRODUCCIÓN

El sector florícola en Ecuador inicio su desarrollo en el año de 1982 y ha ido creciendo sustancialmente alrededor de los años, es el cultivo que mayor a destacado en mano de obra en el área de las rosas este cuenta con muchas variedades en un gran número de hectáreas distribuidas en el país. Existen muchas personas que generan una ganancia estableciendo dichos cultivos, siendo ellos los proveedores tanto para grandes empresas exportadoras como a medianas y pequeñas comercializadoras que las procesan[1],[2].

Ecuador es conocido como el tercer país con más exportación de flores en el mundo, por su excelente calidad, su belleza y la intensidad de sus colores, características que se deben en gran parte a la zona de cultivo y clima donde se cultivan ; su actividad ha ido creciendo y sus métodos para seleccionar lo exportable se refinan día con día[3].Esto fortalece la reputación del país en los mercados internacionales. La producción florícola también ha impulsado y generado empleo para cientos de personas las cuales participan en todo el proceso productivo, desde la cultivación hasta la comercialización.

Las empresas al necesitar un producto de calidad las someten a un procesamiento riguroso con empleados con experiencia en el área para poder generar y exportar productos de calidad. Los procesos de clasificación y embonche son etapas críticas que impactan directamente en el producto final[4]. La clasificación es en la cual separan las que sirven para exportación y las que se dejan para importación (nacional); lo que usan para clasificarlo es si poseen

enfermedades o maltrato, después pasan por la embonchadora quien agrupa todos los tallos ya clasificados y los junta en una lámina de cartón que los envuelve dependiendo el número de tallos establecido para el pedido en el día, por consiguiente, se corta a la medida solicitada. Cada bonche (lamina con los tallos agrupados) le pertenece a una mesa enumerada y con esta información se proporciona el rendimiento de los trabajadores en el día.

El rendimiento concierne en la productividad y competitividad del personal y cada área de trabajo por ello cada empresa realiza procesos de selección y buscan resultados. Es por ello que es importante porque influye directamente en su remuneración porque según la cantidad de boches procesados de manera correcta durante su jornada laboral se realizará el pago ya sea este al final de la semana o del mes dependiendo el acuerdo que exista con la empresa en la que trabajan.

En la comercializadora Comexiger, ubicada en el barrio Joseguango Bajo, los procesos de clasificación y embonche de rosas representan una parte esencial de su actividad productiva. Sin embargo, el registro del rendimiento operativo de los trabajadores todavía se realiza de manera manual, lo que implica que los trabajadores deben contar las pinzas que son los distintivos de cada mesa colocados en los bonches los cuales los ubican en unas varillas y anotan de manera directa la cantidad de bonches procesados en una pizarra. Posteriormente, los dueños ingresan estos datos en hojas de Excel que realizan el cálculo respectivo. Esta práctica, además de ser tediosa y repetitiva, suele generar errores, pérdida de información y retrasos que afectan directamente la toma de decisiones y el control del trabajo realizado por cada trabajador de la empresa.

Razón por la cual se plantea el siguiente problema de investigación:

¿Cómo mejorar el monitoreo y control del rendimiento laboral en los procesos de clasificación y embonche de rosas en la comercializadora Comexiger?

Para dar respuesta a esta problemática surge la propuesta de desarrollar una plataforma informática que facilite el monitoreo y control del desempeño operativo. La idea es combinar una aplicación de web, destinada principalmente para los administradores (dueños) y les ayudará a gestionar y controlar en tiempo real la información de los trabajadores registrados, les permitirá tener al día la disponibilidad y también el rendimiento de los trabajadores sin tener que contarlos uno a uno. Además, permitirá identificar con rapidez posibles errores en los registros.

Como parte del proyecto, se desarrolla una aplicación móvil destinada a los trabajadores. Esta app les permitirá estar mejor informados sobre sus tareas diarias y también visualizar en tiempo real el registro del procesamiento de bonches en cada mesa de trabajo. Habrá un botón para iniciar y finalizar su jornada laboral, lo que ayudará a evitar confusiones con los horarios de entrada y salida, un aspecto clave para calcular su remuneración. Además, la aplicación podrá generar etiquetas que se colocarán en los bonches; al escanearlas, se contabilizará el rendimiento de cada mesa de trabajo.

Con esta herramienta, no solo se busca mejorar la calidad de los procesos y optimizar el tiempo, sino también brindar apoyo a todos los que forman parte de Comexiger con una solución más moderna, rápida y eficiente. Gracias a esta herramienta, la comercializadora podrá acceder a información actualizada al instante, lo que permitirá tomar decisiones rápidas y detectar irregularidades en el rendimiento de los trabajadores. Además, se evitará el gasto en etiquetas y en papel para el control y registro de los inicios de jornada. También se reducirá el tiempo necesario para contar los bonches por mesa, esto ayuda cuando son escaneados, se eliminará la necesidad de colocar y devolver las pinzas a las mesas correspondientes, ahorrando tiempo y recursos. Los principales beneficiarios de esta mejora serán sus dueños, Eveleni Franco y Rafael Vergara, quienes podrán tomar decisiones más informadas sobre el rendimiento laboral, evitando así pérdidas para la empresa. De manera indirecta, el personal operativo también se verá beneficiado, al contar con un sistema confiable para conocer su rendimiento diario.

Para lograr esto, se trabajará con la metodología XP, con ella nos permitirá la comunicación entre los desarrolladores y beneficiarios; posteriormente ayudará a obtener una idea clara y precisa para resolver la problemática que ayudará a comprender las necesidades reales de los usuarios.

2.1 OBJETIVOS

2.1.1 Objetivo general:

Implementar una plataforma informática para la gestión de rendimiento en los procesos de clasificación y embonche de rosas, a través de la metodología XP la cual permitirá optimizar la precisión de los registros dentro de la empresa Comexiger.

2.1.2 Objetivos específicos:

- Establecer las bases teóricas del monitoreo y control del rendimiento laboral en procesos agroindustriales, mediante revisión documental, para obtener bases teóricas.

- Aplicar la metodología XP en el desarrollo del proyecto para definir un proceso ágil y centrado en las necesidades de Comexiger.

- Desarrollar una plataforma informática que automatice el registro, monitoreo y análisis del rendimiento en la clasificación y embonche de rosas para mejorar la eficiencia en Comexiger.

2.2 TAREAS POR OBJETIVO

En la tabla 2.1 se muestran las actividades realizadas por cada objetivo lo cual evidencia la coherencia entre lo planificado y lo ejecutado dentro del proyecto. organizar las actividades en función de cada objetivo demuestra una estructura metodológica clara, esto nos ayuda a facilitar el control del avance y respalda que cada acción que se desarrolló responde directamente a los resultados esperados y técnicas definidas.

Tabla 2.1 Sistema de Tareas por objetivo.

Objetivos específicos	Actividades (tareas)	Resultados esperados	Técnicas, Medios e Instrumentos
Objetivo específico 1			
Establecer las bases teóricas del monitoreo y control del rendimiento laboral en procesos agroindustriales, mediante revisión documental.	Realizar una revisión bibliográfica sobre monitoreo y control de rendimiento.	Marco teórico actualizado sobre procesos agroindustriales.	Bases de datos académicas (Scopus, Redalyc, Google Schoolar).
	Sistematizar los principales conceptos, indicadores y herramientas.	Documento estructurado con fundamentos conceptuales clave.	Word, Zotero.

Objetivos específicos	Actividades (tareas)	Resultados esperados	Técnicas, Medios e Instrumentos
Objetivo específico 2			
Aplicar la metodología XP en el desarrollo del proyecto para definir un proceso ágil y centrado en las necesidades de Comexiger.	Realizar entrevistas y dinámicas de empatía con los trabajadores clave de Comexiger.	Recolección de necesidades reales y expectativas funcionales.	Entrevistas
	Crear y validar prototipos de solución basados en los hallazgos obtenidos.	Bocetos o maquetas funcionales validadas por los usuarios.	Herramientas de prototipado (Figma, papel)
Objetivo específico 3			
Desarrollar una plataforma informática que automatice el registro, monitoreo y análisis del rendimiento en la clasificación y embonche de rosas para mejorar la eficiencia en Comexiger.	Desarrollar la arquitectura del sistema y sus componentes (móvil y web).	Estructura técnica del sistema definida y documentada.	Lenguajes de programación como: Python, con el Framework Django, Dart con el Framework Flutter y diagramas UML.
	Programar módulos para registro de datos, monitoreo y análisis de rendimiento.	Funcionalidades clave implementadas y en pruebas.	IDEs (Visual Studio Code, Visual Studio, Flutter).
	Realizar pruebas piloto en Comexiger y ajustar el sistema según resultados.	Plataforma funcional, ajustada a los procesos reales de clasificación y embonche.	Pruebas de usuario, observación participativa.

3 MARCO TEÓRICO

3.1 ANTECEDENTES

En la investigación sobre el diseño de una aplicación web referente a la evaluación del desempeño laboral, la cual permite evaluar a los trabajadores de manera más ágil y organizada mediante herramientas automatizadas. Esto no solo facilita el proceso, sino que también asegura mayor transparencia en los resultados, mejora la eficiencia en la gestión y contribuye a reducir el impacto ambiental al disminuir el uso de papel y recursos físicos. [5].

Dicha investigación empleó una metodología mixta combinando enfoques cualitativo y cuantitativo. Para la recolección de resultados se aplicaron encuestas, entrevistas semiestructuradas, observación directa y grupos focales (focus groups) con la participación de 45 colaboradores pertinentes a tres empresas del sector. Los resultados evidenciaron que los métodos tradicionales de evaluación presentan deficiencias en objetividad, tiempos de ejecución y retroalimentación efectiva, lo que genera una alta receptividad hacia la digitalización de este proceso. Concluye que su aplicativo web presenta como una solución innovadora, alineada con las actuales tendencias de transformación digital, liderazgo organizacional y responsabilidad social. Además, tiene el potencial de fortalecer la equidad laboral y optimizar la eficiencia en la gestión del talento humano.

Aunque el contexto del estudio está enfocado en el sector metalmecánico, comparte similitudes con el ámbito agrícola-industrial, como la necesidad de mejorar la objetividad en la evaluación, reducir tiempos de ejecución y garantizar una retroalimentación efectiva.

Por otro lado, en la investigación orientada a la implementación de una aplicación móvil para optimizar el monitoreo y control de los riesgos laborales, se desarrolló una herramienta tecnológica enfocada en la gestión de riesgos críticos dentro de la empresa EPSEMHCO, esta aplicación móvil logro mejorar el seguimiento de las condiciones laborales, facilitando la identificación de situaciones de riesgo y contribuyendo a una supervisión más eficiente y organizada.

Esta investigación utilizó la metodología Kanban, la cual facilitó la organización y ejecución de los procesos de planificación y codificación, permitiendo así un flujo de trabajo organizado y eficiente. Se realizaron pruebas de funcionalidad aplicadas demostraron un 73,33% de eficiencia. Asimismo, en cuanto a la portabilidad obtuvo un 88,24% en la adaptabilidad y

facilidad de acceso. En conclusión, el sistema se consolidó como una herramienta efectiva para supervisar y controlar las situaciones de riesgo dentro de la empresa[6].

La metodología mixta utilizada por el autor que incluye encuestas, entrevistas y observación directo también resulta pertinente para este proyecto, esto permite comprender la percepción y aceptación del personal operativo en los procesos de clasificación y embonche de rosas.

3.2 CONTEXTUALIZACIÓN DEL PROBLEMA

3.2.1 Importancia de la floricultura en Ecuador especialmente en Cotopaxi

La floricultura en Ecuador es un sector vital para la economía nacional que aporta al desarrollo social y ambiental, destacando la producción de rosas, esenciales para la exportación del país.

Lo interesante es cómo esta actividad ha florecido en una zona donde las condiciones naturales parecen hechas a medida. En Cotopaxi, la altura favorece tallos más robustos, el clima ofrece contrastes ideales para resaltar los colores, y la luz solar constante ayuda a obtener flores de altísima calidad[7] .

Desde la perspectiva de los investigadores, esto ha convertido a la región en un motor económico local que necesita seguir innovando para mantenerse competitivo. Y es que la floricultura ha transformado la vida en muchas comunidades: ya no se trata solo de cultivar flores, sino de una red de negocios y servicios que gira alrededor de esta industria, generando empleo digno, oportunidades de emprendimiento y desarrollo territorial.

3.3 EL FLUJO DE EXPORTACIÓN: PROCESOS DE POST-COSECHA

Entre 2018 y 2022, Ecuador no solo se consolidó como el segundo mayor exportador de rosas a nivel mundial, sino que también se enfrentó al desafío de mantener estándares de calidad cada vez más altos en el ámbito internacional. Los procesos como la clasificación y el emponche son etapas cruciales en la cadena productiva, por lo que optimizarlos es esencial para asegurar la competitividad del sector y mantener el prestigio de las rosas ecuatorianas en los mercados internacionales[8].

Analizando la investigación se puede aportar que, el liderazgo que ha logrado Ecuador en la exportación de rosas conlleva una responsabilidad continua en cuanto a innovación y mejora de procesos. No se trata solo de mantener el volumen de exportación; es crucial perfeccionar cada etapa operativa para cumplir con las exigencias técnicas y comerciales del mercado

internacional. En este contexto, fortalecer actividades como la clasificación y el emponche se convierte en una estrategia fundamental para garantizar uniformidad, una presentación óptima y un mayor valor agregado, todos elementos clave para mantener la competitividad del sector florícola ecuatoriano.

3.3.1 Clasificación

Una flor es seleccionada después de cumplir con las siguientes características para ser de exportación:

- Corte ideal el cual depende del mercado.
- No debe contener plagas y enfermedades.
- Sus tallos no deben estar torcidos.
- El botón debe ser adecuado de preferencia un punto intermedio de abierto y cerrado.

3.3.2 Embonche

El embonche es un proceso donde se agrupan y organizan los productos en conjuntos conocidos como bonches. Esto se hace siguiendo criterios como el tamaño, la calidad o la cantidad. Esta actividad no solo ayuda a estandarizar la presentación del producto, sino que también facilita su manejo en el proceso operativo. Además, el embonche es clave para controlar el rendimiento, ya que cada bonche se puede registrar y asociar al trabajador que lo elaboró. Así, se mejora tanto la organización como el seguimiento de la producción diaria.

3.3.3 Control

3.3.3.1 Manual

El control manual se refiere al método tradicional de registrar la producción de manera escrita, normalmente en hojas o cuadernos. Aquí se anota cuántos bonches ha elaborado cada trabajador. Este enfoque depende totalmente de la anotación hecha por personas, lo que puede dar lugar a errores, omisiones o inconsistencias en la información. Además, complica la consulta rápida de datos y el seguimiento preciso del rendimiento diario, lo que puede limitar la capacidad de los responsables de la empresa para tomar decisiones a tiempo.

3.3.3.2 Sistema de Control

Este proceso de control realizado con sistema automatizado es el que se implementa y requiere que la persona encargada recoja los bonches en los que vienen etiquetas dentro de los bonches los que poseen toda la información necesaria para enviar datos al sistema. Luego cortan a las medidas enviadas. Por consiguiente, los escaneará y estos se irán registrando dentro del sistema y podrán obtener reportes en tiempo real y determinar en el mismo día el rendimiento realizado por cada mesa de trabajo.

3.4 RENDIMIENTO LABORAL

3.4.1 Definición

Se define como la capacidad de un trabajador de cumplir con sus tareas de manera eficiente, resaltando su impacto en la productividad agroindustrial. Ahora bien, en la floricultura, este rendimiento no se mide solo en cantidad. Es un trabajo que exige precisión, sensibilidad y rapidez, pero también implica aguante físico y capacidad de concentración durante muchas horas[8].

El trabajador se convierte casi experto de las flores: desarrolla un instinto para detectar defectos, sabe qué tipo de rosa puede soportar un viaje largo y cuál no, y todo eso lo hace en condiciones de presión por la alta demanda.

Principales indicadores de rendimiento en agroindustria

Los indicadores clave están en la productividad por hora, la calidad del producto, y el cumplimiento de estándares internacionales. Detrás de cada número hay una historia: si alguien baja su rendimiento, puede deberse a fatiga, problemas personales o falta de motivación. Si la calidad del producto baja, quizás sea necesario revisar los métodos de capacitación o incluso las condiciones del entorno laboral[9].

Para los investigadores los estándares ya no se enfocan solo en lo técnico. Hoy en día, los consumidores también valoran la sostenibilidad y la ética laboral. Un cliente internacional quiere saber no solo si la rosa es bonita, sino si fue producida con responsabilidad. En este sentido, se considera que aplicar un enfoque estable permite identificar mejoras con impacto real y duradero tanto en el rendimiento como en la calidad de vida laboral.

3.5 LIMITACIONES DEL CONTROL MANUAL

Los métodos manuales limitan la recopilación de datos precisos, provocando errores frecuentes y dificultando la implementación de mejoras en los procesos productivos. Esto se muestra en pérdidas de tiempo, falta de claridad sobre qué está funcionando y qué no, y en muchos casos, decisiones tomadas con base en información incompleta o equivocada[10] .

Es frecuente que las fincas pierdan horas buscando un dato perdido entre pilas de papeles, cuando una plataforma digital podría entregar esa información en segundos. Con el razonamiento respectivo se propone que, continuar con controles manuales significa retrasar el crecimiento del sector. Automatizar el monitoreo permitiría detectar cuellos de botella, tomar decisiones con respaldo y responder con agilidad a las exigencias del mercado.

3.6 MONITOREO Y CONTROL DIGITAL

3.6.1 Concepto y beneficios del monitoreo digital

El uso de plataformas digitales permita visualizar en tiempo real el rendimiento laboral, lo que facilita la toma de decisiones estratégicas y mejora la eficiencia operativa .Estas plataformas no solo automatizan, sino que también transforman la forma en que se analiza el trabajo, dado que permiten ver patrones, evaluar tiempos y detectar áreas de mejora con datos reales, no con intuiciones[11].

Desde nuestra perspectiva estos sistemas son claves para avanzar: reducen la carga operativa, permiten a los equipos enfocarse en lo esencial y generan un entorno de trabajo más justo, informado y eficiente. Y por consiguiente también aporta un valor significativo para los dueños o encargados puesto que al tener más precisión se pagaría lo justo y no tendrían incongruencias por valores extras.

3.6.2 Comparación entre métodos manuales y automatizados

Los métodos automatizados superan ampliamente a los manuales, debido a que permiten recopilar datos en tiempo real, reducir errores y optimizar la planificación estratégica ; la automatización no significa reemplazar a las personas, sino potenciar su trabajo[12].

Desde la perspectiva planteada en el estudio, los métodos automatizados ofrecen una ventaja clara sobre los procesos manuales. Facilitan la recopilación de datos en tiempo real, reducen la posibilidad de errores y ayudan a una mejor planificación estratégica. Se cree que la

automatización no tiene como objetivo reemplazar el trabajo humano, sino más bien potenciarlo, proporcionando herramientas que permiten a las personas realizar sus tareas con mayor precisión, eficiencia y un valor añadido.

A continuación, se muestra la comparación de cada enforque en la tabla 3.1.

Tabla 3.1 Comparación entre métodos manuales y automatizados.

ENFOQUE MANUAL	**ENFOQUE AUTOMATIZADO**
Necesita un alto esfuerzo humano para recopilar información.	Recopila datos en tiempo real
Tiende a ser más propenso a errores	Reduce los errores posibles en la información
Su manera de obtener datos lenta y poco eficiente	Genera información precisa de manera rápida y continua.
Los trabajadores reciben poca retroalimentación útil.	Los trabajadores acceden a datos que mejoran su desempeño.
Tardan mucho llenando formularios.	Mejora el trabajo permitiendo centrarse en tareas que requieren atención urgente y reducen tiempo.
No contribuye significativamente a la dignificación del trabajo.	Mejora la productividad y dignifica el trabajo humano.

3.7 TENDENCIAS ACTUALES

El rendimiento laboral desde una perspectiva psicosocial y tecnológica puede mostrar que, en el enfoque cuantitativo, no experimental y de corte transversal, se evidenció una correlación negativa moderada entre el tecnoestrés y el rendimiento laboral, lo que indica que un aumento en los niveles de estrés tecnológico impacta negativamente el desempeño de los empleados.

Aunque dicha investigación se enfoca en un entorno urbano y en trabajadores de TIC , su aporte resulta relevante para el presente proyecto, porque destaca el impacto del entorno tecnológico en el bienestar y productividad de los trabajadores[13]. En el caso de Comexiger, se habla de un contexto agrícola-industrial, el proceso que conlleva al ingreso del rendimiento laboral

manual como ellos lo practican y si lo reemplaza por una plataforma informática para monitoreo se obtienen mejores resultados, pero a la vez puede generar inicialmente niveles de tecnoestrés en los operarios si no se acompaña de una capacitación adecuada y diseño amigable del sistema. En el extranjero, investigaciones similares abordan la digitalización del monitoreo de riesgos y rendimiento, uso de plataformas multiusuario y tecnologías IoT aplicadas a la agroindustria, logrando importantes avances en la prevención y control de accidentes[14].

Por tanto, este estudio invita a considerar el diseño de la plataforma informática, elementos conocidos, formación progresiva en el uso del sistema y monitoreo del impacto emocional del cambio tecnológico, garantizándoles así no solo eficiencia operativa sino también sostenibilidad humana. El sistema ayudará a mejorar tanto el estrés laboral y eficiencia para sus administradores el cual permite obtener su rendimiento en tiempo real y saber lo que se debe mejorar y tener poder evaluar al instante con datos reales instantáneos y no esperar un mes para poder llegar a obtener resultados.

3.8 METODOLOGÍA XP (EXTREME PROGRAMING)

La metodología eXtreme Programming-XP (Programación Extrema), obtuvo su nombre por el proceso de tomar una buena práctica que ya ha demostrado ser efectiva y potenciarla al máximo, llevándola a su nivel más alto de desarrollo y rendimiento, su desarrollador fue Kent Beck, el problema surge cuando los requisitos de software se vuelven cada vez más complejos a lo largo del desarrollo. A medida que aparecen nuevos cambios o ajustes, es necesario aplicarlos de forma ordenada para no afectar el tiempo ni el presupuesto definido[15].

Para criterio investigativo, se trata de un conjunto de procesos que pueden seleccionarse y adaptarse según las necesidades del proyecto, con el objetivo de impulsar una mejora significativa y sostenible, asegurando que el desarrollo continue de manera eficiente y contralada.

La metodología XP se diseñó primordialmente para equipos reducidos, de dos hasta de diez personas, que desempeñan en asignaciones constantes o poco definidas. Para su implementación son necesarias pruebas o versionamientos, XP es una metodología ágil y flexible que enfatiza la interconexión entre las etapas de diseño e implementación[16].

Desde la perspectiva de los investigadores la metodología XP es especialmente adecuada para equipos pequeños que trabajan en proyectos con requisitos cambiantes o poco claros. Su

enfoque ágil y flexible les permite adaptarse rápidamente a nuevas necesidades. Además, la práctica constante de pruebas y versiones no solo mejora la calidad del software, sino que también ayuda a reducir errores a lo largo del proceso. La estrecha relación entre diseño e implementación promueve un desarrollo más dinámico, colaborativo y enfocado en resultados eficientes.

Las etapas que se encontraron dentro de esta metodología son las siguientes:

3.8.1 Planificación

Esta es la primera etapa, en esta etapa se realiza una reunión con los clientes y el equipo para determinar las historias de usuario a través de encuestas o entrevistas por consiguiente una vez de realizado el desarrollo de las historias de usuario el equipo se reúne y las presenta para describir el resultado esperado. Entonces el quipo recolecta las historias y forma un plan que lo dividen en iteraciones importantes para solventar el funcionamiento que se desea parte por parte.

3.8.2 Diseño

Esta etapa nos ayuda a ver a futuro lo que se desea obtener, está relacionado con uno de los principales valores de XP: la simplicidad. Un buen diseño genera lógica y estructura al sistema esto evita contratiempos y repeticiones poco necesarias sin sentido.

3.8.3 Codificación

En esta etapa se crea el código que se empleará en la implementación de la metodología, como estándares para la codificación, programación por pares, integración continua y propiedad colectiva del código las cuales son herramientas esenciales para la obtención de esta etapa a modo que esta fase quede terminada.

3.8.4 Pruebas

Es el núcleo de esta etapa, dado que aquí se realizan amabas pruebas unitarias, pruebas automatizadas para determinar si lo que el cliente espera funciona correctamente también las pruebas de aceptación en donde el cliente probará el sistema y verificará su funcionalidad al 100% y verifica que todos sus requisitos se cumplan[17].

A partir de la investigación se considera que las etapas de esta metodología permiten un desarrollo organizado y enfocado en las necesidades del cliente. Finalmente, llegamos a la fase de pruebas, que es el corazón del proceso, donde realizamos pruebas unitarias, automatizadas y de aceptación para asegurarnos de que el sistema cumpla con todos los requisitos y expectativas del cliente.

3.8.5 Valores

Desde el análisis realizado, los valores de la metodología XP detallados en la Figura 3.1 las personas son el corazón del proceso de desarrollo, por cuanto fomentan el trabajo en equipo, la claridad en los objetivos, la mejora continua y el coraje para enfrentar cambios, asegurando así un ambiente colaborativo y enfocado en la calidad.

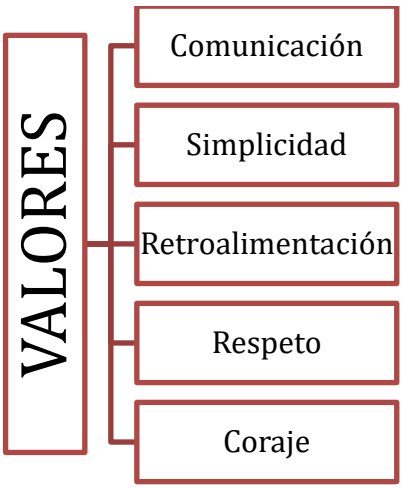


Figura 3.1.Valores de Metodología XP.

3.8.6 Principios

Con base en lo expuesto por los investigadores en la Figura 3.2. Estos principios guían al equipo en su trabajo a través de cambios graduales, asegurando calidad, aceptando la transformación, proporcionando retroalimentación rápida y asumiendo responsabilidades. Esto permite que el desarrollo sea continuo, adaptable y centrado en cumplir de manera eficiente con los requisitos del proyecto.

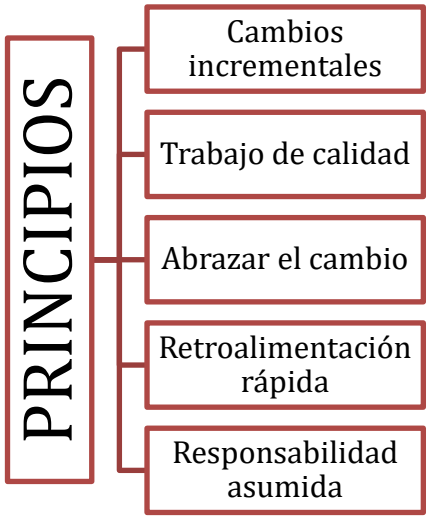


Figura 3.2. Principios de la Metodología XP.

3.8.7 Comparación con otras metodologías

Para la elección significativa de la metodología más apropiada para esta investigación se realiza una comparación entre la metodología XP y algunas más que se detallan en la tabla 3.2.

Tabla 3.2.Comparación entre la Metodología XP y otras conocidas.

CRITTERIO	**XP[17]**	**SCRUM[18]**	**KANBAN[19]**
ENFOQUE PRINCIPAL	Prácticas de ingeniería y excelencia técnica del software.	Gestión del proyecto y trabajo en equipo	Flujo continuo y visualización del proceso.
CICLOS DE TRABAJO	Iteraciones de 1-2 semanas.	Sprints fijos de 2-4 semanas.	Flujo continuo, sin iteraciones.
TAMAÑO DE EQUIPO	2-10 desarrolladores	5-9 personas cross-funcionales	Flexible, cualquier tamaño
PRÁCTICAS CLAVE	Programación en parejas, TDD, Integración continua	Sprint Planning, Daily Standup, Retrospectives	Tablero visual, WIP Limits, Cycle Time

CRITTERIO	**XP[17]**	**SCRUM[18]**	**KANBAN[19]**
FLEXIBILIDAD AL CAMBIO	Cambios en cualquier momento.	Solo entre sprints.	Cambios continuos
PARTICIPACIÓN DEL CLIENTE	El cliente tiene el poder de decidir prioridades es decir qué funciones se hacen primero y definir los criterios de aceptación de cada tarea.	Product Owner representa al cliente	Variable según necesidad
MEJOR PARA	Proyectos con requisitos cambiantes y alta calidad técnica	Proyectos con alcance definido y necesidad de estructura	Mantenimiento, soporte y flujo continuo

Después de analizar los diferentes aspectos de cada metodología, podemos concluir que Extreme Programing (XP) representa la mejor opción a elegir por su efectivísimo en el desarrollo de software; la principal ventaja de XP sobre Scrum y Kanban radica en su enfoque riguroso en las prácticas de ingeniería también en el aspecto que se adapta fácilmente a los cambios.

3.9 TIPOS DE PLATAFORMAS MÓVIL Y WEB.

Analizar estas plataformas nos ayuda a entender mejor sus características, alcances y limitaciones. Esto, a su vez, apoya de manera coherente las decisiones que se toman en el proyecto y nos permite contextualizar el entorno en el que funcionará el sistema que estamos desarrollando.

3.9.1 Aplicaciones Móviles

Durante las últimas 3 décadas el desarrollo tecnológico ha propiciado en que las TIC se han vuelto una herramienta indispensable para obtener mejores condiciones de bienestar y desarrollo entre personas, empresas e instituciones.

Una aplicación móvil, en términos simples, es un programa para celular creado para ayudar a las personas a llevar a cabo tareas específicas o facilitar actividades en dispositivos electrónicos. Puede ser utilizada para trabajar, comunicarse, organizar información o simplemente para disfrutar del entretenimiento.

Por otro lado, una aplicación móvil (App) es un tipo de programa que se descarga de Internet y se instala en dispositivos como teléfonos inteligentes, tabletas o incluso reproductores multimedia. Gracias a la conexión a Internet, estas aplicaciones permiten acceder a una variedad de servicios y funciones de manera rápida y sencilla, adaptándose a lo que el usuario necesita. Hoy en día, las aplicaciones móviles son una parte esencial del marketing digital, siendo uno de los segmentos que más ha crecido en los últimos años.

Esto se debe a que están disponibles en casi todos los teléfonos, incluso en los modelos más básicos, lo que facilita su acceso y uso por un gran número de personas[20].

Para los investigadores, las aplicaciones móviles no solo son herramientas tecnológicas, sino también oportunidades estratégicas para mejorar la comunicación, optimizar procesos y fortalecer la interacción entre las organizaciones y sus usuarios.

3.9.2 Aplicaciones Web

Hoy en día, el desarrollo de sistemas en línea se ha convertido en una de las estrategias más importantes para negocios e instituciones, facilitando el acceso a la información de manera rápida y sencilla.

El crecimiento de las aplicaciones web ha sido impresionante, impulsado por el impacto del internet como un medio clave para difundir información y servicios. A medida que la tecnología avanza, la complejidad en el desarrollo de estas aplicaciones también ha aumentado[21].

A partir del análisis realizado en la investigación, se puede observar que el rápido crecimiento de las aplicaciones web es un claro reflejo de la transformación digital que ha vivido nuestra sociedad en los últimos años. El internet se ha convertido en el canal principal para compartir

información y ofrecer servicios. Ha aumentado las expectativas en cuanto a seguridad, escalabilidad, rendimiento y experiencia del usuario. Como resultado, el desarrollo de aplicaciones web se ha vuelto más complejo, requiriendo metodologías bien estructuradas y conocimientos técnicos especializados para enfrentar de manera efectiva los nuevos retos del entorno digital.

En el ámbito de la ingeniería, se entiende por aplicación web a aquellas que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet, todo ello mediante un navegador. Es decir, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web (HTML, JavaScript, Java, asp.net, etc.) en la que se confía la ejecución al navegador[22].

Por nuestra parte, en el mundo de la ingeniería, una aplicación web se ve como una herramienta que permite a los usuarios acceder a un sistema a través de un navegador, ya sea por Internet o una intranet. Este tipo de aplicación se desarrolla con lenguajes que son compatibles con los navegadores, lo que hace que se ejecute sin necesidad de instalaciones adicionales, convirtiéndola en una solución práctica y accesible para diferentes entornos tecnológicos.

3.10 LENGUAJES Y HERRAMIENTAS UTILIZADAS

3.10.1 Lenguajes

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.

3.10.1.1 Python

Python es un lenguaje de programación que facilita el trabajo rápido y ayuda a integrar sistemas de manera más efectiva con menores costos de mantenimiento, sin importar que el usuario sea nuevo o experimentado en programación es fácil de aprender.

Como se puede observar en la Figura 3.3 la imagen es representativa del lenguaje Python con el propósito de identificar visualmente la herramienta tecnológica utilizada para el desarrollo del proyecto.

Figura 3.3.Lenguaje Python[26].

3.10.1.1.1 Características:

- **Fácil de aprender:** Python tiene documentación en el cual detalla fácilmente como podemos implementar o usar su código.

- **Aplicaciones:** Python aloja miles de módulos de terceros para su biblioteca estándar, así como módulos de la comunidad de Python.

- **Código abierto:** Python está desarrollado con licencia de código abierto, lo cual lo hace fácilmente utilizable y distribuible[23].

Según la interpretación desarrollada en la investigación podemos decir que Python se destaca por su capacidad para facilitar un desarrollo ágil y una integración eficiente de sistemas, todo mientras mantiene los costos de mantenimiento bastante bajos. Su sintaxis sencilla y su documentación clara hacen que aprenderlo sea un paseo, y la gran cantidad de módulos disponibles, junto con su naturaleza de código abierto, lo convierten en una herramienta flexible y adaptable para diferentes entornos tecnológicos.

3.10.1.2 Dart

Es un lenguaje de programación accesible, portátil y productivo para aplicaciones de alta calidad en cualquier plataforma. Este es gratuito y de código abierto.

La Figura 3.4, la imagen es representativa del lenguaje Dart con el propósito de identificar visualmente la herramienta de programación utilizada en el desarrollo de la aplicación móvil.

Figura 3.4.Lenguaje Dart[28].

3.10.1.2.1 Características:

- **Accesible:** Permite desarrollar un estilo consistente, conciso y escrito un lenguaje que ofrece características modernas como seguridad y coincidencia de patrones.

- **Portátil y rápido:** Compila en máquinas ARM, x64 y RISC-V y código para dispositivos móviles, de escritorio y backend.

- **Productivo:** Realiza cambios de forma iterativa viendo cambios instantáneos en su aplicación en ejecución[24].

Desde el análisis que hemos realizado, se puede decir que Dart es un lenguaje que destaca por ser accesible, portátil y muy productivo, ideal para desarrollar aplicaciones de alta calidad en diversas plataformas. Su naturaleza gratuita y de código abierto amplía las posibilidades de uso y adaptación. Además, permite mantener un estilo de programación claro y moderno, y facilita la visualización inmediata de los cambios durante el desarrollo, lo que realmente optimiza el proceso de creación de aplicaciones.

3.10.2 Herramientas

3.10.2.1 Django

Django es un framework de desarrollo web basado en Python, este propone el desarrollo veloz con un diseño intacto.

Fue creado por desarrolladores con gran experiencia, este resuelve molestias del desarrollo web, nos ayuda a centrarnos en nuestras aplicaciones sin tantas complejidades. Es muy rápido y eficaz para ser gratis y también de código abierto.

La Figura 3.5 presenta el logotipo del framework Django, para que podamos identificar visualmente la tecnología utilizada en la creación y administración del backend del sistema.

Figura 3.5. Framework Django[25].

3.10.2.1.1 Características:

- **Es rápido:** Django fue desarrollado para ser eficaz, siendo este capaz de llevar las aplicaciones desde concepto a fin de manera rápida.

- **Muy seguro:** Django es muy serio con la seguridad y también ayuda a los desarrolladores a esquivar errores de seguridad que son bastante comunes.

- **Es escalable:** Muchos sitios web activos se ayudan de Django para escalar de manera rápida y flexible[26].

A continuación, en la tabla 3.3, se presenta un cuadro comparativo justificando la elección del framework Django:

Tabla 3.3.Comparación entre el Framework Django y otros frameworks.

	Django[26]	**Laravel**[27]	**Express.js**[28]
Lenguaje	Python, lenguaje legible, versátil y con gran soporte para desarrollo web y científico.	PHP, lenguaje orientado a la web, ampliamente usado, pero con sintaxis menos moderna.	JavaScript, permite trabajar frontend y backend con un mismo lenguaje, pero puede volverse complejo por su asincronía.
Rendimiento	Alto rendimiento, estable y eficiente gracias a su arquitectura bien estructurada.	Rendimiento medio, puede requerir optimización adicional en proyectos grandes.	Muy alto rendimiento por el motor V8, pero depende del desarrollador mantener la estabilidad.
Seguridad	Muy alta seguridad con protección automática contra CSRF, XSS, SQL Injection y autenticación robusta. Ideal para sistemas críticos.	Alta seguridad, aunque requiere configuraciones adicionales en algunos casos.	Seguridad variable; depende mucho de librerías externas y configuraciones manuales.
Base de datos	ORM incorporado muy potente, compatible con PostgreSQL, MySQL, Oracle y más, simplificando el acceso y la gestión de datos.	Eloquent ORM, intuitivo, pero más lento y pesado en algunos escenarios.	No incluye ORM; se debe instalar y configurar todo manualmente.
Ventajas	Rápido de desarrollar, altamente seguro,	Sintaxis elegante, buena	Súper flexible y rápido para APIs,

	Django[26]	**Laravel**[27]	**Express.js**[28]
	escalable, incluye panel administrativo automático y una estructura clara para proyectos grandes.	documentación y comunidad activa.	ideal para microservicios.
Desventajas	Menos flexible en proyectos pequeños por su estructura estricta, curva de aprendizaje media.	Puede consumir más recursos y volverse lento en aplicaciones de gran escala.	Requiere demasiadas librerías, poca estructura, más riesgo de errores y malas prácticas.

Por lo cual entre los Frameworks evaluados hemos elegido Django como nuestro Framework ideal.

En resumen, Se utilizó Django puesto que ofrece una mejor seguridad, una mejor estabilidad, estructura que los otros Frameworks. Al trabajar con Python nos da un entorno más moderno y escalable. Su rendimiento es alto y nos ayuda con ataques; su protección viene integrada por defecto.

3.10.2.2 PostgreSQL

PostgreSQL es un sistema de base de datos relacional de código abierto que utiliza el lenguaje SQL y ofrece diferentes funciones para almacenar y gestionar grandes cantidades de datos de forma segura y eficiente[29].

En el marco de esta investigación, PostgreSQL se presenta como una herramienta muy confiable para gestionar información, ya que facilita la organización, almacenamiento y consulta de datos de forma segura. Además, su habilidad para manejar grandes volúmenes de información lo hace ideal para el desarrollo de aplicaciones que necesitan estabilidad y un buen rendimiento.

La Figura 3.6 presenta el logotipo del motor de base de datos PostgreSQL, para identificar visualmente la tecnología utilizada en la creación y administración del sistema.

Figura 3.6.Motor de base de datos PostgreSQL.

3.10.2.2.1 Características

- Tipos de datos variados: permite trabajar con datos numéricos, texto, booleanos, fechas, JSON y otros formatos.
- Integridad de datos: utiliza claves primarias, claves foráneas y restricciones como único y no nulo para asegurar la consistencia de la información.
- Indexación avanzada: mejora la velocidad de búsqueda y consulta dentro de la base de datos.
- Control de concurrencia (MVCC): permite que varios usuarios trabajen al mismo tiempo sin afectar el rendimiento.
- Transacciones seguras: garantiza que las operaciones con datos se realicen de forma correcta y completa
- Seguridad: cuenta con sistemas de autenticación y control de acceso para proteger la información.
- Replicación y recuperación: incluye mecanismos para respaldo y recuperación de datos ante fallos[29].

3.10.2.3 Visual Studio Code

Visual Studio Code (VS Code) un editor ultrarrápido, ideal para el uso diario. Tiene herramientas que ayudan al desarrollador a ejecutar sus ideas sin distracciones. Con soporte

para muchos tipos de lenguajes, ayudando así a ser muy productivo con resaltado de sintaxis, como es en sangría, coincidencia de corchetes etc.

En esta sección, la Figura 3.7 presenta la imagen representativa de Visual Studio Code, con el fin de destacar el entorno de desarrollo que se utiliza para programar y editar el código del proyecto.

Figura 3.7. Editor de código Visual Studio Code[30].

3.10.2.3.1 Características:

- **Depuración:** Visual Studio Code incluye un depurador interactivo lo que recorre el código fuente así también con inspección de variables, llamados y comandos de consola.

- **Arquitectura:** Visual Studio Code combina y usa tecnologías como JavaScript y Node.js junto a la rapidez y adaptabilidad de las aplicaciones nativas[31].

A continuación, en la tabla 3.4, se presenta un cuadro comparativo justificando la elección del editor de código Visual Studio Code:

Tabla 3.4. Comparación entre el editor de código Visual Studio Code y otros editores de código.

	Visual Studio Code[31]	**Sublime Text**[32]	**Atom**[33]
Licencia / Costo	Gratis	De pago (USD $99, licencia única)	Gratis (Open Source, descontinuado)

	Visual Studio Code[31]	**Sublime Text**[32]	**Atom**[33]
Rendimiento	Muy buen rendimiento; rápido y estable en proyectos medianos y grandes	El más rápido; extremadamente ligero	El más rápido; extremadamente ligero
Lenguajes Soportados	Compatible con casi todos los lenguajes gracias a extensiones: Python, Django, PHP, JS, TS, C++, Java, Flutter, etc.	Soporta varios lenguajes mediante plugins	Soporta muchos lenguajes, pero sin actualizaciones
Extensiones y Personalización	Miles de extensiones, temas e integraciones profesionales	Menos extensiones que VS Code; buena personalización básica	Muy personalizable, pero sin soporte oficial
Consumo de Recursos	Equilibrado entre potencia y rendimiento	Muy bajo; ideal para PCs de bajos recursos	Alto; tiende a ralentizar el sistema
Ventajas	• Totalmente gratis • Marketplace enorme • Autocompletado inteligente (IntelliSense) • Depurador integrado • Git y GitHub nativos • Integración con Docker, bases de datos y servidores	• Ultra ligero • Arranque casi instantáneo • Muy estable • Perfecto para escribir código rápido	• Muy personalizable • Interfaz agradable • Fácil de usar para principiantes

	Visual Studio Code[31]	**Sublime Text**[32]	**Atom**[33]
	• Excelente para cualquier lenguaje y tipo de proyecto • Actualizaciones constantes		
Desventajas	• Puede consumir más recursos si se instalan muchas extensiones • Algo pesado en PCs muy antiguas	• No es gratuito • Pocas extensiones modernas	• Descontinuado • Rendimiento pobre • No recomendable para proyectos actuales

Basándonos en la información recopilada y en el análisis comparativo entre Visual Studio Code y otros editores de código, hemos llegado a la conclusión de que Visual Studio Code es la opción más adecuada para el desarrollo de este proyecto. Esta elección se apoya en el hecho de que es una herramienta gratuita, recibe actualizaciones constantes y cuenta con un extenso marketplace de extensiones que permite trabajar con varios lenguajes como Python, Django, JavaScript y Flutter. Además, ofrece características avanzadas como IntelliSense, un depurador integrado, control de versiones con Git y compatibilidad con entornos como Docker y bases de datos, lo que realmente potencia la productividad y la organización en el desarrollo.

3.10.2.4 Flutter

Flutter es un marco de desarrollo que te permite crear aplicaciones multiplataforma compiladas de forma nativa a partir de una única base de código.

Se presenta en la Figura 3.8 una imagen que representa el framework Flutter, la cual sirve como un recurso visual para reconocer la tecnología empleada en el desarrollo de la aplicación móvil.

Figura 3.8. Framework Flutter[34].

3.10.2.4.1 Características:

- **Rápido:** El código se traduce a código de máquina, ya sea para ARM o INTEL, y también se convierte en JavaScript. Esto nos ayuda a lograr un rendimiento rápido en cualquier dispositivo.
- **Productivo:** Construye y prueba rápidamente con reload. Cada vez que actualices el código, los cambios aparecerán al instante.
- **Flexible:** Controla los pixeles para crear diseños personalizados y que sean adaptables a cualquier pantalla[35].

A partir del análisis realizado en la investigación, se puede afirmar que Flutter se ha consolidado como un marco de desarrollo muy eficiente para crear aplicaciones multiplataforma a partir de una sola base de código, lo que ayuda a optimizar tanto el tiempo como los recursos. Su capacidad para compilar de manera nativa permite lograr un alto rendimiento en diversos dispositivos. Además, la función de recarga rápida hace que sea posible realizar ajustes inmediatos durante el desarrollo, lo que mejora la productividad. Su flexibilidad en el diseño ofrece un control total sobre la interfaz, permitiendo crear aplicaciones que son adaptables y visualmente atractivas en diferentes tamaños de pantalla.

3.10.2.5 Escáner TERA de código de barras inalámbrico 2D D5100

Es un dispositivo de mano en forma de pistola diseñado para leer, decodificar y transmitir información desde códigos de barras a un sistema informático. Automatiza y acelera el proceso de captura de datos, eliminando la necesidad de introducir manualmente largas series numéricas o alfanuméricas y reduciendo significativamente los errores humanos.

La Figura 3.9 que se muestra nos ayuda a identificar el escáner TERA inalámbrico 2D D5100, un dispositivo que se utiliza para leer y gestionar códigos de barras dentro del sistema que se ha implementado.

Figura 3.9. Escáner TERA de código de barras inalámbrico 2D D5100[36].

3.10.2.5.1 Características:

- **Tecnología 1D/2D/QR:** Lee todos los formatos de códigos de barras, código de barras 1D, códigos QR 2D, etc.

- **Conectividad dual:** Funciona en modo inalámbrico con un USB que se conecta al computador con capacidad de 100 metros de distancia. También funciona de manera alámbrica es decir por medio de cable para puestos fijos o para cargar su batería [37].

En opinión de los investigadores podemos decir que el escáner de códigos de barras es una herramienta fundamental para mejorar los procesos de registro y control de información en el sistema. Al automatizar la lectura y decodificación de datos, se reducen significativamente los errores que suelen ocurrir con el ingreso manual, lo que acelera la captura de información. Su capacidad para leer formatos 1D, 2D y códigos QR amplía su utilidad en diferentes entornos laborales. Además, la conectividad dual, tanto inalámbrica como por cable, ofrece flexibilidad operativa y facilita su integración en diversos escenarios de uso.

3.10.2.6 Impresora Térmica de Etiquetas Nelko PL70e-BT

Impresora de etiquetas profesional que utiliza tecnología térmica. Su función principal es producir etiquetas de envío de tamaño estándar (4x6 pulgadas) de manera rápida y confiable, pero es capaz de imprimir una variedad de otros tamaños de etiquetas para múltiples usos.

Se incluye la Figura 3.10 de la impresora térmica Nelko PL70e-BT para que podamos identificar el equipo que se utiliza en el proyecto para imprimir etiquetas.

Figura 3.10. Impresora Térmica de Etiquetas Nelko PL70e-BT[38].

3.10.2.6.1 Características:

- **Tecnología sin tinta:** Utiliza tecnología térmica directa. Solo se necesita rollos o pilas de etiquetas térmicas, lo que elimina el gasto, desorden y mantenimiento de cartuchos de tinta.

- **Conectividad:** Se conecta inalámbricamente a dispositivos móviles o computadoras a través de Bluetooth con su aplicación dedicada. También se puede conectar por cable a computadoras para una conexión estable y rápida[39].

Según nuestra perspectiva, la impresora térmica de etiquetas juega un papel clave al permitir la creación rápida y confiable de etiquetas en formatos estándar y otros tamaños, adaptándose a diversas necesidades operativas. Su tecnología de impresión térmica directa elimina la necesidad de tinta, lo que reduce los costos de mantenimiento y simplifica su uso. Además, su conectividad inalámbrica a través de Bluetooth y la opción de conexión por cable ofrecen versatilidad y estabilidad, facilitando su integración con dispositivos móviles y equipos informáticos en el entorno laboral.

3.10.2.7 WebSocket

WebSocket es una tecnología innovadora que establece una conexión de comunicación interactiva entre el navegador del usuario y un servidor. Esta API permite enviar mensajes al servidor y recibir respuestas de manera controlada, sin necesidad de hacer consultas constantes para obtener información[40].

Se incluye la Figura 3.11 representativa de la tecnología WebSocket para identificar el protocolo de comunicación que permite la transmisión de datos en tiempo real entre el cliente y el servidor dentro del sistema desarrollado.

Figura 3.11. WebSocket[41].

En la tabla 3.5 se puede visualizar la comparación realizada entre WebSocket y Ajax para realizar la elección ideal para sea óptima para el proyecto.

Tabla 3.5. Comparación entre la tecnología WebSocket y AJAX.

Categoría	**WebSocket**[40]	**AJAX**[42]
Tipo de Tecnología	Protocolo de comunicación bidireccional y persistente. Permite conexión abierta entre cliente y servidor.	Técnica basada en HTTP que permite enviar y recibir datos de forma asíncrona sin recargar la página.
Comunicación	Bidireccional en tiempo real: el servidor puede enviar datos al cliente sin que este los solicite.	Unidireccional bajo demanda: el cliente debe solicitar la información al servidor mediante peticiones.
Persistencia de Conexión	Conexión persistente (se mantiene abierta mientras dure la sesión).	Conexión no persistente (cada petición abre y cierra conexión).

Categoría	**WebSocket**[40]	**AJAX**[42]
Rendimiento	Excelente rendimiento para datos en tiempo real y comunicación continua.	Bueno para operaciones puntuales. El rendimiento baja si se hacen muchas peticiones frecuentes.
Compatibilidad	Amplio soporte en navegadores modernos.	Totalmente compatible incluso con navegadores más antiguos.
Ventajas	• Comunicación en tiempo real. • Menor overhead en conexiones continuas. • Mayor interactividad.	• Fácil de implementar. • Ideal para solicitudes eventuales. • Basado en HTTP estándar.
Desventajas	• Configuración más compleja. • Requiere servidor compatible. • No siempre necesario para apps simples.	• No es tiempo real puro. • Requiere múltiples peticiones • Mayor overhead en uso frecuente.

Desde un enfoque técnico en la investigación, WebSocket se presenta como una tecnología que permite establecer una comunicación bidireccional y continua entre el navegador y el servidor, lo que facilita el intercambio de datos en tiempo real. A diferencia de los métodos tradicionales, elimina la necesidad de hacer solicitudes repetitivas para obtener información, mejorando así el rendimiento y la eficiencia del sistema. Esta característica es especialmente valiosa en aplicaciones que requieren actualizaciones inmediatas de datos y una interacción constante con el usuario.

3.10.2.8 Listener

Es un objeto en Java que escucha y responde a componentes generados por la interfaz de usuario. Estos son los mecanismos generales para que una aplicación GUI sea interactiva[43].

Ejemplos:

- Hacer clic a un botón.
- Escribir texto.
- Mover el mouse.

Nuestra perspectiva es que, en Java, un listener se define como un objeto que se encarga de detectar y reaccionar a los eventos que surgen de la interfaz gráfica de usuario, lo que permite que la aplicación sea interactiva. Acciones como hacer clic en un botón, escribir texto o mover el mouse activan estos mecanismos, facilitando la comunicación entre el usuario y el sistema.

3.10.2.9 API REST

3.10.2.9.1 API

Es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar sistemas de software en las aplicaciones. Se considera como un contrato entre el usuario y el proveedor de información, donde se establece el contenido que necesita el consumidor (la llamada) y el que requiere el productor (la respuesta).

3.10.2.9.2 API REST

Se trata de una interfaz de programación de aplicaciones (API) que se adhiere a los principios de diseño de la arquitectura REST. REST, que significa transferencia de estado representacional, es un conjunto de pautas y recomendaciones para crear una API web[44] .

Consideramos que se trata de un conjunto de definiciones y protocolos que permiten que diferentes sistemas de software se integren y se comuniquen entre sí. Actúan como un acuerdo que define cómo se hace una solicitud y qué tipo de respuesta se puede esperar. En este contexto, la API REST se basa en los principios de la arquitectura REST, que está diseñada para crear servicios web que sean estructurados, escalables y organizados, facilitando así un intercambio eficiente de información entre el cliente y el servidor.

3.10.2.10 Daphne

Es un servidor ASGI en Python para HTTP, HTTP/2 y WebSocket, desarrollado por el equipo de Django. Creado originalmente como parte del proyecto Django Channels para permitir que se maneje conexiones en tiempo real[45].

A continuación, en la tabla 3.6, se presenta un cuadro comparativo entre Daphne y el servidor principal de Django (WSGI)**:**

Tabla 3.6.Comparación entre el servidor Daphne y WSGI.

Categoría	**Daphne**[45]	**Servidor original de Django (WSGI)**[46]
Tipo de Servidor	Servidor ASGI diseñado para manejar conexiones asíncronas y en tiempo real.	Servidor WSGI de desarrollo para aplicaciones tradicionales basadas en peticiones HTTP.
Soporte de Protocolos	Soporta HTTP, HTTP/2 y WebSockets.	Solo soporta HTTP bajo WSGI. No maneja WebSockets.
Comunicación Asíncrona	Sí, soporta asincronía nativa. Ideal para apps en tiempo real.	No soporta asincronía real. Funciona de forma síncrona.
Rendimiento	Mejor rendimiento en apps con alta concurrencia o eventos en tiempo real.	Adecuado para cargas pequeñas en desarrollo; rendimiento limitado.
Soporte para WebSockets	Sí.	No.
Ventajas	• Maneja WebSockets y async. • Ideal para chats, notificaciones, streaming.	• Muy simple de usar. • Viene integrado.

Categoría	**Daphne**[45]	**Servidor original de Django (WSGI)**[46]
	• Pensado para alta concurrencia.	• Ideal para desarrollo rápido.
Desventajas	• Requiere configuración adicional. • Mayor complejidad.	• No soporta WebSockets. • No es asíncrono. • No apto para producción.

Desde la perspectiva de la investigación, la elección de Daphne surge de la necesidad de contar con un servidor que pueda gestionar tanto la comunicación en tiempo real como los procesos asíncronos. Al ser un servidor ASGI que es compatible con HTTP y WebSockets, ofrece ventajas significativas en comparación con el servidor WSGI tradicional de Django, que solo opera de manera síncrona. Dado que el proyecto requiere una mayor concurrencia y respuestas inmediatas, Daphne se presenta como una opción más adecuada, ya que optimiza el rendimiento y la escalabilidad del sistema, alineándose perfectamente con los requisitos tecnológicos establecidos.

3.10.2.11 GitHub

GitHub es una plataforma basada en la nube donde puedes almacenar, compartir y trabajar junto con otros usuarios para escribir código además de permitir llevar el control de cambios que se realizan a nivel de código[47].

Dentro del desarrollo de este estudio, GitHub se presenta como una plataforma digital que permite almacenar proyectos de programación en la nube. Con esta herramienta, los usuarios tienen la posibilidad de compartir su código y colaborar con otros en el desarrollo de software, al mismo tiempo que mantienen un registro de los cambios realizados en el proyecto.

3.10.3 Servidor

Según[48] un servidor web es un programa creado para transferir datos de hipertexto, es decir, aplicaciones web que contienen elementos como: textos, widgets, CSS, etc. Detalla estos son alojados en una computadora que posee internet; él espera la petición de algún navegador para

poder acceder a la aplicación y responde dicha petición. La respuesta que envía es en código HTML mediante transferencia de datos.

Desde el punto de vista de los investigadores, un servidor web no solo envía la información; también facilita la comunicación entre los clientes y las aplicaciones en línea. Gracias a su ejecución, las personas pueden acceder a páginas web y disfrutar de diversos servicios digitales. Además de responder a solicitudes en HTML, también gestiona procesos dinámicos y consultas a bases de datos. Por esta razón, el servidor web se considera una parte esencial en el funcionamiento de las aplicaciones y sistemas web que utilizamos hoy en día.

3.10.3.1 Render

Render es un servidor web que ejecuta aplicaciones dentro de servidores basados en Linux y ayuda alojar aplicaciones web en frameworks como: Node.js, Node.js con Express, Python con Django o FastAPI lo que sea. Render compila e implementa su código con cada envío a su rama Git vinculada. Cada servicio web de Render obtiene un valor único onrender.com subdominio[49].

Según el análisis render es una plataforma que facilita la publicación de aplicaciones web en internet de una manera mucho más sencilla. Es compatible con diversas tecnologías de desarrollo, lo que hace que los proyectos puedan ejecutarse sin necesidad de configuraciones complicadas en un servidor. Además, se integra con repositorios de código, lo que significa que cada vez que actualizas el proyecto, los cambios se reflejan automáticamente en la aplicación. Así, también se genera una dirección web única que permite acceder al sistema desde cualquier lugar.

3.10.4 Draw.io

Draw.io es una herramienta gratuita de código abierto para crear diagramas técnicos, como diagramas de flujo, UML, redes, infraestructura en la nube y organogramas. Funciona en el navegador, como una aplicación de escritorio, e incluso cuando estás fuera de línea Tihs cosa funciona bien con Confluence, Jira, GitHub, y Google Drive Guardar los archivos en sus propios servidores, elegir dónde mantenerlos Tihs permite que la gente trabaje juntos en vivo y puede crear diagramas con ayuda de IA[50].

Con el análisis realizado se puede decir que draw.io es muy útil para organizar y visualizar ideas. En lugar de explicar todo con palabras, podemos crear un diagrama de flujo rápidamente y de forma sencilla. Esto hace que el trabajo en equipo y la planificación de proyectos sean mucho más fáciles.

4 MÉTODOS Y PROCEDIMIENTOS

4.1 TIPO DE INVESTIGACIÓN

Para desarrollar este proyecto se decidió usar la Investigación Aplicada, porque en primera instancia analizamos el problema sobre la falta de automatización tecnológica para proporcionar un rendimiento en tiempo real en la comercializadora Comexiger, obteniendo esta información requerida se podrá obtener resultados claros y sin errores para evitar así un descontento entre los trabajadores y también reducir perdidas para la empresa. Mediante los conocimientos adquiridos en el transcurso de la carrera e información recolectada por medio de diversas fuentes bibliográficas es factible el desarrollo de una Plataforma informática para de este modo resolver el problema encontrado.

4.2 MÉTODO DE INVESTIGACIÓN

Para llevar a cabo el proyecto en la empresa Comexiger, después de evaluar las características de cada enfoque, se decidió optar por el método de Investigación-Acción. Este enfoque permite intervenir directamente en los problemas identificados, implementar soluciones y evaluar los resultados en el mismo contexto donde surge la necesidad.

La Investigación-Acción se basa en la idea de que la investigación no debe limitarse a observar y describir un problema, sino que debe proponer e implementar mejoras concretas. En el caso de Comexiger, este método permitió que tanto los propietarios como los trabajadores no solo fueran sujetos de análisis, sino que también participaran activamente en la identificación de necesidades, el diseño del sistema informático y la validación de su funcionamiento en el entorno real.

Además, este enfoque facilitó una retroalimentación continua durante el desarrollo de la plataforma, permitiendo realizar ajustes progresivos a medida que se obtenían resultados en cada fase.

Fue fundamental elegir un método de investigación que se ajustara tanto a las características del entorno organizacional como a los objetivos del sistema que se quería implementar. La selección del método se hizo teniendo en cuenta varios criterios, como el nivel de participación de los beneficiarios, la capacidad de adaptarse a cambios durante el desarrollo, la conexión entre teoría y práctica, y la posibilidad de validar los resultados en un contexto real de operación.

4.3 ENFOQUE

El proyecto se llevó a cabo con un enfoque mixto, combinando tanto elementos cualitativos como cuantitativos para analizar y resolver el problema que se presentó en la empresa Comexiger.

Desde la perspectiva cualitativa, se realizaron observaciones directas de los procesos de postcosecha y se llevaron a cabo entrevistas informales con los propietarios y trabajadores. Esto permitió entender mejor la dinámica operativa, las dificultades que enfrentan y las necesidades reales del entorno laboral.

En cuanto al enfoque cuantitativo, se aplicó al evaluar los resultados tras la implementación del sistema informático, teniendo en cuenta indicadores como la reducción de errores en el registro, la mejora en el control del rendimiento y la optimización del tiempo en los procesos operativos.

La fusión de ambos enfoques no solo facilitó una comprensión más profunda de la problemática desde una perspectiva humana y organizacional, sino que también permitió medir de manera objetiva el impacto de la solución tecnológica que se implementó.

4.3.1 Técnicas

4.3.1.1 Entrevista

La entrevista se aplicó a los administradores de la empresa(dueños) con el propósito de conocer directamente sus dificultades posteriormente se observó el proceso y necesidades del personal de trabajo. Esta herramienta permitió obtener información cualitativa sobre la manera en que se administran las actividades diarias así identificando problemas en la organización y control de información. Gracias a los resultados de esta herramienta obtuvimos una base para definir los requerimientos iniciales de la plataforma informática.

4.3.1.2 Encuesta

Fue un medio que nos ayudó a recopilar la información de manera estructurada, permitiendo así validar y ordenar las necesidades de la empresa. La función que cumplió fue calcular los problemas detectados y determinar los requisitos funcionales y no funcionales. Los datos obtenidos nos ayudaron a la elaboración de las historias de usuario y diseño del sistema

4.3.2 Instrumentos

Los instrumentos de investigación que se utilizaron fueron clave para recopilar información valiosa que nos ayuda a entender la situación actual del control de rendimiento en la empresa Comexiger. Gracias a ellos, se pudieron identificar problemas en el registro manual de datos y las necesidades del área de trabajo. Estos instrumentos fueron diseñados teniendo en cuenta las técnicas de investigación aplicadas, con el objetivo de obtener información clara y útil para analizar el problema. Con estos datos, se pudo guiar el desarrollo del sistema propuesto.

4.3.2.1 Guion de entrevista

Se utilizó una guía de preguntas semiestructurada dirigida a los propietarios o responsables del área. Las preguntas permitieron conocer cómo se lleva a cabo el registro del rendimiento de los trabajadores y las dificultades que presenta el método manual.

4.3.2.2 Cuestionario

Se utilizó un cuestionario estructurado dirigido a los trabajadores de la empresa. Las preguntas se enfocaron en entender el método actual de registro, las dificultades que enfrentan a lo largo de la jornada y su opinión sobre el uso de la tecnología. Este instrumento facilitó la recopilación de información de manera ordenada y comparable, además de ayudar a identificar necesidades comunes entre los trabajadores. De esta forma, se logró obtener una visión más clara de los problemas en el proceso actual.

4.4 POBLACIÓN Y MUESTRA

4.4.1 Población

La población del estudio está compuesta por todas las personas que participan en el proceso operativo de postcosecha en la empresa Comexiger. Esto incluye a los propietarios y trabajadores que se encargan de las actividades relacionadas con el proceso de control de

rendimiento. En total, hay alrededor de 12 personas involucradas en este proceso, quienes interactúan de manera directa o indirecta con las actividades que se mejoraron gracias al desarrollo del sistema informático.

4.4.2 Muestra

Dado que la población total es bastante pequeña, con alrededor de 12 personas, se decidió trabajar con todos los integrantes, utilizando un muestreo no probabilístico de tipo intencional. Este enfoque se eligió porque los participantes fueron seleccionados según su conexión directa con el proceso operativo que se quería mejorar.

Al incluir a todos los trabajadores y responsables del área, se logró una evaluación completa del sistema en condiciones reales de uso. Trabajar con la población completa permitió obtener resultados más precisos, ya que no se hizo una selección parcial de sujetos, sino que se evaluó el impacto del sistema en todos los usuarios finales.

4.5 ARQUITECTURA DEL SISTEMA

4.5.1 Arquitectura MVC + API REST

Para poder continuar con el progreso de la plataforma informática de Comexiger, se ha optado por una arquitectura cliente-servidor que junta el patrón MVC (Modelo Vista Controlador) con una API RESTful como capa de servicios. Esta decisión fue tomada por la necesidad de integrar dos aplicaciones cliente diferentes (web y móvil) que comparten los mismos datos.

4.5.2 Componentes de la Arquitectura

4.5.2.1 Backend (Servidor) - API REST con patrón MVC

El backend implementará una API REST siguiendo dicha arquitectura, que será el núcleo central del sistema.

4.5.2.1.1 Modelo

El modelo muestra la representación de los datos que usará el sistema, sus mecanismos y la lógica de negocio el modelo incluirá:

- Entidades de datos
- Lógica de negocio

- Persistencia de datos

4.5.2.1.2 Vista

Hablando de una API REST, la vista se convierte en las respuestas JSON que el servidor envía a los clientes. REST sigue un modelo de arquitectura cliente-servidor, donde el cliente envía solicitudes y el servidor responde con los datos que se han solicitado los cuales son:

- Datos estructurados en formato JSON
- Códigos de estado HTTP
- Mensajes de error o confirmación

4.5.2.1.3 Controlador

Esta capa es la que se encuentra en el medio de Modelo y Vista, administrando la secuencia de información entre ellas, transformando y adecuando datos a sus necesidades y estos controladores manejarán:

- Endpoints REST
- Métodos HTTP: GET, POST, PUT, DELETE
- Validación de peticiones y respuestas en formato JSON
- Autenticación y autorización de usuarios

4.5.2.2 Frontend Web (Cliente Administradores)

La aplicación web para los encargados de administrar de Comexiger también seguirá el patrón MVC del lado del cliente:

- **Modelo:** Objetos que representan Trabajadores, Rendimiento, Disponibilidad.
- **Vista:** Interfaces HTML/CSS para visualizar tablas de control, reportes y estadísticas en tiempo real.
- **Controlador:** Lógica que consume la API REST mediante peticiones con WebSockets por cuanto este ayuda a procesar la respuesta y actualizar la vista dinámica y al instante.

- **Integración con Escáner 1:** Conectado a estación de control de calidad para registrar bonches procesados, sumando al rendimiento del trabajador y también a la disponibilidad de stock.
- **Escáner 2:** Vinculado a la estación de cuarto frio donde se almacenan los bonches para registrar bonches que salen disponibilidad, restando únicamente a ahí sin afectar a rendimiento.

4.5.2.3 Frontend Móvil (Cliente Trabajadores)

La aplicación móvil para los trabajadores consumirá la misma API REST:

- Pantallas para iniciar/finalizar jornada.
- Generador de etiquetas.
- Integración de Impresora: Conexión para imprimir etiquetas con código QR que incluyen información necesaria.
- Visualización de rendimiento personal y disponibilidad en tiempo real.

4.6 METODOLOGÍA XP

Después de haber comparado y analizado las diferentes metodologías presentadas en la tabla 2.2, se llegó a la conclusión que para que el desarrollo de nuestra plataforma informática destinad al rendimiento laboral de los trabajadores en Comexiger, se usaría la metodología XP. Esta metodología se complementa perfectamente con el objetivo de desarrollo iterativo y entrega continua de valor.

Con XP es posible trabajar en iteraciones cortas de 1-2 semanas, entregando versiones funcionales del sistema de manera continua. Las historias de usuario representan las funcionalidades que el cliente desea para el sistema, conlleva ventaja una vez que su implementación se realiza mediante programación en parejas, lo que garantiza código de mayor calidad y evita errores. El desarrollo manejado por pruebas permite que cada funcionalidad sea verificada automáticamente antes de su integración y al contrario la integración continua permite que los cambios sean agregados rápidamente tanto en la aplicación móvil como en la web.

La principal razón es porque el entorno de trabajo de Comexiger está en constante cambio: los procesos de clasificación y embonche pueden modificarse, los indicadores de rendimiento pueden requerir ajustes.

Por lo tanto, utilizar la metodología XP, es la mejor opción, debido a que, se adapta muy fácilmente a los cambios mediante feedback continuo y entregas frecuentes, es perfecta para el desarrollo del proyecto, asegurando así las funcionalidades actualizadas y sistema robusto que cumpla con las necesidades reales de los usuarios.

4.6.1 Roles

Los roles fueron identificados de acuerdo al cargo de cada trabajador y analizando quienes serán los que interactuarán con los sistemas propuestos; los roles hallados fueron los siguientes detallados en la tabla 4.1.

Tabla 4.1. Roles usados para el desarrollo.

ROL	DESCRIPCIÓN
Administrador/Dueños	La persona encargada de administrar el sistema se asegura de que todo funcione correctamente. Tiene la capacidad de crear y gestionar usuarios, hacer las configuraciones necesarias y supervisar las actividades como rendimiento y disponibilidad, que se llevan a cabo dentro del sistema. Además, cuenta con acceso a los reportes de rendimiento y disponibilidad, lo que le permite analizar la productividad y ayudar en la toma de decisiones.
Trabajador	El trabajador puede echar un vistazo a su propio rendimiento dentro del sistema. Así, podrá ver cuántos bonches ha registrado durante su jornada laboral, lo que le ayudará

ROL	DESCRIPCIÓN
	a tener un mejor control de su trabajo y a tener una idea más clara de su productividad.
Trabajador Área de control	Es la persona responsable de registrar la producción a lo largo de la jornada laboral. Para hacerlo, utiliza el escaneo de códigos QR de los bonches, lo que permite que la información de la producción se registre automáticamente en el sistema mientras se lleva a cabo el trabajo.
Empacador	Es la persona responsable de registrar la salida la cual surge después de escanear otro escáner.

4.7 HISTORIAS DE USUARIO

Se emplearon para recoger y organizar las necesidades reales de la empresa desde la perspectiva del usuario final es decir los dueños. El papel que jugó dentro de este proyecto fue, traducir los requerimientos expresados durante la entrevista en descripciones claras y resumidas, tales que permitieron definir qué debía hacer el sistema y qué valor aportaría a la empresa. Se plantea el siguiente diseño para el desarrollo de las historias de usuario, mostrado en la tabla 4.2.

Tabla 4.2.Diseño de Historias de usuario.

Historia de usuario		**ID HU**	
Puntos de historia		**Usuario**	
Prioridad		**Iteración asignada**	
Descripción			

Criterios de aceptación	
Definición de hecho	
Programador/a responsable:	

4.8 DIAGRAMA DE FLUJO

Para realizar el diagrama de flujo se utilizó la herramienta Draw.io la cual nos ayuda eficazmente a diseñar varios diagramas y cuenta con herramientas ideales para realizarlos.

Dicha herramienta se utilizó para representar gráficamente el proceso de funcionamiento de la plataforma informática. Su objetivo fue mostrar de forma precisa y clara el flujo de acciones que realizará el usuario, facilitando tanto una mejor compresión como representación del proceso general y su correcta funcionalidad.

4.9 PROTOTIPADO

Para el empezar el desarrollo es recomendable partir de prototipos que reflejen la idea de lo que se espera realizar para marcar el diseño y colores que se emplearán en la misma. Para el desarrollo de los prototipos necesarios tanto para la aplicación Web como para la móvil se utilizaron las siguientes herramientas descritas en la tabla 4.3.

Tabla 4.3. Herramientas de Prototipado.

Herramientas	Aplicación	Propósito
Figma	Web	Se usó como herramienta principal para el diseño visual del prototipo, preparación de apariencia de las pantallas antes de iniciar el desarrollo. Esto nos ayudó para conocer como el usuario interactuaría con el sistema, se usó también como un avance y validarlo con la empresa y realizar ajustes tempranos sin necesidad de programar.

Herramientas	Aplicación	Propósito
Flutter Flow	Móvil	Se utilizó esta herramienta para el desarrollo de un prototipo interactivo de la aplicación móvil mismo que simula el comportamiento real del sistema, ayudándonos a probar la navegación entre pantallas y flujo completo.

4.10 MÉTODO DE DESARROLLO DE LA APLICACIÓN WEB

La plataforma informática, diseñada para la recolección automatizada del rendimiento laboral conjuntamente con la disponibilidad de la comercializadora Comexiger fue desarrollada con las siguientes tecnologías para la aplicación Web las cuales se muestran en la tabla 4.4.

Tabla 4.4. Herramientas utilizadas para el sistema web.

Tecnología	Uso en el Sistema Web
Django	Se utilizó Django por su alta seguridad con protección automática contra CSRF, XSS, SQL Injection, su ORM incorporado potente, su estabilidad y estructura clara para proyectos grandes.
Python	Se utilizó Python por ser un lenguaje legible, versátil y con gran soporte para desarrollo web y científico, proporcionando un entorno moderno y escalable para el sistema.
SQLite	Se utilizó SQLite por su rendimiento excepcional en operaciones locales, cero configuraciones, ser ultraligera, confiable y rápida.
Visual Studio Code	Se utilizó VS Code por ser totalmente gratis, depurador integrado, integración nativa con Git y GitHub, y excelente compatibilidad con Python y Django.
Escáner TERA D5100	Se utilizó para automatizar y acelerar el proceso de captura de datos mediante códigos Qr.

Tecnología	Uso en el Sistema Web
WebSocket	Se utilizó WebSocket para la comunicación en tiempo real del sistema, permitiendo actualizaciones instantáneas sin necesidad de recargar la página.
Listener	Se utilizó para capturar los y procesar eventos de usuario en la interfaz del sistema web, permitiendo interactividad inmediata al responder a acciones como clics en botones, escaneo de códigos de barras y envío de formularios.
API REST	Se utilizó API REST para establecer la comunicación entre el frontend y backend del sistema, permitiendo operaciones CRUD sobre el inventario.
Daphne	Se utilizó Daphne para habilitar el soporte de WebSockets en Django, permitiendo la comunicación asíncrona y en tiempo real necesaria para las notificaciones instantáneas y actualizaciones automáticas del sistema de inventario.
GitHub	Se uso GitHub, para el versionamiento y trazabilidad de la aplicación web subiéndola a un repositorio para gestionar el código fuente este permitió registrar cada cambio realizado durante el desarrollo.

4.11 MÉTODO DE DESARROLLO DE LA APLICACIÓN MÓVIL

Las tecnologías usadas para realizar la aplicación Móvil fueron las detalladas a continuación en la tabla 4.5:

Tabla 4.5. Herramientas usadas para el desarrollo de la aplicación Móvil.

Tecnología	Uso en el Sistema Móvil
Flutter	Se utilizó Flutter para el desarrollo de la aplicación móvil, permitiendo crear una única base de código que funciona en Android.
Dart	Se utilizó Dart como lenguaje de programación principal de Flutter para la aplicación móvil.
Impresora Nelko PL70e-BT	Se utilizó para la impresión de etiquetas directamente desde la aplicación móvil mediante conexión Bluetooth. Elimina el gasto de tinta, permite impresión rápida de códigos de barras y etiquetas de inventario, facilitando la gestión de productos en tiempo real desde dispositivos móviles.
Visual Studio Code	Se utilizó VS Code por ser totalmente gratis, depurador integrado, integración nativa con Git y GitHub, y excelente compatibilidad con Dart y Flutter.
GitHub	Se uso GitHub, para el versionamiento y trazabilidad de la aplicación móvil subiéndola a un repositorio para gestionar el código fuente este permitió registrar cada cambio realizado durante el desarrollo.

4.12 PRUEBAS UNITARIAS

Las pruebas unitarias se aplicarán durante el desarrollo del sistema para verificar el correcto funcionamiento de cada módulo de manera individual. Su función principal es encontrar errores en etapas tempranas con esto podremos asegurar la fiabilidad del código y permitir todo cumpla con lo establecido.

Los campos describen lo siguiente:

ID: Código del caso de prueba

Módulo / Función: Nombre de la función o parte del sistema que se prueba

Caso de prueba: Que es lo que queremos comprobar

Datos de entrada: Datos usados para realizar la prueba

Resultado esperado: Lo que debería pasar

Resultado obtenido: Lo que sucedió realmente

¿Pasa? (Sí/No): "Si" en caso de que coincida con lo esperado y "No" si falla.

Para mantener el orden se utilizó el siguiente diseño de la tabla 4.6.

Tabla 4.6. Diseño para realizar las pruebas Unitarias.

ID	Módulo / Función	Caso de prueba	Datos de entrada	Resultado esperado	Resultado obtenido	¿Aceptado? (Sí/No)

4.13 PRUEBAS DE ACEPTACIÓN

Las pruebas de aceptación se realizaron al finalizarán al finalizar cada iteración del sistema, con el fin de verificar que las funcionalidades que se desarrollaron cumplen satisfactoriamente con las necesidades de la empresa. Estas pruebas permitirán validar el sistema desde el punto de vista del usuario, asegurando un producto final útil, comprensible y fiel a los requerimientos planteados.

Los campos describen lo siguiente:

ID: Código del escenario

Historia de usuario: Requisito o la historia de usuario

Escenario: Situación específica

Pasos a realizar: Acciones que realiza el usuario paso a paso

Resultado esperado: Lo que debería pasar

Resultado obtenido: Lo que sucedió realmente

¿Aceptado? (Si/No): "Si" en caso de que coincida con lo esperado y "No" si falla.

Para mantener el orden se utilizó el siguiente diseño de la tabla 4.7.

Tabla 4.7. Diseño para realizar las Pruebas de Aceptación.

ID	Historia de usuario	Escenario	Pasos a realizar		Resultado esperado	Resultado obtenido	¿Aceptado? (Sí/No)

4.14 CONTROL DE VERSIONES E INTEGRACIÓN CONTINUA

5 ANÁLISIS DE RESULTADOS

Se recopilo cierta información con respecto al proceso que realiza la empresa Comexiger antes de la exportación.

5.1 PROCESO POST-COSECHA EN COMEXIGER

5.1.1 Clasificación

Se deben tomar en cuenta muchos factores como la medida, enfermedades y si las flores están cerradas o abiertas llamado esto el punto de corte ideal mostrado en la figura 5.1.

Figura 5.1.Proceso de clasificación (Comexiger).

En la figura 5.2 lo clasifican por las medidas en las que pueden llegar a tomar los bonches.

Figura 5.2 Clasificación (Comexiger).

Estos procesos son mucho más que operaciones técnicas: cada rosa debe cumplir con requisitos precisos y los trabajadores tienen que tomar decisiones en segundos. Clasificar no es solo separar por color o tamaño, es entender los destinos del producto y anticipar las preferencias del cliente final.

5.1.2 Embonche

Los trabajadores encargados llevan las flores hasta su puesto de trabajo y ubican las flores como se ve en la figura 5.3.

Figura 5.3 Flores a su puesto de trabajo (Comexiger).

Como en la figura 5.4 se coloca el cartón y separadores que también son de cartón.

Figura 5.4 Colocación de cartones y separadores (Comexiger).

Después esta es la parte fundamental de este procedimiento. Se arman bonches de 20 o 25 rosas y la longitud depende del pedido realizado por del cliente. Aquí realizan un nuevo control de la calidad de la rosa como en la figura 5.5.

Figura 5.5 Armado y control de calidad de rosas (Comexiger).

Luego colocan los separadores entre las capas de rosas en la figura 5.6 se evidencia.

Figura 5.6 Colocación de separadores (Comexiger).

El siguiente paso consiste en la colocación de grapas para mantener cerrado firmemente el cartón y el bonche se vea uniforme visto en la figura 5.7.

Figura 5.7 Colocación de grapas (Comexiger).

En la figura 5.8 colocan una liga esto sirve para lograr uniformidad del bonche.

Figura 5.8 Colocación de ligas (Comexiger).

Identificación del bonche, en la figura 5.9 usan una pinza de color distintivo que identifica que mesa elaboró el ramo.

Figura 5.9 Identificar el bonche con pinza (Comexiger).

En particular, marca el momento en que se decide qué flores viajan y cuáles no. Es un proceso tan importante como delicado.

5.1.3 Proceso de Control

Es la medición del rendimiento operativo del personal ya sea manual o través de un sistema el cuál se quiere implementar en esta ocasión.

5.1.3.1 Manual

Este proceso requiere que la persona encargada recoja los bonches tal y como se ve en la figura 5.10 en los que vienen las pinzas o identificadores cada mesa.

Figura 5.10 Recolección de bonches con pinza o identificadores (Comexiger).

En la figura 5.11 estas pinzas son recogidas y colocadas en un lugar en el que sean visibles para su respectivo conteo cuando el espacio se acabe.

Figura 5.11 Colocación de pinzas en un lugar visible (Comexiger).

Luego en la figura 5.12 depositan el bonche en la banda para poder cortar en la máquina según la medida solicitada.

Figura 5.12 Colocación de bonche en banda para corte según su medida (Comexiger).

Una vez las cuente serán anotadas en un pizarrón dependiendo el número de mesa como en la figura 5.13 y por último las devolverá para que el ciclo empiece otra vez hasta que las labores de los trabajadores concluyan.

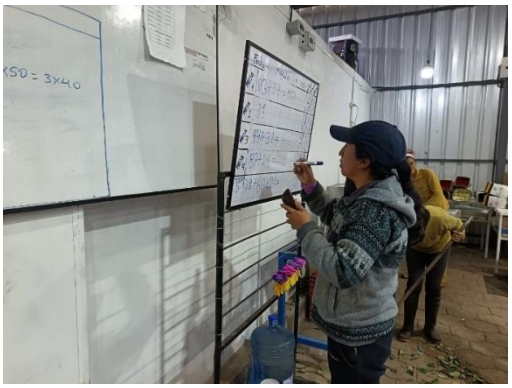


Figura 5.13 Conteo de rendimiento (Comexiger).

Estos son transferidos a hojas de cálculo Excel para que, con base en estos datos diarios de rendimiento, se determina la remuneración correspondiente a cada trabajador.

5.2 FASE DE PLANIFICACIÓN

En esta etapa, se llevaron a cabo reuniones de manera presencial con los dueños de la empresa Comexiger posteriormente se observó tanto el trabajo como el procesamiento que realizan los trabajadores, se pudo observar que existen procesos como los de control que no se encuentran automatizados, problemas como este se evidencian en cada historia de usuario.

5.2.1 Resultado de la entrevista

Nombre del entrevistado/a: Eveleni Franco

Lugar: Comercializadora Comexiger

1.- ¿Cómo se realiza actualmente el registro de rendimiento de los trabajadores?

Lo hacemos de manera manual. Los trabajadores colocan las pinzas en los bonches, luego cuentan y anotan en una pizarra. Al final del día recolectamos la información con una foto.

2.- ¿Qué problemas se presentan con el método manual de conteo y registro?

Se cometen errores y en ocasiones se olvidan de tomar la foto esto causa una pérdida de datos o en algunos casos no coinciden los conteos con lo que realmente se trabajó y lleva mucho tiempo a causa de esto tiene que sacar las pinzas colocarlas en un lugar y luego al llenarse el lugar se deben contar, se sacan y se devuelve a la mesa a la que pertenecen.

3.- ¿Con que frecuencia ocurren errores en los datos registrados?

No es muy seguido, pero si ocurre con un poco de frecuencia, sobre todo cuando se acumulan los bonches en el lugar de control.

4.- ¿Cuánto tiempo se invierte diariamente en contar y transcribir la información?

De una 1 hora en adelante.

5.- ¿Qué información considera más importante para controlar el rendimiento del personal?

Saber cuántos bonches hace cada mesa durante toda su jornada, a qué hora iniciaron su jornada, y si hubo algún retraso o problema durante el día.

6.- ¿Cómo influye el rendimiento en el pago de los trabajadores**?**

A más bonches más rendimiento y el pago es mayor

7.- ¿Qué dificultades tiene al tomar decisiones con los datos actuales**?**

No siempre se pueden se pueden tomar pronto porque se requiere de pasar los datos y se demora.

8.- ¿Le gustaría visualizar el rendimiento en tiempo real? ¿Por qué**?**

Me permitiría saber cómo estuvo el día de los trabajadores porque no tendría que esperar a que lo contabilicen.

9.- ¿Qué funciones considera indispensables en un sistema informático**?**

Registro automático de los bonches que sería el rendimiento y la disponibilidad, identificación por mesa, reportes diarios y mensuales visualizando en tiempo real la información.

10.- ¿Cree que una aplicación móvil ayudaría a los trabajadores a mejorar su rendimiento en su jornada**?**

Si porque evitarían confusiones y verían su rendimiento en tiempo real y sabrían cómo mejorar al día siguiente y evitar el conteo a mano de la disponibilidad.

5.2.1.1 Análisis

Según lo que se comentó en la entrevista con la empresa Comexiger, en la actualidad, el rendimiento de los trabajadores se registra de forma manual, utilizando pinzas y una pizarra. Este método puede dar lugar a errores en el conteo y, en ocasiones, a la pérdida de información, además de que consume bastante tiempo para anotar los datos. También se destacó que estos registros son cruciales, ya que afectan directamente el pago de los trabajadores. La persona entrevistada señaló que contar con un sistema informático facilitaría el registro de la información de manera más rápida y organizada. Además, permitiría ver el rendimiento en tiempo real y mejorar el control del trabajo.

5.2.2 Resultado de la encuesta

La encuesta se realizó a 10 personas dentro del personal de la empresa.

Lugar: Comercializadora Comexiger

2.- ¿Ha tenido errores al contar o registrar los bonches**?**

- Nunca
- A veces
- Frecuentemente

Los resultados se ven en la tabla 5.1.

Tabla 5.1. Resultados de la pregunta 2.

Opciones	**Total de Respuestas**
Nunca	**2**
A veces	**2**
Frecuentemente	**6**

Las estadísticas se encuentran en la figura 5.14.

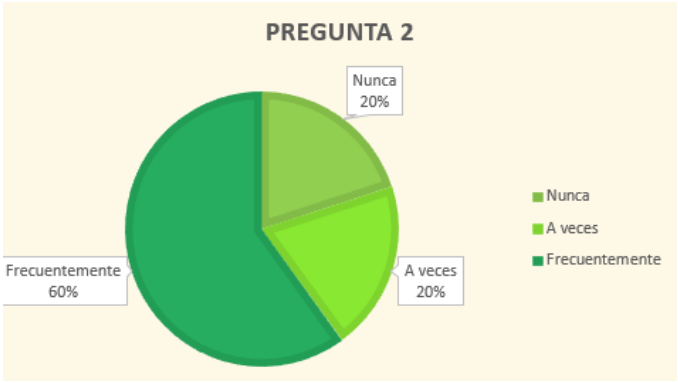


Figura 5.14.Estadística de la pregunta 2.

Según lo que dicen las personas que participaron en la encuesta, 6 de cada 10 personas (60%) dicen que a menudo han cometido errores al contar o registrar los bonches, mientras que 2 personas (20%) dicen que a veces y 2 personas (20%) afirman que nunca han tenido errores.

5.- ¿Le gustaría conocer su rendimiento en tiempo real?

- Sí
- No

Los resultados se ven en la tabla 5.2.

Tabla 5.2. Resultados de la pregunta 5.

Opciones	**Total de Respuestas**
Sí	**9**
No	**1**

Las estadísticas se encuentran en la figura 5.15.

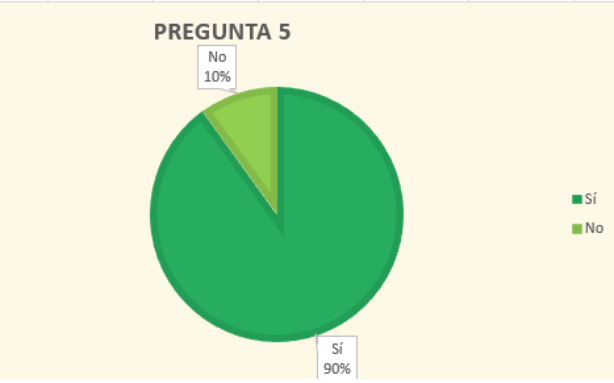


Figura 5.15.Estadística de la pregunta 5.

Según los resultados de la encuesta, 9 de 10 encuestados (90%) dicen que sí quieren saber su rendimiento en tiempo real, mientras que 1 persona (10%) dice que no.

Esto muestra que la mayoría del personal está interesado en tener nuestra plataforma de informática, que les permitirá ver su rendimiento laboral de inmediato y con precisión.

9.- ¿Cree que un sistema digital reduciría errores en su pago?

- Sí
- No

Los resultados se ven en la tabla 5.3.

Tabla 5.3.Resultados de la pregunta 9.

Opciones	**Total de Respuestas**
Si	7
No	3

Las estadísticas se encuentran en la figura 5.16.

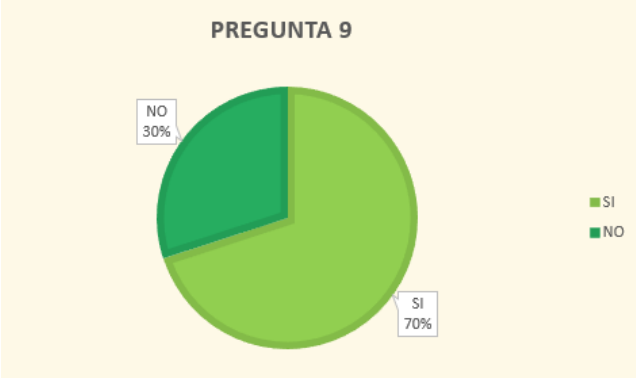


Figura 5.16. Estadística de la pregunta 9.

Según una encuesta reciente, la mayoría de los encuestados, un total de 7 de cada 10, piensan que un sistema digital ayudaría a reducir los errores en los pagos. Por otro lado, 3 de cada 10 personas encuestadas creen que no.

Estos resultados apoyan la puesta en marcha de nuestra plataforma de informática. Gracias a ella, esperamos mejorar la precisión en el registro y cálculo del rendimiento laboral.

5.2.2.1 Análisis

Para cerrar el análisis de la encuesta, es importante seleccionar las preguntas que realmente reflejan la necesidad y la aceptación del sistema. Las que más destacan para respaldar la viabilidad son:

Pregunta 2: errores al contar o registrar bonches.

Pregunta 5: interés en conocer el rendimiento en tiempo real.

Pregunta 9: la percepción de que un sistema digital podría disminuir los errores en los pagos.

Estas tres preguntas evidencian el problema actual, la aceptación de la solución y el beneficio esperado.

Los resultados de la encuesta muestran que implementar el sistema en la empresa es totalmente viable. Por un lado, varios empleados comentaron que han tenido problemas al contar o registrar los bonches, lo que indica que el método manual no siempre es confiable. Además, la mayoría expresó su deseo de conocer su rendimiento en tiempo real, lo que demuestra un interés en utilizar una herramienta tecnológica que les brinde un mejor control sobre su trabajo. Por

último, muchos piensan que un sistema digital podría ayudar a reducir los errores en los pagos, lo que aumentaría la confianza en el proceso. En resumen, estos resultados evidencian la necesidad de mejorar el registro actual y que los trabajadores ven con buenos ojos la implementación de la plataforma informática.

5.2.3 Historias de Usuario

Tabla 5.4. HU-M01

Historia de usuario		**ID HU**	**HU-M01**
Puntos de historia	8	**Usuario**	Trabajador
Prioridad	Alta	**Iteración asignada**	1
Descripción	Como trabajador, quiero iniciar sesión en la app móvil con mi usuario y contraseña, para poder acceder a mis funciones de registro y visualización.		
Criterios de aceptación	• La app solicita usuario y contraseña válidos. • Los datos incorrectos muestran un mensaje de error.		
Definición de hecho	El trabajador puede iniciar sesión correctamente y acceder solo a sus funciones.		
Programador/a responsable:	Anthony Monta		

Tabla 5.5. HU-M02

Historia de usuario		**ID HU**	**HU-M02**
Puntos de historia	5	**Usuario**	Trabajador
Prioridad	Alta	**Iteración asignada**	1

Descripción	Como trabajador, quiero registrar el inicio de mi jornada desde la app, para que quede registrado el horario de entrada automáticamente.
Criterios de aceptación	• Botón de "Iniciar jornada" funcional. • La hora de inicio se guarda en la base de datos y se refleja en la web. • No permite iniciar una jornada si ya está activa.
Definición de hecho	La hora registrada coincide con la hora real del sistema y se muestra al administrador.
Programador/a responsable:	Anthony Monta

Tabla 5.6.HU-M03

Historia de usuario		**ID HU**	**HU-M03**
Puntos de historia	5	**Usuario**	Trabajador
Prioridad	Alta	**Iteración asignada**	1
Descripción	Como trabajador, quiero registrar el fin de mi jornada, para que se registre automáticamente la hora de salida y se pueda calcular mi remuneración.		
Criterios de aceptación	• Botón de "Finalizar jornada" funcional. • La hora de fin se guarda automáticamente y se refleja en la web. • Impide cerrar jornada si no se ha iniciado.		
Definición de hecho	El sistema registra correctamente la jornada completa de cada trabajador.		

Programador/a responsable:	Anthony Monta

Tabla 5.7.HU-M04

Historia de usuario		**ID HU**	**HU-M04**
Puntos de historia	5	**Usuario**	Trabajador
Prioridad	Alta	**Iteración asignada**	1
Descripción	Como trabajador, quiero ver mi rendimiento y el de mi mesa en tiempo real, para conocer mi productividad durante la jornada.		
Criterios de aceptación	• Se muestran los bonches procesados por mesa. • Se indica el total de bonches de la jornada.		
Definición de hecho	La visualización coincide con los registros en la web.		
Programador/a responsable:	Anthony Monta		

Tabla 5.8. HU-M05

Historia de usuario		**ID HU**	**HU-M05**
Puntos de historia	5	**Usuario**	Trabajador
Prioridad	Alta	**Iteración asignada**	1
Descripción	Como trabajador, quiero imprimir etiquetas para cada bonche desde la app, para identificar correctamente los bonches y evitar errores en el conteo.		

Criterios de aceptación	• Se pueden generar etiquetas con la medida y numero de mesa.
Definición de hecho	Cada etiqueta generada corresponde a un ID único el cuál impedirá repetición de los mismos.
Programador/a responsable:	Anthony Monta

Las demás historias de usuario se pueden ver en el Anexo 2.

5.2.4 Diagrama de flujo

El flujo de las aplicaciones con los implementos necesarios se propuso de la siguiente manera.

5.2.4.1 Inicio de sesión en la aplicación Móvil

Rol: Administrador

Los administradores realizan el proceso de iniciar sesión con sus credenciales se visualiza los casos en el que se validan las credenciales cuando es si y no he ingresan al sistema como se puede ver en el flujo de la figura 5.17 esto dependiendo el flujo en el caso de que sí.

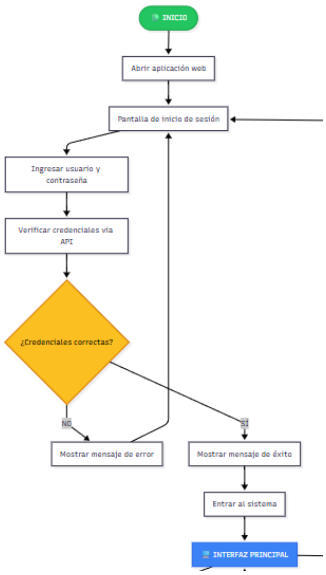


Figura 5.17. Diagrama de la Aplicación Web Inicio de sesión.

Una vez en la interfaz principal hay opciones como temas y salir que acabaría con todo el flujo; como se evidencia en el diagrama de la figura 5.18 y módulo de Administración.

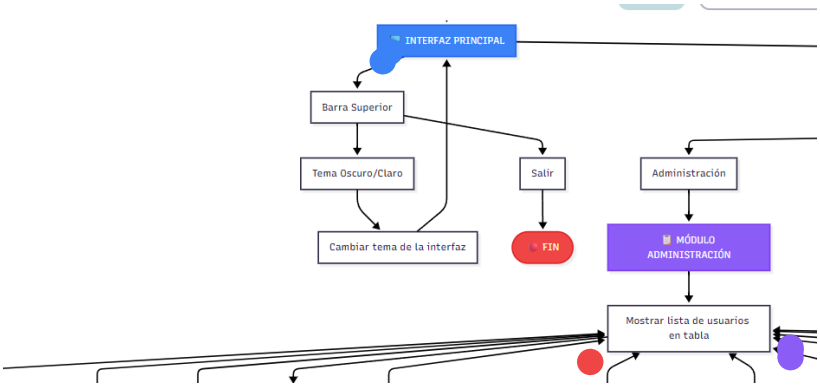


Figura 5.18.Diagrama de la Aplicación Web Módulo de Administración.

5.2.4.1 Registro de usuarios en la Aplicación Móvil

Rol: Administrador

Selecciona Administración en el cual muestra el listado de los usuarios que existen y se pueden realizar los procesos como agregar en el que la acción del usuario es Nuevo Usuario aquí se abre la interfaz de registros y se llenan los campos aquí entra la decisión si los campos son válidos y están completos va a si y se guardan caso contrario muestra que faltan datos o que debe corregirlos dicho puede ser visualizado en la figura 5.19 y 5.20.

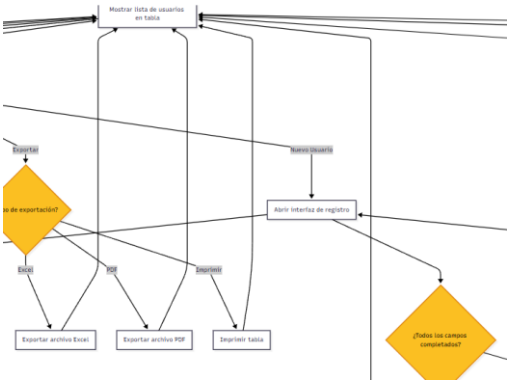


Figura 5.19.Diagrama de la Aplicación Web Registro de Usuario.

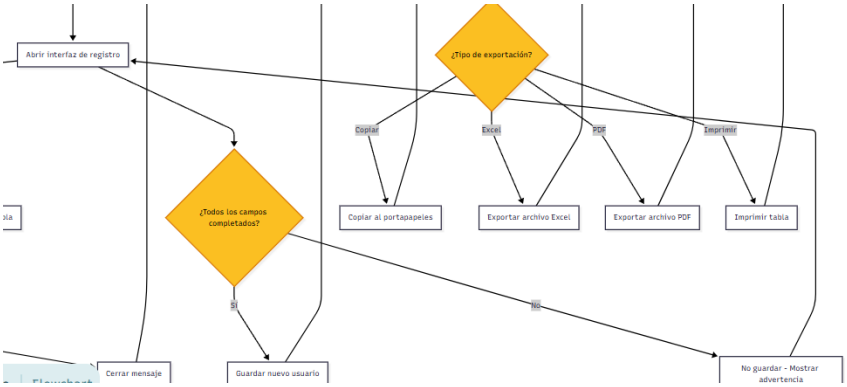


Figura 5.20.Diagrama de la Aplicación Web Registro y validación de usuario.

Otra acción que se le permite a los administradores es la de exportar en la que se elige el tipo de exportación de su preferencia visto en la figura 5.21.

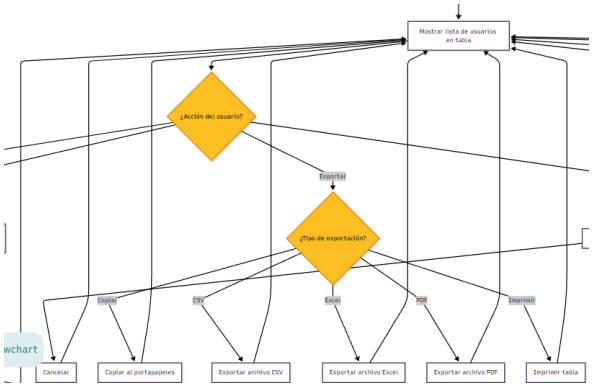


Figura 5.21.Diagrama de la Aplicación Web selección de Exportación.

Por acción del usuario administrador en el caso de eliminar muestra un mensaje de confirmación en el cual entrará en la decisión de confirmarlo o cancelarlo como se ve en la figura 5.22.

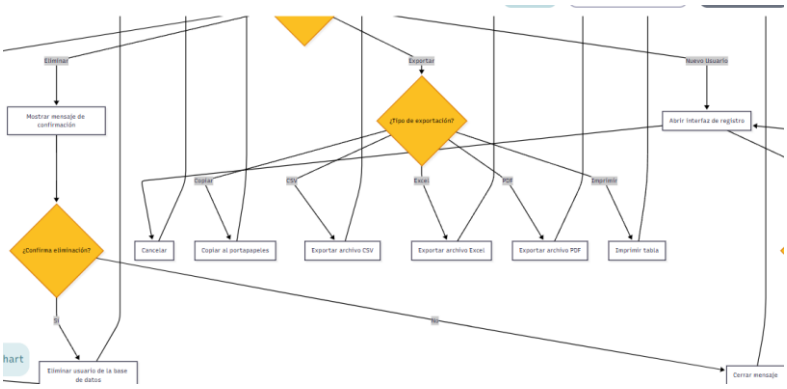


Figura 5.22.Diagrama de la Aplicación Web para eliminar un Usuario.

Cuando el usuario administrador decide editar, el sistema presenta un formulario que muestra la información actual del registro seleccionado, lo que le permite hacer las modificaciones necesarias. Después de realizar los cambios, el usuario tiene la opción de guardar la información actualizada o cancelar la operación, según lo que considere más adecuado, tal como se ilustra en la figura 5.23.

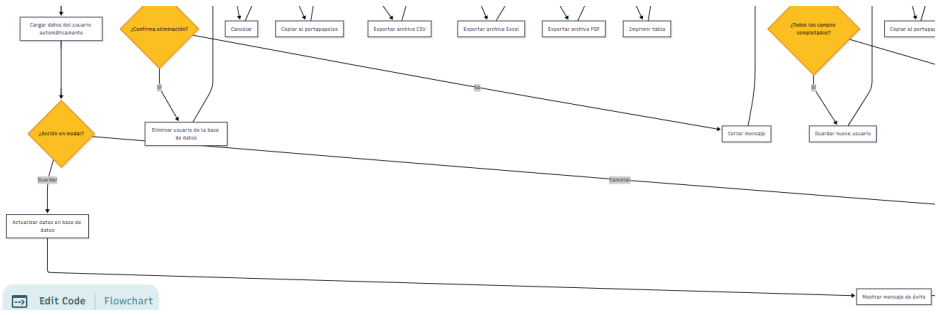


Figura 5.23. Diagrama de la Aplicación Web para editar el Usuario.

5.2.4.2 Inicio de sesión en la aplicación Móvil

Rol: Trabajador

El proceso comienza cuando el trabajador abre la aplicación móvil e ingresa sus credenciales, es decir, su usuario y contraseña, en la pantalla de inicio de sesión. Si las credenciales no son válidas, aparece un mensaje de error que le indica al trabajador que lo intente de nuevo. Si son correctas, el sistema le da acceso y lo lleva al menú principal de la aplicación dicho flujo se refleja en la figura 5.24.

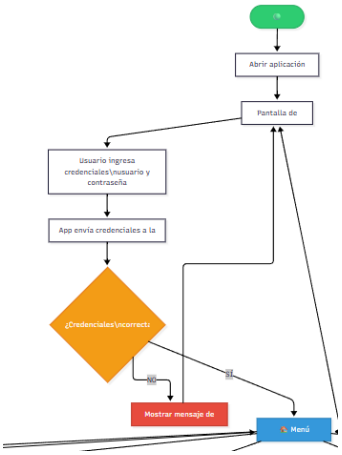


Figura 5.24. Diagrama de la Aplicación Móvil del Inicio de Sesión.

5.2.4.2.1 Inicio de Jornada

Rol: Trabajador

El proceso comienza cuando el trabajador registra el inicio de su jornada laboral a través de la aplicación móvil. Esta acción se guarda en el sistema, a lo largo de la jornada, el sistema sigue registrando la actividad del trabajador. Finalmente, al concluir sus tareas, el trabajador marca el final de la jornada tal como se muestra en la figura 5.25, lo que permite guardar la hora de salida y cerrar el registro del día en el sistema.

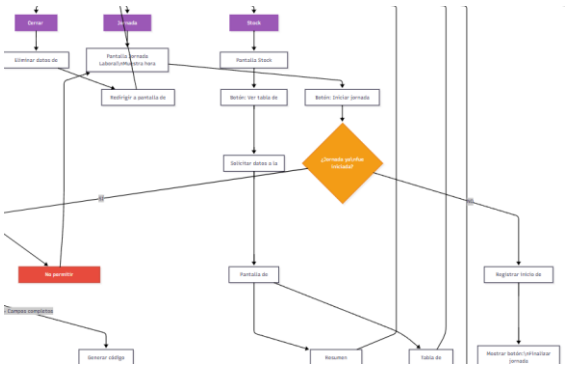


Figura 5.25. Diagrama de la Aplicación Móvil del inicio de Jornada.

5.2.4.2.2 Impresión de etiquetas

Rol: Trabajador Área de control

Para la impresión de etiquetas empieza cuando el usuario accede al módulo de Etiquetas, después de ingresar toda la información necesaria que pide, el sistema genera automáticamente las etiquetas para asi enviarlas a la impresora aquí entran a la decisión si los datos estan completos se envian por otro lado no tal como se evidencia en la figura 5.26.

Figura 5.26.Diagrama de la Aplicación Móvil para la impresión de Etiquetas.

5.2.4.3 Modo de escaneo

Rol: Trabajador del Área de control

Una vez llegado a este punto estas etiquetas se pegan en el bonche y serán escaneadas por el escáner 1 y esta información irá sumando tanto a disponibilidad donde corresponda la medida con variedad y a rendimiento según donde corresponda la mesa, fecha.

Rol: Empacador

De la misma manera con el escáner 2 que restará con la diferencia que solo lo hará en disponibilidad. Transcurrido este proceso se podrá visualizar en tiempo real y consultarse en ambas aplicaciones tanto web como móvil.

5.2.4.4 Consulta de rendimiento y disponibilidad

5.2.4.4.1 Web

- **Rendimiento**

Rol: Administrador

Una vez con datos este módulo se puede consultar; se ingresa a la aplicación móvil. En la figura 5.27 se evidencia que seleccionar este módulo ya implica que debe iniciar jornada como anteriormente se mencionó. Aquí se visualiza el rendimiento que se sigue procesando.

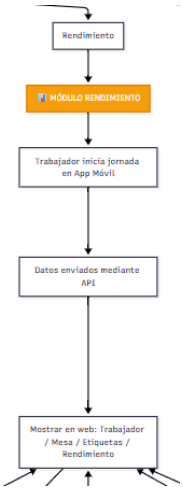


Figura 5.27.Diagrama de la Aplicación Web para la consulta de rendimiento.

Aquí se toma una decisión correspondiente a la acción del usuario ya sea editar rendimiento que abre un modal en el que conlleva a tomar la decisión de guardar o cancelar. Para eliminar solo se necesita de la decisión de confirmar o cancelar como se ve en la figura 5.28.

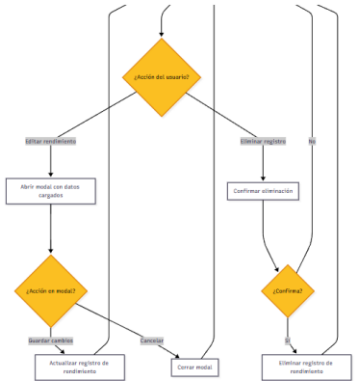


Figura 5.28. Diagrama de la Aplicación Web para editar y eliminar el rendimiento.

- **Disponibilidad**

Rol: Administrador

El proceso para verificar la disponibilidad comienza cuando el usuario entra a la aplicación web y accede al módulo correspondiente. En esta parte, el sistema presenta la información registrada sobre los bonches disponibles, basada en los datos que tiene almacenados en la base de datos. Aquí se edita la variedad por la medida y también se eliminan. También el flujo es el mismo para la exportación y tomando decisiones en cada una de las acciones realizada por el usuario este proceso se puede visualizar en las figuras 5.29 y 5.30.

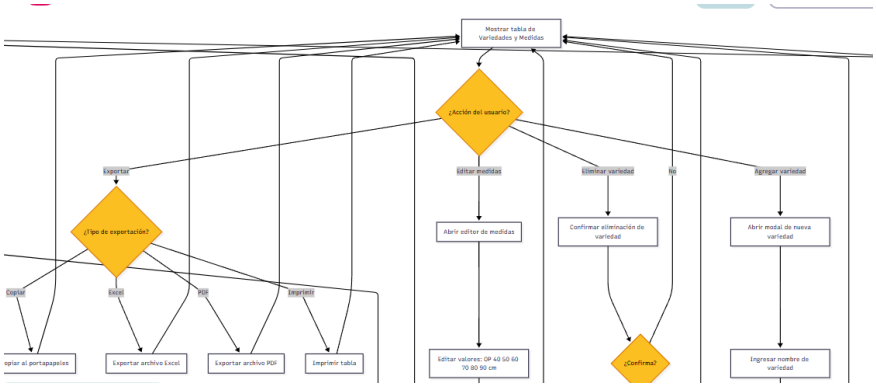


Figura 5.29.Diagrama de la Aplicación Web para visualizar la Disponibilidad.

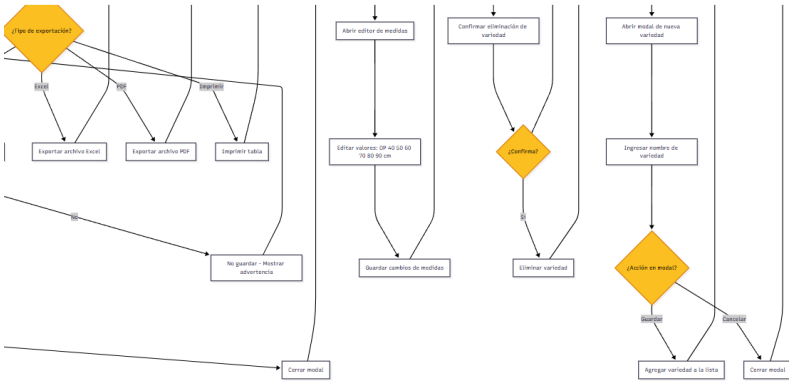


Figura 5.30.Diagrama de la Aplicación Web para editar y eliminar la Disponibilidad.

5.2.4.4.2 Móvil

- **Rendimiento**

Rol: Trabajador

Dentro del menú selecciona el apartado de rendimiento y este se filtra por mesa el cual pertenece se mostrarán los datos simplificados el proceso es el que se muestra en la figura 5.31.

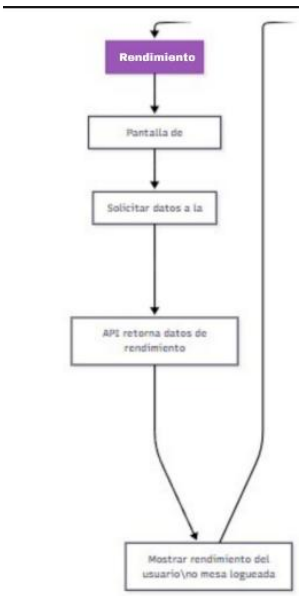


Figura 5.31.Diagrama de la Aplicación Móvil para visualizar el Rendimiento.

- **Disponibilidad**

Rol: Trabajador

Dentro del menú, elige la opción de disponibilidad, donde la información se muestra. En esta sección, los datos se presentan de manera simplificada para que sea más fácil consultarlos y hacer un seguimiento. El proceso se puede evidenciar en la figura 5.32.

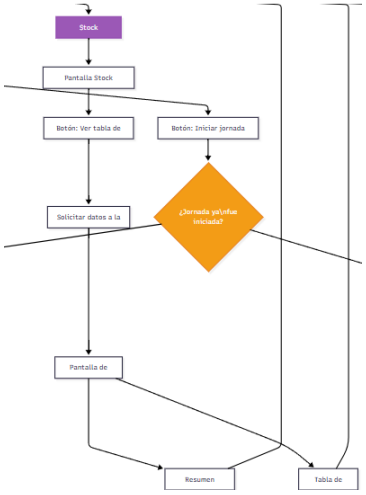


Figura 5.32.Diagrama de la Aplicación Móvil para visualizar la Disponibilidad.

5.2.4.5 Diagrama de arquitectura

En la figura 5.33 se muestra la arquitectura del sistema se utilizó.

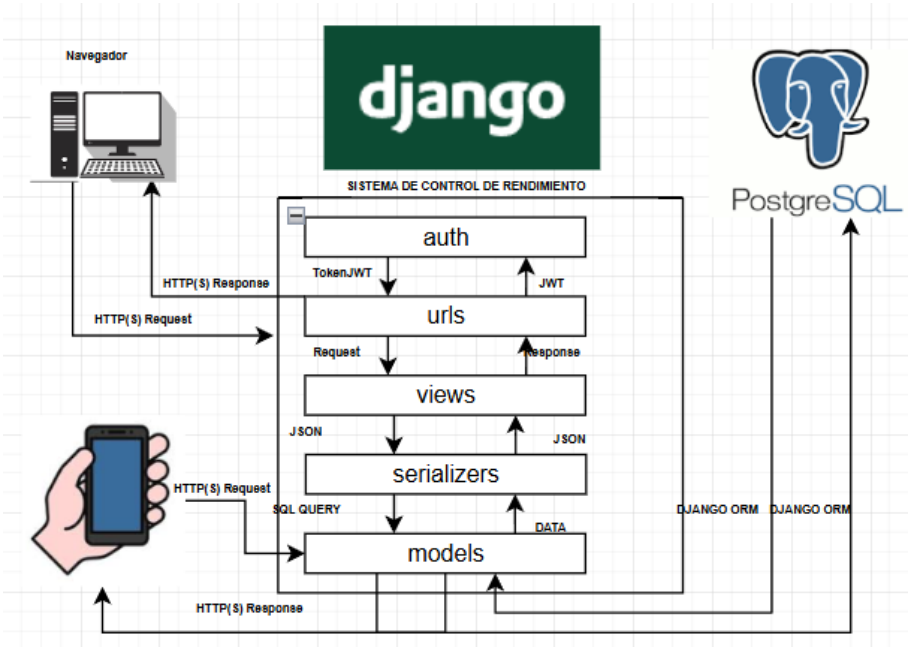


Figura 5.33.Arquitectura del sitema de control de rendimiento

5.3 FASE DE DISEÑO

5.3.1 Esquema de base de datos

Esquema de bases de datos del sistema web y móvil se muestra en la Figura 5.34 el cual comparte la misma base de datos, lo que permite la sincronización entre ambas plataformas, permitiendo que los datos se envíen bidireccionalmente de web a móvil y de móvil a web.

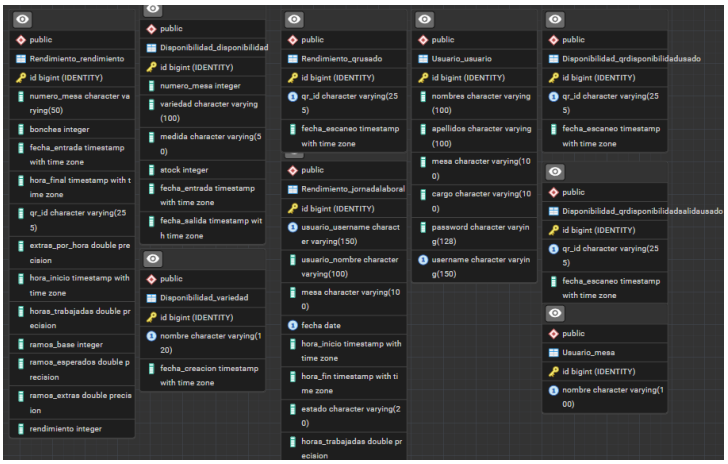


Figura 5.34.Esquema de base de datos de PostgreSQL.

5.3.2 Diseño de la aplicación web

5.3.2.1 Prototipado

Esta interfaz mostrada en la Figura 5.35 representa la funcionalidad del inicio de sesion que se implementará en el sistema.

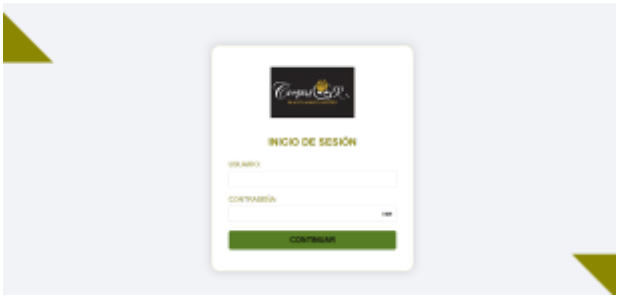


Figura 5.35.Prototipo de inicio de sesión.

Esta interfaz de la Figura 5.36 representa la funcionalidad rendimiento de los trabajadores que se implementará en el sistema.

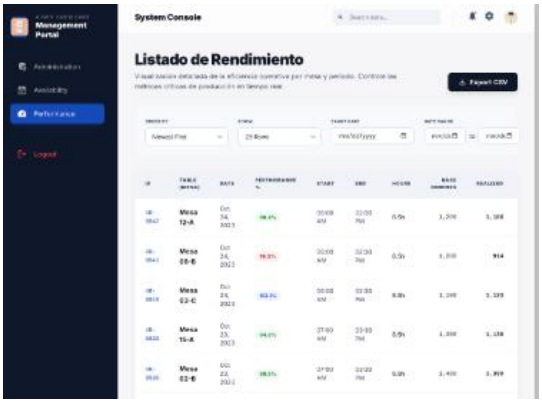


Figura 5.36. Prototipo del módulo de rendimiento.

Esta interfaz de la Figura 5.37 representa la funcionalidad de la disponibilidad que se implementará en el sistema.

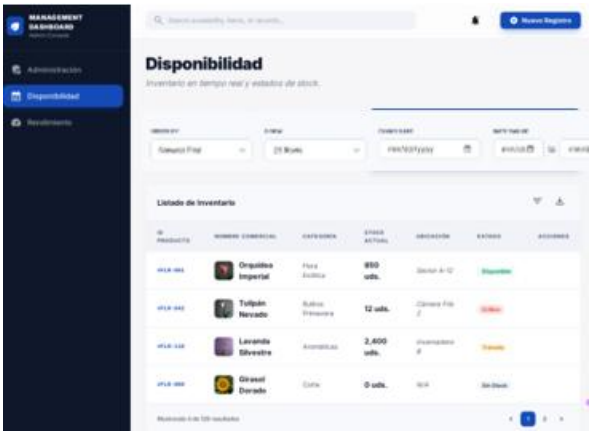


Figura 5.5.37. Prototipo del módulo de disponibilidad.

Esta interfaz de la Figura 5.38 representa la funcionalidad los usuarios registrados que se implementará en el sistema.

Figura 5.38.Prototipo del módulo de administración de Usuarios.

Esta interfaz de la Figura 5.39 representa la funcionalidad registrar un usuario nuevo que se implementará en el sistema.

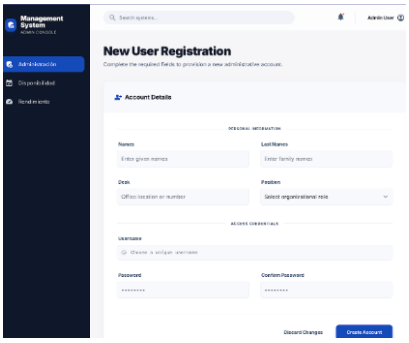


Figura 5.39.Prototipo del formulario de registro de usuarios.

El modal de la Figura 5.40 representa la funcionalidad agregar una nueva mesa que se implementará en el sistema.

Figura 5.40.Prototipo del formulario para agregar una mesa.

5.3.3 Diseño de la aplicación móvil

5.3.3.1 Prototipado

Esta interfaz de la Figura 5.41 representa la funcionalidad del inicio de sesion que se implementará en el app movil.

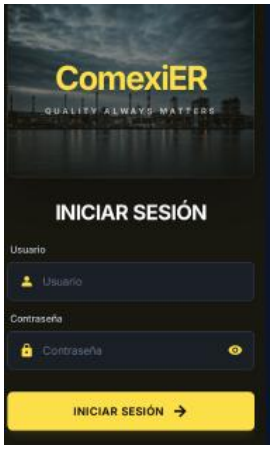


Figura 5.41.Prototipo para el inicio de sesión de los trabajadores.

Esta interfaz de la Figura 5.42 representa la funcionalidad del menu principal que se implementará en el app movil.

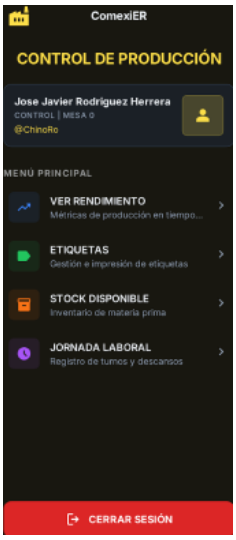


Figura 5.42.Prototipo del menú.

Esta interfaz de la Figura 5.43 representa la funcionalidad del rendimiento que se implementará en el app movil.

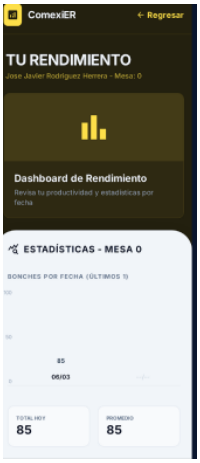


Figura 5.43.Prototipo del módulo de visualización de rendimiento de cada trabajador.

Esta interfaz de la Figura 5.44 representa la funcionalidad de creacion de Qrs a imprimir que se implementará en el app movil.

Figura 5.44.Prototipo del formulario para la creación de etiquetas.

Esta interfaz de la Figura 5.45 representa la funcionalidad de inicio y fin de la jornada laboral que se implementará en el app movil.

Figura 5.45.Prototipo para el inicio y fin de la jornada laboral

Esta interfaz de la Figura 5.46 representa la funcionalidad vizualizar la disponibilidad que se implementará en el app movil.

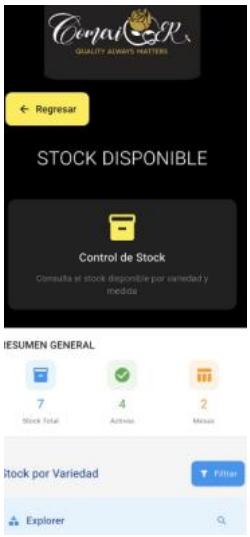


Figura 5.46.Prototipo para la visualización de disponibilidad.

5.4 FASE DE CODIFICACIÓN O DESARROLLO

5.4.1 Aplicación web

5.4.1.1 Estructura de vistas y modelos

En esta sección de la Figura 5.47 se definió la organización del proyecto por aplicaciones, separando responsabilidades por módulos. Los modelos se diseñaron para representar entidades

del sistema.

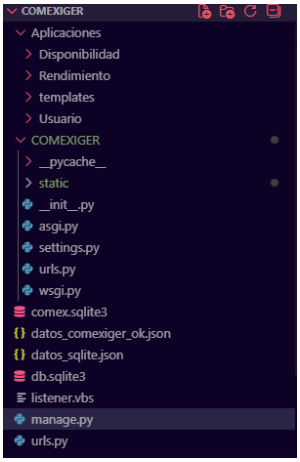


Figura 5.47.Estructura de aplicaciones con sus vistas y modelos.

5.4.1.1.1 Disponibilidad

Representa el estado y control de bonches realizados disponibles. Su función principal es registrar información relacionada con la cantidad, de cada variedad en un momento determinado, permitiendo así al sistema reconocer si este puede ser utilizado o asignado en para alguna otra venta su estructura se presenta en la Figura 5.48.

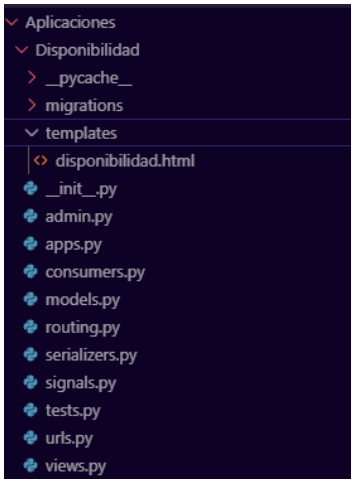


Figura 5.48. Estructura de la aplicación Disponibilidad en Django.

- **Modelo de Disponibilidad**

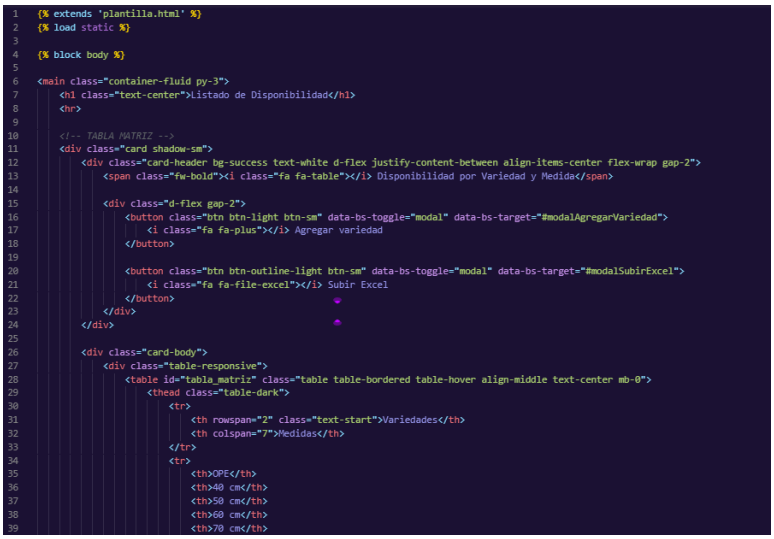
En la Figura 5.49 se puede visualizar el modelo utilizado en el Framework Django el cual será utilizado para todo el desarrollo del proyecto.

```python
from django.db import models
from django.utils import timezone

class QRDisponibilidadUsado(models.Model):
    qr_id = models.CharField(max_length=255, unique=True)
    fecha_escaneo = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.qr_id

class QRDisponibilidadSalidaUsado(models.Model):
    qr_id = models.CharField(max_length=255, unique=True)
    fecha_escaneo = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.qr_id

class Disponibilidad(models.Model):
    numero_mesa = models.PositiveIntegerField(verbose_name="Número de mesa")
    variedad = models.CharField(max_length=100)
    medida = models.CharField(max_length=50)

    stock = models.PositiveIntegerField(default=0)

    fecha_entrada = models.DateTimeField()
    fecha_salida = models.DateTimeField(null=True, blank=True)

    def __str__(self):
        return f"Mesa {self.numero_mesa} - {self.variedad} ({self.stock})"
class Variedad(models.Model):
    nombre = models.CharField(max_length=120, unique=True)

    fecha_creacion = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.nombre
```

Figura 5.49.Modelo Disponibilidad en Django.

- **Template de Disponibilidad**

En la Figura 5.50 se puede visualizar el template de disponibilidad utilizado en el Framework Django el cual será utilizado para todo el desarrollo del proyecto.



```
{% extends 'plantilla.html' %}
{% load static %}

{% block body %}

<main class="container-fluid py-3">
    <h1 class="text-center">Listado de Disponibilidad</h1>
    <hr>

    <!-- TABLA MATRIZ -->
    <div class="card shadow-sm">
        <div class="card-header bg-success text-white d-flex justify-content-between align-items-center flex-wrap gap-2">
            <span class="fw-bold"><i class="fa fa-table"></i> Disponibilidad por Variedad y Medida</span>

            <div class="d-flex gap-2">
                <button class="btn btn-light btn-sm" data-bs-toggle="modal" data-bs-target="#modalAgregarVariedad">
                    <i class="fa fa-plus"></i> Agregar variedad
                </button>

                <button class="btn btn-outline-light btn-sm" data-bs-toggle="modal" data-bs-target="#modalSubirExcel">
                    <i class="fa fa-file-excel"></i> Subir Excel
                </button>
            </div>
        </div>

        <div class="card-body">
            <div class="table-responsive">
                <table id="tabla_matriz" class="table table-bordered table-hover align-middle text-center mb-0">
                    <thead class="table-dark">
                        <tr>
                            <th rowspan="2" class="text-start">Variedades</th>
                            <th colspan="7">Medidas</th>
                        </tr>
                        <tr>
                            <th>OPE</th>
                            <th>40 cm</th>
                            <th>50 cm</th>
                            <th>60 cm</th>
                            <th>70 cm</th>
```

Figura 5.50.Fragmento de código del Template de Disponibilidad.

En la Figura 5.51 se puede visualizar el resultado del frontend de disponibilidad utilizado en el Framework Django el cual será utilizado para todo el desarrollo del proyecto.

Figura 5.51.Panel de Gestión de Disponibilidad.

5.4.1.1.2 Rendimiento

Se utiliza para registrar y analizar el desempeño de los procesos realizados dentro del sistema. Permite llevar un control de los resultados obtenidos en un período determinado, facilitando la evaluación del funcionamiento general y apoyando la toma de decisiones, su estructura se representa en la Figura 5.52.

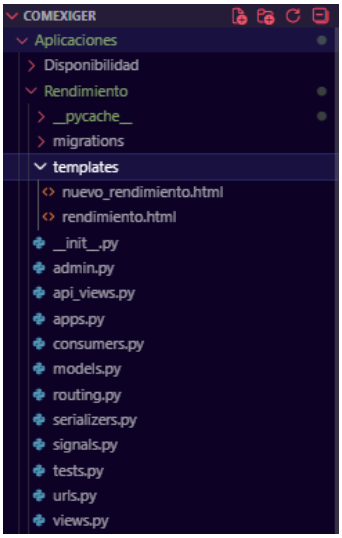


Figura 5.52.Estructura de la aplicación Rendimiento.

- **Modelo Rendimiento**

En la Figura 5.53 se puede visualizar el modelo de rendimiento utilizado en el Framework Django el cual será utilizado para todo el desarrollo del proyecto.

Figura 5.53.Fragmento de código del Modelo Rendimiento.

- **Template de Rendimiento**

En la Figura 5.54 se puede visualizar el template de rendimiento utilizado en el Framework Django el cual será utilizado para todo el desarrollo del proyecto.

Figura 5.54.Fragmento de código del template de Rendimiento.

En la Figura 5.55 se puede visualizar el resultado del frontend de disponibilidad utilizado en el Framework Django el cual será utilizado para todo el desarrollo del proyecto.

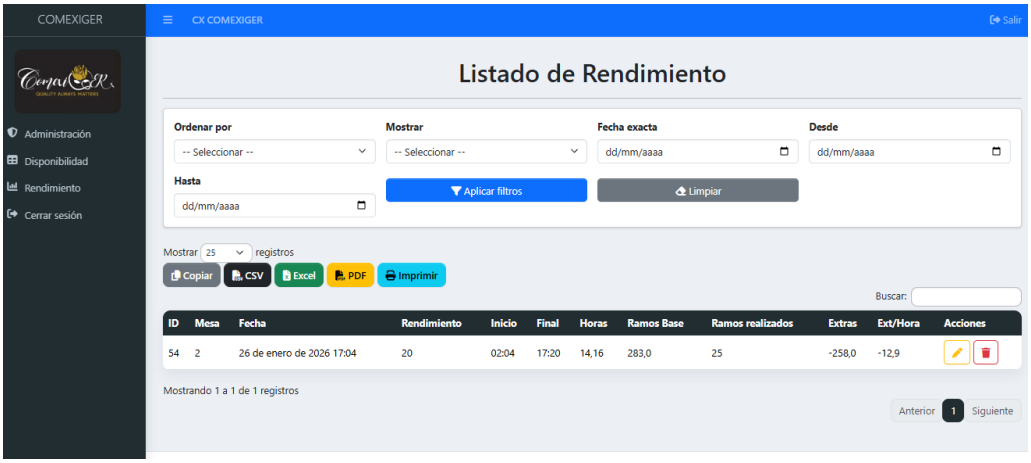


Figura 5.55.Panel de gestión de Rendimiento.

5.4.1.1.3 Usuario

Aquí se almacena la información de las personas que utilizarán la aplicación el que permite identificar a cada usuario y gestionar su acceso, su estructura se muestra en la Figura 5.56.

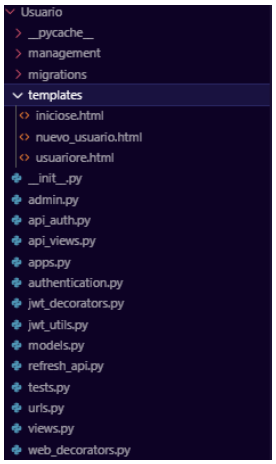


Figura 5.56.Estructura de la aplicación de Usuario.

- **Modelo de Usuario**

En la Figura 5.57 se puede visualizar el modelo de usuario utilizado en el Framework Django el cual será utilizado para todo el desarrollo del proyecto.

```python
from django.db import models
from django.contrib.auth.hashers import make_password, check_password
class Mesa(models.Model):
    nombre = models.CharField(max_length=100, unique=True)

    def __str__(self):
        return self.nombre

class Usuario(models.Model):
    nombres = models.CharField(max_length=100)
    apellidos = models.CharField(max_length=100)
    mesa = models.CharField(max_length=100)
    cargo = models.CharField(max_length=100)
    username = models.CharField(max_length=150, unique=True)
    password = models.CharField(max_length=128)



    def __str__(self):
        return f"{self.nombres} {self.apellidos} ({self.username})"

    def set_password(self, raw_password):
        self.password = make_password(raw_password)

    def check_password(self, raw_password):
        return check_password(raw_password, self.password)
    @property
    def is_authenticated(self):
        return True
```

Figura 5.57.Fragmento de código del Modelo de Usuario.

- **Template Usuario**

En la Figura 5.58 se puede visualizar el template de usuario utilizado en el Framework Django el cual será utilizado para todo el desarrollo del proyecto.

```
{% extends 'plantilla.html' %}
{% load static %}

{% block head %}
{{ block.super }}

<link rel="stylesheet" href="https://cdn.datatables.net/1.13.7/css/dataTables.bootstrap5.min.css">
<link rel="stylesheet" href="https://cdn.datatables.net/responsive/2.5.0/css/responsive.bootstrap5.min.css">

<link rel="stylesheet" href="https://cdn.datatables.net/buttons/2.4.2/css/buttons.bootstrap5.min.css">

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/sweetalert2@11/dist/sweetalert2.min.css">

<style>
  table.dataTable thead th { white-space: nowrap; }

  .dataTables_wrapper .dataTables_filter {
    text-align: right;
    padding: 12px 16px 0 16px;
  }
  .dataTables_wrapper .dataTables_filter label {
    font-weight: 600;
    color: #2c3e50;
    display: inline-flex;
    align-items: center;
    gap: 10px;
    margin: 0;
  }
  .dataTables_wrapper .dataTables_filter input {
    border-radius: 12px !important;
    border: 2px solid #e9ecef !important;
    padding: 10px 14px !important;
    outline: none;
    transition: all 0.25s ease;
```

Figura 5.58.Fragmento de código del template de Usuario.

En la Figura 5.59 se puede visualizar el resultado del frontend de usuario utilizado en el Framework Django el cual será utilizado para todo el desarrollo del proyecto.

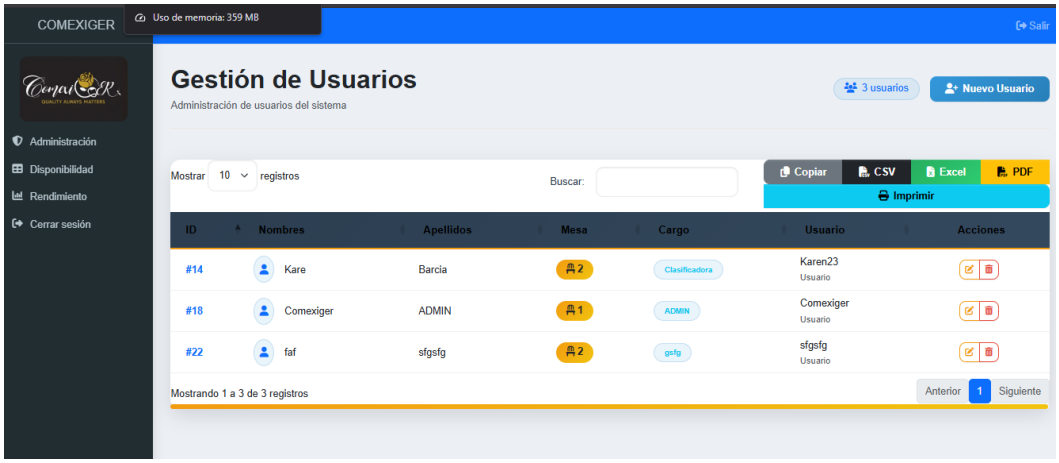


Figura 5.59. Panel de gestión de Usuario.

5.4.1.2 Implementación de la lógica de negocio en las vistas

La lógica de negocio del sistema se desarrolló controlando de manera ordenada la creación, edición y validación de registros. Cuando una operación requiere modificar varios datos del mismo tiempo. Además, en la interfaz web se incorporaron mensajes de confirmación para el usuario.

- **Disponibilidad:** se puede visualizar el código de la figura 5.60.

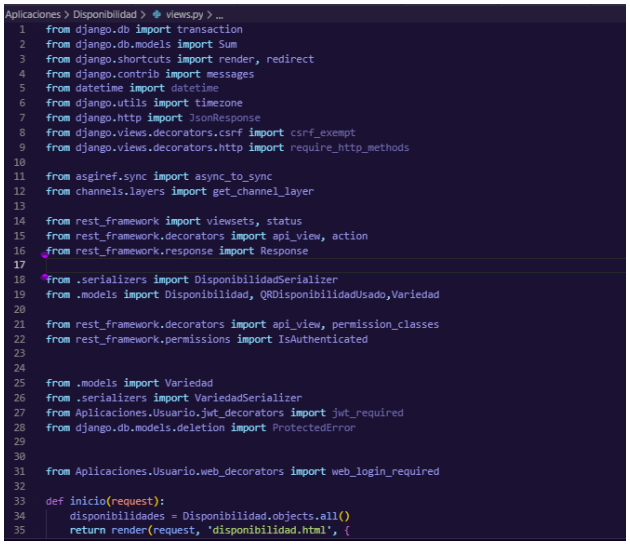
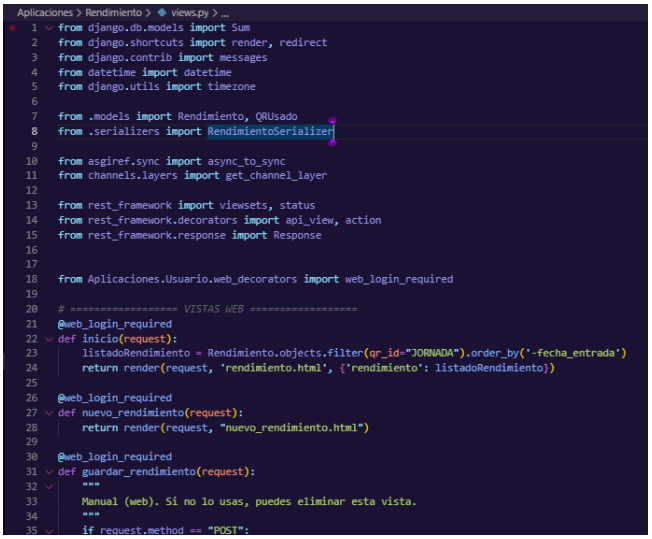


Figura 5.60. Fragmento de código del views de Disponibilidad.

- **Rendimiento:** se representa el código en la figura 5.61.



```python
from django.db.models import Sum
from django.shortcuts import render, redirect
from django.contrib import messages
from datetime import datetime
from django.utils import timezone

from .models import Rendimiento, QRUsado
from .serializers import RendimientoSerializer

from asgiref.sync import async_to_sync
from channels.layers import get_channel_layer

from rest_framework import viewsets, status
from rest_framework.decorators import api_view, action
from rest_framework.response import Response


from Aplicaciones.Usuario.web_decorators import web_login_required

# ================= VISTAS WEB =================
@web_login_required
def inicio(request):
    listadoRendimiento = Rendimiento.objects.filter(qr_id="JORNADA").order_by('-fecha_entrada')
    return render(request, 'rendimiento.html', {'rendimiento': listadoRendimiento})

@web_login_required
def nuevo_rendimiento(request):
    return render(request, "nuevo_rendimiento.html")

@web_login_required
def guardar_rendimiento(request):
    """
    Manual (web). Si no lo usas, puedes eliminar esta vista.
    """
    if request.method == "POST":
```

Figura 5.61.Fragmento de código del views de Rendimiento.

- **Usuario:** el fragmento del código se representa en la figura 5.62

```python
def cerrarsesion(request):
    request.session.flush()
    messages.success(request, "Sesión cerrada correctamente.")
    return redirect("iniciose")
@web_login_required
def dispo(request):
    disponibilidades = Disponibilidad.objects.all().order_by("-fecha_entrada")
    return render(request, "disponibilidad.html", {
        "disponibilidades": disponibilidades,
        "web_username": request.session.get("web_username"),
    })

@web_login_required
def inicios(request):
    listadoUsuarios = Usuario.objects.all()
    mesas = Mesa.objects.all().order_by("nombre")
    return render(request, "usuariore.html", {"usuario": listadoUsuarios, "mesas": mesas})


@web_login_required
def nuevo_usuario(request):
    mesas = Mesa.objects.all().order_by("nombre")
    return render(request, "nuevo_usuario.html", {"mesas": mesas})
@web_login_required
def guardar_mesa(request):
    if request.method == "POST":
        nombre = (request.POST.get("nombre") or "").strip()

        if not nombre:
            messages.error(request, "Ingrese el nombre de la mesa.")
            return redirect("nuevo_usuario")

        if Mesa.objects.filter(nombre__iexact=nombre).exists():
            messages.warning(request, "Esa mesa ya existe.")
            return redirect("nuevo_usuario")
```

Figura 5.62.Fragmento de código del views de Usuario.

5.4.1.3 Gestión de autenticación y sesión

Para acceder al sistema se implementó la autentificación como se ve en la figura 5.63 es decir un Login que cubre tanto el uso desde la web como desde servicios externos. El inicio sesión permite crear una sesión válida que habilita el acceso a las funcionalidades protegidas.

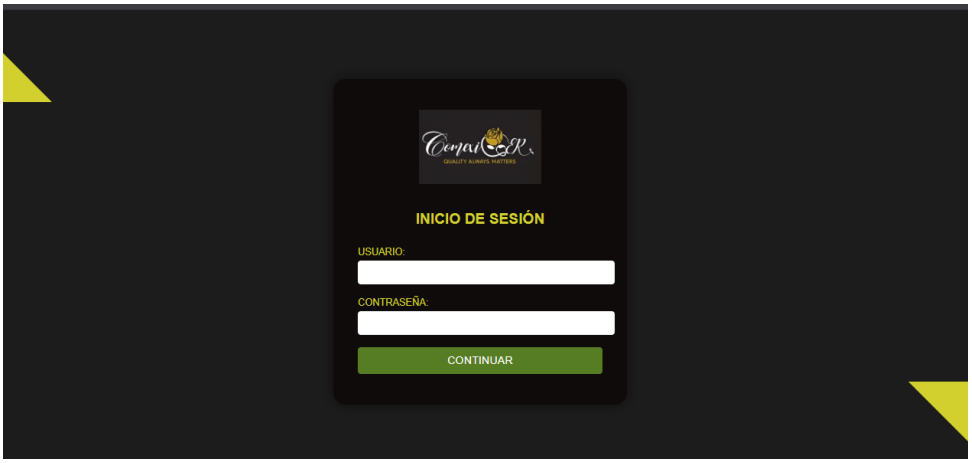


Figura 5.63.Módulo de inicio de sesión.

Fragmento del código de inicio de sesión en la figura 5.64.

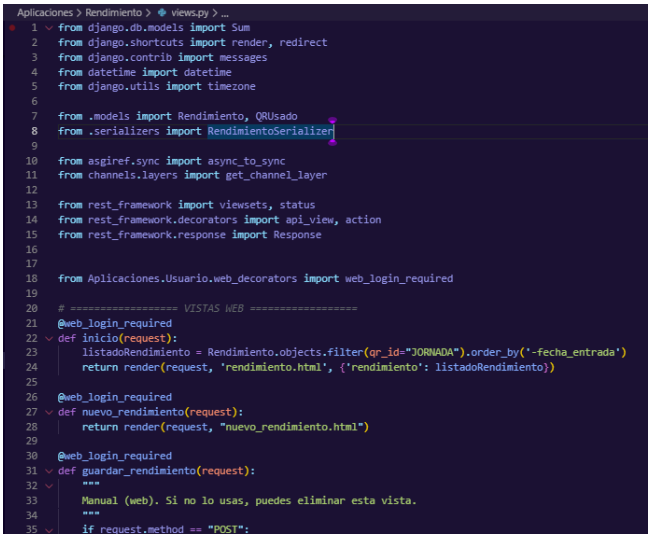


Figura 5.64.Fragmento de código del views de Usuario.

5.4.1.4 Construcción de servicios REST APIs en Django

Asimismo, se desarrollaron servicios REST utilizando Django REST Framework para permitir la gestión de la información del sistema. Realizar esto nos ayudará a una mejor conexión para nuestra aplacación móvil.

5.4.1.4.1 Disponibilidad

Se representa el api de disponibilidad en el fragmento de código de la figura 5.65.

```python
class DisponibilidadViewSet(viewsets.ModelViewSet):
    def por_mesa(self, request):
        return Response(self.get_serializer(qs, many=True).data)


# ========================
# API MANUAL
# ========================
@api_view(['GET', 'POST'])
def api_disponibilidad_list(request):
    print(" user:", request.user, "auth:", request.user.is_authenticated)
    print(" cookies:", request.COOKIES)
    print("session keys:", list(request.session.keys()))

    if request.method == 'GET':
        ordenar = request.query_params.get("ordenar")
        fecha = request.query_params.get("fecha")
        desde = request.query_params.get("desde")
        hasta = request.query_params.get("hasta")
        reciente = request.query_params.get("reciente")

        qs = Disponibilidad.objects.all()

        if fecha:
            qs = qs.filter(fecha_entrada__date=fecha)

        if desde and hasta:
            qs = qs.filter(fecha_entrada__date__range=[desde, hasta])

        campos = {
            "mesa": "numero_mesa",
            "variedad": "variedad",
            "medida": "medida",
            "fecha": "fecha_entrada"
        }
```

Figura 5.65.Fragmento de código de la API de Disponibilidad.

- **Serializer**

Se representa el Serializer en el fragmento de código de la figura 5.66.

```python
from rest_framework import serializers
from .models import Disponibilidad

from .models import Variedad

class VariedadSerializer(serializers.ModelSerializer):
    class Meta:
        model = Variedad
        fields = ["id", "nombre", "fecha_creacion"]
class DisponibilidadSerializer(serializers.ModelSerializer):
    class Meta:
        model = Disponibilidad
        fields = '__all__'


class DisponibilidadCreateSerializer(serializers.ModelSerializer):
    class Meta:
        model = Disponibilidad
        fields = [
            'numero_mesa',
            'variedad',
            'medida',
            'stock',
            'fecha_entrada',
            'fecha_salida'
        ]

```

Figura 5.66.Fragmento de código del Serializer de la API Disponibilidad.

- **Rutas**

Se representa el Serializer en el fragmento de código de la figura 5.67.

```
    path('api/', include(router.urls)),
    path('api/disponibilidades/', views.api_disponibilidad_list, name='api-disponibilidad-list'),
    path('api/disponibilidades/<int:pk>/', views.api_disponibilidad_detail, name='api-disponibilidad-detail'),
    path('api/disponibilidades/stats/', views.api_disponibilidad_stats, name='api-disponibilidad-stats'),

    path('api/disponibilidades/salida/', api_disponibilidad_salida),



    path("api/variedades/excel/", views.variedades_excel_api, name="variedades_excel_api"),
]

```

Figura 5.67.Fragmento de código de las rutas de la API Disponibilidad.

5.4.1.4.2 Rendimiento

Se representa el rendimiento en el fragmento de código de la figura 5.68.

```
# Rendimiento/api_views.py
from django.http import JsonResponse
from django.views.decorators.csrf import csrf_exempt
from django.utils import timezone
import json
from asgiref.sync import async_to_sync
from channels.layers import get_channel_layer
from Aplicaciones.Usuario.jwt_decorators import jwt_required
from .models import Rendimiento
from .serializers import RendimientoSerializer

def _broadcast_rendimiento(rendimiento):
    channel_layer = get_channel_layer()

    data = RendimientoSerializer(rendimiento).data


    async_to_sync(channel_layer.group_send)(
        "rendimientos",
        {
            "type": "nuevo_rendimiento",
            "data": data
        }
    )

@csrf_exempt
@jwt_required
def iniciar_jornada_api(request):
    if request.method != "POST":
        return JsonResponse({"success": False, "error": "Método no permitido. Use POST"}, status=405)

    try:
        data = json.loads(request.body.decode("utf-8"))
        mesa = (data.get("mesa") or data.get("numero_mesa") or "").strip()
        if not mesa:
```

Figura 5.68.Fragmento de código de la API de Rendimiento.

- **Serializer**

Se representa el Serializer en el fragmento de código de la figura 5.69.

```
# Rendimiento/serializers.py
from rest_framework import serializers
from .models import Rendimiento, JornadaLaboral

class RendimientoSerializer(serializers.ModelSerializer):
    class Meta:
        model = Rendimiento
        fields = '__all__'
        read_only_fields = ['horas_trabajadas', 'ramos_esperados', 'ramos_extras', 'extras_por_hora']

class JornadaLaboralSerializer(serializers.ModelSerializer):
    class Meta:
        model = JornadaLaboral
        fields = [
            'id', 'usuario_username', 'usuario_nombre', 'mesa',
            'fecha', 'hora_inicio', 'hora_fin', 'estado', 'horas_trabajadas'
        ]
        read_only_fields = ['fecha', 'horas_trabajadas']
```

Figura 5.69.Fragmento de código del Serializer de la API Rendimiento.

Rutas: Se representa el Serializer en el fragmento de código de la figura 5.70.

```
Aplicaciones > Rendimiento > urls.py > ...
17      path('api/', include(router.urls)),
18      path('api/rendimientos/', views.api_rendimiento_list, name='api-rendimiento-list'),
19      path('api/rendimientos/<int:pk>/', views.api_rendimiento_detail, name='api-rendimiento-detail'),
20      path('api/rendimientos/stats/', views.api_rendimiento_stats, name='api-rendimiento-stats'),
21  ]
```

Figura 5.70.Fragmento de código de las rutas de la API Rendimiento.

5.4.1.4.3 Usuario

Se representa el api de usuario en el fragmento de código de la figura 5.71.

```
from django.http import JsonResponse
from django.views.decorators.csrf import csrf_exempt
from django.contrib.auth.hashers import make_password, check_password
import json

from .models import Usuario, Mesa
from .jwt_utils import crear_access_token, crear_refresh_token
from Aplicaciones.Usuario.jwt_decorators import jwt_required

@csrf_exempt
def registrar_usuario_api(request):
    """
    API para registro desde Flutter
    POST: /api/registrar/
    """
    if request.method != "POST":
        return JsonResponse({"success": False, "error": "Método no permitido. Use POST"}, status=405)

    try:
        data = json.loads(request.body.decode("utf-8"))

        campos_requeridos = ["nombres", "apellidos", "mesa", "cargo", "username", "password"]
        faltantes = [c for c in campos_requeridos if not str(data.get(c, "")).strip()]
        if faltantes:
            return JsonResponse(
                {"success": False, "error": f"Campos requeridos faltantes: {', '.join(faltantes)}"},
                status=400
            )

        username = data["username"].strip()
        if Usuario.objects.filter(username__iexact=username).exists():
            return JsonResponse({"success": False, "error": f"El usuario '{username}' ya existe"}, status=400)
```

Figura 5.71.Fragmento de código de la API de Usuario.

Rutas

Se representa las rutas en el fragmento de código de la figura 5.72.

```
    path('api/registrar/', registrar_usuario_api, name='api_registrar'),
    path('api/login/', login_usuario_api, name='api_login'),
    path('api/mesas/', obtener_mesas_api, name='api_mesas'),
    path('api/verificar_mesa/', verificar_mesa_api, name='api_verificar_mesa'),


    path('api/jornada/iniciar/', iniciar_jornada_api, name='api_jornada_iniciar'),
    path('api/jornada/finalizar/', finalizar_jornada_api, name='api_jornada_finalizar'),
    path('api/jornada/actual/', obtener_jornada_actual_api, name='api_jornada_actual'),
    path('api/jornada/historial/', obtener_historial_jornadas_api, name='api_jornada_historial'),

    path('api/variedades/', listar_variedades_api, name='api_variedades'),

    path("api/token/refresh/", refresh_token_api, name="api_token_refresh"),


]
```

Figura 5.72.Fragmento de código de las rutas de la API Usuario.

5.4.1.5 Consumo de APIs

En el sistema, el consumo de APIs se realiza desde la interfaz web mediante peticiones HTTP asíncronas (por ejemplo, usando fetch) para enviar y recibir información en formato JSON. Este enfoque permite actualizar componentes de la interfaz como tablas, formularios y modales sin necesidad de recargar toda la página, mejorando la experiencia del usuario.

- **Seguridad JWT**

Para asegurar el acceso a los recursos, los endpoints protegidos requieren autenticación mediante JWT**.** Cuando el usuario inicia sesión en el flujo correspondiente, el sistema obtiene un token y, en cada solicitud hacia endpoints protegidos, el frontend lo envía. Con esto se evita que usuarios no autenticados puedan consumir funcionalidades sensibles.

- Validación de JWT en endpoints protegidos en la figura 5.73.

```python
# Aplicaciones/Usuario/jwt_decorators.py

from functools import wraps
from django.http import JsonResponse
from Aplicaciones.Usuario.jwt_utils import decodificar_token
from Aplicaciones.Usuario.models import Usuario

def jwt_required(view_func=None, *, allowed_cargos=None, enforce_mesa=False):
    def decorator(func):
        @wraps(func)
        def wrapper(request, *args, **kwargs):
            auth = request.headers.get("Authorization", "")

            if not auth:
                return JsonResponse({"success": False, "error": "Token requerido"}, status=401)

            parts = auth.split()
            if len(parts) != 2 or parts[0].lower() != "bearer":
                return JsonResponse(
                    {"success": False, "error": "Formato Authorization inválido. Usa: Bearer <token>"},
                    status=401
                )

            token = parts[1].strip()

            try:
                payload = decodificar_token(token)
            except Exception:
                return JsonResponse({"success": False, "error": "Token inválido o expirado"}, status=401)

            if payload.get("type") != "access":
                return JsonResponse({"success": False, "error": "Token incorrecto (se requiere access token)"}, status=401)

            usuario_id = payload.get("sub")
```

Figura 5.73.Fragmento de código de validación de JWT en endpoints protegidos.

- Login API que genera/retorna el token en la figura 5.74.

```python
from functools import wraps
from django.http import JsonResponse
from .jwt_utils import decodificar_token
from .models import Usuario

def jwt_required(view_func):
    """
    Requiere header:
      Authorization: Bearer <access_token>

    Inyecta request.api_user (Usuario) o request.api_admin=True si fuera admin.
    """
    @wraps(view_func)
    def _wrapped(request, *args, **kwargs):
        auth = request.headers.get("Authorization", "")
        if not auth:
            return JsonResponse({"success": False, "error": "Falta Authorization: Bearer <token>"}, status=401)

        parts = auth.split()
        if len(parts) != 2 or parts[0].lower() != "bearer":
            return JsonResponse({"success": False, "error": "Formato Authorization inválido (usa Bearer <token>)"}, status=401)

        token = parts[1].strip()

        try:
            payload = decodificar_token(token)
        except Exception:
            return JsonResponse({"success": False, "error": "Token inválido o expirado"}, status=401)

        if payload.get("type") != "access":
            return JsonResponse({"success": False, "error": "Token incorrecto (se requiere access token)"}, status=401)

        # (opcional) admin si lo usas
        if payload.get("tipo") == "admin":
            request.api_admin = True
```

Figura 5.74.Fragmento de código del login API que genera/retorna el token.

5.4.1.6 Servidor Daphne

Para soportar la comunicación en tiempo real, se ejecuta sobre ASGI usando Daphne como servidor de aplicación es decir que se reemplazó el servidor WSGI por la importante utilización de Channels. Gracias a esto se logra una sincronización y permite que el sistema escuche y responda vista en la figura 5.75.

```python
COMEXIGER > asgi.py > ...
import os
from django.conf import settings
from django.core.asgi import get_asgi_application
from django.contrib.staticfiles.handlers import ASGIStaticFilesHandler

from channels.auth import AuthMiddlewareStack
from channels.routing import ProtocolTypeRouter, URLRouter

os.environ.setdefault("DJANGO_SETTINGS_MODULE", "COMEXIGER.settings")

django_asgi_app = get_asgi_application()


import Aplicaciones.Rendimiento.routing
import Aplicaciones.Disponibilidad.routing


if settings.DEBUG:
    django_asgi_app = ASGIStaticFilesHandler(django_asgi_app)

application = ProtocolTypeRouter({
    "http": django_asgi_app,

    "websocket": AuthMiddlewareStack(
        URLRouter(
            Aplicaciones.Rendimiento.routing.websocket_urlpatterns
            + Aplicaciones.Disponibilidad.routing.websocket_urlpatterns
        )
    ),
})
```

Figura 5.75.Fragmento de código del cambio de servidor Daphne.

5.4.1.7 Configuración del WebSockets

Se configuraron WebSockets con Django Channels definiendo una conexión que ayuda a manejar el ciclo de vida y comunicación que se organiza por grupos para enviar notificaciones. Esto es lo que permite la comunicación en tiempo real.

- Disponibilidad

Se representa la configuración del websokect en el fragmento de código de la figura 5.76.

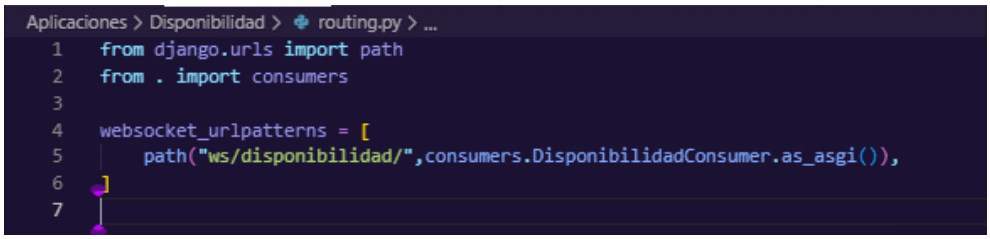


Figura 5.76.Websokects de Disponibilidad.

- Rendimiento

Se representa la configuración del websokect en el fragmento de código de la figura 5.77.

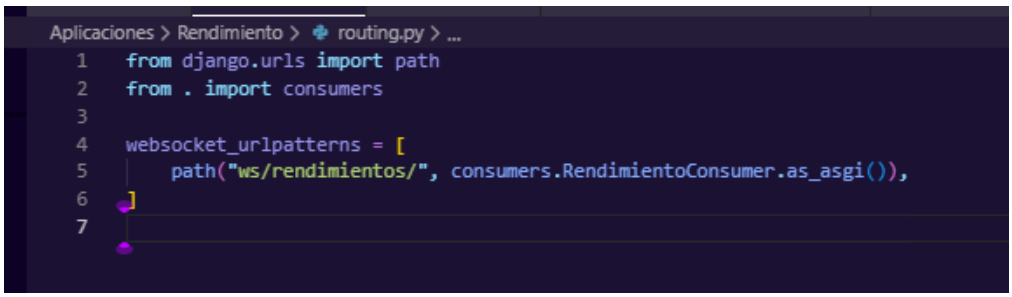


Figura 5.77.Websokects de Rendimiento.

5.4.2 Creación y conexión del Listener

Es el encargado de conectar la aplicación web con los Escáneres y el que escuchara los eventos WebSocket visto en la figura 5.78.

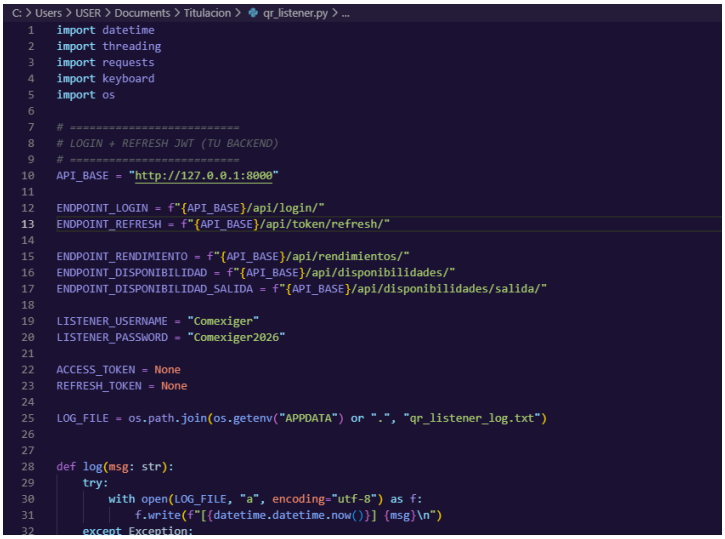


Figura 5.78.Fragmento de código del Listener configurado.

- **Disponibilidad**

Envía la notificación como se en la figura 5.79.

Figura 5.79.Fragmento de código de Disponibilidad para la conexión del listener.

- R**endimiento**

Envía la notificación vista en la figura 5.80.

```python
from channels.generic.websocket import AsyncWebsocketConsumer
import json

class RendimientoConsumer(AsyncWebsocketConsumer):
    async def connect(self):
        await self.channel_layer.group_add("rendimientos", self.channel_name)
        await self.accept()

    async def disconnect(self, close_code):
        await self.channel_layer.group_discard("rendimientos", self.channel_name)

    async def nuevo_rendimiento(self, event):
        await self.send(text_data=json.dumps({
            "type": "nuevo_rendimiento",
            "data": event.get("data", {})
        }))

    async def send_rendimiento(self, event):
        await self.send(text_data=json.dumps({
            "type": "send_rendimiento",
            "data": event.get("data", {})
        }))
    async def nuevo_rendimiento(self, event):
        await self.send(text_data=json.dumps(event["data"]))
```

Figura 5.80.Fragmento de código de Rendimiento para la conexión del listener.

5.4.3 Configuración de los escáneres 1 y 2

Para diferenciar entre la suma y resta de optó por usar prefijos y así separar la resta de la suma, parar lograr realzar este proceso se tomó en cuenta los manuales de cada uno. en el cual vienen los códigos que se deben escanear para añadir correctamente los prefijos.

- **Escáner 1** destinado a ser usado como suma a este se le añadió el prefijo IN, se configuro a través de los códigos del manual el Auto escaneo conjuntamente con el enter al final para que gracias a él todo se envía a disponibilidad y a rendimiento.
- **Escáner 2** este es destinado para la resta aquí se añadió el prefijo OUT se configuro a través de los códigos del manual el Auto escaneo conjuntamente con el enter al final este solo modifica la disponibilidad.

5.4.4 Aplicación Móvil

5.4.4.1 Estructura de pantallas y componentes

La aplicación móvil hecha en Flutter está organizada como se puede ver en la figura 5.81; en la carpeta lib. Esto se hace de forma modular, lo que significa que cada parte tiene su propia función. Esto es útil porque hace que la aplicación sea más fácil de mantener y de hacer más grande. También se pueden reutilizar componentes de la aplicación.

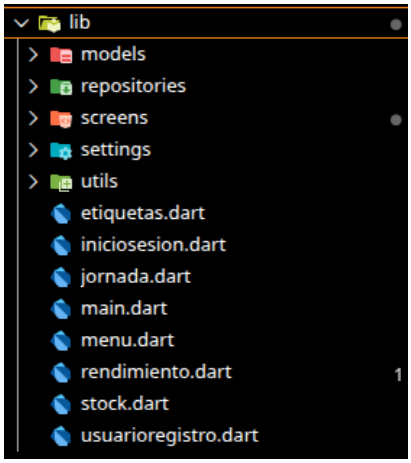


Figura 5.81. Estructura del App Móvil.

5.4.4.2 Gestión del estado y modelos de datos

En la figura 5.82 se muestra una carpeta; en esta se encuentran las estructuras de datos del sistema. Estas estructuras de datos incluyen los usuarios, las mesas, el stock, las jornadas, el rendimiento y las variedades.

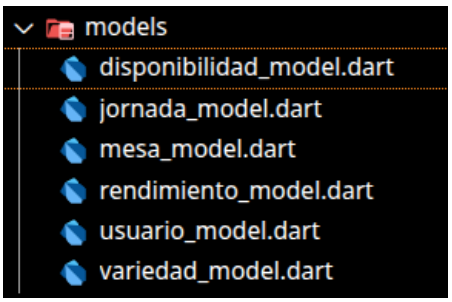


Figura 5.82. Modelos creados.

5.4.4.2.1 Modelos

Estas estructuras de datos permiten llevar la información que se recibe desde la API a Dart y de Dart a la API. Además, ayudan a centralizar la lógica básica, como las validaciones y los cálculos. De esta manera, se garantiza un correcto manejo de los datos en la aplicación móvil.

- **Disponibilidad_model.dart:** en la figura 5.83 se muestra el fragmento del código del modelo para disponibilidad.

```dart
import 'package:flutter/material.dart';

class JornadaModel {
  final int id;
  final String mesa;
  final DateTime fechaEntrada;
  final DateTime horaInicio;
  final DateTime? horaFin;
  final double? horasTrabajadas;

  JornadaModel({
    required this.id,
    required this.mesa,
    required this.fechaEntrada,
    required this.horaInicio,
    this.horaFin,
    this.horasTrabajadas,
  });


  static DateTime _parseDateTimeSafe(dynamic value) {
  if (value == null) return DateTime.now();
  final s = value.toString().trim();
  if (s.isEmpty) return DateTime.now();

  return DateTime.parse(s).toLocal();
}
```

Figura 5.83. Fragmento de código del modelo Disponibilidad.

- **mesa_model.dart:** en la figura 5.84 se visualiza un fragmento de código del modelo utilizado para mesa.

```dart
class MesaModel {
  int id;
  String nombre;

  MesaModel({required this.id, required this.nombre});

  factory MesaModel.fromJson(Map<String, dynamic> json) {
    return MesaModel(id: json['id'] ?? 0, nombre: json['nombre'] ?? '');
  }

  Map<String, dynamic> toJson() {
    return {'id': id, 'nombre': nombre};
  }

  @override
  bool operator ==(Object other) {
    if (identical(this, other)) return true;
    return other is MesaModel && other.id == id && other.nombre == nombre;
  }

  @override
  int get hashCode => id.hashCode ^ nombre.hashCode;

  @override
  String toString() {
    return 'Mesa{id: $id, nombre: $nombre}';
  }
}
```

Figura 5.84. Fragmento de código del modelo Mesa.

- **rendimiento_model.dart:** en la figura 5.85 se visualiza un fragmento de código del modelo utilizado para visualizar el rendimiento.

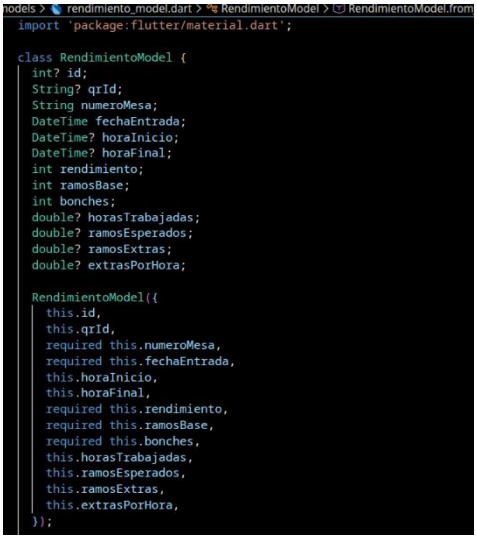


Figura 5.85. Fragmento de código del modelo Rendimiento.

- **usuario_model.dart:** en la figura 5.86 se visualiza un fragmento de código del modelo utilizado para mostrar e identificar el usuario.

Figura 5.86. Fragmento de código del modelo Usuario.

- **variedad_model.dart:** en la figura 5.87 se visualiza un fragmento de código del modelo utilizado para mesa.

```dart
class VariedadModel {
  final int id;
  final String nombre;

  VariedadModel({required this.id, required this.nombre});

  factory VariedadModel.fromJson(Map<String, dynamic> json) {
    return VariedadModel(
      id: (json['id'] as num).toInt(),
      nombre: (json['nombre'] ?? '').toString(),
    );
  }
}
```

Figura 5.87. Fragmento de código del modelo Variedad.

5.4.4.2.2 Repositorios

En la figura 5.88 se muestra una carpeta con los repositories; en esta se encuentran gestionados la comunicación entre la aplicación móvil, la aplicación web y la API. Esto significa que centralizamos el envío de solicitudes al servidor. También nos encargamos de la autenticación cuando sea necesaria. Además, gestionamos las respuestas del servidor. Luego, transformamos las respuestas del backend en estructuras que la aplicación puede usar. También nos ocupamos de los errores de conexión.

Figura 5.88. Carpeta repositories.

- **disponibilidad_repository.dart:**

El proceso ayuda a controlar la disponibilidad dentro de la aplicación comunicándose con la API del sistema. Este módulo permite obtener registros de disponibilidad. También se pueden consultar estadísticas. La información se puede filtrar por mesa o por estado activo. Se puede registrar la salida de stock mediante códigos QR. Dicho flujo se refleja en la figura 5.89.

```dart
import '../utils/api_client.dart';
import '../models/disponibilidad_model.dart';

  static Future<Map<String, dynamic>> obtenerTodasDisponibilidades({
    String? fecha,
    String? desde,
    String? hasta,
    String? ordenar,
    bool? reciente,
  }) async {
    final queryParams = <String, String>{};

    if (fecha != null && fecha.isNotEmpty) queryParams['fecha'] = fecha;

    if (desde != null && hasta != null && desde.isNotEmpty && hasta.isNotEmpty) {
      queryParams['desde'] = desde;
      queryParams['hasta'] = hasta;
    }

    if (ordenar != null && ordenar.isNotEmpty) {
      queryParams['ordenar'] = ordenar;
      if (reciente != null) queryParams['reciente'] = reciente.toString();
    }

    try {
      final res = await ApiClient.get(
        '/api/disponibilidades/',
        auth: true,
        query: queryParams.isEmpty ? null : queryParams,
      );

      final status = res['status'] as int;
      final data = res['data'] as Map<String, dynamic>;

      final rawList = data['data'];

      if (status == 200 && rawList is List) {
        return {
          'success': true,
          'disponibilidades': rawList.map((d) => DisponibilidadModel.fromJson(d)).toList(),
          'count': rawList.length,
        };
      }
```

Figura 5.89. Proceso de controlar la disponibilidad.

- **jornada_repository.dart:**

El proceso permite gestionar la jornada laboral de cada mesa dentro de la aplicación. Puedes iniciar una jornada, finalizarla y revisar la jornada actual. También puedes consultar el historial de jornadas registradas. Esto se hace a través de la conexión con la API del sistema. En la figura 5.90, se puede ver cómo funciona este proceso.

```dart
import '../utils/api_client.dart';

class JornadaRepository {
  static Future<Map<String, dynamic>> iniciarJornada({
    required String mesa,
    String? usuarioUsername,
    String? usuarioNombre,
  }) async {
    print('🟢 [JORNADA] Iniciando jornada (mesa=$mesa) usuario=${usuarioUsername ?? "-"}');

    try {
      final res = await ApiClient.post(
        '/api/jornada/iniciar/',
        auth: true,
        body: {'mesa': mesa},
      );

      final status = res['status'] as int;
      final responseData = res['data'] as Map<String, dynamic>;

      print('🚢 Código: $status');
      print('🚢 Respuesta: $responseData');

      if (status == 201) {
        return {
          'success': true,
          'message': responseData['message'],
          'data': responseData['data'],
        };
      }

      if (status == 409 || status == 400) {
        return {
          'success': false,
          'message': responseData['error'] ?? 'No se pudo iniciar jornada',
          'data': responseData['data'],
        };
      }
```

Figura 5.90. Proceso de gestión de jornada laboral.

- **mesa_repository.dart:**

El proceso consiste en obtener una lista de mesas registradas en el sistema a través de la API. La aplicación solicita la información al servidor. Luego, el servidor proporciona los datos

disponibles. La aplicación procesa estos datos para mostrarlos dentro del sistema. Este proceso se puede ver en la figura 5.91.

```dart
import '../utils/api_client.dart';

class MesaRepository {
  static Future<Map<String, dynamic>> obtenerMesas() async {
    try {
      final res = await ApiClient.get('/api/mesas/', auth: true);

      final status = res['status'] as int;
      final data = res['data'] as Map<String, dynamic>;

      print('🪑 [MESAS] Código: $status');
      print('🪑 [MESAS] Respuesta: $data');

      if (status == 200) {
        final rawList = (data['data'] ?? []) as List;

        final mesas = rawList.map((e) => (e['nombre'] ?? '').toString()).where((x) => x.isNotEmpty).toList();

        return {'success': true, 'mesas': mesas, 'count': mesas.length};
      }

      return {'success': false, 'message': data['error'] ?? 'Error ($status)', 'raw': data};
    } catch (e) {
      return {'success': false, 'message': 'Error al obtener mesas: $e'};
    }
  }
}
```

Figura 5.91. Proceso de obtención de mesas.

- **rendimiento_repository.dart:**

El proceso ayuda a manejar la información de rendimiento en el sistema a través de la API. Con este módulo, puedes obtener registros de rendimiento, ver estadísticas, buscar información por fecha o mesa, y agregar nuevos rendimientos con códigos QR. Esto se muestra en la figura 5.92.

```dart
import '../utils/api_client.dart';
import '../models/rendimiento_model.dart';

class RendimientoRepository {

  static Future<Map<String, dynamic>> obtenerTodosRendimientos({
    String? fecha,
    String? desde,
    String? hasta,
    String? ordenar,
    bool? reciente,
    String? mesa,
  }) async {
    final queryParams = <String, String>{};

    if (fecha != null && fecha.isNotEmpty) queryParams['fecha'] = fecha;

    if (desde != null && hasta != null && desde.isNotEmpty && hasta.isNotEmpty) {
      queryParams['desde'] = desde;
      queryParams['hasta'] = hasta;
    }

    if (ordenar != null && ordenar.isNotEmpty) {
      queryParams['ordenar'] = ordenar;
      if (reciente != null) queryParams['reciente'] = reciente.toString();
    }

    if (mesa != null && mesa.isNotEmpty) queryParams['mesa'] = mesa;

    print('📊 [REPOSITORY] Obteniendo rendimientos');
    print('📋 Query: $queryParams');

    try {
      final res = await ApiClient.get(
        '/api/rendimientos/',
        auth: true,
        query: queryParams.isEmpty ? null : queryParams,
      );

      final status = res['status'] as int;
      final data = res['data'];
```

Figura 5.92. Proceso de manejo de información de rendimiento.

- **variedad_repository.dart:**

El proceso nos ayuda a obtener la información de las variedades que están registradas en el sistema. Esto se hace hablando con la API. La aplicación pide los datos al servidor, los procesa y los prepara para que se puedan usar dentro del sistema. Esto se puede ver en la figura 5.93.

```dart
import '../utils/api_client.dart';
import '../models/variedad_model.dart';

class VariedadRepository {
  static Future<Map<String, dynamic>> obtenerVariedades() async {
    try {
      final res = await ApiClient.get('/api/variedades/', auth: true);

      final status = res['status'] as int;
      final data = res['data'] as Map<String, dynamic>;

      print('🌸 [VARIEDADES] Código: $status');
      print('🌸 [VARIEDADES] Respuesta: $data');

      if (status == 200) {
        final rawList = (data['data'] ?? []) as List;
        final variedades = rawList
            .map((e) => VariedadModel.fromJson(e as Map<String, dynamic>))
            .toList();

        return {'success': true, 'variedades': variedades};
      }

      return {'success': false, 'message': data['error'] ?? 'Error ($status)'};
    } catch (e) {
      return {'success': false, 'message': 'Error al obtener variedades: $e'};
    }
  }
}
```

Figura 5.93. Obtención de información de las variedades.

5.4.4.3 Implementación de la lógica funcional del cliente móvil

En esta carpeta llamada utils se encuentra la lógica funcional del cliente móvil. Esta lógica se basa en la gestión de la comunicación con la API. Hay dos archivos importantes aquí: api_client.dart y usuario_preferences.dart.

El archivo api_client.dart es muy útil. Centraliza las solicitudes HTTP que se envían al servidor. Además, automáticamente agrega el token de autenticación cuando es necesario. Procesa las respuestas que vienen en formato JSON y también gestiona los errores que puedan ocurrir.

- **api_client.dart:**

El proceso ayuda a controlar cómo se comunican la aplicación móvil y la API del sistema. Con este módulo, podemos hacer solicitudes al servidor, enviar y recibir datos en formato JSON, y gestionar la autenticación utilizando tokens para acceder a los servicios del sistema. Esto se muestra en la figura 5.94.

```dart
import 'dart:convert';
import 'package:http/http.dart' as http;
import 'usuario_preferences.dart';

class ApiClient {
  ApiClient._();

  static const String baseUrl = "http://10.198.32.101:8000";

  static const Map<String, String> _baseHeaders = {
    'Content-Type': 'application/json; charset=UTF-8',
    'Accept': 'application/json',
  };

  static Uri _u(String path, [Map<String, String>? query]) {
    final p = path.startsWith('/') ? path : '/$path';
    return Uri.parse('$baseUrl$p').replace(queryParameters: query);
  }

  static Future<Map<String, String>> _headers({bool auth = true}) async {
    final h = Map<String, String>.from(_baseHeaders);
    if (!auth) return h;

    final token = await UsuarioPreferences.obtenerAccessToken();
    if (token != null && token.trim().isNotEmpty) {
      h['Authorization'] = 'Bearer $token';
    }
    return h;
  }

  static Map<String, dynamic> _decode(http.Response r) {
    try {
      final txt = utf8.decode(r.bodyBytes);
      final d = jsonDecode(txt);
      return d is Map<String, dynamic> ? d : {'data': d};
    } catch (_) {
      return {'raw': utf8.decode(r.bodyBytes)};
    }
  }
```

Figura 5.94. Control de comunicaciones con la API.

- **usuario_preferences.dart:**

El proceso sirve para controlar cómo se almacena la información del usuario dentro de la aplicación. Con este módulo, se pueden guardar los datos del usuario y los tokens de autenticación después de que el usuario inicia sesión. Esto permite comprobar si el usuario está autenticado y obtener su información cuando se necesite. Esto se muestra en la figura 5.95.

```dart
import 'package:shared_preferences/shared_preferences.dart';

class UsuarioPreferences {

  static Future<void> guardarUsuarioLogin(Map<String, dynamic> usuarioData) async {
    final prefs = await SharedPreferences.getInstance();

    await prefs.setString('usuario_id', usuarioData['id']?.toString() ?? '');
    await prefs.setString('usuario_username', usuarioData['username'] ?? '');
    await prefs.setString('usuario_nombre', usuarioData['nombres'] ?? '');
    await prefs.setString('usuario_apellidos', usuarioData['apellidos'] ?? '');
    await prefs.setString('usuario_mesa', usuarioData['mesa'] ?? '');
    await prefs.setString('usuario_cargo', usuarioData['cargo'] ?? '');
    await prefs.setBool('is_logged_in', true);

    print('✅ Datos de usuario guardados en SharedPreferences');
    print('   Usuario: ${usuarioData['username']}');
    print('   Nombre: ${usuarioData['nombres']} ${usuarioData['apellidos']}');
    print('   Mesa: ${usuarioData['mesa']}');
    print('   Cargo: ${usuarioData['cargo']}');
  }

  static Future<void> guardarTokens({
    required String access,
    required String refresh,
  }) async {
    final prefs = await SharedPreferences.getInstance();
    await prefs.setString('access_token', access);
    await prefs.setString('refresh_token', refresh);
    print('✅ Tokens guardados en SharedPreferences');
  }

  static Future<String?> obtenerAccessToken() async {
    final prefs = await SharedPreferences.getInstance();
    final t = prefs.getString('access_token');
    return (t != null && t.trim().isNotEmpty) ? t : null;
  }

  static Future<String?> obtenerRefreshToken() async {
    final prefs = await SharedPreferences.getInstance();
    final t = prefs.getString('refresh_token');
    return (t != null && t.trim().isNotEmpty) ? t : null;
  }
```

Figura 5.95. Control de la información de usuario.

5.4.4.4 Gestión de autenticación y sesión

La gestión de la sesión y la autenticación se hace en la pantalla de InicioSesion. Aquí, el usuario puede introducir sus credenciales para acceder al sistema. Esto se hace a través del UsuarioRepository.

El proceso de inicio de sesión es el siguiente: la aplicación envía el nombre de usuario y la contraseña al backend. Luego, el backend responde. Si la respuesta es positiva, se almacenan los datos del usuario y los tokens en SharedPreferences. Esto se hace a través del UsuarioPreferences para mantener la sesión activa. Después, el usuario es redirigido al menú principal.

iniciosesion.dart:

El trabajador comienza iniciando sesión con su usuario y contraseña en la pantalla de inicio esta aplicación verifica si estas credenciales son correctas enviándolas al sistema. Esto se puede ver en la figura 5.96.

```dart
import 'package:flutter/material.dart';
import '../repositories/usuario_repository.dart';
import '../utils/usuario_preferences.dart';

class InicioSesion extends StatefulWidget {
  const InicioSesion({super.key});

  @override
  State<InicioSesion> createState() => _InicioSesionState();
}

class _InicioSesionState extends State<InicioSesion> {
  final TextEditingController _usernameController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();
  bool _isLoading = false;
  bool _obscurePassword = true;
  String _errorMessage = '';

  Future<void> _iniciarSesion() async {
    if (_usernameController.text.isEmpty || _passwordController.text.isEmpty) {
      setState(() => _errorMessage = 'Por favor complete todos los campos');
      return;
    }

    setState(() {
      _isLoading = true;
      _errorMessage = '';
    });

    try {
      final resultado = await UsuarioRepository.login(
        username: _usernameController.text.trim(),
        password: _passwordController.text,
      );

      if (resultado['success'] == true) {
        final Map<String, dynamic> usuarioData =
            (resultado['data'] ?? {}) as Map<String, dynamic>;
```

Figura 5.96. Verificación de credenciales correctas.

5.4.4.5 Consumo de la API REST

El consumo de la API REST se realiza mediante una arquitectura organizada. Esta arquitectura se basa en el ApiClient y en los distintos repositories. El ApiClient es el encargado de centralizar todas las peticiones HTTP, como GET, POST, PUT y DELETE. Además, construye las rutas a partir de una URL base y controla automáticamente los headers. Incluye el token de autenticación cuando es necesario.

Los repositories utilizan este cliente para enviar peticiones al servidor y procesar las respuestas. Transforman los datos JSON en estructuras que la aplicación puede manejar. También se controla el manejo de errores, los códigos de estado y la renovación automática del token en caso de que expire.

De esta manera, la comunicación entre la aplicación móvil y el backend es segura y está estructurada. Además, es mantenible y permite integrar de forma eficiente los servicios REST del sistema.

5.4.4.6 Integración con impresora de etiquetas

La conexión con la impresora se hace creando etiquetas con códigos QR a partir de la variedad, la mesa y la medida que se eligen. Se obtiene esta información y se saca mediante una aplicación de programación a través de las partes correspondientes. La aplicación crea un identificador

único, de modo que el código QR se genera automáticamente. Luego, se obtiene la etiqueta como imagen a través de un controlador de capturas de pantalla y se guarda en el dispositivo, y después se comparte mediante una herramienta de compartir, permitiendo la opción de enviar directamente la etiqueta a la impresora compatible. De esta manera, la impresión de etiquetas a partir de la información de la variedad, la mesa y la medida correspondientes permite crear e imprimir etiquetas rápidamente para identificar y rastrear la producción.

etiquetas.dart:

El proceso permite crear etiquetas con códigos QR dentro de la aplicación. Primero, el usuario elige la variedad, la mesa y la medida. Luego, el sistema crea automáticamente un código QR con esta información. Este código QR se puede compartir o imprimir como una etiqueta para usar durante el proceso de trabajo. En la figura 5.97 se puede ver cómo funciona este proceso.

```dart
import 'dart:io';
import 'package:flutter/material.dart';
import 'package:qr_flutter/qr_flutter.dart';
import 'package:screenshot/screenshot.dart';
import 'package:path_provider/path_provider.dart';
import 'package:share_plus/share_plus.dart';

import 'repositories/mesa_repository.dart';
import 'repositories/variedad_repository.dart';

class Etiquetas extends StatefulWidget {
  const Etiquetas({super.key});

  @override
  State<Etiquetas> createState() => _EtiquetasState();
}

class _EtiquetasState extends State<Etiquetas> {
  final ScreenshotController screenshotController = ScreenshotController();

  String? _variedadSeleccionada;
  String? _mesaSeleccionada;
  String? _medidaSeleccionada;

  String _qrData = "Esperando datos...";

  List<String> _variedades = [];
  bool _cargandoVariedades = true;

  List<String> _mesas = [];
  bool _cargandoMesas = true;

  final List<String> _medidas = ['40', '50', '60', '70', '80', '90', 'OPE'];
  Future<void> _cargarVariedades() async {
    setState(() => _cargandoVariedades = true);

    final res = await VariedadRepository.obtenerVariedades();

    if (res['success'] == true) {
      final variedades = (res['variedades'] as List)
          .map((v) => v.nombre.toString())
          .toList();

```

Figura 5.97. Conexión de impresora y generación de códigos Qr.

5.5 FASE DE PRUEBAS

5.5.1 Aplicación web

5.5.1.1 Prueba Unitarias

Tabla 5.9.Pruebas Unitarias de la Aplicación Web.

ID	Módulo / Función	Caso de prueba	Datos de entrada	Resultado esperado	Resultado obtenido	¿Aceptado? (Sí/No)
PU -01	Login web	Acceso administrador	Usuario admin contraseña válida.	El administrador accede al sistema.	El acceso fue correcto	**SI**
PU -02	Login web	Acceso administrador	Credenciales inválidas.	El administrador no accede al sistema.	El acceso fue incorrecto.	SI
PU -03	Login web	Acceso administrador	Credenciales de usuarios trabajadores.	El usuario no accede al sistema.	Accedo denegado.	SI
PU -04	Crear usuario	Registrar nuevo trabajador	Nombres completos, Apellidos completos usuario y contraseña.	El usuario se guarda en BD.	El usuario fue registrado.	SI
PU -05	Crear usuario	Registrar nuevo trabajador	Campos incompletos	El sistema pide ingresar campos completos.	El usuario no fue registrado.	SI
PU -06	Crear mesa	Registrar nueva mesa	ID mesa 01	La mesa se registra en el sistema.	La mesa fue creada correctamente.	SI

ID	Módulo / Función	Caso de prueba	Datos de entrada	Resultado esperado	Resultado obtenido	¿Aceptado? (Sí/No)
PU-07	Crear mesa	Registrar nueva mesa	Campos incompletos	La mesa no es registrada.	La mesa no fue creada.	SI

5.5.1.1 Prueba de Aceptación

Tabla 5.10.Pruebas de aceptación de la Aplicación Web.

ID	Historia de usuario	Escenario	Pasos a realizar	Resultado esperado	Resultado obtenido	¿Aceptado? (Sí/No)
PA-01	HU-M01	Inicio de sesión trabajador	1 abrir app, 2 Ingresar usuario, 3 Ingresar contraseña, 4 Presionar iniciar.	El trabajador accede a la app.	Acceso realizado correctamente.	SI
PA-02	HU-M02	Registro inicio jornada	1 iniciar sesión, 2 Presionar iniciar jornada.	La hora de inicio se registra.	La hora registrada correctamente.	SI
PA-03	HU-M03	Registro fin jornada	1 presionar finalizar jornada.	El sistema guarda hora de salida.	La hora registrada correctamente.	SI
PA-04	HU-M04	Visualizar rendimiento	1 iniciar sesión, 2 Abrir	Se muestran bonches y mesa.	Información visible correctamente.	SI

ID	Historia de usuario	Escenario	Pasos a realizar	Resultado esperado	Resultado obtenido	¿Aceptado? (Sí/No)
			módulo rendimiento.			
PA-05	HU-M05	Generar etiquetas	1 seleccionar mesa 2 Generar etiqueta	El sistema crea QR único	QR generado correctamente	SI
PA-06	HU-M06	Visualizar rendimiento	1 iniciar sesión, 2 Abrir módulo disponibilidad.	Se muestran bonches de cada variedad.	Información visible correctamente	SI

5.5.2 Listener

5.5.2.1 Pruebas unitarias

Tabla 5.11.Pruebas Unitarias del Listener

ID	Módulo / Función	Caso de prueba	Datos de entrada	Resultado esperado	Resultado obtenido	¿Aceptado? (Sí/No)
PU-01	Escaneo QR con Escáner 1.	Registrar bonche y se incrementar rendimiento y disponibilidad	Escaneo de QR válido con escáner 1	El sistema registra el bonche.	Bonche registrado correctamente.	SI
PU-02	Escaneo QR con Escáner 1.	Registrar bonche	Escaneo de QR válido	El sistema no registra el bonche.	Bonche se registra.	SI

ID	Módulo / Función	Caso de prueba	Datos de entrada	Resultado esperado	Resultado obtenido	¿Aceptado? (Sí/No)
			con escáner 1			
PU-03	Escaneo QR con Escáner 2.	Reducir disponibilidad	Escaneo de QR válido con escáner 2.	La disponibilidad disminuye.	La disponibilidad se redujo correctamente.	SI

5.5.2.2 Pruebas de aceptación

Tabla 5.12.Pruebas de Aceptación del Listener.

ID	Historia de usuario	Escenario	Pasos a realizar	Resultado esperado	Resultado obtenido	¿Aceptado? (Sí/No)
PA-01	HU-M014	Escaneo con escáner 1.	1 escanear QR.	Se incrementa el rendimiento.	El rendimiento y disponibilidad aumentó.	SI
PA-02	HU-M15	Escaneo con escáner 2.	1 escanear QR para salida.	Se reduce disponibilidad.	El stock se redujo.	SI

5.5.3 Aplicación Móvil

5.5.3.1 Prueba Unitarias

Tabla 5.13.Pruebas Unitarias de la Aplicación Móvil.

ID	Módulo / Función	Caso de prueba	Datos de entrada	Resultado esperado	Resultado obtenido	¿Aceptado? (Sí/No)
PU-01	Login móvil	Verificar acceso con credenciales correctas.	usuario: Nancy contraseña: 123456.	El sistema permite el acceso a la aplicación.	El sistema permitió el acceso correctamente.	SI
PU-02	Login móvil	Verificar acceso con credenciales incorrectas	usuario: Nancy contraseña: 00000.	El sistema muestra mensaje de error.	El sistema no permitió el ingreso.	SI
PU-03	Inicio de jornada	Registrar inicio de jornada	Presionar botón iniciar jornada.	El sistema registra la hora de inicio.	La hora se registró correctamente.	SI
PU-04	Inicio de jornada	Validar que no se pueda iniciar jornada duplicada.	Intentar iniciar jornada dos veces.	El sistema bloquea el segundo intento.	El sistema cambió de iniciar a finalizar.	SI
PU-05	Fin de jornada	Registrar fin de jornada	Presionar botón finalizar jornada.	El sistema registra hora de salida	La hora se registró correctamente	SI

5.5.3.2 Prueba de Aceptación

Tabla 5.14.Pruebas de Aceptación de la Aplicación Móvil.

ID	Historia de usuario	Escenario	Pasos a realizar	Resultado esperado	Resultado obtenido	¿Aceptado? (Sí/No)
PA-01	HU-M08	Crear usuario	1 ingresar al panel admin, 2 Registrar trabajador.	El usuario se crea y puede iniciar sesión en la app móvil.	Usuario registrado correctamente.	SI
PA-02	HU-M09	Crear mesa	1 acceder módulo mesas, 2 Registrar nueva mesa.	La mesa queda disponible.	Mesa registrada correctamente.	SI
PA-03	HU-M10	Gestión de stock	1 agregar variedad, 2 Definir stock.	El sistema guarda disponibilidad.	Stock registrado correctamente.	SI

5.5.4 Despliegue

5.5.4.1 Web

5.5.4.1.1 Configuración del proytecto para producción

Para subir la aplicación al servidor, es necesario preparar el proyecto para producción.

En el archivo settings.py se agregó la lectura de variable de entorno que será DATABASE_URL=os.getenv("DATABASE_URL", "").strip() como se puede ver en la figura 5.98.

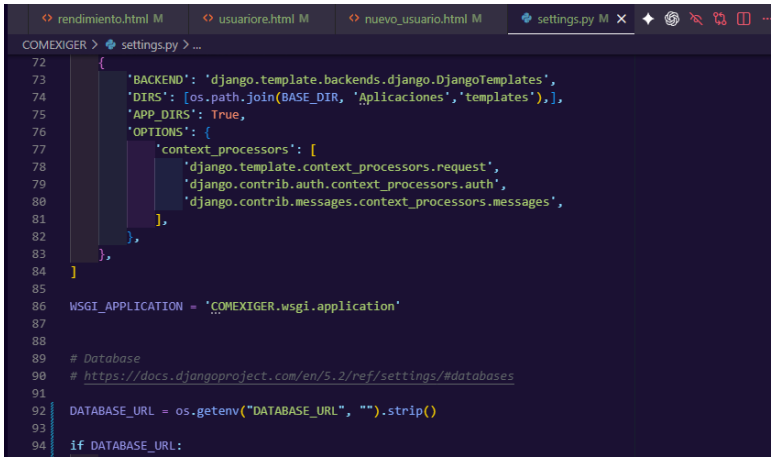


Figura 5.98. Fragmento de código de la configuración realizada.

Se configuro las variables de entorno en el mismo archivo como lo son SECRET_KEY, DEBUF, ALLOWED_HOSTS estas salen de env en la figura 5.99 se muestran los detalles.

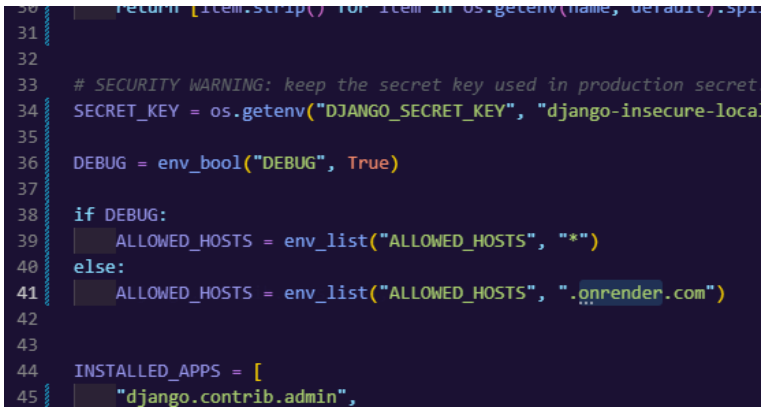


Figura 5.99.Fragmento de código de las variables de entorno.

También para WebSockets y dejarlo listo para los siguientes pasos si existe REDIS_URL, usa channels_redis y en la figura 5.100 se analiza el resultado.

```
]

ASGI_APPLICATION = "COMEXIGER.asgi.application"

REDIS_URL = os.getenv("REDIS_URL", "").strip()
if REDIS_URL:
    CHANNEL_LAYERS = {
        "default": {
            "BACKEND": "channels_redis.core.RedisChannelLayer",
            "CONFIG": {
                "hosts": [REDIS_URL],
            },
        }
    }
else:
    CHANNEL_LAYERS = {
```

Figura 5.100.Fragmento de código de la configuración para WebSocket.

Los estáticos se activó ASGIStaticFilesHandler y también en prod con SERVE_STATIC=true como la figura 5.101.

```
serve_static = os.getenv("SERVE_STATIC", "true").strip().lower() in {"1", "true", "yes", "on"}
if settings.DEBUG or serve_static:
    django_asgi_app = ASGIStaticFilesHandler(django_asgi_app)

application = ProtocolTypeRouter({
    "http": django_asgi_app,

    "websocket": AuthMiddlewareStack(
        URLRouter(
            Aplicaciones.Rendimiento.routing.websocket_urlpatterns
            + Aplicaciones.Disponibilidad.routing.websocket_urlpatterns
        )
    ),
})
```

Figura 5.101.Fragmento de código de la configuración para servir estáticos.

Luego se creó el archivo render.yaml aquí se definió cómo render construye e inicia la app el contenido del mismo se puede ver en la figura 5.102.

```
services:
  - type: web
    name: comexiger-web
    runtime: python
    plan: starter
    buildCommand: "pip install -r requirements.txt && python manage.py collectstatic --noinput"
    startCommand: "daphne -b 0.0.0.0 -p $PORT COMEXIGER.asgi:application"
    envVars:
      - key: PYTHON_VERSION
        value: 3.12.8
      - key: DJANGO_SECRET_KEY
        generateValue: true
      - key: DEBUG
        value: "false"
      - key: DATABASE_SSL_REQUIRE
        value: "true"
      - key: SERVE_STATIC
        value: "true"
      - key: ALLOWED_HOSTS
        value: ".onrender.com"
      - key: CSRF_TRUSTED_ORIGINS
        value: "https://*.onrender.com"
```

Figura 5.102. Archivo render.yaml creado.

Con el siguiente comando se generó el archivo requirements en donde se muestran las dependencias necesarias para el despliegue de la app; las dependencias se pueden ver en la figura 5.103.

Figura 5.103.Archivo requirements.txt para producción.

5.5.4.1.2 Configuración del servidor

En este punto se contemplaron las siguientes características para ponerlo en producción en Render, más adelante se presentan en la tabla 5.15.

Tabla 5.15. Especificaciones de los servicios utilizados en Render.

Servicio	CPU	RAM	Almacenamiento	Justificación
Web Service	0.5 CPU	512MB	Administrado	Este servicio se utiliza para ejecutar la aplicación web y el backend del sistema. Se eligió este plan porque ofrece los recursos necesarios para el funcionamiento del sistema sin generar un costo elevado.
PostgreSQL	0.1 CPU	256MB	5GB iniciales	PostgreSQL se utiliza para almacenar toda la información que genera el sistema, como los datos de los usuarios y los registros de los procesos. Se estableció una

Servicio	CPU	RAM	Almacenamiento	Justificación
				capacidad inicial de 5 GB para garantizar que haya suficiente espacio para guardar los datos y permitir que la información crezca a medida que se use el sistema.
Key Value	-	256MB	Administrado	Este servicio permite guardar datos temporales en la memoria, lo que facilita que ciertas operaciones se realicen más rápidamente. Al utilizarlo, se mejora el rendimiento del sistema y se pueden manejar múltiples solicitudes al mismo tiempo sin afectar directamente la base de datos principal.

- **Creación de la base de datos**

Ahora dentro del servidor crearemos la base de datos en. Ingresamos a render en caso de no tener una cuenta iniciamos sesión o nos registramos con Google o GitHub mostrado en la figura 5.104.

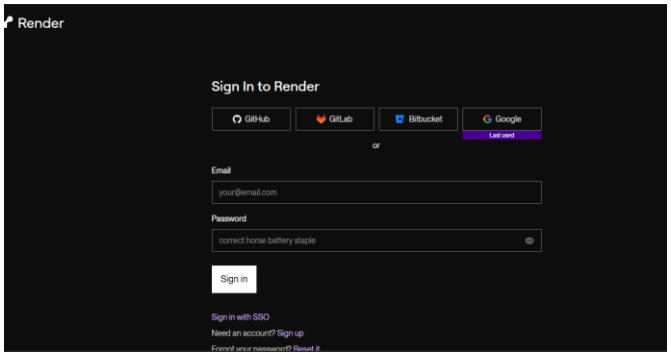


Figura 5.104. Modo de inicio de sesión en Render.

Se inicia sesión en la cuenta que ya tenemos creada como se puede ver en la figura 5.105.

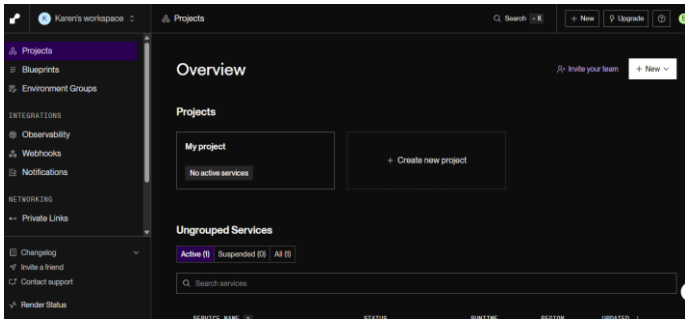


Figura 5.105.Inicio de sesión en Render.

Una vez aquí seleccionamos New y seleccionamos PostgreSQL como se ve en la figura 5.106.

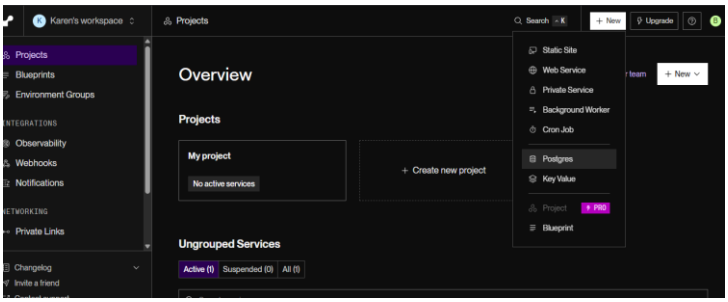


Figura 5.106. Creación de la base de datos en Render.

Ahora llenaremos los datos necesarios para la creación de nuestra base de datos como se muestra en la figura 5.107.

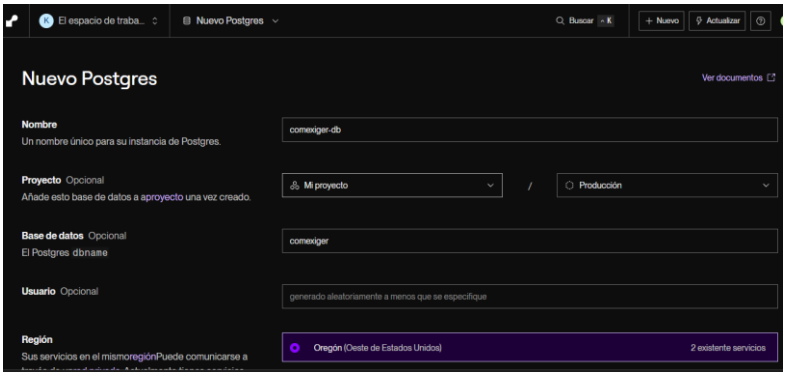


Figura 5.107. Campos a llenar en Render.

Aquí solo se llena el campo de la versión de PostgreSQL y lo demás por defecto tal y como se observa en la figura 5.108.

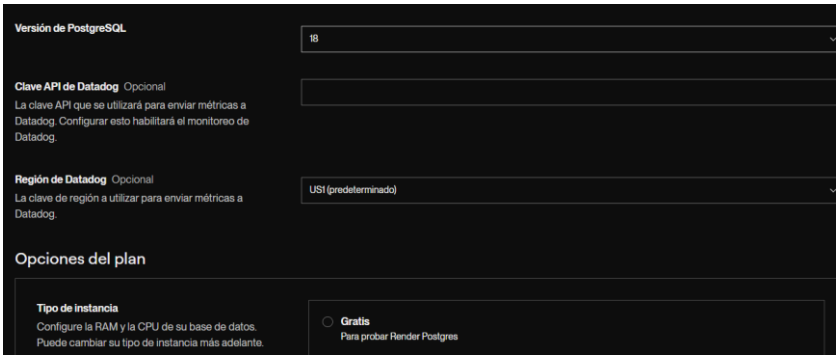


Figura 5.108. Selección de versión de PostgreSQL.

En este campo elegiremos ese Basic-256mb de 256 MB (RAM) 0.1 CPU visto la figura 5.109.

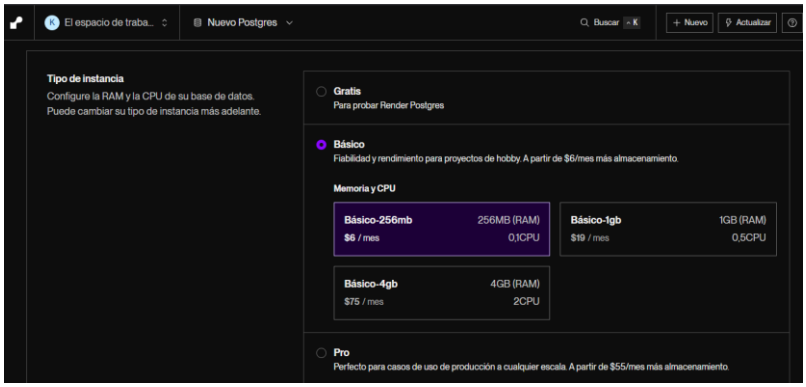


Figura 5.109.Selección del plan.

Este paso es muy importante ya que hay que considerar el espacio con el que podemos empezar que será 5 y dejamos activado Storage porque si la base llega cerca del límite, Render sube el storage automáticamente; el siguiente campo lo dejamos por defecto visualizado en la figura 5.110.

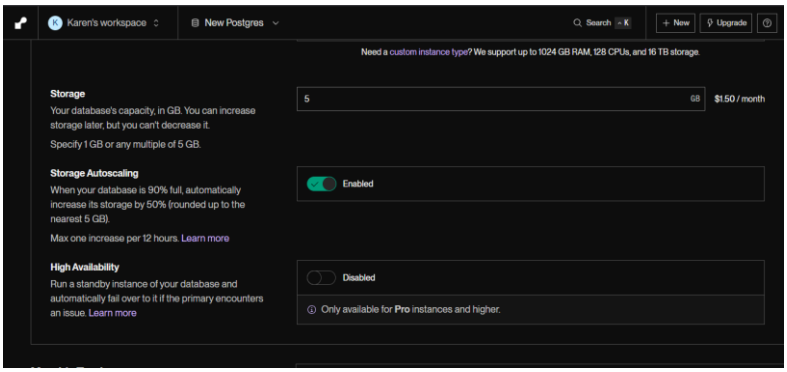


Figura 5.110. Selección de almacenamiento.

Ahora pagaremos agregando una tarjeta una vez hecha eso se mostrará la siguiente interfaz como la figura 5.111.

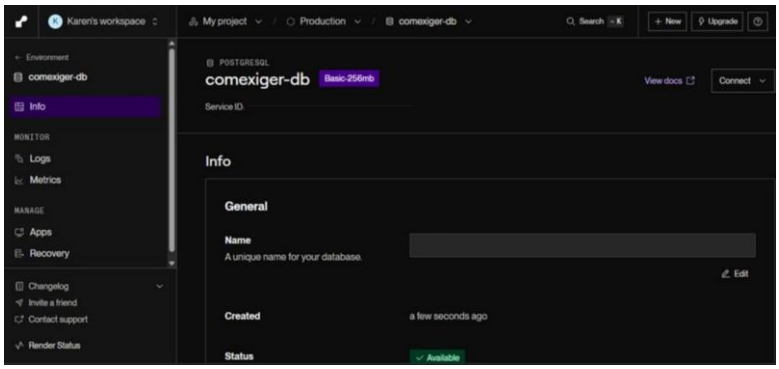


Figura 5.111. Proceso de facturación en Render.

- **Creación de Key Value**

Este ayuda a la estabilización del Websokect, se realizó en el mismo proceso para crear, pero aquí se elige Key Value como en la figura 5.112.

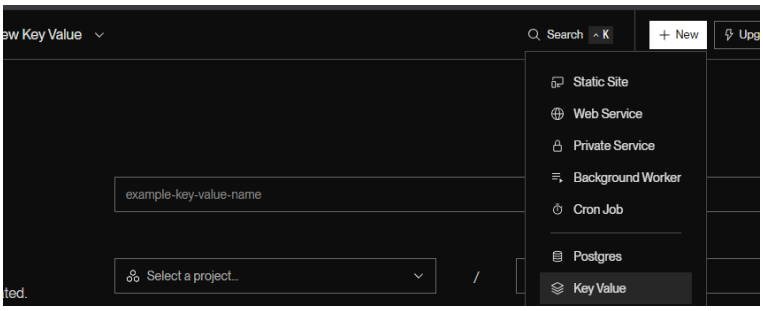


Figura 5.112. Selección de Key Value.

Una vez en aquí la mayoría van por defecto como en la figura 5.113.

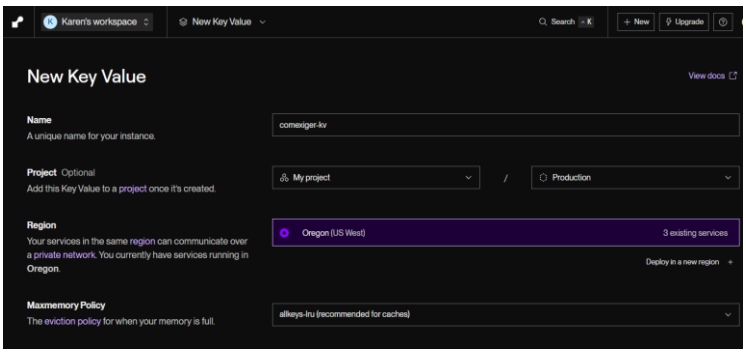


Figura 5.113. Formulario para Key Value.

Aquí se seleccionó el plan y generamos el cual es de las características propuestas en la figura 5.114.

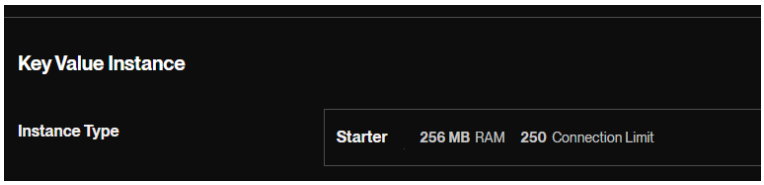


Figura 5.114.Selección del plan para Key Value.

Se copió también el internal de aquí el cual se mostrará tal y como esta en la figura 5.115.

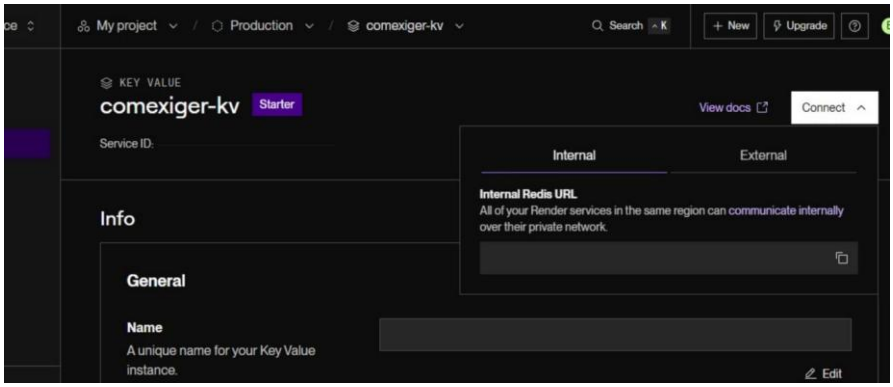


Figura 5.115.Link de la Key Value.

- **Publicación en entorno productivo**

Ahora en la base de datos creada en Render se selecciona Connect y se copió Internal para después agregarlo al servicio web que se muestra en la figura 5.116.

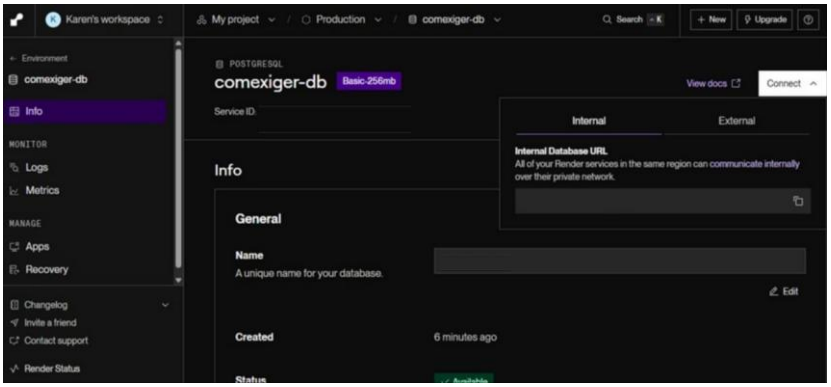


Figura 5.116. Link de la base de datos.

Se seleccionó New nuevamente y se eligió Web Service como en la figura 5.117.

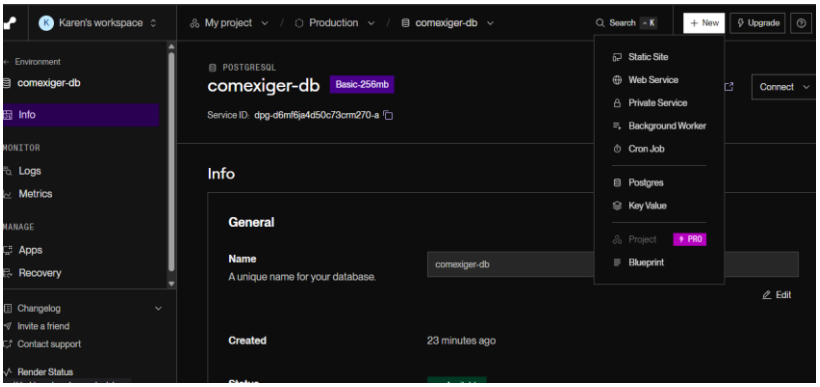


Figura 5.117.Creación del Servicio Web.

Aquí se escogió el proyecto previamente subido a GitHub visualizado en la figura 5.118.

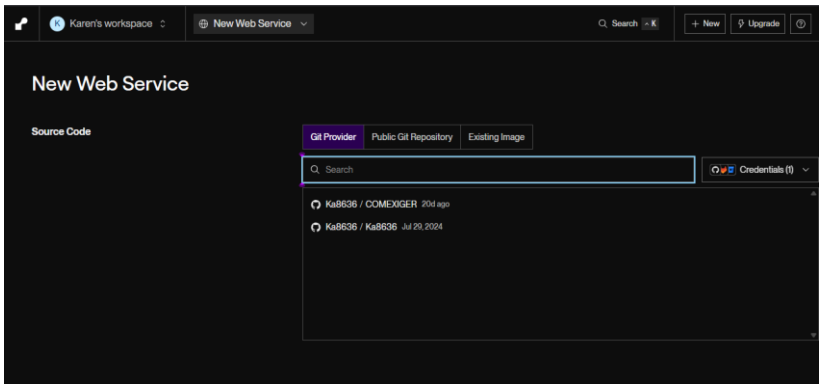


Figura 5.118.Selección del proyecto subido en GitHub.

Se llenó los siguientes campos de acorde al proyecto como se visualiza en la figura 5.119.

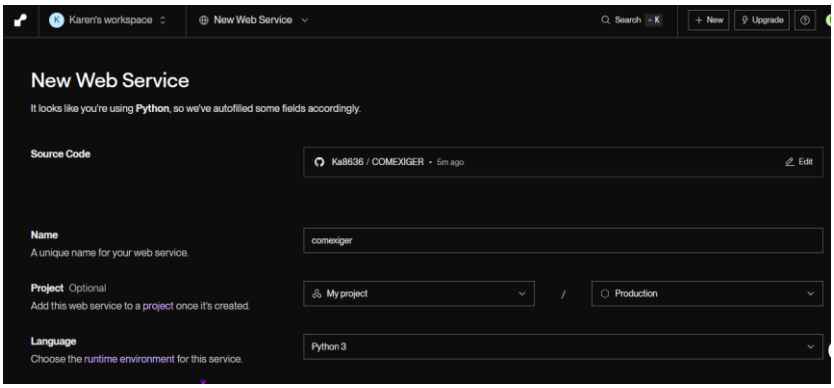


Figura 5.119. Formulario para Web Service.

En los siguientes campos se agregaron los comandos necesarios ya que se tomó en cuenta porque se usó Daphne tales vistos en la figura 5.120.

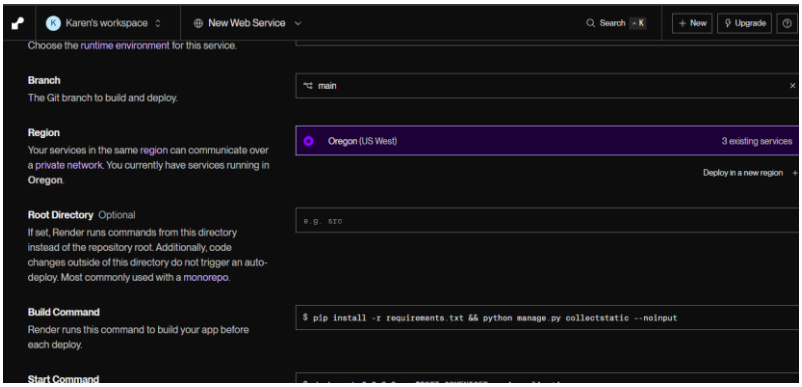


Figura 5.120.Comandos para el correcto despliegue en Render.

En la figura se muestra el plan seleccionado en la figura 5.121.

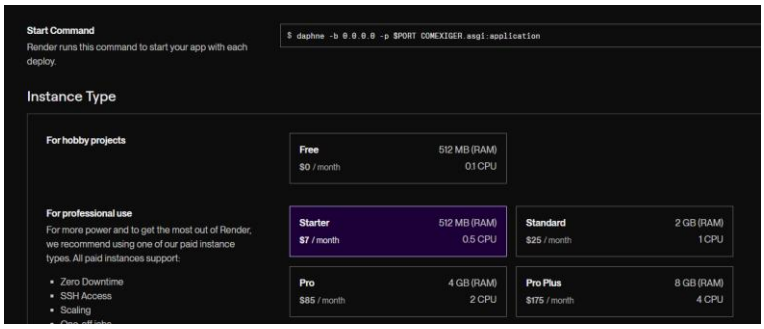


Figura 5.121.Selección del plan.

En la siguiente figura 5.122 se muestran las variables usadas y necesarias para el despliegue.

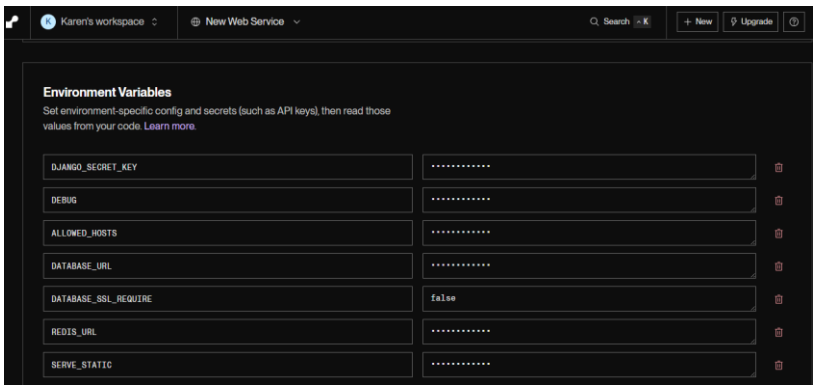


Figura 5.122.Variables para el servicio Web.

Ahora en Advanced solo se cambiaron los campos que se ven en la figura 5.123.

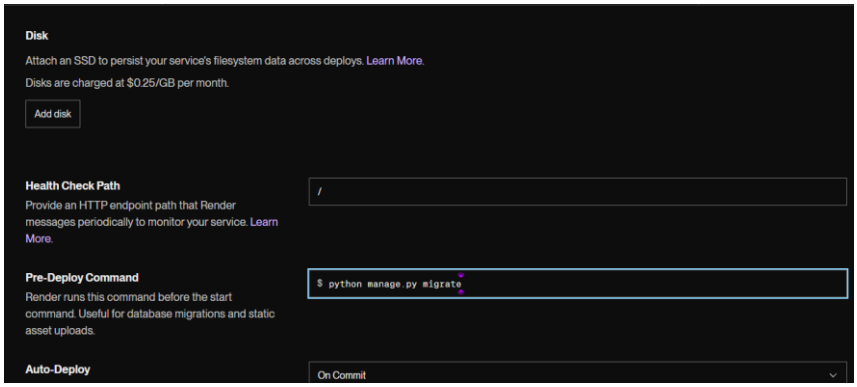


Figura 5.123.Cambio de en los campos.

Se selecciono generar y empezó el proceso como se puede ver en la figura 5.124.

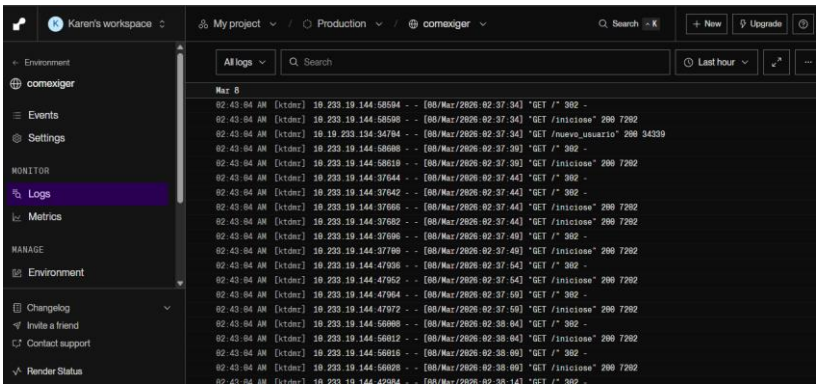


Figura 5.124. Logs del server.

Una vez finalizado se mostró un mensaje de éxito y un icono que muestra que la aplicación se subió con éxito tal como se puede ver en la figura 5.125. Datos como el link del sistema se encuentran ahí.

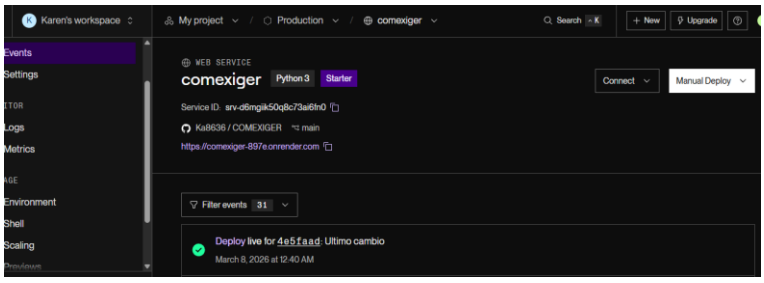


Figura 5.125. Despliegue completo.

Se copió el link e ingresó y se evidenció en la figura 5.126 que concluye con éxito.

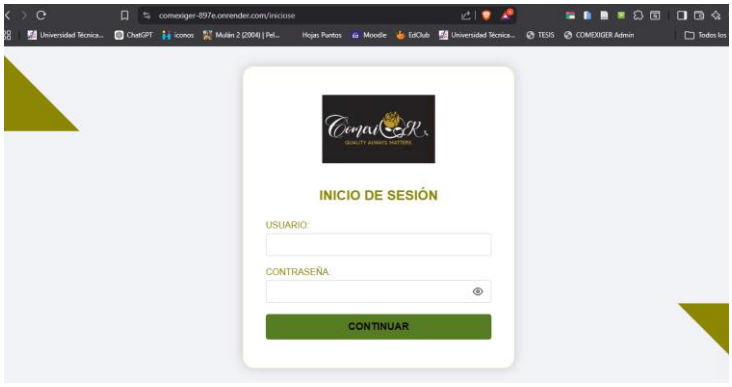


Figura 5.126.Confirmación de despliegue.

5.5.4.2 Móvil

Para el siguiente despliegue se necesita de una cuenta de desarrollador en play store para empezar la producción de la aplicación.

5.5.4.2.1 Preparación para publicación

- **Política de Privacidad de la App**

La aplicación tiene una política de privacidad diseñada para mantener a los usuarios informados sobre cómo se utilizan y protegen sus datos personales. Siguiendo las directrices de Google Play Store, cualquier aplicación que recolecte información debe proporcionar un enlace a su política de privacidad. Este proceso para acceder y ver la política de privacidad se muestra en la figura 5.127.

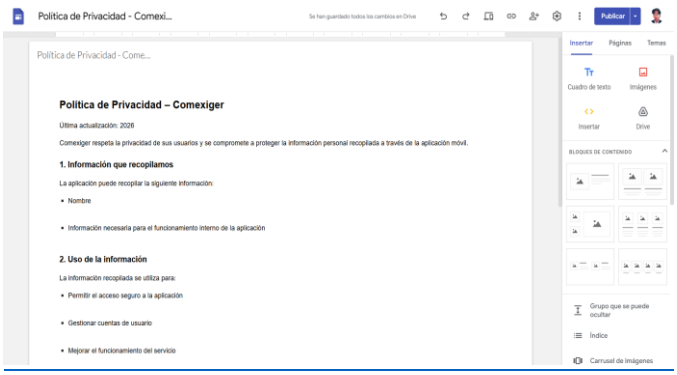


Figura 5.127. Política de privacidad de Google.

Por esta razón, se ha creado una página web que detalla qué datos recoge la aplicación, cómo se utilizan y de qué manera se protege la información de los usuarios. Este acceso a la política de privacidad y su visualización se refleja en la figura 5.128.

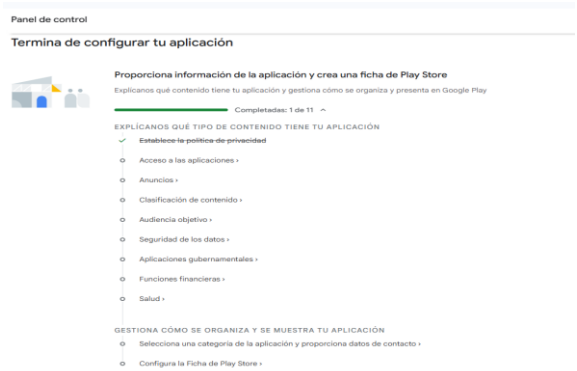


Figura 5.128. Acceso a la política de privacidad.

Ahora incluimos nuestras políticas que creamos en google sites como lo representa la figura 5.129.

Figura 5.129. Insertar link de página de políticas y privacidad.

5.5.4.2.2 Público objetivo de la Aplicación

En la Google Play Console, definimos quién puede usar nuestra aplicación. Seleccionamos la opción “Mayores de 18 años". Esto significa que nuestra aplicación está diseñada para adultos. Esta configuración se muestra en la figura 5.130.

Figura 5.130. Definición de quien usa la aplicación.

Esta configuración es necesaria para seguir las reglas de publicación de Google en la Google Play Store, como se observa en la figura 5.131.

Figura 5.131. Configuración necesaria para publicación.

5.5.4.2.3 Seguridad y Gestión de Datos

En esta parte de la Google Play Console, hay información sobre la privacidad y cómo se manejan los datos de los usuarios, como se ilustra en la figura 5.132.

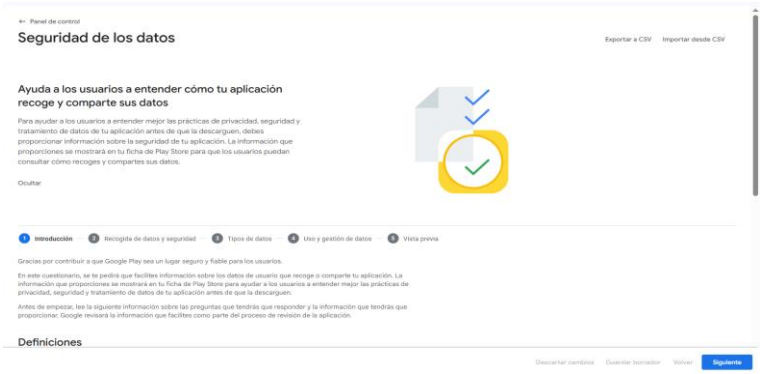


Figura 5.132. Manejo de información sobre privacidad de usuario.

La aplicación recopila información básica, como el nombre y lo que los usuarios hacen dentro de la aplicación, y no la comparte con nadie más, tal como se muestra en la figura 5.133.

Figura 5.133. Recopilación de información básica.

Esta información se muestra a los usuarios y a los consumidores en la página de la tienda para que sepan cómo se usan sus datos, como se representa en la figura 5.134.

Figura 5.134. Información mostrada a usuarios.

5.5.4.2.4 Configuración de la Ficha en Play Store

En esta sección de Google Play Console, configuramos la información básica de la aplicación para publicarla en Google Play Store. Esto se debe a que su función principal es ayudar a gestionar procesos empresariales. También registramos un correo electrónico de contacto: comexiger@gmail.com. De esta forma, los usuarios pueden comunicarse con el desarrollador si lo necesitan, como se ilustra en la figura 5.135.

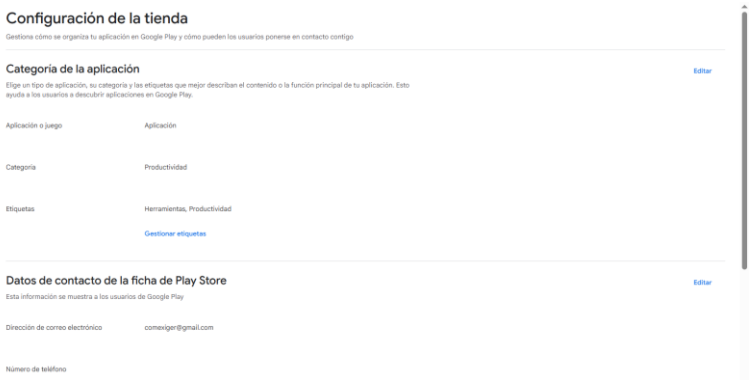


Figura 5.135. Nueva configuración de información básica para publicación.

5.5.4.2.5 Fichero de la Aplicación en Google Play

En la Google Play Console, configuramos el fichero de la aplicación que los usuarios verán en la Google Play, como se ilustra en la figura 5.136.

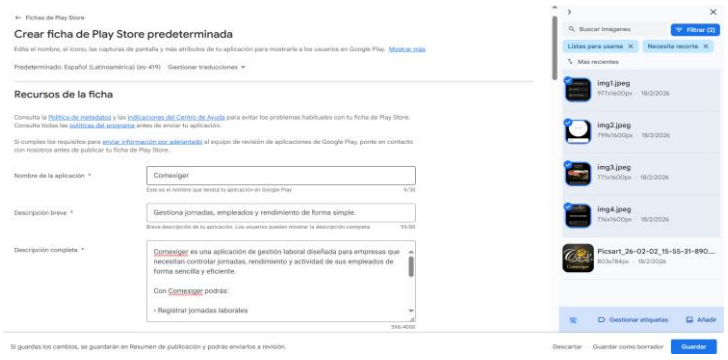


Figura 5.136. Configuración de fichero de aplicación.

Añadimos el nombre de la aplicación, una descripción breve y una descripción completa y capturas de pantalla, como podemos observar en la figura 4.137.

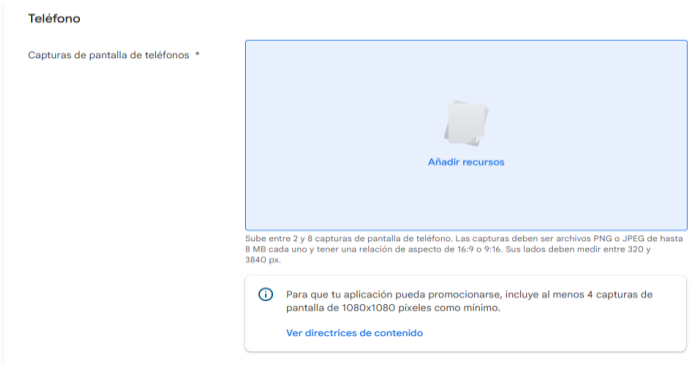


Figura 5.137. Descripción y captura de pantalla.

También subimos capturas de pantalla que muestran cómo funciona el sistema. De esta manera, los usuarios pueden ver las funciones principales de la aplicación antes de descargarla, así como se representa en la figura 5.138.

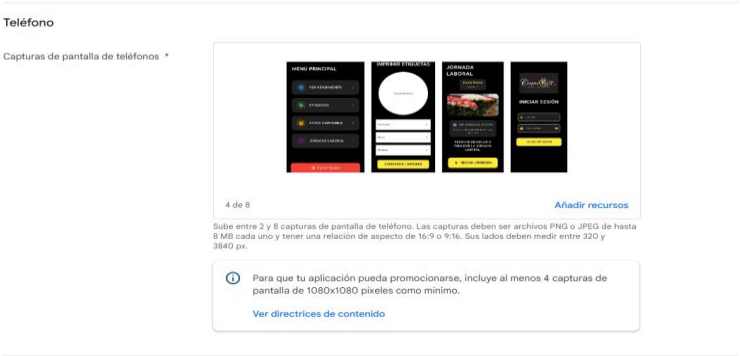


Figura 5.138. Capturas de pantalla de funcionalidad.

5.5.4.2.6 Pruebas Cerradas de la App

En la sección de la Google Play Console, creamos una versión Beta para pruebas cerradas, que también se conoce como Alpha. La idea es poner a prueba la aplicación antes de lanzarla oficialmente en la Google Play Store, como se representa en la figura 5.139.

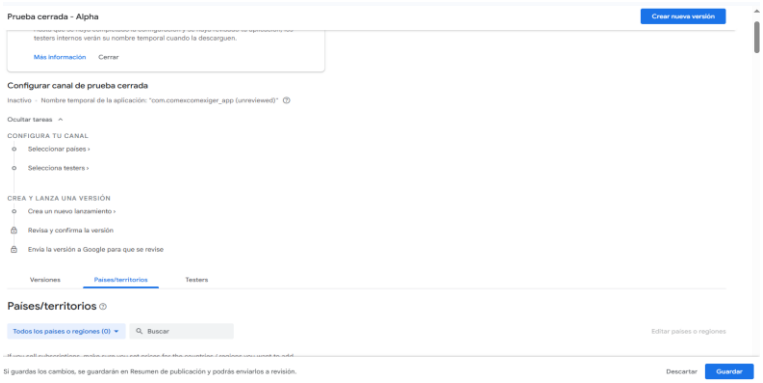


Figura 5.139. Creación de versión beta.

Creamos una versión especial para estas pruebas y elaboramos una lista de personas que la probarán. Para ello, utilizamos sus direcciones de correo electrónico, lo que nos permite seleccionar a los usuarios que descargarán y probarán la aplicación. Queremos asegurarnos de que todo funcione a la perfección antes de solicitar que la revisen, como se muestra en la figura 5.140.

Figura 5.140. Añadir lista de testers.

Elaboración de la versión de prueba de la aplicación

En esta parte de la Google Play Console, creamos una versión de prueba cerrada de la aplicación. También configuramos el nombre de la versión y las notas de la actualización, como se ilustra en la figura 5.141.

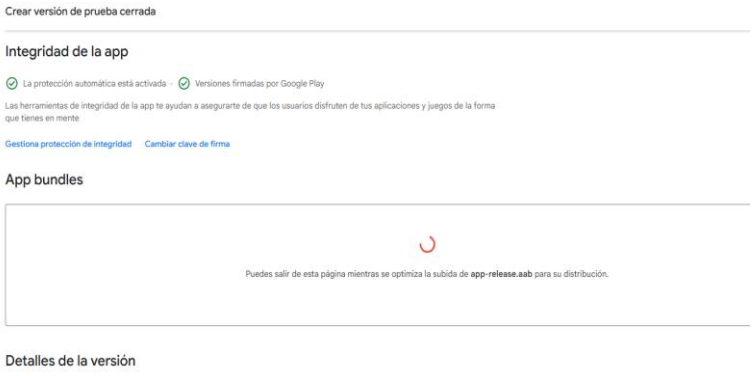


Figura 5.141. Creación de prueba cerrada.

Subimos el archivo App Bundle, que es un archivo con extensión .aab, para que se pueda distribuir en pruebas, como se ilustra en la figura 5.142.

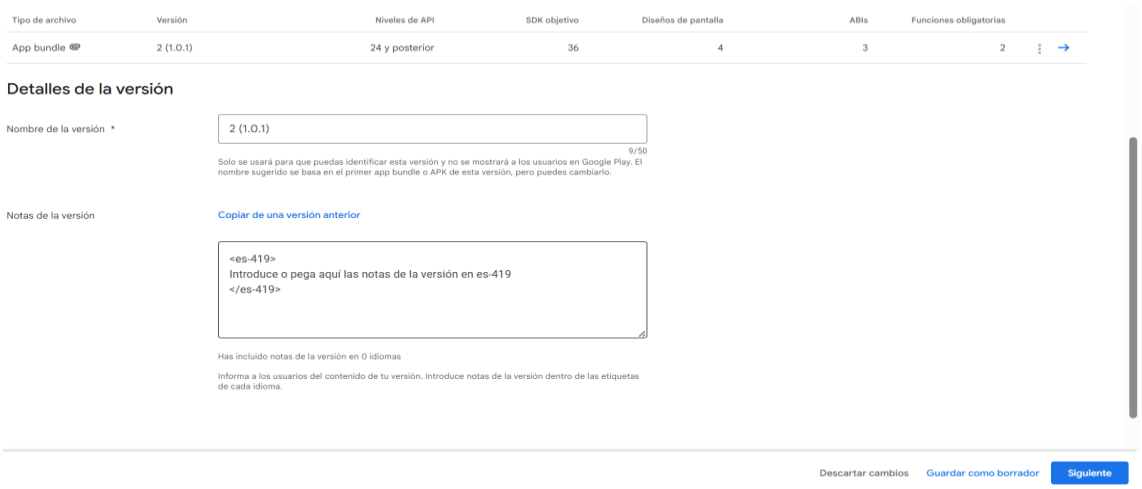


Figura 5.142. Añadir archivo bundle.

De esta manera, la aplicación optimizada está lista para ser revisada y publicada después en la Google Play Store, como se representa en la figura 5.143.

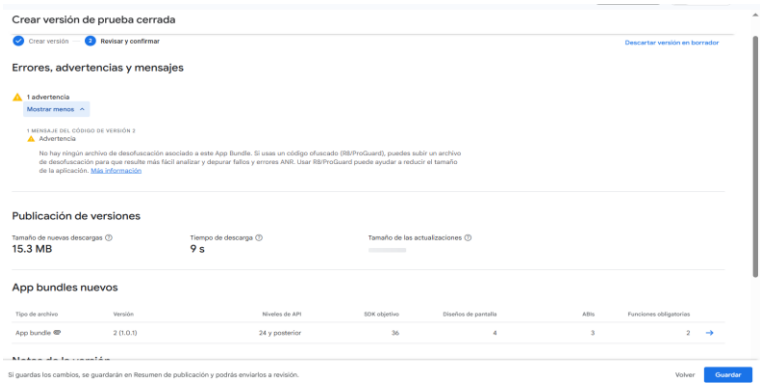


Figura 5.143. Aplicación optimizada y lista para publicación.

Resumen de Publicación de la Aplicación

En la Google Play Console, hay una sección donde podemos ver un resumen con los cambios que hemos hecho en la aplicación antes de enviarla para su revisión, como se muestra en la figura 5.144.

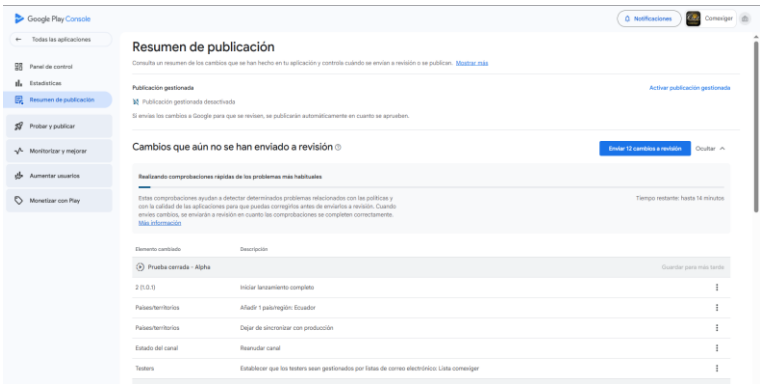


Figura 5.144. Sección de resumen de cambios.

En esta sección de la Google Play Console, podemos verificar algunas cosas importantes como las versiones de la aplicación, los países donde la aplicación estará disponible y el canal de pruebas. Esto se muestra en la figura 5.145.

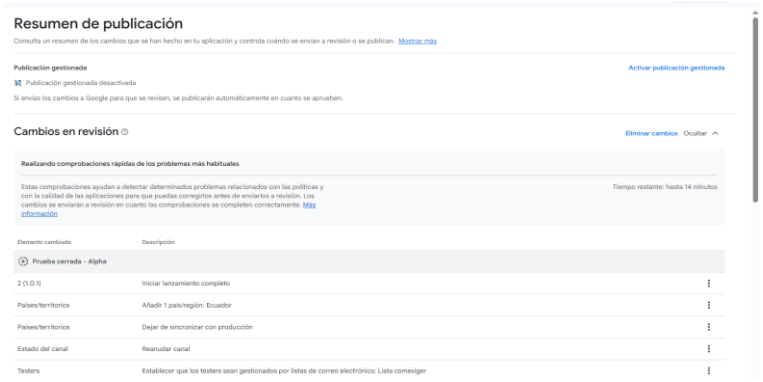


Figura 5.145. Verificación de requisitos para revisión.

Requisitos para Publicar en Producción

Antes de solicitar acceso a producción, es necesario realizar una prueba cerrada. En esta prueba, se requiere un mínimo de 12 personas que actúen como evaluadores durante al menos 14 días. Esto permite verificar que la aplicación funciona correctamente antes de su publicación, como se observa en la figura 5.146.

Figura 5.146. Requisitos para publicación.

5.5.4.3 Sistema de control implementado

Una vez que el trabajador deje los bonches en el lugar en donde realiza el corte se dirige a el área en donde se encuentra la impresora una vez ahí ingresa a la aplicación móvil y generará las etiquetas necesarias mostrado en la figura 5.147.

Figura 5.147. Impresión de etiquetas.

Reemplazando las pinzas y el identificador que se realizaba a mano, se agregó una etiqueta de un código QR en cada bonche que contiene toda la información necesaria. Esta etiqueta será agregada por la persona que trae los bonches como se ve en la figura 5.148.

Figura 5.148. Pegado de etiquetas en los bonches.

Ahora lo escaneará con el escáner 1 ya que este es el que suma esto automáticamente se agrega al sistema con su respectiva mesa ya que este es el identificador de cada trabador también la información necesaria la cual viaja en el código QR. Esta información se agrega tanto en rendimiento como en disponibilidad visto en la figura 5.149.

Figura 5.149. Escaneo de los bonches.

Una vez cortado y con capuchón guardan la flor y ahí es donde el escáner 2 realizará su función en el caso de enviar pedidos se escanean los bonches y estos serán restados en disponibilidad para tener al día la totalidad de la misma. Todos estos datos se visualizan y se pueden consultar tanto en la aplicación móvil como en la web con una clara diferencia visto en la figura 5.150.

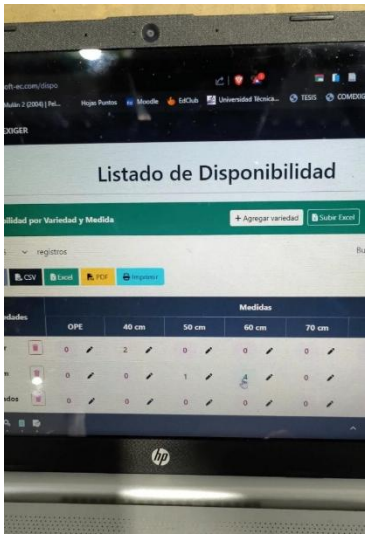


Figura 5.150. Visualización de resultados de Disponibilidad en la aplicación Web.

Como en la figura 5.151 la aplicación móvil es única y exclusivamente para los trabajadores ya que ahí pueden visualizar su rendimiento y la disponibilidad dicha información no podrá ser manipulada por ellos.

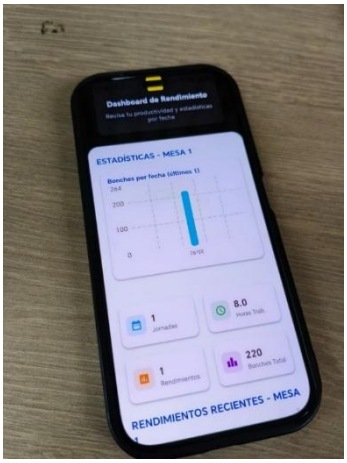


Figura 5.151.Visualización de resultados de Rendimiento en la aplicación Móvil.

5.6 RESULTADOS

En la tabla 5.16 se muestran los resultados del Método manual y el sistema implementado

Tabla 5.16.Resultados del sistema implementado.

Etapa del proceso	**Método manual**	**Tiempo Aproximado**	**Sistema propuesto**	**Tiempo aproximado**
1.- Identificación del bonche	La embonchadora colocan las pinzas manualmente en los bonches durante la jornada.	0 min	El trabajador del Área de control imprime las etiquetas con código QR desde el sistema.	1 min
2.-	Se retiran las pinzas de los bonches	4 min	El trabajador coloca directamente la etiqueta en el bonche correspondiente	2 min

Etapa del proceso	**Método manual**	**Tiempo Aproximado**	**Sistema propuesto**	**Tiempo aproximado**
3.-	Las pinzas se colocan y se realiza el conteo manual	5 min	El sistema registra la información de los códigos Qr escaneados.	1 min
4.-	Las pinzas se retiran y se devuelven a la mesa de trabajo.	3 min	Esta actividad no se realiza.	0 min
5.-	El resultado se anota en una pizarra.	3 min	La información se queda registrada automáticamente en la base de datos.	0.03 min
Tiempo total del proceso		15		5.03

Los resultados indican que el método manual toma alrededor de 15 minutos para contar, registrar y reorganizar las pinzas cuando la barrilla está llena. Por otro lado, el sistema propuesto permite registrar la información de manera directa a través de la impresión y colocación de etiquetas con código QR, lo que reduce el tiempo total del proceso a aproximadamente 5.03 minutos.

Esta diferencia representa una reducción notable en el tiempo operativo, además de mejorar la precisión del registro y facilitar que la información esté disponible en tiempo real dentro del sistema.

6 CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

Se crearon bases sólidas para el proyecto. La revisión de documentos permitió construir un marco teórico que incluye la importancia de la floricultura en Ecuador y los procesos post-cosecha, como clasificar y embonchar. También se estudiaron conceptos clave como el rendimiento laboral, el monitoreo digital y las tecnologías utilizadas, como Python, Django, Flutter y WebSockets. Esto no solo ayudó a entender el problema de Comexiger, sino que también justificó la elección de las herramientas y métodos utilizados, demostrando que son adecuados para la industria agroindustrial.

La aplicación de la metodología XP fue muy importante para el éxito del proyecto. Esta metodología se centra en trabajar de manera ágil y en estar siempre pendiente de lo que el cliente necesita. En nuestro caso, hubo una comunicación constante con los dueños y trabajadores de Comexiger a través de entrevistas, encuestas y la validación de prototipos. Esto nos permitió saber exactamente qué necesitaban. Las historias de usuario que creamos a partir de estas conversaciones fueron fundamentales para entender sus necesidades reales. También planeamos el trabajo en pequeñas partes y recibimos retroalimentación constantemente. De esta manera, pudimos adaptar el desarrollo de la plataforma a los cambios y requisitos del entorno de Comexiger, lo que aseguró que el producto final fuera lo que ellos esperaban.

Se creó y puso en marcha con éxito una plataforma informática. Esta plataforma tiene dos partes: una aplicación web para los administradores y una aplicación móvil para los trabajadores. Con esta solución, el registro del rendimiento en la clasificación y el embonche se hace de forma automática. Esto se logra escaneando códigos QR. De esta manera, se reemplaza el proceso manual de contar pinzas y anotar en pizarras, que es lento y puede tener errores.

6.2 RECOMENDACIONES

El cambio de un proceso manual a uno digital puede dar miedo o hacer que algunos trabajadores se resistan, sobre todo si no conocen bien la tecnología. Los dueños de Comexiger deben estar cerca de sus empleados durante los primeros días que usen el sistema, felicitándolos por lo que hagan bien y teniendo paciencia cuando cometan errores. Hay que recordarles a los trabajadores

que la plataforma existe para ayudarlos, no para vigilarlos, y que tener un registro claro de su trabajo también es beneficioso para ellos cuando se calcula su salario.

Ahora que podemos ver cómo va el trabajo en tiempo real, es una buena idea usar esta información para dos cosas: llevar un control y motivar al equipo. Por ejemplo, podríamos elegir cada semana al trabajador que lo ha hecho mejor y darle un reconocimiento. También podríamos compartir lo que el equipo ha logrado al final del día. Esto ayuda a que los trabajadores se sientan valorados y vean que su trabajo es importante para la empresa.

Sería útil tener un lugar donde los trabajadores puedan decir lo que piensan o lo que les preocupa sobre la aplicación móvil y los escáneres. Las personas que trabajan con estas herramientas todos los días a menudo tienen ideas muy buenas para mejorar las cosas. Si los escuchamos de verdad, no solo podremos arreglar pequeños problemas con el sistema, sino que también haremos que los trabajadores se sientan más parte del equipo y más seguros. De esta manera, la tecnología nos ayudará en lugar de ser algo que nos impongan.

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», *Econ. Negocios*, vol. 15, n.º 1, pp. 124-143, ene. 2024, doi: 10.29019/eyn.v15i1.1270.

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

[3] «a20v41n10p02.pdf». Accedido: 3 de noviembre de 2025. [En línea]. Disponible en: https://revistaespacios.com/a20v41n10/a20v41n10p02.pdf

[4] E. L. Mamani, D. F. M. Huancollo, L. W. C. Cari, R. F. Aguilar, y K. A. Apaza, «ESTRÉS LABORAL Y RENDIMIENTO LABORAL DE LOS TRABAJADORES EN ENTIDADES FINANCIERAS», en *Memória, cultura e sociedade 2*, 1.ª ed., Atena Editora, 2021, pp. 95-109. doi: 10.22533/at.ed.8742118109.

[5] M. Macias López, «Diseño de una aplicación web para la evaluación del desempeño laboral en la industria metalmecánica como estrategia para la competitividad empresarial en Barranquilla.», mar. 2025, Accedido: 18 de julio de 2025. [En línea]. Disponible en: https://hdl.handle.net/11323/14190

[6] G. P. S. Ccapa, «Programa Especial de Titulación».

[7] F. J. B. Ordóñez, «Impacto y desafíos del sector florícola en Ecuador: entre certificaciones de responsabilidad social y realidades laborales», *Rev. InveCom ISSN En Línea 2739-0063*, vol. 5, n.º 2, Art. n.º 2, 2025, doi: 10.5281/zenodo.13381972.

[8] M. G. M. Garcia, J. C. Michalus, y S. Maldonado, «MODELO Y PROCEDIMIENTOS PARA LA GESTIÓN DE LA INNOVACIÓN TECNOLÓGICA EN ASOCIACIONES CAMPESINAS AGRÍCOLAS DE COTOPAXI, ECUADOR», *Rev. Científica Visión Futuro*, vol. 29, n.º 2, Art. n.º 2, jul. 2025, Accedido: 17 de julio de 2025. [En línea]. Disponible en: https://visiondefuturo.fce.unam.edu.ar/index.php/visiondefuturo/article/view/886

[9] M. Mantilla, K. Gancino, y E. Valencia, «(PDF) CAPITAL DE TRABAJO Y RENTABILIDAD EN EL SECTOR FLORÍCOLA DEL ECUADOR: UN ANÁLISIS

ESTADÍSTICO MULTIVARIANTE», ResearchGate. Accedido: 17 de julio de 2025. [En línea]. Disponible en: https://www.researchgate.net/publication/377659149_CAPITAL_DE_TRABAJO_Y_RENTABILIDAD_EN_EL_SECTOR_FLORICOLA_DEL_ECUADOR_UN_ANALISIS_ESTADISTICO_MULTIVARIANTE

[10] «SciELO Brazil - Nuevo modelo de acumulación y agroindustria: las implicaciones ecológicas y epidemiológicas de la floricultura en Ecuador Nuevo modelo de acumulación y agroindustria: las implicaciones ecológicas y epidemiológicas de la floricultura en Ecuador». Accedido: 17 de julio de 2025. [En línea]. Disponible en: https://www.scielo.br/j/csc/a/HxPL8t8dHqV5SBVxWWbYkCR/?lang=es

[11] L. C. Sangoquiza Chiluisa y J. E. Toapanta Pilatasig, «Impacto de la productividad y desempeño financiero de las empresas florícolas de la zona 3 del Ecuador», 2023, Accedido: 17 de julio de 2025. [En línea]. Disponible en: http://repositorio.utc.edu.ec/handle/27000/10068

[12] J. Espin-Castro *et al.*, «Energy Optimization of the Post-Harvest Area of Roses in Quiroga, Ecuador: A Comparative Analysis», en *Proceedings of the 3rd International Symposium on Automation, Information and Computing*, Beijing, China, China: SCITEPRESS - Science and Technology Publications, 2022, pp. 138-149. doi: 10.5220/0011907200003612.

[13] E. Arboleda, «RELACIÓN ENTRE LOS NIVELES DE TECNOESTRÉS Y EL RENDIMIENTO LABORAL… 2».

[14] N. A. C. Baraja, «IDENTIFICACIÓN Y EVALUACIÓN DE LOS RIESGOS LABORALES Y SU PLAN DE MEJORA EN EL ÁREA OPERATIVA DE LA FLORÍCOLA ROSAHEN».

[15] E. Bautista-Villegas, «Metodologías agiles XP y Scrum, empleadas para el desarrollo de páginas web, bajo MVC, con lenguaje PHP y framework Laravel», *Rev. Amaz. Digit.*, vol. 1, n.º 1, pp. e168-e168, ene. 2022, doi: 10.55873/rad.v1i1.168.

[16] R. Fojtik, «Extreme Programming in development of specific software», *Procedia Comput. Sci.*, vol. 3, pp. 1464-1468, 2011, doi: 10.1016/j.procs.2011.01.032.

[17] «Extreme Programming (XP): Values, Principles, and Practices», AltexSoft. Accedido: 5 de enero de 2026. [En línea]. Disponible en: https://www.altexsoft.com/blog/extreme-programming-values-principles-and-practices/

[18] 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.

[19] 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/

[20] A. I. A. L. Presenta:, «Impacto de las aplicaciones móviles como herramienta de promoción y publicidad dentro del ITSMT».

[21] «content». Accedido: 7 de enero de 2026. [En línea]. Disponible en: https://repositorio.puce.edu.ec/server/api/core/bitstreams/a91efd3a-bf58-4020-8c43-30288134e031/content

[22] S. Monsalve, K. Quinónez, D. García Arango, y C. Henao Villa, «Identificación del desarrollo de aplicaciones web en la actualidad», 2020, pp. 272-282.

[23] «Welcome to Python.org», Python.org. Accedido: 6 de enero de 2026. [En línea]. Disponible en: https://www.python.org/about/

[24] «Dart programming language». Accedido: 6 de enero de 2026. [En línea]. Disponible en: https://dart.dev/index.html

[25] *Framework Django*. Accedido: 6 de enero de 2026. [En línea]. Disponible en: https://www.caktusgroup.com/blog/2016/12/14/django-boring-or-why-tech-startups-should-use-django/

[26] «Django», Django Project. Accedido: 6 de enero de 2026. [En línea]. Disponible en: https://www.djangoproject.com/

[27] «Installation | Laravel 12.x - The clean stack for Artisans and agents». Accedido: 8 de marzo de 2026. [En línea]. Disponible en: https://laravel.com/docs/12.x/installation

[28] «Express - Node.js Marco de aplicación web». Accedido: 8 de marzo de 2026. [En línea]. Disponible en: https://expressjs.com/es/

[29] «PostgreSQL: Acerca de». Accedido: 8 de marzo de 2026. [En línea]. Disponible en: https://www.postgresql.org/about/

[30] *Editor de Codigo Visual Studio Code*. Accedido: 6 de enero de 2026. [En línea]. Disponible en: https://commons.wikimedia.org/wiki/File:Visual_Studio_Code_1.35_icon.svg

[31] «Why did we build Visual Studio Code?» Accedido: 6 de enero de 2026. [En línea]. Disponible en: https://code.visualstudio.com/docs/editor/whyvscode

[32] «Sublime Text - the sophisticated text editor for code, markup and prose». Accedido: 8 de marzo de 2026. [En línea]. Disponible en: https://www.sublimetext.com/

[33] «Atom: un IDE para el desarrollador web | SEIDOR». Accedido: 8 de marzo de 2026. [En línea]. Disponible en: https://www.seidor.com/es-ec/blog/atom-un-ide-para-el-desarrollador-web

[34] *Framework Flutter*. Accedido: 6 de enero de 2026. [En línea]. Disponible en: https://brandlogos.net/tag/flutter

[35] «Flutter - Build apps for any screen». Accedido: 6 de enero de 2026. [En línea]. Disponible en: //flutter.dev/

[36] *Escáner TERA de código de barras inalámbrico 2D D5100*. Accedido: 6 de enero de 2026. [En línea]. Disponible en: https://www.amazon.com/-/es/Tera-inal%C3%A1mbrico-indicador-port%C3%A1til-compacto/dp/B07M68LS2N/ref=sr_1_1?sr=8-1

[37] «Escáner de código de barras inalámbrico QR 2D Tera D5100: escaneo rápido y confiable», Tera Digital. Accedido: 6 de enero de 2026. [En línea]. Disponible en: https://tera-digital.com/es/products/2d-barcode-scanner-d5100

[38] *Impresora Térmica de Etiquetas Nelko PL70e-BT*. Accedido: 6 de enero de 2026. [En línea]. Disponible en: https://m.media-amazon.com/images/I/71gbC8UthxL._AC_SL1500_.jpg

[39] «Amazon.com: Nelko Impresora térmica Bluetooth de etiquetas de envío, impresora de etiquetas de envío inalámbrica de 4 x 6 para paquetes de envío, compatible con Android, iPhone y Windows, ampliamente utilizada : Productos de Oficina». Accedido: 6 de enero de 2026. [En línea]. Disponible en: https://www.amazon.com/-/es/Impresora-Bluetooth-inal%C3%A1mbrica-compatible-ampliamente/dp/B0BRBY6888/ref=sr_1_1?sr=8-1&th=1

[40] «WebSockets - API web | MDN», MDN Web Docs. Accedido: 6 de enero de 2026. [En línea]. Disponible en: https://developer.mozilla.org/es/docs/Web/API/WebSockets_API

[41] *WebSocket*. Accedido: 6 de enero de 2026. [En línea]. Disponible en: https://sunpoint.com.ua/home/blogpage?id=118

[42] «¿Qué es AJAX? - Explicación de JavaScript y XML asincrónicos - AWS», Amazon Web Services, Inc. Accedido: 8 de marzo de 2026. [En línea]. Disponible en: https://aws.amazon.com/es/what-is/ajax/

[43] «Introduction to Event Listeners (The Java™ Tutorials > Creating a GUI With Swing > Writing Event Listeners)». Accedido: 6 de enero de 2026. [En línea]. Disponible en: https://docs.oracle.com/javase/tutorial/uiswing/events/intro.html

[44] «¿Qué es una API de REST?» Accedido: 6 de enero de 2026. [En línea]. Disponible en: https://www.redhat.com/es/topics/api/what-is-a-rest-api

[45] *django/daphne*. (1 de enero de 2026). Python. Django. Accedido: 6 de enero de 2026. [En línea]. Disponible en: https://github.com/django/daphne

[46] «How to deploy with WSGI | Django documentation», Django Project. Accedido: 8 de marzo de 2026. [En línea]. Disponible en: https://docs.djangoproject.com/en/6.0/howto/deployment/wsgi/

[47] «Acerca de GitHub y Git - Documentación de GitHub», GitHub Docs. Accedido: 8 de marzo de 2026. [En línea]. Disponible en: https://docs-internal.github.com/es/get-started/start-your-journey/about-github-and-git

[48] C. Neira, «Implementación de un servidor web y un diseño de una página utilizando herramientas de software libre para el dispensario Sagrada Familia de la ciudad de Guayaquil.», 2016.

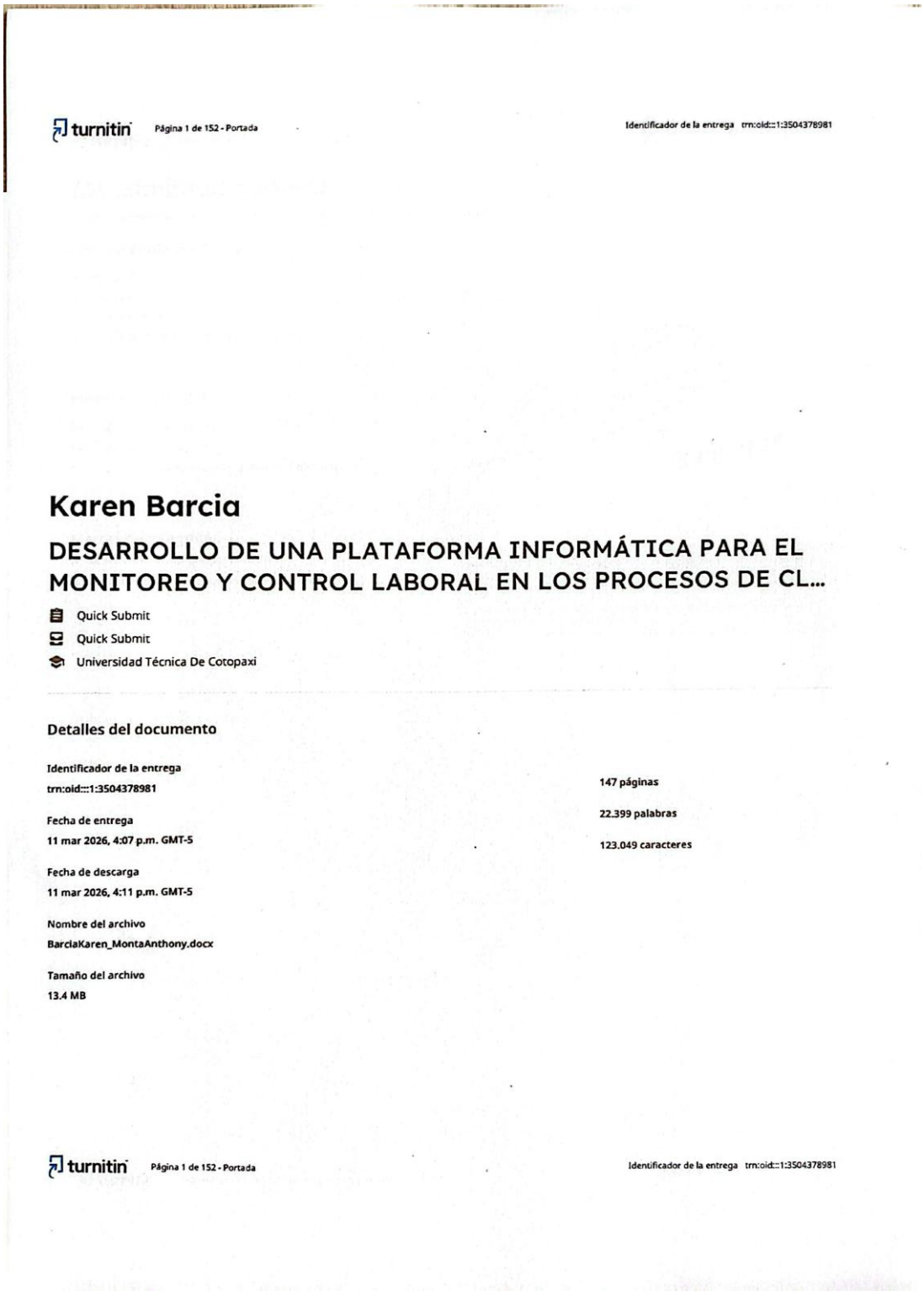
[49] «Web Services», Render. Accedido: 8 de marzo de 2026. [En línea]. Disponible en: https://render.com/docs/web-services

[50] «draw.io Blog». Accedido: 9 de marzo de 2026. [En línea]. Disponible en: https://www.drawio.com/blog

8 ANEXOS

8.1 ANEXO 1

Porcentaje de anti plagio pasado por turtining.



turnitin Página 1 de 152 - Portada Identificador de la entrega trn:oid:::1:3504378981

Karen Barcia

DESARROLLO DE UNA PLATAFORMA INFORMÁTICA PARA EL MONITOREO Y CONTROL LABORAL EN LOS PROCESOS DE CL...

Quick Submit

Quick Submit

Universidad Técnica De Cotopaxi

Detalles del documento

Identificador de la entrega
trn:oid:::1:3504378981

Fecha de entrega
11 mar 2026, 4:07 p.m. GMT-5

Fecha de descarga
11 mar 2026, 4:11 p.m. GMT-5

Nombre del archivo
BarciaKaren_MontaAnthony.docx

Tamaño del archivo
13.4 MB

147 páginas

22.399 palabras

123.049 caracteres

turnitin Página 1 de 152 - Portada Identificador de la entrega trn:oid:::1:3504378981

turnitin Página 2 de 152 - Descripción general de integridad Identificador de la entrega trn:oid:::1:3504378981

6% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

Filtrado desde el informe

- Bibliografía
- Texto citado
- Texto mencionado
- Coincidencias menores (menos de 12 palabras)

Fuentes principales

8%	Fuentes de Internet
0%	Publicaciones
5%	Trabajos entregados (trabajos del estudiante)

Marcas de integridad

N.º de alertas de integridad para revisión

No se han detectado manipulaciones de texto sospechosas.

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.

turnitin Página 2 de 152 - Descripción general de integridad Identificador de la entrega trn:oid:::1:3504378981

8.2 ANEXO 2

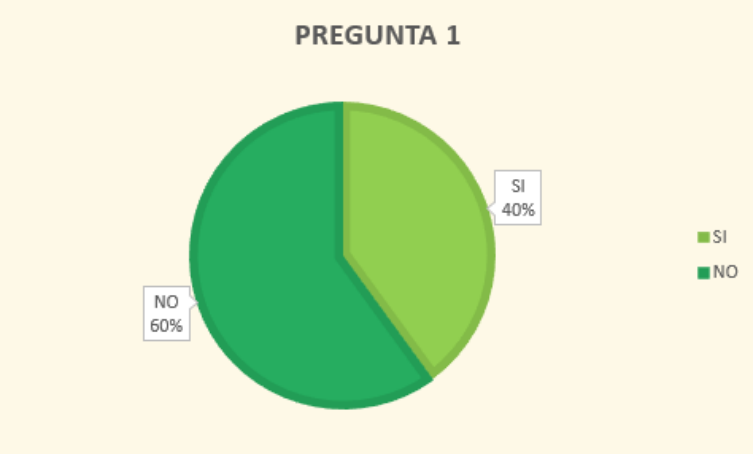
Resultados de la encuesta realizada en el punto 5.2.2.

1.- ¿El conteo manual de bonches es fácil de realizar**?**

SI

NO

Opciones	**Total de Respuestas**
SI	4
NO	6



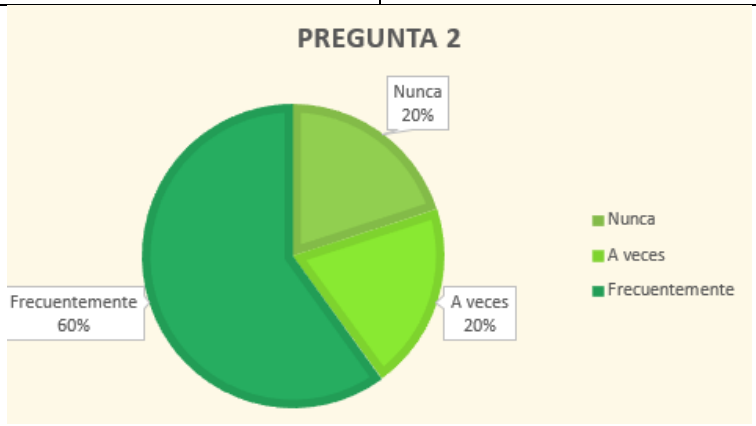
En la estadística de la figura 4.14 la mayoría de las personas que respondieron a la encuesta dicen que contar bonches a mano es difícil. De cada diez personas, 6 dicen que no es fácil, mientras que 4 personas dicen que si.

Esto muestra que contar a mano tiene problemas. Por eso, sería una buena idea usar nuestra plataforma informática para controlar y registrar el conteo. De esta manera, se puede trabajar de forma más eficiente y reducir los errores.

2.- ¿Ha tenido errores al contar o registrar los bonches**?**

- Nunca
- A veces
- Frecuentemente

Opciones	Total de Respuestas
Nunca	**2**
A veces	**2**
Frecuentemente	**6**



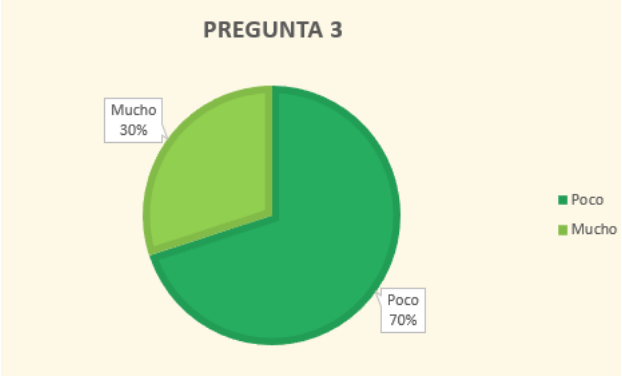
Según lo que dicen las personas que participaron en la encuesta, 6 de cada 10 personas (60%) dicen que a menudo han cometido errores al contar o registrar los bonches, mientras que 2 personas (20%) dicen que a veces y 2 personas (20%) afirman que nunca han tenido errores.

3.- ¿Considera justo el método actual para calcular su rendimiento?

- Poco
- Mucho

Opciones	Total de Respuestas
Poco	**7**
Mucho	**3**

Las estadísticas se encuentran en la figura 4.16.



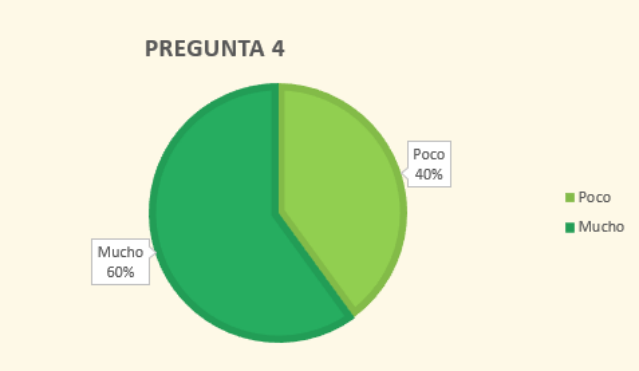
La mayoría de las personas que respondieron a la encuesta no están de acuerdo con la forma en que se calcula su rendimiento actualmente. De cada diez personas, siete dicen que no es justo. Solo tres personas piensan que es justo.

Esto muestra que mucha gente no está satisfecha con el método actual. Por eso, nuestra plataforma informática puede ayudar a mejorar la situación. Con ella, se pueden hacer cálculos del rendimiento laboral que sean más precisos, más abiertos y más confiables.

4.- ¿Cree que se pierde tiempo en el conteo manual?

- Poco
- Mucho

Opciones	**Total de Respuestas**
Poco	**4**
Mucho	**6**



Según los resultados de la encuesta, seis de diez encuestados consideran que se pierde mucho

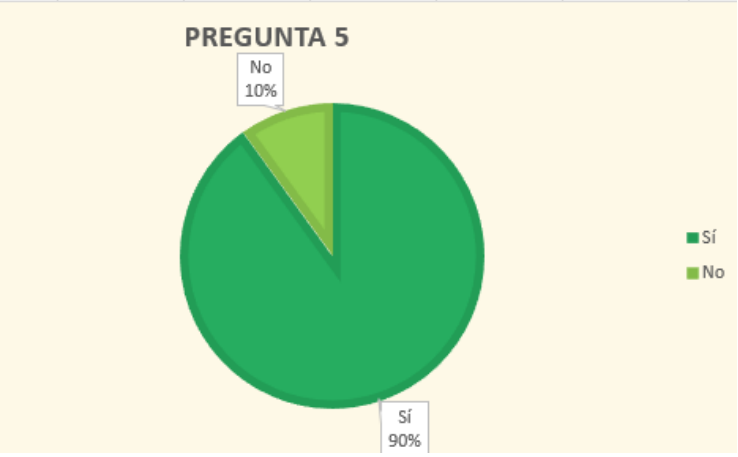
tiempo en el conteo manual, mientras que cuatro personas indican que se pierde poco tiempo.

Estos resultados muestran que el proceso manual afecta la eficiencia. La implementación de nuestra plataforma informática permitirá optimizar el tiempo y mejorar el control del proceso.

5.- ¿Le gustaría conocer su rendimiento en tiempo real?

- Sí
- No

Opciones	**Total de Respuestas**
Sí	**9**
No	**1**



Según los resultados de la encuesta, 9 de 10 encuestados (90%) dicen que sí quieren saber su rendimiento en tiempo real, mientras que 1 persona (10%) dice que no.

Esto muestra que la mayoría del personal está interesado en tener nuestra plataforma de informática, que les permitirá ver su rendimiento laboral de inmediato y con precisión.

6.- ¿Usa con facilidad aplicaciones en su teléfono móvil?

- Poco
- Mucho

Opciones	Total de Respuestas
Poco	**2**
Mucho	**8**

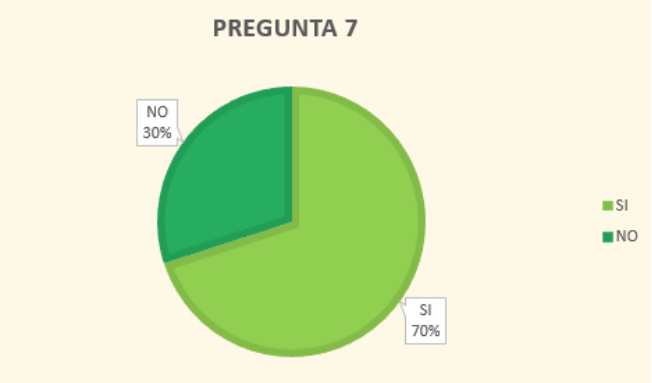
Según lo que dicen los resultados de la encuesta, 8 de 10 personas que respondieron dicen que usan aplicaciones en su teléfono móvil con facilidad. En cambio, 2 personas dicen que lo hacen poco.

Esto muestra que la mayoría de las personas que respondieron la encuesta saben usar aplicaciones móviles. Esto es bueno porque significa que podrán usar nuestra plataforma informática sin problemas.

7.- ¿Considera útil una aplicación para iniciar y finalizar su jornada laboral?

- Sí
- No

Opciones	Total de Respuestas
Sí	**7**
No	**3**



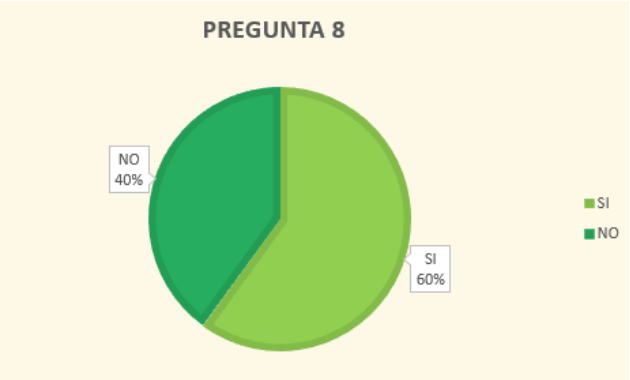
Según los resultados de la encuesta, siete de diez encuestados consideran útil una aplicación para iniciar y finalizar su jornada laboral, mientras que tres personas indican que no.

Estos resultados respaldan la implementación de nuestra plataforma informática, la cual permitirá a los empleados registrar de forma digital el inicio y fin de la jornada laboral.

8.- ¿Le parecería mejor escanear una etiqueta en lugar de usar pinzas?

- Sí
- No

Opciones	**Total de Respuestas**
Sí	**6**
No	**4**



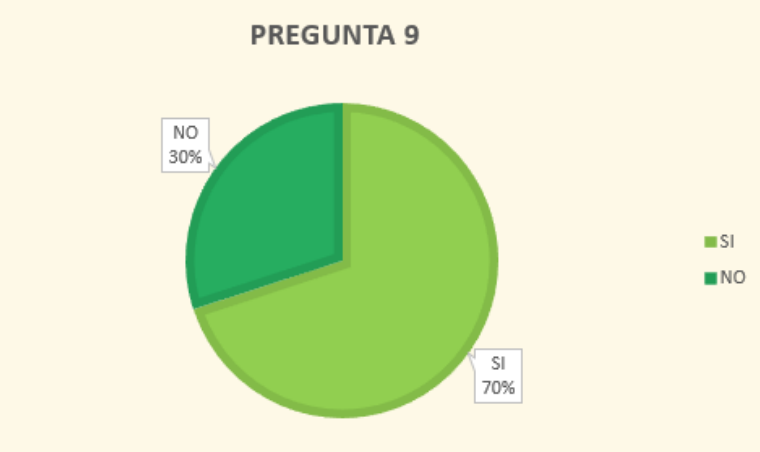
Según los resultados de la encuesta, 6 de 10 encuestados piensan que escanear una etiqueta es mejor que usar pinzas. Esto es el 60% de las personas que respondieron. Por otro lado, 4 personas, que son el 40%, no están de acuerdo con esto.

Estos resultados indican que la gente está de acuerdo con nuestra plataforma informática. Esta plataforma sugiere escanear etiquetas para mejorar la forma en que se registran los bonches.

9.- ¿Cree que un sistema digital reduciría errores en su pago?

- Sí
- No

Opciones	**Total de Respuestas**
Si	7
No	3



Según una encuesta reciente, la mayoría de los encuestados, un total de 7 de cada 10, piensan que un sistema digital ayudaría a reducir los errores en los pagos. Por otro lado, 3 de cada 10 personas encuestadas creen que no.

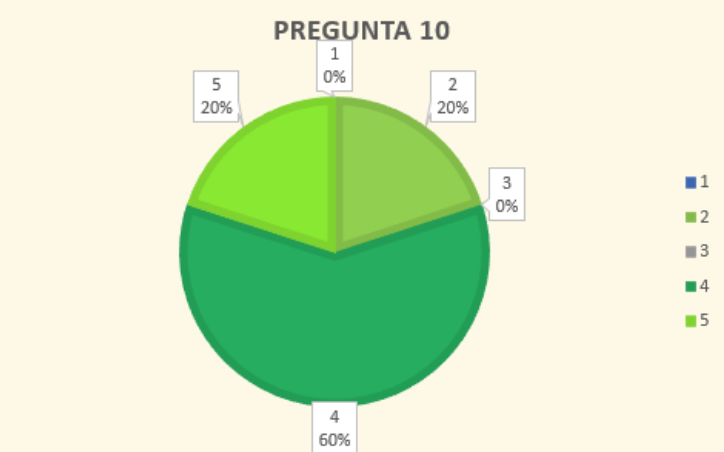
Estos resultados apoyan la puesta en marcha de nuestra plataforma de informática. Gracias a ella, esperamos mejorar la precisión en el registro y cálculo del rendimiento laboral.

10.- En una escala del 1 al 5, ¿qué tan necesario considera modernizar el proceso?

- 1
- 2
- 3

- 4
- 5

Opciones	Total de Respuestas
1	0
2	2
3	0
4	6
5	2



Según los resultados de la encuesta, seis de cada diez personas que participaron calificaron con un cuatro la necesidad de modernizar el proceso. Por otro lado, dos personas le dieron un cinco y otras dos personas le dieron un dos.

8.3 ANEXO 3

Historia de usuario		ID HU	HU-M06
Puntos de historia	5	**Usuario**	Trabajador
Prioridad	Alta	**Iteración asignada**	1

Descripción	Como trabajador, quiero ver la disponibilidad desde la app, para evitar errores en la planificación de envíos.
Criterios de aceptación	• La app permite visualizar el stock. • El stock disponible se muestra tanto en la app y en la web.
Definición de hecho	Los datos reflejan correctamente la disponibilidad de stock para todos los usuarios.
Programador/a responsable:	Anthony Monta

Historia de usuario		**ID HU**	**HU-M07**
Puntos de historia	5	**Usuario**	Administrador
Prioridad	Alta	**Iteración asignada**	1
Descripción	Como administrador, quiero iniciar sesión en la web con mi usuario y contraseña, para poder gestionar la información de trabajadores y mesas.		
Criterios de aceptación	• Permite login con usuario y contraseña válidos. • Muestra mensaje de error si los datos son incorrectos.		
Definición de hecho	El administrador accede solo a las funciones disponibles para su rol como Administrador; por el contrario, otro usuario, aunque se encuentre registrado no podrá acceder.		
Programador/a responsable:	Karen Barcia		

Historia de usuario		**ID HU**	**HU-M08**
Puntos de historia	5	**Usuario**	Administrador
Prioridad	Alta	**Iteración asignada**	1
Descripción	Como administrador, quiero crear nuevos usuarios para la app móvil, para que los trabajadores puedan iniciar sesión y revisar su rendimiento.		
Criterios de aceptación	• Se puede registrar nombre, rol y credenciales. • Los usuarios creados pueden iniciar sesión en la app móvil.		
Definición de hecho	Los nuevos usuarios aparecen correctamente en el sistema y pueden iniciar sesión en la app móvil.		
Programador/a responsable:	Karen Barcia		

Historia de usuario		**ID HU**	**HU-M09**
Puntos de historia	5	**Usuario**	Administrador
Prioridad	Alta	**Iteración asignada**	1
Descripción	Como administrador, quiero crear nuevas mesas, para organizar los bonches y asociarlos correctamente a cada mesa y trabajador.		
Criterios de aceptación	• Se pueden registrar mesas con un identificador único. • Las mesas aparecen disponibles para la app móvil y la web al ser seleccionada en las etiquetas.		
Definición de hecho	Cada bonche registrado se asocia correctamente a la mesa correspondiente y cada trabajador.		

Programador/a responsable:	Karen Barcia

Historia de usuario		**ID HU**	**HU-M10**
Puntos de historia	5	**Usuario**	Administrador
Prioridad	Alta	**Iteración asignada**	1
Descripción	Como administrador, quiero agregar, editar y eliminar la disponibilidad de las variedades en stock, para mantener actualizada la información de producción y recursos disponibles.		
Criterios de aceptación	• Se puede agregar una nueva variedad con su disponibilidad inicial. • Se puede editar la disponibilidad de una variedad existente. • Se puede eliminar una variedad cuando ya no esté en producción. • Los cambios realizados se reflejan en los reportes diarios y mensuales. • La disponibilidad también se visualizará tras el escaneo que se realice desde el rendimiento al ser escaneado por el primer escáner. • La disponibilidad se restará al dar de baja o vender esto surgirá al ser escaneados por el segundo escáner.		
Definición de hecho	• La información actualizada coincide con los registros almacenados en la base de datos.		

	• Los cambios se sincronizan correctamente con la app móvil. • No se permiten valores negativos en la disponibilidad.
Programador/a responsable:	Karen Barcia

Historia de usuario		**ID HU**	**HU-M11**
Puntos de historia	5	Usuario	Administrador
Prioridad	Alta	Iteración asignada	1
Descripción	Como administrador, quiero visualizar y editar el rendimiento de los trabajadores, incluyendo horas trabajadas y bonches realizados, para garantizar un control preciso de la producción.		
Criterios de aceptación	Se puede visualizar el rendimiento por trabajador y por mesa. Se puede editar la hora registrada de trabajo. Se puede modificar la cantidad de bonches realizados. Se puede eliminar un registro de rendimiento en caso de error. Los cambios se reflejan automáticamente en los reportes diarios y mensuales. El rendimiento también se visualizará tras el escaneo que se realice por el primer escáner.		
Definición de hecho	La información coincide con los datos registrados en la app móvil. Las modificaciones quedan almacenadas correctamente en la base de datos. El sistema recalcula automáticamente los totales después de una edición.		

Programador/a responsable:	Karen Barcia

Historia de usuario		**ID HU**	**HU-M12**
Puntos de historia	5	**Usuario**	Trabajador
Prioridad	Alta	**Iteración asignada**	1
Descripción	Como administrador, quiero generar y descargar informes de rendimiento de los trabajadores y disponibilidad de stock, para analizar la productividad y tomar decisiones sobre pagos y recursos.		
Criterios de aceptación	• Se pueden generar informes diarios, semanales y mensuales. • Los informes incluyen: bonches procesados por trabajador y mesa, horas de jornada, y disponibilidad de stock. • Los informes se pueden descargar en formatos PDF y Excel. • Los datos reflejan fielmente lo registrado en la app móvil y la web.		
Definición de hecho	• La descarga es funcional y los datos están completos y correctos.		
Programador/a responsable:	Karen Barcia		

Historia de usuario		**ID HU**	**HU-M13**
Puntos de historia	5	**Usuario**	Trabajador

Prioridad	Alta	**Iteración asignada**	1
Descripción	Como trabajador, quiero que el sistema registre automáticamente un bonche al escanear su código QR, para evitar errores manuales y garantizar precisión en los datos.		
Criterios de aceptación	• El sistema permite escanear el código QR del bonche. • El QR contiene información del trabajador, mesa, variedad y fecha. • El registro se almacena automáticamente en la base de datos. • No se permite registrar el mismo QR más de una vez.		
Definición de hecho	• El registro coincide con los datos enviados desde la app móvil. • Se valida que el código QR sea único. • El sistema almacena fecha y hora automática del escaneo.		
Programador/a responsable:	Karen Barcia		

Historia de usuario		**ID HU**	**HU-M14**
Puntos de historia	5	**Usuario**	Trabajador
Prioridad	Alta	**Iteración asignada**	1
Descripción	Como trabajador, quiero que el Escáner 1 sume automáticamente un bonche al rendimiento del trabajador y también a disponibilidad según su variedad al escanear el código QR, para llevar un control exacto de producción.		

Criterios de aceptación	• Al escanear con el Escáner 1, el sistema incrementa en +1 el número de bonches realizados. • Se actualiza automáticamente el rendimiento del trabajador. • Se actualiza el rendimiento por mesa. • Se registra la hora exacta del escaneo. • Se recalculan los totales diarios.
Definición de hecho	• El rendimiento y disponibilidad mostrado coincide con los registros almacenados. • No se permite duplicar el escaneo del mismo QR. • El sistema refleja los cambios en tiempo real.
Programador/a responsable:	Karen Barcia

Historia de usuario		**ID HU**	**HU-M15**
Puntos de historia	5	**Usuario**	Trabajador
Prioridad	Alta	**Iteración asignada**	1
Descripción	Como trabajador, quiero que el Escáner 2 reduzca automáticamente la disponibilidad de la variedad al escanear el código QR del bonche, para mantener actualizado el control de stock.		
Criterios de aceptación	• Al escanear con el Escáner 2, el sistema descuenta automáticamente la cantidad correspondiente de la disponibilidad. • Se actualiza el stock disponible.		

	• No se permite que la disponibilidad quede en valores negativos.
Definición de hecho	• La disponibilidad coincide con los registros reales. • El sistema valida que exista stock antes de descontar. • La información se sincroniza con la app móvil.
Programador/a responsable:	Karen Barcia

Historia de usuario		**ID HU**	**HU-M16**
Puntos de historia	5	**Usuario**	Trabajador
Prioridad	Alta	**Iteración asignada**	1
Descripción	• Como trabajador, quiero que el sistema diferencie el Escáner 1 (suma rendimiento) y Escáner 2 (resta disponibilidad), para asegurar que los movimientos de producción y stock sean correctos.		
Criterios de aceptación	• El sistema identifica desde qué escáner se realiza el registro. • El Escáner 1 solo incrementa rendimiento. • El Escáner 2 solo reduce disponibilidad.		
Definición de hecho	• El sistema ejecuta correctamente la lógica según el escáner utilizado. • No existen inconsistencias entre rendimiento y disponibilidad.		
Programador/a responsable:	Karen Barcia		