UNIVERSITARIO RUMIÑAHUI

Pregrado

Carrera: Sistemas y Gestión de Data

Asignatura (UIC): Ejecución de Proyectos en TIC'S

Trabajo de titulación previo a la obtención del

Título en:

Tecnólogo Universitario en Sistemas y

Gestión de Data

Tema: Sistema web para la planificación y control de la producción del personal en el Grupo S&M

Autor/s: Santiago Miguel Gualotuña Gualotuña

Edgar David Caillagua Jimenez

Tutor: Mg. Danny Santiago Páez Oscullo

Fecha: septiembre 2025

Autor: Santiago Miguel Gualotuña Gualotuña

Título a obtener: Tecnólogo Universitario en Sistemas y Gestión de Data.

Matriz: Sangolquí -Ecuador

Correo electrónico: santiago.gualotuna@ister.edu.ec

Autor: Edgar David Caillagua Jimenez

Título a obtener: Tecnólogo Universitario en Sistemas y Gestión de Data.

Matriz: Sangolquí -Ecuador

Correo electrónico: edgardavid.caillagua@ister.edu.ec

Dirigido por: Páez Oscullo Danny Santiago

Título: Ing. Computación Gráfica.

Mg. Lógica, Computación e Inteligencia Artificial.

Matriz: Sangolquí -Ecuador

Correo electrónico: danny.paez@ister.edu.ec

Santiago Miguel Gualotuña Gualotuña

Edgar David Caillagua Jimenez

SISTEMA WEB PARA LA PLANIFICACIÓN Y CONTROL

DE LA PRODUCCIÓN DEL PERSONAL EN EL GRUPO S&M

CARTA DE CESIÓN DE DERECHOS DEL TRABAJO DE TITULACIÓN

CT-ANX-2025-ISTER-2-2.3

Sangolquí, (02) de (septiembre) del 2025

MSc. Elizabeth Ordoñez
DIRECTORA DE DOCENCIA

MSc. Mónica Loachamín
COORDINADORA DE TITULACIÓN

INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO

Presente

Por medio de la presente, yo, Gualotuña Gualotuña Santiago Miguel declaro y acepto en forma expresa lo siguiente: Ser autor del trabajo de titulación denominado Sistema web para la planificación y control de la producción del personal en el Grupo S&M, de la Tecnología Superior Universitaria Sistemas y Gestión de Data; y a su vez manifiesto mi voluntad de ceder al Instituto Superior Tecnológico Rumiñahui con condición de Universitario los derechos de reproducción, distribución y publicación de dicho trabajo de titulación, en cualquier formato y medio, con fines académicos y de investigación.

Esta cesión se otorga de manera no exclusiva y por un periodo indeterminado. Sin embargo, conservo los derechos morales sobre mi obra.

En fe de lo cual, firmo la presente.

Atentamente,

SANTIAGO MIGUEL GUALOTUÑA GUALOTUÑA
C.I.: 1727661256

CARTA DE CESIÓN DE DERECHOS DEL TRABAJO DE TITULACIÓN

CT-ANX-2025-ISTER-2-2.3

Sangolquí, (02) de (septiembre) del (2025)

MSc. Elizabeth Ordoñez
DIRECTORA DE DOCENCIA

MSc. Mónica Loachamín
COORDINADORA DE TITULACIÓN

INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO

Presente

Por medio de la presente, yo, Caillagua Jimenez Edgar David declaro y acepto en forma expresa lo siguiente: Ser autor del trabajo de titulación denominadoSistema web para la planificación y control de la producción del personal en el Grupo S&M, de la Tecnología Superior Universitaria Sistemas y Gestión de Data; y a su vez manifiesto mi voluntad de ceder al Instituto Superior Tecnológico Rumiñahui con condición de Universitario los derechos de reproducción, distribución y publicación de dicho trabajo de titulación, en cualquier formato y medio, con fines académicos y de investigación.

Esta cesión se otorga de manera no exclusiva y por un periodo indeterminado. Sin embargo, conservo los derechos morales sobre mi obra.

En fe de lo cual, firmo la presente.

Atentamente,

EDGAR DAVID CAILLAGUA JIMENEZ
C.I.:1721896866

FORMULARIO PARA ENTREGA DE PROYECTOS EN BIBLIOTECA INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO

CT-ANX-2025-ISTER-3

CARRERA: **TECNOLOGÍA UNIVERSITARIA EN SISTEMAS Y GESTIÓN DE DATA**

AUTOR/ES: **SANTIAGO MIGUEL GUALOTUÑA GUALOTUÑA - EDGAR DAVID CAILLAGUA JIMENEZ**

TUTOR: **DANNY SANTIAGO PAEZ OSCULLO**

CONTACTO ESTUDIANTE: **0969047436 - 0960137242**

CORREO ELECTRÓNICO: smggdrag@hotmail.com , davidcaillagua020@gmail.com

TEMA: **SISTEMA WEB PARA LA PLANIFICACIÓN Y CONTROL DE LA PRODUCCIÓN DEL PERSONAL EN EL GRUPO S&M.**

OPCIÓN DE TITULACIÓN: UNIDAD DE INTEGRACIÓN CURRICULAR

RESUMEN EN ESPAÑOL:

En el presente trabajo de titulación se abordan metodologías para el desarrollo de sistemas de control y planificación de actividades en entornos de producción, con el objetivo de analizar datos en tiempo real dentro de una empresa mediana, específicamente el Grupo S&M. El estudio parte de una problemática real en un contexto de producción diaria, caracterizado por el cumplimiento de metas y la gestión de diversas actividades y desafíos operativos, lo que se busca es implementar un sistema que permita controlar en tiempo real el flujo de unidades producidas en la confección de prendas de vestir,facilitando asi la planificación anticipada de actividades, contribuyendo a cumplir con los plazos de entrega establecidos y optimizar los procesos productivos.

Se hace énfasis en la necesidad de desarrollar un sistema escalable, y de fácil uso para los usuarios dentro de la empresa, este proyecto se sustenta en información real proveniente de los procesos internos, lo cual garantiza la pertinencia y aplicabilidad del sistema propuesto.

El alcance del proyecto contempla la creación de una herramienta tecnológica que aporte significativamente a la mejora continua del Grupo S&M, mediante la recolección y análisis de datos productivos diarios y semanales. Este análisis permitirá identificar oportunidades de mejora y optimización en los procesos, elevando el nivel de eficiencia y productividad de la organización.

PALABRAS CLAVE:
Sistemas de control, planificación de producción, análisis de datos, tiempo real, escalabilidad, mejora continua, Grupo S&M.

ABSTRACT:

In the present degree project, methodologies for the development of control and planning systems for activities in production environments are addressed, with the objective of analyzing data in real time within a medium-sized company, specifically Grupo S&M. The study starts from a real problem in a context of daily production, characterized by the fulfillment of goals and the management of various activities and operational challenges. What is sought is to implement a system that allows real-time control of the flow of units produced in the manufacturing of clothing, thus facilitating the anticipated planning of activities, contributing to meeting the established delivery deadlines and optimizing production processes.

Emphasis is placed on the need to develop a scalable and user-friendly system for the employees within the company. This project is based on real information from internal processes, which guarantees the relevance and applicability of the proposed system.

The scope of the project contemplates the creation of a technological tool that significantly contributes to the continuous improvement of Grupo S&M, through the collection and analysis of daily and weekly production data. This analysis will allow the identification of opportunities for improvement and optimization in the processes, raising the level of efficiency and productivity of the organization.

KEYWORDS:

Control systems, production planning, data analysis, real time, scalability, continuous improvement, Grupo S&M.

Firma del Estudiante (Autor)

Santiago Miguel Gualotuña Gualotuña

C. I.: 1727661256

Firma del Estudiante (Autor)

Edgar David Caillagua Jimenez

C.I.: 1721896866

SOLICITUD DE PUBLICACIÓN DEL TRABAJO DE TITULACIÓN

CT-ANX-2025-ISTER-4

Sangolquí, (02) de (septiembre) del 2025

Sres.-
INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO

Presente

Yo **CAILLAGUA JIMENEZ EDGAR DAVID**, con C.I.: **1721896866** alumno de la Carrera TECNOLOGÍA SUPERIOR UNIVERSITARIA SISTEMAS Y GESTIÓN DE DATA, cedo al INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO, los derechos de publicaciones del presente trabajo de Titulación en el Repositorio Institucional para hacer uso de todos los contenidos con fines estrictamente académico o de investigación.

Atentamente,

_____
Firma del Estudiante
CAILLAGUA JIMENEZ EDGAR DAVID
C.I.: **1721896866**

SÓLO PARA USO DEL ISTER

Han sido revisadas las similitudes del trabajo en el software "TURNITING" y cuenta con un porcentaje de; motivo por el cual, el Proyecto Técnico de Titulación es publicable. (EL PORCENTAJE DE SIMILITUD DEBE SER MÁXIMO DE 15%)

_____	_____
MSc. Elizabeth Ordoñez	MSc. Mónica Loachamín
DIRECTORA DE DOCENCIA	COORDINADORA DE TITULACIÓN

Fecha del Informe _____/_____/_____

SOLICITUD DE PUBLICACIÓN DEL TRABAJO DE TITULACIÓN

CT-ANX-2025-ISTER-4

Sangolquí, (02) de (septiembre) del 2025

Sres.-
INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO

Presente

Yo **GUALOTUÑA GUALOTUÑA SANTIAGO MIGUEL**, con C.I.: **1727661256** alumno de la Carrera TECNOLOGÍA SUPERIOR UNIVERSITARIA SISTEMAS Y GESTIÓN DE DATA, cedo al INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO, los derechos de publicaciones del presente trabajo de Titulación en el Repositorio Institucional para hacer uso de todos los contenidos con fines estrictamente académico o de investigación.

Atentamente,

_____
Firma del Estudiante
GUALOTUÑA GUALOTUÑA SANTIAGO MIGUEL
C.I.: **1727661256**

SÓLO PARA USO DEL ISTER

Han sido revisadas las similitudes del trabajo en el software "TURNITING" y cuenta con un porcentaje de; motivo por el cual, el Proyecto Técnico de Titulación es publicable. (EL PORCENTAJE DE SIMILITUD DEBE SER MÁXIMO DE 15%)

_____	_____
MSc. Elizabeth Ordoñez	MSc. Mónica Loachamín
DIRECTORA DE DOCENCIA	COORDINADORA DE TITULACIÓN

Fecha del Informe _____/_____/_____

SISTEMA WEB PARA LA PLANIFICACIÓN Y CONTROL
DE LA PRODUCCIÓN DEL PERSONAL EN EL GRUPO S&M

Dedicatoria:

En dedicación a mi familia ya que han sido los escalones fundamentales y la mayor fuente de inspiración en mi vida, así mismo le doy las gracias por enseñarme su ejemplo de superación y a no rendirme ante las dificultades, por inculcarme los valores y virtudes que me han guiado en cada paso de mi camino. Este logro se los dedico a ustedes, ya que sin su apoyo perseverante nada de esto sería posible.

También al Grupo S&M, por abrirme las puertas de su empresa y brindarme el apoyo para ejecutar este proyecto de titulación, permitiéndome crecer personal y profesionalmente dentro del ámbito laboral. Y de manera especial al Ing.Milton de la Cruz ya que con su liderazgo, conocimiento y valiosa guía fue la pieza clave en el éxito de este proyecto, ya que siendo un gran Líder ha logrado dejar en mí enseñanzas que llevaré siempre conmigo.

Santiago Miguel Gualotuña Gualotuña

En dedicatoria a mis padres porque me apoyaron incondicionalmente y han sido un ejemplo de perseverancia, fueron las personas que me motivaron a continuar en todas las etapas de mi estudio y vida personal. Para mis hermanos porque son una fuente de perseverancia y fuerza en los momentos más importantes de este camino.

A Mis amigos, con su propio apoyo hicieron que el proceso fuera más tolerante. Y especialmente con Dios, porque me dio la salud, la fuerza y la sabiduría necesarias para completar este proyecto.

Edgar David Caillagua Jimenez

Agradecimientos:

Agradezco primeramente a Dios, por mantenerme firme ante las dificultades de la vida y por darme la fortaleza necesaria para culminar esta carrera, en la cual he forjado no solo mi formación profesional, sino también un valioso crecimiento personal.

Al docente de la materia y tutor de este proyecto de titulación, Ing. Danny Páez, por su valioso aporte en el desarrollo de esta obra y por ser un pilar fundamental en la transmisión del conocimiento dentro de una institución tan prestigiosa como lo es el Instituto Universitario Rumiñahui. Su orientación constante, compromiso y dedicación en cada etapa del proceso fueron una guía esencial en mi formación como desarrollador de sistemas y analista de datos.

Y, finalmente, a todos aquellos que, a lo largo de esta carrera, nos han motivado con su ejemplo, transmitiéndonos su conocimiento, su pasión por esta profesión y su forma de ejercerla, dejando en nosotros una huella imborrable que perdurará en nuestra vida profesional.

Santiago Miguel Gualotuña Gualotuña

Muchas gracias al Instituto tecnológico Rumiñahui por abrir una puerta de capacitación profesional. Mi gratitud al grupo S&M, una compañía en la que se desarrolla este proyecto porque me dio la oportunidad de usar mi conocimiento en el contexto real y contribuir a las soluciones tecnológicas para sus procesos de producción. Un agradecimiento especial a mis maestros, aquellos que, con liderazgo, y paciencia se lograr con mis objetivos.

Mis compañeros de clase y amigos, intercambiando experiencias, conocimiento y apoyo continuo en este proceso. Al final, agradezco a mi familia, un motivo básico de mi vida por mi fe y estuvo presente en cada etapa de mi aprendizaje y mi camino personal.

Edgar David Caillagua Jimenez

Tabla de contenido

INDICE DE ILUSTRACIONES

1. INTRODUCIÒN

En la actualidad, las empresas dedicadas a la confección de prendas de vestir formales se enfrentan a un entorno cada vez más competitivo y cambiante, en el que la eficiencia operativa y la capacidad de adaptarse rápidamente a los requerimientos del mercado son factores clave para su supervivencia y crecimiento (Perez, 2024). Desde este punto de vista , la correcta planificación y control de la producción adquieren una relevancia fundamental, especialmente en empresas medianas como el Grupo S&M, que deben cumplir con cronogramas de entrega ajustados, gestionar recursos limitados y mantener altos estándares de calidad.

El equipo de S&M de Ecuador, una compañía textil, ha estado obteniendo sus procesos de planificación muy tradicionales, utilizando hojas de cálculo para realizar un seguimiento de los pedidos y dividir el trabajo en la planta. Aunque estos recursos han sido útiles en las etapas iniciales, actualmente se evidencian limitaciones que afectan el rendimiento y la organización de las actividades diarias, ya que la falta de un sistema tecnológico que permitá visalizar cantidades en tiempo real, monitoree la producción, evite cuellos de botella y gestione las cargas de trabajo, ha resultado en una planificación improvisada, con tareas empíricas de tareas y baja trazabilidad de los resultados.

La falta de información influyen de manera directa en la efectividad operacional, ya que generan la duplicidad de los esfuerzos, la descoordinación entre los sectores, los retrasos en los trámites y las dificultades para el cumplimiento de los plazos acordados con los clientes.

Por otro lado, la inexistencia de un registro sistematizado de datos de producción impide el análisis comparativo y evaluación de objetivos, lo que dificulta la toma de decisiones acertadas y limita la capacidad de aplicar la mejora continua.

Basado en este problema, la firma necesita un sistema digital capaz de vigilar el proceso de fabricación de prendas, detectar cualquier colapso y gestionar los trabajos en función de los datos sólidos y actualizados. Este sistema no solo será fácil de usar, sino que también será flexible, por lo que debe ser fácil para el equipo capacitarse y manejar cualquier cosa que venga en su camino en el flujo de trabajo.

El grupo S&M busca una implementación de este sistema para mejorar su organización interna dentro de su producción para aumentar su producción y poder cumplir con sus compromisos comerciales por lo tanto este sistema mostrara una tabla visual con un diseño amigable en el cual podra observar la cantidad total de prendas que se sera producidas así como el flujo de sus unidades ya que se realizara mediante módulos de preparación así ayudará a la decisión en el tiempo.

La importancia de esta propuesta está a la vista. Más allá de su aplicación inicial, con este montaje buscamos sentar las bases para lograr una transformación digital progresiva en S&M. También buscamos demostrar que, al sistematizar procesos y usar herramientas de análisis de datos, el grupo S&M es capaz de anticipar problemas, hacer una mejor planificación y mejorar su rendimiento en un nivel general. También contaremos con una recopilación práctica de datos históricos que son excelentes para hacer informes, verificar información y mejorar siempre los procesos.El punto principal del estudio se trata de establecer un buen sistema de planificación de producción para una planta textil de tamaño mediano, observando cómo funciona el grupo S&M durante tres meses. Nuestro estudio no va involucrar procesos de finanzas, inventarios o mecanismos de la cadena de suministro. Se usará un método descriptivo y la metodología SCRUM, mezclándolo con algunos análisis cualitativos y cuantitativos de los procesos actuales de la empresa y el diseño del sistema.

1.1 Antecedentes

La sostenibilidad de las pequeñas y medianas empresas (PYMES) en Ecuador se ve constantemente desafiada por una serie de factores, siendo la mala administración y la deficiente gestión de la productividad los más críticos. Según estudios como el de Calva et al. (2017), la mitad de los fracasos empresariales en el país están ligados a problemas de liquidez y una alta carga financiera. Esta vulnerabilidad se agrava en sectores como la industria manufacturera, donde, como señala Orteaga (2013), una gestión ineficiente de la productividad y la ausencia de mecanismos formales de control comprometen la viabilidad de las operaciones. La escasa capacidad de estas empresas para optimizar sus activos y gestionar adecuadamente la relación entre ingresos y costos les impide exponer su potencial productivo en eficiencia, lo que a menudo provoca en perdidas.

En la actualidad, la empresa que no tiene un control interno fuerte, su viabilidad y crecimiento serán difíciles de lograr en el futuro. Según el académico Vivanco (2017) y Macias (2022), el sistema de control interno permite tener una idea clara sobre la realidad de la empresa, así como también mide y gestiona controlando y planificando procesos, el control no sólo mejora la solvencia y capacidad de la empresa para cumplir con sus obligaciones, sino que aclara los procedimientos y estándares de operación que permite a los empleados tener una sensación de pertenencia y cultura. A medida que han crecido el erro humano, así como la complejidad del negocio, el control interno se convierte en una necesidad para asegurar que la información productiva que emiten las empresas sea fiable y para asegurar la viabilidad a largo plazo.

Esto debes de saberlo para sobrevivir: Las PYMES que implementan controles internos robustos tienen más posibilidades de sobrevivir y, sobre todo, crecer en el futuro. Un sistema de control interno permite a la empresa obtener visibilidad sobre su situación real, medir y gestionar y planificar procesos (Vivanco, 2017; Macias, 2022). Un control que funcione bien mejorará la solvencia y la capacidad de la empresa para cumplir con sus obligaciones. Reglas claras y estándares de funcionamiento fijan la dirección y responsabilidad de los empleados. Gracias a que minimizan riesgos de fraude y errores y contribuyen a la fiabilidad de la información financiera un control interno merece considerarse como un instrumento fundamental para garantizar la sostenibilidad a largo en un entorno empresarial incierto y competitivo.

1.2 Problematica:

El Grupo S&M actualmente realiza la programación de su producción manualmente mediante el uso de hojas de cálculo, que resulta limitada para un seguimiento dinámico, se evidencia que esta herramienta no es capaz de llevar a cabo la asignación y seguimiento de las actividades del personal en tiempo real. Esto es crítico cuando hay cuellos de botella por acumulación de trabajo en una zona determinada de la planta, también la falta de tecnología obliga a los responsables a adoptar enfoques mecánicos y manuales, sin tener claridad sobre qué es lo importante, cuánto tardarán ni en qué estado va lo que se engancha.

Esta situación trae como resultado directo y negativo la duplicación de tareas, la pérdida de tiempo, baja productividad y recursos humanos mal distribuidos. Además, se complica poder entender si hay un problema operativo en este lugar y, si existe, poder entender a fondo la situación real en este punto. Desde la base de todo esto, la empresa no puede aplicar estrategias de mejora continua que sean basadas en datos reales y que estén actualizados. La situación descrita pone de manifiesto que es necesario diseñar un sistema de control y planificación de producción que sirva para organizar de forma eficiente la actividad diaria y que además facilite la toma de decisiones.

1.3 Justificación

Esta implementación de un sistema de control y planificación de la producción en el Grupo S&M se justifica por los notables beneficios que traerá tanto a nivel técnico como social, resolviendo una problemática que afecta directamente en la eficiencia y control de la planta de producción.

1.3.1 Justificación Social

El proyecto tendrá un fuerte impacto social al mejorar las condiciones de trabajo de los empleados. La ausencia de una herramienta digital conduce a la incertidumbre y a la carga de trabajo, lo que debilita la estabilidad emocional y la calidad de vida de las personas. El mejorado sistema de distribución de la carga laboral recortará los tiempos de inactividad y volverá más predecible y justo este ámbito de la vida que los empleados ya no necesitarán temer. Considero que proporcionar un lugar de trabajo mejorado no solo ayudará a los empleados, sino que permitirá a los empresarios controlar mejor a los subordinados y a los propios trabajadores. Este esta teoria se alinea con la responsabilidad social empresarial al priorizar el desarrollo integral del recurso humano y fortalecer el tejido productivo local (De la cruz, et al, 2019).

1.3.2 Justificación Técnica

Técnicamente, el proyecto es fundamental para procesos de gestión más modernos. Las hojas de cálculo son una limitante, pues aumentan la probabilidad de error y dificultan el análisis del histórico. El recurso digital unificará toda la información en un solo lugar, permitirá conocer el estado de la producción a todo momento y tomar decisiones en base a datos confiables. El sistema se creará mediante modernas herramientas de programación web, lo cual asegura una interfaz de usuario tanto organizada como visualmente placentera para el personal técnico y administrativo.

Por si fuera poco, sus atribuciones avanzadas le darán la posibilidad de crear reportes automáticos, detectar desviaciones en la planificación y medir el rendimiento por módulo de producción. Todo esto permitirá una mejor planificación y organización de las operaciones, al facilitar la mejora continua y seguir los pasos de cada producto desde el principio hasta el final.

Objetivo General

Desarrollar un sistema de control y planificación de producción para el personal del Grupo S&M que permita monitorear en tiempo real las cantidades y el flujo de confección de prendas de vestir, optimizando la asignación de actividades del personal y mejorando la eficiencia operativa.

Objetivos Específicos

- Evaluar la situación actual del proceso de planificación y control de producción en el Grupo S&M para ello se identificara los principales cuellos de botella y áreas de mejora.
- Implementar un sistema escalable y fácil de usar que registre y gestione los datos de producción en tiempo real utilizando python como lenguaje de programación.
- Proporcionar funcionalidades que permitan la planificación anticipada de actividades y el seguimiento del desempeño de los modulos de trabajo.
- Diseñar una interfaz visual tipo tablero digital que permita visualizar de forma clara y dinámica las cantidades producidas y el estado de avance por módulo y proceso.
- Generar reportes descargables en formato excel sobre la producción diaria, semanal o mensual, para facilitar el análisis de eficiencia y el cumplimiento de metas productivas.

Capítulo 2: Marco Teórico y Variables de Investigación

2.1 Marco Político y Social

El proyecto se alinea estratégicamente con las iniciativas de desarrollo del país. Se enmarca en el Plan Nacional de Desarrollo 2021-2025, específicamente en el Eje 2: "Economía Sostenible y Empleo Digno", que promueve la modernización tecnológica del sector productivo. De manera complementaria, la Agenda Digital Productiva 2021-2025 del Ministerio de Producción fomenta la adopción de soluciones de la Industria 4.0 en las PYMES manufactureras, lo que valida la propuesta de digitalizar los procesos de producción del Grupo S&M.

A nivel social, los sistemas han contribuido a reducir las largas jornadas laborales y el equilibrio de la carga de trabajo, lo que colectivamente contribuye a mejorar las condiciones laborales. Está vinculado a los Objetivos de Desarrollo Sostenible (ODS), especialmente con el objetivo 8.8, que busca proteger los derechos laborales y promover entornos de trabajo seguros. Debido a la transparencia de los indicadores de rendimiento proporcionados por el sistema, se fomentan oportunidades de capacitación más específicas y se promueve una distribución más equitativa de los roles.

2.2 Marco Legal

La propuesta de desarrollo e implementación del sistema se adhiere a la normativa ecuatoriana vigente para garantizar su legalidad y la protección de los derechos de los colaboradores y la información de la empresa.

Constitución de la República del Ecuador (2008): El proyecto se fundamenta en el derecho al trabajo digno (Art. 326), al ofrecer herramientas que mejoran la organización y planificación de la producción para un ambiente laboral más justo.

Código de Trabajo (2005): El sistema apoya el cumplimiento de los artículos 41 y 42, que regulan la jornada laboral y los registros de horas, ya que facilita un seguimiento preciso y transparente de la asignación de actividades.

Ley Orgánica de Protección de Datos Personales (2021): El sistema está diseñado para cumplir con esta ley, implementando medidas de seguridad y confidencialidad para la gestión de los datos personales de los empleados y clientes.

Ley de Comercio Electrónico, Firmas Electrónicas y Mensajes de Datos (2002): Esta ley otorga validez jurídica a los registros digitales y reportes que se generarán con el sistema, lo que es fundamental para la toma de decisiones y la documentación interna.

Reglamento de Seguridad y Salud de los Trabajadores (Decreto Ejecutivo 2393): Al permitir una asignación informada de tareas y prevenir la sobrecarga laboral, el sistema contribuye indirectamente a la observancia de este reglamento.

2. Marco Teorico.

En este capítulo se presentan los marcos teóricos y conceptuales básicos que sustentan el desarrollo del Sistema de Control y Planificación de la Producción (SCPP) para el Grupo S&M. Se abarca todo, desde definiciones generales inherentes a la tecnología y los problemas en cuestión, hasta las herramientas y enfoques específicos que se aplicarán, incluyendo estudios previos relevantes y la conexión explícita de estos conceptos con el proyecto.

2.1. Definiciones Generales: Conceptos Base

Comprender el contexto del proyecto requiere la explicacion de términos fundamentales que definen su base tecnológica y la problemática que busca resolver.

2.1.1. Tecnología en el Contexto Empresarial

A primera vista, la tecnología se ve como un conjunto de herramientas; en su esencia, sin embargo, comprende una aplicación sistemática del conocimiento científico y organizacional para alcanzar objetivos prácticos (Drucker, 1985). Así, en los negocios, las Tecnologías de la Información y la Comunicacion (TIC) desempeñan un papel estratégico, transformando los procesos empresariales, optimizando recursos y haciendo posible la toma de decisiones informadas (Laudon & Laudon, 2020). Para el Grupo S&M, un SCPP digital representa lo mas esencial de las TIC: un instrumento para superar limitaciones operativas y reforzar la coordinación al hacer visibles los cuellos de botella (Kagermann, Wahlster & Helbig, 2013).

2.1.2. Fenómeno y Problemática de la Gestión de la Producción

El tema principal a abordar se refiere a las ineficiencias de producción en la industria textil, especialmente en las empresas medianas. Esta ineficiencia se manifiesta como un problema sistemico: falta de visibilidad en tiempo real, planificación reactiva y uso subóptimo de los recursos humanos (Orlicky, 1975). Es la dependencia de herramientas rudimentarias, por ejemplo, hojas de cálculo, lo que genera cuellos de botella junto con retrasos en las entregas y la incapacidad de adaptarse a las demandas del mercado, comprometiendo la competitividad y sostenibilidad del negocio (Porter, 1985). En Ecuador, se ha encontrado que la falta de control interno adecuado y la gestion limitada de la productividad son razones principales para el fracaso empresarial, particularmente para las pymes manufactureras (Calva et al., 2017; Orteaga, 2013).

2.2. Tecnologías o Enfoques Aplicados

El desarrollo del SCPP(Sistema de control de la planificacion de producción) se fundamenta en un conjunto de tecnologías y enfoques de vanguardia en el desarrollo de software y la gestión de la producción.

2.2.1. *Metodologías de Gestión de Producción*

La base conceptual para la optimización operativa del pryecto SCPP se inicia centrandose en filosofías de mejora continua y planificación avanzada:

Manufactura Esbelta: Este enfoque se centra en la eliminación sistemática de desperdicios (Muda) y la optimización del flujo de valor a través de la estandarización y la retroalimentación continua (Womack, Jones & Roos, 1990). Los principios de flujo continuo y nivelación de la producción son clave para el diseño del sistema que busca identificar y reducir

actividades que no agregan valor al proceso de manufactura. El SCPP, al visualizar cuellos de botella y permitir la redistribución de tareas, apoya directamente la filosofía Lean, reflejada en mejoras en la Eficiencia General de los Equipos (OEE) y la puntualidad (Womack & Jones, 1996).

Kaizen: Un enfoque japonés de mejora continua e incremental en cada nivel de la organización. La implementación de SCPP facilita la recopilación de datos y la visualización de métricas, una parte fundamental de las fases de mejora de Kaizen; pequeños ajustes basados en evidencia que resultan en ganancias significativas de eficiencia (Imai, 1986).

Planificación de Requerimientos de Materiales (MRP-II): Una nueva versión de MRP que se centra en planificar todos los recursos de su empresa (producción, finanzas e ingeniería). La lógica de SCPP se ajusta al sistema MRP-II ya que gestionan directamente la demanda de recursos humanos y el flujo de materiales a nivel operativo y permiten la toma de decisiones en tiempo real (Orlicky, 1975).

Gestión de Procesos de Negocio (BPM): Modelado, análisis, optimización y automatización de procesos de negocio son algunos de los aspectos incluidos en su disciplina. El SCPP se utiliza como una herramienta de apoyo para BPM a través de una visión general del flujo de trabajo de personalización, lo que ayuda a descubrir oportunidades para la optimización y estandarización de procedimientos operativos (Dumas et al., 2018).

2.2.2. *Arquitectura y Tecnologías de Desarrollo Web*

El sistema está construido bajo una arquitectura cliente-servidor, utilizando un conjunto de tecnologías robusto para asegurar la escalabilidad y facilidad de uso.

2.2.2.1. *Desarrollo del Front-End (Interfaz de Usuario)*

La facilidad de uso y la experiencia del usuario (UX) se considerarán un aspecto fundamental del diseño de la interfaz, lo cual es importante en este caso con respecto a la implementación con el personal del Grupo S&M (Nielsen, 1993) y siguiendo el Modelo de Aceptación de Tecnología (Davis, 1989) que fomenta la 'utilidad percibida' y la 'facilidad de uso'.

HTML5 (Lenguaje de Marcado de Hipertexto 5): Es la base para el desarrollo web, siendo la edición más reciente del lenguaje de marcado HTML. Su uso se justifica por su capacidad para superar las incompatibilidades de versiones anteriores y ofrecer una forma más eficiente de crear aplicaciones web (Prescott, 2015). Su principal beneficio es que soporta multimedia y funcionalidades de forma nativa y no requiere una multitud de plugins/APIs. Eso mejora la velocidad de carga y la consistencia entre navegadores. Sus tres principales fortalezas son características más versátiles y la capacidad de operar sin conexión, así como una mejor interacción con los servidores, por lo tanto, es una base lógica para SCPP.

Tailwind CSS: En comparación con los marcos convencionales que proporcionan componentes predefinidos, se selecciona Tailwind CSS como un marco de diseño web centrado en utilidades. Este método se basa en clases reutilizables que actúan como bloques de construcción, proporcionando al entorno de desarrollo interfaces totalmente personalizadas construidas en torno a los requisitos específicos del proyecto (Gerchev, 2022). La mayor fortaleza de Tailwind es que

cada clase representa una propiedad CSS, lo que hace que la personalización sea rápida y precisa sin cambiar estilos globalmente. Podremos desarrollar un diseño fresco y original para el SCPP, libre de cualquier estilo establecido. Además, cuenta con compatibilidad con navegadores modernos y ha sido una bendición para la construcción de patrones responsivos en el desarrollo web moderno también.

JavaScript: Es un lenguaje de programación interpretado que es un lenguaje orientado a objetos construido sobre el estándar ECMAScript (Luna, 2019). Su concepto principal es que puede usarse para crear páginas web interactivas que incrustan su código en documentos HTML, recibiendo directamente las acciones de los usuarios como clics o movimientos del ratón y adaptándose a esas acciones. Debido a que puede capturar eventos en vivo, es una herramienta esencial para construir un comportamiento dinámico del sistema en un navegador. Con la evolución de JavaScript, este lenguaje superó sus primeras limitaciones, gracias a la ayuda de recursos como Node.js, que extiende las aplicaciones de JavaScript más allá del lado del cliente e incluye operaciones del lado del servidor que se considerarían menos aplicables en su lenguaje nativo. Su popularidad proviene de su facilidad de uso con múltiples lenguajes de programación y plataformas diferentes, así como de un léxico que se basa fuertemente en la sintaxis tomada de otros lenguajes como C, Java, y es fácil de aprender. Su capacidad para comunicarse con variables, objetos y funciones de manera sensata ha hecho de JavaScript (el lenguaje principal para transformar la experiencia del usuario en programas interactivos con una función efectiva) el punto central de todos los lenguajes de programación. React: Es una biblioteca JS de aplicaciones web creada por Facebook utilizada para crear interfaces de usuario potentes y reutilizables (Spinola, 2023). React, el proyecto de desarrollo Front-End más utilizado en el mundo (Hoyo, 2024), fue

elegido dada su dominancia en el desarrollo declarativo. Este enfoque permite a los desarrolladores especificar lo que los usuarios quieren de la interfaz, lo cual es reflejado automáticamente por la biblioteca en la pantalla. Sus principales contribuciones son el uso de componentes reutilizables, que dividen la interfaz en bloques de código independientes, la utilización de JSX y el DOM virtual, que mejora el rendimiento con actualizaciones bien organizadas y predecibles. Y estas ideas de diseño ayudan a superar problemas generalizados relacionados con la manipulación del DOM en aplicaciones y la sincronización de estado, por lo tanto, es ampliamente utilizado en plataformas de medios populares como Netflix y Airbnb (Hoyo, 2024).

Vite: es una herramienta de construcción y un entorno rápido para desarrollar una aplicación que ayuda a mejorar la experiencia de los desarrolladores, especialmente si sus proyectos utilizan bibliotecas como React (Khider Ismail, 2025). A diferencia de las soluciones convencionales, que crean un proyecto completo por adelantado, Vite utiliza un modelo de servidor de desarrollo basado en módulos ES. Este enfoque permite un inicio casi en tiempo real y una recarga en caliente a una tasa de eficiencia tal que las aplicaciones desarrolladas con React y otros frameworks pueden desarrollarse más rápido. Es una alternativa moderna y poderosa para el desarrollo de SCPP con una configuración mínima y buena compatibilidad con JSX.

2.2.2.2. *Desarrollo del Back-End (Lógica de Negocio y Gestión de Datos)*

El Back-End del SCPP es el núcleo del sistema, responsable de la lógica de negocio, la exposición de las APIs, la interacción con la base de datos y el procesamiento de la información, permitiendo la comunicación fluida con el Front-End. Para este fin, eligio y se construirá con las siguientes tecnologías:

Python: Python es un lenguaje de programacion de alto nivel con una sintaxis simple e intuitiva que facilita la creación de código legible y eficiente. Es bueno para la ciencia de datos, el aprendizaje automático y, lo más importante para este proyecto, el análisis estadístico y la visualización de datos (Holguín Londoño, 2024), ya que su estructura orientada a objetos y su multifuncionalidad lo convierten en la solución. Con su extensa comunidad y bibliotecas especializadas como NumPy y Pandas, Python acomoda varios tipos de datos numéricos y tabulares. Herramientas como Matplotlib y Seaborn, a su vez, facilitan la producción de visualizaciones claras y personalizables, un aspecto esencial para los informes del sistema y los paneles de control. Esto permite su integración sin problemas con otros lenguajes y también permite su integracion con varias bases de datos y plataformas web; por lo tanto, es la solución flexible y escalable adecuada para construir un sistema que necesite procesar datos en tiempo real mientras automatiza tareas. Python también proporciona la capacidad de transformar información en conocimiento útil necesario para la toma de decisiones estratégicas.

Django: Ofrecimos Django, un marco de desarrollo web gratuito y de código abierto escrito en Python. Es una modificación del patrón clásico MVC de diseño de bases de datos y se basa en una arquitectura de diseño basada en el patrón MTV (Modelo-Plantilla-Vista). El Modelo es para el acceso a datos, la Vista ejecuta la lógica de negocio y la Plantilla es la capa de

presentación (Rosenbloom, 2018; Viejo Pomata, 2020). Django es particularmente adecuado ya que utiliza la modularidad y cuenta con un fuerte sistema de mapeo objeto-relacional (ORM) que facilita la interacción con la base de datos. También prioriza a sus usuarios por encima de todo, debido a su enfoque en la seguridad, con actualizaciones regulares y herramientas para evitar las vulnerabilidades estándar que vienen con la tecnología de la información, y por lo tanto es una opción segura y estable para un sistema empresarial. La amplia popularidad de Django entre empresas globales como Instagram y Spotify confirma su robustez y escalabilidad.

Django REST Framework (DRF): Para el desarrollo de API que se utilizará para la comunicación entre el Front-End (React) y el Back-End (Python/Django), la API se creará utilizando Django REST Framework (DRF). Esta extensión de Django amplía el estándar de Django para apoyar el desarrollo de Interfaces de Programación de Aplicaciones (APIs) construidas a lo largo de la interfaz REST para servicios web impulsados por API. DRF permite el desarrollo rápido de APIs fuertes, seguras y mantenibles, lo cual es excelente para trabajos que separan explícitamente la lógica del front-end y del servidor. (Rosenbloom, 2018; Viejo Pomata, 2020)

DRF se ha consolidado como el estándar de facto para el desarrollo de APIs en Django, con una adopcion masiva en empresas como Sentry y Robinhood, muestra sus caracteristicas clave se incluyen una interfaz web navegable para facilitar las pruebas de los endpoints, soporte para diferentes métodos de autenticación (como tokens y OAuth), tambien posee una capacidad de personalizacion profunda que permite adaptar la lógica de negocio a las necesidades específicas del SCPP, DRF en su arquitectura modular y la sólida comunidad que la respalda garantizan un desarrollo flexible y seguro, alineado con los mismos principios de seguridad que el núcleo de Django.(Rosenbloom, 2018; Viejo Pomata, 2020)

SQLITE: Como el sistema de datos persistentes es gestionado por SQLite, una base de datos relacional de código abierto. Esta es una base de datos que no requiere un proceso de servidor intermedio para la transmision de datos. Su im plementación es almacenar toda la base de datos de tablas, índices y registros en un solo archivo en disco. Dado que la aplicación en sí esta incrustada, no requiere configuraciones complicadas que la conviertan en una capa de abstracción entre el usuario y los datos, accediendo directamente a través de una biblioteca de programación. Como señaló Pawlaszczyk, esta forma de simplicidad y portabilidad ha hecho que SQLite sea adecuada para aplicaciones de escritorio, moviles e integradas, lo que la convierte en la plataforma perfecta para sistemas locales como el SCPP. El archivo de base de datos SQLite comienza con un encabezado de tamaño de 100 bytes con el resto dividido en páginas de tamaño uniforme que son la unidad de almacenamiento base del archivo. Así, estas páginas se presentan en una estructura de árbol B+ para un acceso rápido a los datos (Pawlaszczyk, 2022). La primera pagina es el esquema de la base de datos, que se encuentra dentro de una estructura especial llamada tabla SQLite_Master para encontrar las páginas raíz de cada tabla. En un árbol B+, las páginas hoja contienen datos y las páginas interiores presentan enlaces que permiten la navegacion. Todos los datos adicionales que se almacenaron en páginas de desbordamiento están vinculados entre sí para gestionar una gran cantidad de datos si el registro en una parte era demasiado grande para la página. Además, SQLite optimiza su espacio reutilizando automáticamente las páginas liberadas después de eliminar registros. Adicionalmente, dada su portabilidad y facilidad de mantenimiento, esta arquitectura es ex tremadamente poderosa en entornos con recursos limitados, al tiempo que también proporciona la versatilidad para almacenar toda la información en un solo archivo.

VS Code (Visual Studio Code): No es una tecnología de desarrollo directa, pero VS Code es un entorno de desarrollo integrado (IDE) ligero y extensible que será la herramienta principal para cualquiera que quiera construir. Proporciona resaltado de sintaxis, depuración, integración con

control de versiones (Git) y una amplia gama de extensiones diseñadas para mejorar el flujo de trabajo para Python, JavaScript y sus frameworks (Microsoft, 2024).

2.2.2.3. *Herramientas de Inteligencia de Negocios y Visualización*

El análisis y la presentación de datos son importantes para la toma de decisiones y gracias a un modelo de autoservicio efectivo y una compatibilidad cruzada muy eficaz, la plataforma es considerada un líder en el cuadrante mágico de Gartner y proporciona retornos lucrativos en un período de recuperación de menos de seis meses para las empresas que la implementan, según afirman los estudios. Como puede reunir datos de múltiples bases de datos (bases de datos SQL, Excel y APIs), esta es una herramienta ideal para que el equipo de gestión de S&M Group tome decisiones tácticas y estratégicas.

Tableros de control: los tableros de control son una representación visual de los KPIs y métricas relacionadas con un proceso u organización, estos tienen el propósito principal de rastrear rápidamente los desarrollos y resaltar anomalías (Few, 2006). Este proyecto utilizará dos tipos de tableros de control que son necesariamente adecuados:

OPD(Dashboard Operativo): SCPP tendrá un tablero de control interactivo dentro de su aplicación web (React) para el registro y monitoreo diario de operaciones, este se basa en estos cálculos, este tablero mostrará información de producción en tiempo real, distribución de tareas y el estado de los módulos de trabajo individuales. Estos son algunos de los tableros estratégicos. Los tableros se generarán para un análisis más profundo e informes estratégicos. Los tableros

basados en plantas integrarán datos de plantas con otros datos significativos, para que la gestión de S&M Group pueda comparar el rendimiento pasado, evaluar el rendimiento a largo plazo y tomar decisiones más inteligentes basados en analisis de datos. Esta división de tableros proporciona estratégicamente al personal operativo una herramienta ágil para su trabajo, mientras que la gestión está bien posicionada para la evaluación estratégica.

2.3. Estudios Previos

Los estudios de literatura y proyectos proporcionan un marco de referencia útil para lo que funciona bien, los desafíos comunes y las oportunidades de mejora. MES para PYMES textiles: Un MES, una combinación de hardware (sensores y dispositivos IoT) y software que es capaz de recopilar, procesar y presentar los datos de las plantas en tiempo real, es un facilitador para la mejora en la operación (Kagermann, Wahlster & Helbig, 2013). Un estudio, como el de Ko, Lee & Cho (2022) en la industria de la moda rápida, demostró una disminución del 18% en el tiempo de cambio y un aumento del 12% en la productividad tras la introducción de la colaboración MES.

Los meta-análisis de la Industria 4.0 muestran de manera similar una disponibilidad promedio de máquinas y ganancias de rendimiento del 15% y 11%, respectivamente, tras la implementación de MES vinculados a la nube (Firouzi, Maroufi & Sarhadi, 2023). Estos resultados empíricos prueban la hipótesis de que la visibilidad de recursos limitantes del MES permite la gestión de cuellos de botella, aumentando así el rendimiento general, alineado con la Teoría de Restricciones (Goldratt & Cox, 2004).

Aceptación de Tecnología en Entornos Industriales: El Modelo de Aceptación de Tecnología (Davis, 1989) explica la utilidad percibida y la facilidad de uso, que son los dos determinantes más comunes de la intención de adoptar nuevas tecnologías. Usando MES

modernos (tableros visuales combinados con alertas configurables), estas características entran en juego, maximizando su impacto organizacional y la probabilidad de una implementación exitosa de SCPP dentro del Grupo S&M. El papel de los tableros en la toma de decisiones empresariales: Los tableros y herramientas de Inteligencia de Negocios como Davenport y Harris (2007) han mostrado cómo conducen a una transformación de la toma de decisiones de una actividad empresarial intuitiva a una actividad basada en datos que puede aumentar la eficiencia y la competitividad. Justifica el uso de tableros dinámicos en el sitio web del proyecto como un medio para crear informes gerenciales.

2.4. Relación con el Proyecto: Aplicación de Conceptos

Los conceptos y tecnologías introducidos a lo largo del marco teórico tienen una importancia directa y fundamental que impacta directamente en el desarrollo e implementaci ón del Sistema de Control y Planificación de la Producción (SCPP) para el Grupo S&M. **Resolviendo el problema:**

Como Sistema de Ejecución de Manufactura (MES), el SCPP resuelve directamente el problema de la planificación empírica y, en consecuencia, la falta de control en tiempo real. La centralización de datos, la automatización de procesos y la visualización en tiempo real a través de paneles de control ayudan a eliminar la duplicación de esfuerzos y son parte del proyecto de mejora estratégica, para optimizar la asignación de recursos y, en última instancia, optimizar la sostenibilidad de la empresa (Vivanco, 2017).

Concepto de diseño de soporte material: Se utilizarán principios de Manufactura Esbelta y Kaizen para diseñar las funcionalidades del sistema, el enfoque del sistema estará en la eliminación de desperdicios y la mejora continua del flujo de producci on. El SCPP también debe ser una herramienta de recopilación con módulos que ayudarán a reconocer áreas de mejora

de acuerdo con estas filosofías que, a su vez, beneficiarán la mejora de los indicadores de eficiencia. Full-stack: Esto facilitará una interfaz de usuario potente en HTML, Tailwind CSS, React y Vite que son adecuados para la implementación por parte del personal operativo. Python con Django y Django REST Framework proporcionarán un backend eficiente y escalable con la capacidad de manejar volúmenes masivos de datos y ejecutar lógica de negocio compleja. La base de datos SQLite proporcionará esta información crítica. Una solución completa de alto rendimiento debido a esta sinergia tecnológica. Relevancia del contexto y trabajos previos: En estudios y teorías pasadas como el Modelo de Aceptación de Tecnología, se reconoce la relevancia para la solución. Experiencias similares de otras pymes que adoptan sistemas similares y los beneficios del monitoreo en tiempo real en el sector textil confirman el enfoque del SCPP, y confirman que la implementación requerida debe adaptarse estratégicamente a los requisitos del Grupo S&M.

Relevancia del Contexto y Estudios Previos: Los estudios previos y las teorías como el Technology Acceptance Model confirman la pertinencia de la solución propuesta. La experiencia de otras PYMES en la adopción de sistemas similares y los beneficios del monitoreo en tiempo real en la industria textil, validan el enfoque del SCPP y resaltan la importancia de una implementación cuidadosa y adaptada a las necesidades específicas del Grupo S&M.

2.5. Planteamiento e Identificación de las Variables de Investigación

Esta sección define las variables clave que serán objeto de estudio para evaluar el impacto del sistema propuesto.

2.5.1. *Definición de Variable*

En esta investigación tecnológica realizada, una variable es una caracteristica, propiedad o atributo que puede variar y que es susceptible de ser medida, observada o controlada en un estudio, a continuación mostramos los tipos de variables aplicadas:

2.5.2. *Tipos de Variables*

Variable Independiente (VI): Es aquella que el investigador manipula o introduce en el estudio para observar sus efectos sobre la variable dependiente. Se presume que es la causa o el factor que influye en el resultado.

Variable Dependiente (VD): Es aquella que se mide o se observa y que cambia como resultado de la manipulación de la variable independiente. Es el efecto o el resultado que se quiere investigar.

2.5.3. *Identificación de Variables en el Proyecto Variable Independiente (VI):*

Nombre: Implementación de un sistema digital de control y planificación de la producción.

Definición Conceptual: Plataforma socio-técnica que busca integrar un software (y potencialmente hardware, como indicadores) para capturar, procesar y visualizar datos de una planta de producción en tiempo real, facilitando la asignación de recursos y la toma de decisiones basadas en evidencia (Kagermann, Wahlster & Helbig, 2013). Su 'utilidad percibida' y 'facilidad de uso' son críticas para su adopción y maximizar su impacto organizacional (Davis, 1989).

Definición Operacional: Presencia y funcionalidad de un tablero digital activo en la planta del Grupo S&M que:

- Registra manualmnete y automáticamente datos de cada módulo de confección (inicio/fin de tarea, unidades producidas).
- Proporciona funcionalidades de asignación de tareas y monitoreo de cuellos de botella.
- Permite la generación de reportes descargables

Variable Dependiente (VD):

Nombre: Eficiencia operativa del proceso de confección.

Definición Conceptual: Grado en que la planta convierte insumos (material, mano de obra y tiempo) en prendas terminadas, cumpliendo estándares de calidad y plazos de entrega. Es un indicador clave para competir en costo y rapidez en el modelo 'fast fashion' (Slack & Lewis, 2020).

Definición Operacional: Medida a través de los siguientes indicadores, calculados y monitoreados por el MES:

(a) Overall Equipment Effectiveness (OEE): OEE=Disponibilidad×Desempeño×Calidad, calculado para cada línea o módulo de producción (Nakajima, 1989). Un incremento en OEE se asocia con reducción de costos unitarios y lead time (Boston Consulting Group, 2024).

(b) Cumplimiento del cronograma de producción semanal: Porcentaje de órdenes de producción terminadas y entregadas en la fecha establecida. Este indicador captura la perspectiva cliente-entrega y se alinea con indicadores OTIF (On-Time In-Full) de la cadena de suministro.

2.6. Relación Hipotética

La relación entre las variables se formaliza en la siguiente hipótesis:

H1: La implementación del sistema digital de control y planificación (Manufacturing Execution System, MES) incrementará significativamente la eficiencia operativa del proceso de confección en la planta del Grupo S&M, al reducir tiempos muertos, equilibrar la carga de trabajo y habilitar la toma de decisiones basada en datos.

Capítulo 3: Metodología de la Investigación

Este capítulo detalla el enfoque metodológico que guiará el diseño, desarrollo e implementación del Sistema de Control y Planificación de la Producción (SCPP) en el Grupo S&M. Aborda la justificación de las metodologías seleccionadas, la descripción de las fases del proceso, las herramientas y recursos utilizados, el cronograma de actividades y los procedimientos de validación, asegurando un marco estructurado y sistemático para la investigación aplicada.

3.1. Enfoque y Tipo de Investigación

El método de este estudio es de naturaleza mixta, por lo que se recopilarán y analizarán datos cuantitativos, como la Eficiencia General de los Equipos - OEE y la adherencia al programa de producción, para evaluar la eficiencia operativa junto con datos cualitativos para el diagnóstico inicial, la comprensión de los procesos y la evaluación de la usabilidad del sistema.

La investigación de este proyecto es de naturaleza aplicada, ya que su principal objetivo es el desarrollo e implementación de una solución tecnologica que aborde un problema real en el Grupo S&M y genere un impacto tangible en su eficiencia operativa. Junto a esto, el estudio es descriptivo, ya que explica el estado de los procesos de producción antes de la intervención.

3.2. Diseño de la Investigación

Se empleará un diseño cuasi-experimental de pre-prueba/post-prueba con un solo grupo dentro de la planta de producion. Esto implica la medición de las variables dependientes (indicadores de eficiencia operativa) antes de la intervención (fase diagnóstica inicial) y nuevamente después de la implementación y período de estabilización del SCPP. Este diseño permitirá observar y analizar los cambios en la eficiencia operativa atribuidos directamente a la implementación del sistema.

3.3. Población, Muestra e Instrumentos de Recolección de Datos

La población de estudio comprende a todo el personal del Grupo S&M directamente involucrado en los procesos de confección formal (áreas de corte, costura, acabados y despachos), así como al personal administrativo y de supervisión de la producción. Se trabajará con la población completa de estas áreas específicas debido al tamaño manejable del grupo y la necesidad de una implementación integral del sistema para evaluar su impacto.

Los instrumentos de recolección de datos incluyen:

Observación Directa: Para el levantamiento de información sobre el flujo de trabajo actual, tiempos de ciclo, identificación de cuellos de botella y gestión de actividades in situ.

Entrevistas Semi-estructuradas: Dirigidas al personal clave (gerentes de producción, supervisores, operarios experimentados) para el diagnóstico profundo de la problemática, la elicitación de

requisitos funcionales y no funcionales, y la obtención de retroalimentación cualitativa durante las fases de desarrollo y prueba.

Análisis de Documentos: Revisión de los registros de producción existentes (hojas de cálculo, informes manuales) para obtener datos históricos que servirán como base para la prueba previa.

Registro Automatizado SCPP: El sistema en sí será el principal instrumento para la recolección de datos en tiempo real una vez implementado, registrando auto máticamente las unidades producidas por modulo, los tiempos de actividad y los incidentes, lo que alimentará los cálculos de OEE y el seguimiento de horarios.

Cuestionarios de Usabilidad: Aplicados después de la implementación para evaluar la facilidad de uso, la utilidad percibida y la satisfaccion general del u suario con el SCPP, complementando el análisis cuantitativo con la percepción del usuario.

3.4. Justificación de la Metodología Seleccionada

En el proceso de desarrollo del SCPP, se eligió un enfoque con la metodologia Scrum para la gestión de proyectos de software, recomendado y eficiente para este tipo de proyectos.

Scrum: La elección de este enfoque agil se justifica debido al hecho de que los proyectos de desarrollo de software en industrias basadas en campos dinámicos, como la industria textil, son cambiantes y complejos por naturaleza. Un enfoque Scrum permite la entrega iterativa e incremental de funciones con aportes regulares del Grupo S&M, una rápida adaptación a nuevos requisitos y la mitigación de riesgos. Esto garantiza que el sistema final no solo funcione, sino que también se corresponda bien con las necesidades operativas y estratégicas de la empresa, y así pueda aumentar la probabilidad de una implementación y adopción exitosa.

La preparación, modelado, evaluación y despliegue, aqui se puede ver su naturaleza cíclica se alinea con la mejora continua y asegura que los datos recolectados por el sistema sean transformados en accionables y utiles para la toma de decisiones de los encargados y gerenciales.

3.5. Descripción de Fases o Etapas del Proceso

El proceso metodológico se estructura en fases que integran los principios de Scrum

3.5.1. Fase 1: Diagnóstico y Comprensión (Integración Scrum)

Esta fase inicial se enfoca en el levantamiento de requisitos y la comprensión profunda del negocio y los datos existentes.

Actividades:

Diagnóstico de la situación actual: Observación directa de los procesos de confección y entrevistas con el personal para identificar cuellos de botella y la problemática de la planificación empírica (enfoque cualitativo).

Comprensión del Negocio : Definición clara de los objetivos de negocio del Grupo S&M (e.g., aumentar eficiencia, reducir tiempos de entrega), y cómo el SCPP contribuirá a ellos.

Comprensión de los Datos : Análisis de los datos históricos de producción disponibles (hojas de cálculo) para entender su estructura, calidad y limitaciones.

Definición de Requisitos (Scrum): Elicitación y priorización de las funcionalidades del SCPP, creando un Product Backlog inicial.

3.5.2. Fase 2: Diseño y Planificación del Desarrollo (Integración Scrum)

Se traduce el entendimiento en un plan de acción y un diseño técnico.

Actividades:

Diseño de la Arquitectura del SCPP: Definición de la arquitectura cliente-servidor, componentes Front-End y Back-End, y la API RESTful.

Diseño de Base de Datos: Modelado de la base de datos SQLite, definiendo tablas, relaciones y esquemas de datos para el almacenamiento de la información de producción.

Diseño de Interfaz de Usuario (UI/UX): Prototipado y diseño de la interfaz visual del tablero de control y los módulos del sistema (basado en principios de usabilidad y Tailwindcss).

Planificación de Sprints (Scrum): Definición de la duración de los sprints y planificación del primer sprint con las funcionalidades de mayor prioridad del Product Backlog.

Preparación de Datos : Definición de cómo se limpiarán, integrarán y transformarán los datos para ser consumidos por el sistema y mostrarlos en resportes.

3.5.3. Fase 3: Desarrollo Iterativo (Scrum)

El núcleo de la construcción del sistema, desarrollado en sprints.

Actividades (por Sprint):

Desarrollo del Front-End: Construcción de la interfaz de usuario con React, Vite , HTML y Tailwindcss, implementando la visualización del tablero en tiempo real y los módulos de registro/asignación.

Desarrollo del Back-End: Implementación de la lógica de negocio con Python/Django, creación de la API RESTful con Django REST Framework y manejo de la persistencia de datos con SQLite.

Esto incluye validaciones

Pruebas Unitarias e Integración Continua: Realización de pruebas constantes para asegurar la funcionalidad y la integración entre los componentes.

Daily Scrums: Reuniones diarias para seguimiento del progreso y resolución de impedimentos.

3.5.4. Fase 4: Evaluación y Despliegue (Integración Scrum)

Esta fase se centra en la validación de las funcionalidades y la puesta en producción.

Actividades:

Pruebas de Integración y Sistema: Verificación de la funcionalidad completa del SCPP, incluyendo la comunicación entre front-end y back-end, y la integridad de la base de datos.

Evaluación : Validación de que el sistema cumple con los objetivos de negocio definidos, calculando y presentando los KPIs (OEE, cumplimiento de cronograma) de manera precisa.

Pruebas Piloto y Retroalimentación (Scrum): Despliegue del sistema en un entorno controlado dentro del Grupo S&M con un grupo reducido de usuarios para obtener retroalimentación temprana.

Despliegue (Deployment): Puesta en producción del SCPP para todas las áreas definidas, asegurando su operatividad y estabilidad.

Capacitación de Usuarios: Entrenamiento del personal operativo y administrativo en el uso del SCPP y la interpretación de los dashboards.

3.5.5. Fase 5: Monitoreo y Mejora Continua (Post-implementación)

Esta fase garantiza la sostenibilidad y evolución del sistema.

Actividades:

Monitoreo de la Eficiencia Operativa: Recolección continua de datos del SCPP para el cálculo de los indicadores de eficiencia (OEE, cumplimiento) durante un periodo de observación de dos semanas.

Análisis Post-Implementación: Comparación de los datos post-implementación con la línea base (pre-prueba) para evaluar el impacto del SCPP.

Retroalimentación Continua (Scrum): Establecimiento de canales para que los usuarios reporten problemas y sugieran mejoras.

Retrospectivas Periódicas (Scrum): Revisiones post-implementación para identificar lecciones aprendidas y oportunidades para futuras iteraciones o mejoras del sistema.

3.6. Herramientas y Recursos Utilizados

Además de las tecnologías de desarrollo detalladas en el Marco Teórico (Python, DjangoRest framework,D React, Vite, SQLite, VS Code, HTML, CSS, JavaScript), se emplearán los siguientes recursos y herramientas para la gestión y ejecución del proyecto:

Herramientas de Gestión de Proyectos: Tablas en excel para la gestión del Product Backlog, Sprint Backlog y seguimiento de tareas bajo la metodología Scrum.

Control de Versiones: Git y GitHub/GitLab para el control de versiones del código fuente, facilitando la colaboración y la gestión de cambios.

Comunicación: Plataformas como Microsoft Teams para la comunicación diaria del equipo de desarrollo y con los stakeholders del Grupo S&M.

Entorno de Desarrollo: Máquinas de desarrollo configuradas con los entornos necesarios para Python/Django y React.

Servidores: Entorno de servidor para el despliegue del sistema (en una máquina virtual local o en la nube para pruebas y producción).

3.7. Cronograma de Actividades

Se propone un cronograma de actividades desglosado en fases, a ser gestionado mediante sprints de duración fija (2-3 semanas por sprint). A continuación, se presenta una estimación general para un periodo de cuatro meses de implementación y observación:

Columna1	Columna2	Columna3	Columr	Columr	Columr	Columr
Fase de la Metodología	**Actividades Clave**	**Duración Estimada**	**Mes 1**	**Mes 2**	**Mes 3**	**Mes 4**
1. Diagnóstico y Comprensión	Levantamiento de requisitos, entrevistas, análisis de procesos, definición de KPIs, Product Backlog inicial.	2-3 semanas	X			
2. Diseño y Planificación	Diseño de arquitectura, base de datos, UI/UX, prototipado, planificación de primeros sprints.	2-3 semanas	X			
3. Desarrollo Iterativo (Sprints 1-4)	Desarrollo Front-End, Back-End, API RESTful, integración SQLite, pruebas unitarias.	8 semanas (4 sprints)	X X X	X X X		
4. Evaluación y Despliegue	Pruebas de integración/sistema, despliegue piloto, capacitación, integración Power BI, despliegue final.	3-4 semanas		X		
5. Monitoreo y Mejora Continua	Recolección de datos post-implementación, cálculo de OEE y cumplimiento, análisis de impacto, retroalimentación.	16 semanas (continua)			X X X	
6. Análisis y Redacción Final	Análisis de resultados, conclusiones, redacción de la tesis.	Posterior a las fases anteriores				

Ilustración 1. Tabla de Estimación general para un periodo de 4 meses de implementación, observación y pruebas

Ilustración 2. Cronograma de etapas 1 Y 2 a seguir de la metodología SCRUM.

Como se aprecia en la imagen, se representan las dos primeras fases del proyecto: la preparación de requerimientos y la definición de la base de datos. Además, se muestra la segunda fase, que corresponde al diseño y la planificación. Esta última incluye una serie de actividades que cuentan con un plazo específico para su cumplimiento, de acuerdo con el cronograma elaborado en ExceL

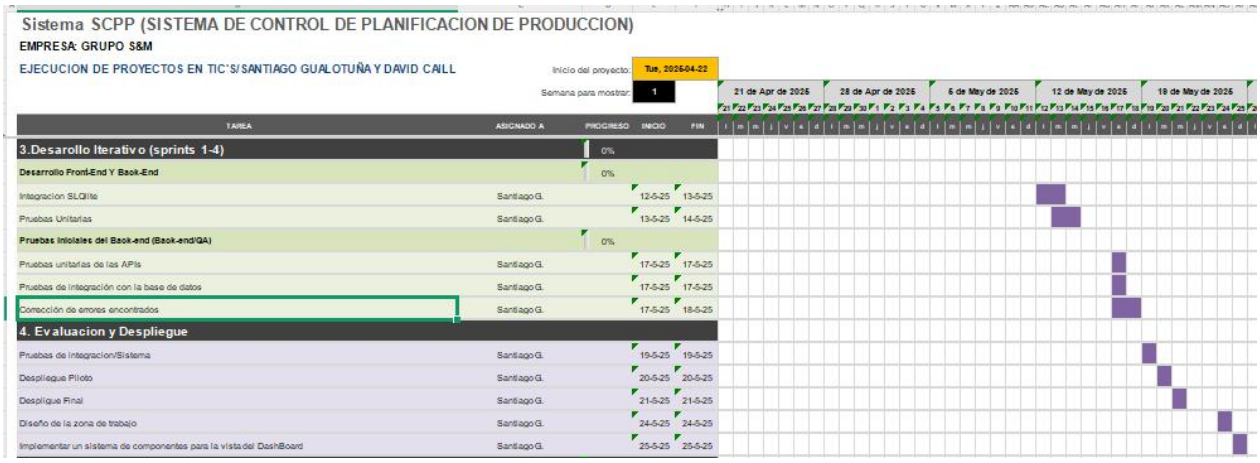


Ilustración 3. Cronograma de etapas 3 y 4 a seguir de la metodología SCRUM.

Como se aprecia en la imagen, se representan las fases tres y cuatro del proyecto: el desarrollo iterativo de sprints y la evaluación y despliegue. Estas fases incluyen una última serie de actividades que cuentan con un plazo específico para su cumplimiento, de acuerdo con el cronograma elaborado en Excel

Ilustración 4. Cronograma de etapas 5 y 6 a seguir de la metodología SCRUM.

Como se observa en la imagen, se muestran las fases cinco y seis del proyecto: Monitoreo y mejora continua y Desarrollo de funcionalidades y restricciones dentro de cada módulo. En estas fases se incluyen actividades como la recolección y análisis de datos, retroalimentación, corrección de errores, así como la implementación de funciones de filtrado y control dentro de los módulos. Cada una de estas tareas cuenta con plazos definidos para su cumplimiento, de acuerdo con el cronograma establecido en Excel.

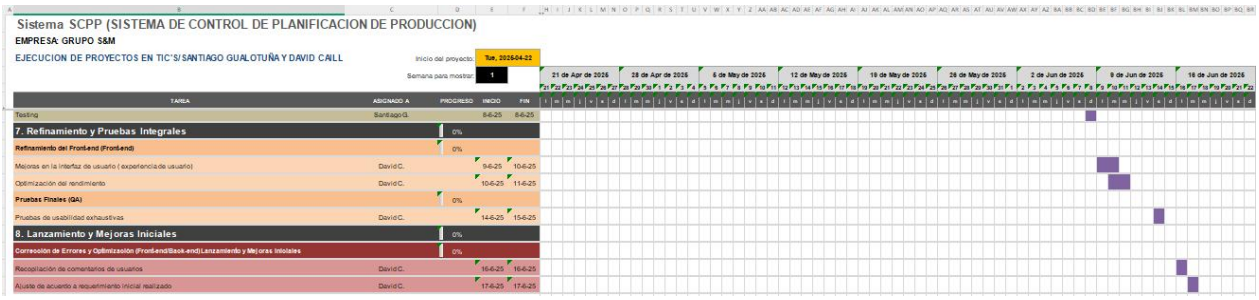


Ilustración 5. Cronograma de etapas 7 y 8 a seguir de la metodología SCRUM.

Como se puede ver en la imagen número 5, las fases siete y ocho del proyecto se pueden resumir de la siguiente manera: Refinamiento y pruebas exhaustivas, y Lanzamiento y mejoras iniciales. Durante estas fases, se consideran la optimización del rendimiento, la mejora de la experiencia del usuario, pruebas de calidad ex haustivas, así como la corrección de errores y el ajuste final de acuerdo con los requisitos iniciales. Estas tareas están programadas con un cronograma definido, basado en el calendario establecido en Excel, lo que garantiza un cierre del proyecto suave y eficiente.

Nota: La duración de los sprints y el número total se ajustarán dinamicamente según la complejidad de las funcionalidades y los comentarios del Grupo S&M.

3.8. Procedimientos de Validación

La validación del SCPP y de sus resultados obtenidos al probar el sistema se realizará mediante una combinación de testeo técnico, métricas de rendimiento y retro alimentacion de los usuarios.

3.8.1. Testeo y Calidad del Software

Pruebas Unitarias: Se realizarán pruebas automatizadas para validar la funcionalidad de componentes individuales (funciones, métodos, clases) en el Front-End (JavaScript/React) y Back-End (Python/Django).

Pruebas de Integración: Verificación de la correcta comunicación entre los módulos del sistema (Front-End con Back-End vía API RESTful, Back-End con SQLite).

Pruebas de Sistema: Evaluación integral del SCPP como un todo, asegurando que cumple con los requisitos funcionales y no funcionales (rendimiento, seguridad, responsividad).

Pruebas de Aceptación del Usuario (UAT): Realizadas por el personal del Grupo S&M para validar que el sistema satisface sus necesidades operativas y es intuitivo de usar.

3.8.2. Métricas de Rendimiento y Eficiencia

Métricas de Productividad: El SCPP calculará automáticamente:

Overall Equipment Effectiveness (OEE): Como se definió en el Marco Teórico, para cada módulo o línea de producción.

Unidades Producidas por Hora/Módulo/Empleado: Para evaluar la capacidad de producción y la eficiencia individual.

Tiempo de Ciclo de Producción: Reducción del tiempo que toma una prenda en pasar por las etapas de confección.

Métricas de Cumplimiento:

Porcentaje de Órdenes Completadas a Tiempo: Comparación entre la fecha de entrega prometida y la fecha de finalización real registrada por el sistema.

Reducción de Cuellos de Botella: Medición de la frecuencia y duración de las acumulaciones de trabajo en puntos específicos, a través de los datos del sistema.

3.8.3. Retroalimentación y Mejora Continua

Sesiones de Retroalimentación: Se realizarán reuniones periódicas con los usuarios clave para recoger sus impresiones, identificar puntos de mejora y ajustar el sistema según sea necesario. Esto es fundamental para la aceptación del usuario (Davis, 1989).

Análisis Comparativo (Pre vs. Post): Comparación de las métricas de eficiencia operativa recolectadas antes y después de la implementación del sistema para validar la hipótesis planteada.

3.9. Consideraciones Éticas

La Este estudio se implementará bajo rigurosos estándares éticos. Se asegurará la confidencialidad de la información sensible del Grupo S&M y los datos personales de los colaboradores. Se obtendrá el consentimiento informado de todos los participantes involucrados dentro del sistema asegurando su participación voluntaria y su derecho a retirarse en cualquier momento.

CAPÍTULO IV: DISEÑO Y DESARROLLO DE LA SOLUCIÓN

El diseño y desarrollo de un sistema informático es una de las partes más cruciales de cualquier proyecto tecnológico. Esta aplicación se inició para realizar el Sistema de Control y Planificación de la Producción (PCPS) para el Grupo S&M con el fin de revolucionar el modelo operativo en cómo la empresa maneja el proceso de producción, pasando de una solución manual y dispersa a una auto matizada y en línea. Este capítulo discute las decisiones de diseño seleccionadas, la justificación de las tecnologías identificadas utilizadas, el modelado de bases de datos, el diseño de la interfaz de usuario y la fase de desarrollo de acuerdo con la metodología ágil Scrum, permitiendo la entrega de una solución ágil y en continua evolución. Además, se discuten las medidas tomadas en relación con la seguridad del sistema, la im plementación y la capacitación del personal, que son muy necesarias e importantes para facilitar un proyecto exitoso.

4.1. Diseño General de la Solución

El SCPP se diseñó bajo una arquitectura cliente-servidor para garantizar la separación de responsabilidades, la escalabilidad y el mantenimiento del sistema. En esta estructura, el Front-End, desarrollado con React + Vite, HTML5, Tailwind CSS y JavaScript, se encarga de la interacción con el usuario y del diseño responsivo e intuitivo. La elección de React se basó en su modularidad, la reutilización de componentes y el amplio soporte de la comunidad, características que facilitaron la construcción de una interfaz dinámica. Por su parte, el Back-End fue construido con Python, Django y Django REST Framework, lo que proporcionó un entorno robusto y escalable para gestionar la lógica de negocio, la validación de reglas y el almacenamiento de datos.

La base de datos, un componente fundamental, se gestiona con SQLite, una opción ligera y portable ideal para entornos locales y proyectos que priorizan la simplicidad y la rapidez en el despliegue. La comunicación entre el Front-End y el Back-End se establece a través de una API RESTful, lo que asegura la independencia entre las capas y la posibilidad de integrar futuros servicios externos sin afectar la arquitectura general. Este diseño modular permite incorporar mejoras progresivas, como migrar a una base de datos más robusta o integrar funcionalidades predictivas sin alterar la arquitectura fundamental del sistema.

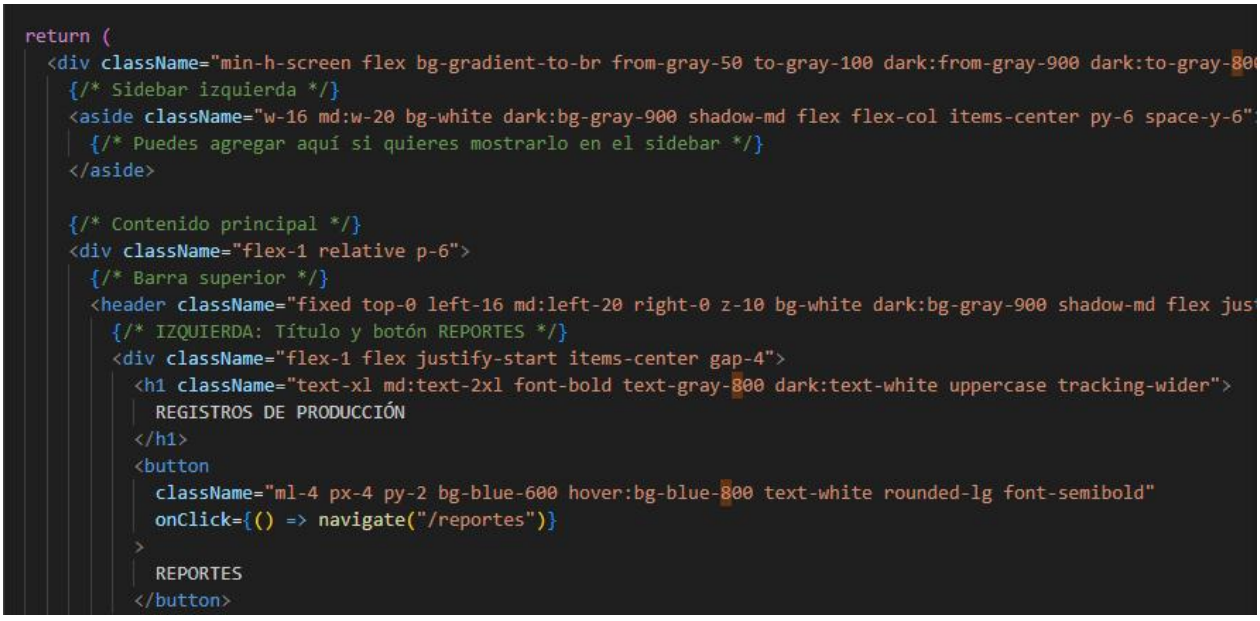
4.2. Diseño de la Base de Datos

La base de datos también sirve como una parte clave del sistema ya que mantiene todos los datos de producción relevantes, este concepto fue creado utilizando las entidades principales del proceso de fabricación, incluyendo la configuración de tablas (Productos (la identificación y cantidad planificada de cada prenda), Módulos (los procesos de producción), Registros de Producción (cantidades por turno y operador), Usuarios (por ejemplo, credenciales y datos básicos del personal), y Roles (permisos de acceso controlado)). El papel de la integridad referencial se aseguró mediante el uso de claves foráneas y los registros eran consistentes. Y luego se necesitan algunas restricciones, como verificar los datos cuantitativos, para evitar la sobreproducción o entradas negativas y asegurar la fiabilidad de la información. El Diagrama de Entidad-Relación (E-R) se utilizó como un mapa conceptual para producir las tablas mientras servía como un factor importante para verificar que la información del sistema permaneciera

intacta.

4.3. Diseño de la Interfaz y Experiencia de Usuario (UI/UX)

En el diseño de la interfaz de usuario se basó en principios de usabilidad (UX) y accesibilidad para permitir que el personal de la planta se adaptara rapidamente a la nueva herramienta, despues se creó una presentación gráfica de la producción diaria, el cumplimiento de módulos y la identificación de cuellos de botella para el panel principal. También se desarrollaron formularios de registro para permitir la entrada o edición de cantidades de producción por turno con validaciones en tiempo real. Se desarrollaron notificaciones y alertas para advertir sobre inconsistencias o superación de límites. La compatibilidad responsiva fue un factor crítico que permitió que la aplicación se utilizara en computadoras, y en televisores mediante conexión HDMI. Se priorizó la simplicidad del diseño, el uso de botones claros y mensajes de confirmación para que los usuarios finales pudieran entender el proceso fácilmente.



```
return (
  <div className="min-h-screen flex bg-gradient-to-br from-gray-50 to-gray-100 dark:from-gray-900 dark:to-gray-80
    {/* Sidebar izquierda */}
    <aside className="w-16 md:w-20 bg-white dark:bg-gray-900 shadow-md flex flex-col items-center py-6 space-y-6"
      {/* Puedes agregar aquí si quieres mostrarlo en el sidebar */}
    </aside>

    {/* Contenido principal */}
    <div className="flex-1 relative p-6">
      {/* Barra superior */}
      <header className="fixed top-0 left-16 md:left-20 right-0 z-10 bg-white dark:bg-gray-900 shadow-md flex just
        {/* IZQUIERDA: Título y botón REPORTES */}
        <div className="flex-1 flex justify-start items-center gap-4">
          <h1 className="text-xl md:text-2xl font-bold text-gray-800 dark:text-white uppercase tracking-wider">
            REGISTROS DE PRODUCCIÓN
          </h1>
          <button
            className="ml-4 px-4 py-2 bg-blue-600 hover:bg-blue-800 text-white rounded-lg font-semibold"
            onClick={() => navigate("/reportes")}
          >
            REPORTES
          </button>
```

Ilustración 6. Estructura para la interfaz visual incial del usuario.

En la imagen numero 6 se puede observar la estructura del código que define la interfaz principal del sistema, donde se implementa un diseño con React y TailwindCSS, mostramos en codigo la barra lateral y el contenido central tambien esta el botón de reportes, lo que evidencia la organización modular que facilita la navegación y la interacción del usuario. Esta parte del código mostramos cómo se buscó mantener claridad y usabilidad en el desarrollo del sistema.

```
{/* Lista de registros (resumen) */}
<div className="mt-10">
  <h2 className="text-2xl font-bold mb-4">GRUPO S&M🍀</h2>
  {/* Filtro de producto */}
  <input
    type="text"
    value={filtroProducto}
    onChange={e => setFiltroProducto(e.target.value)}
    placeholder="Buscar por producto..."
    className="mb-4 p-2 border border-blue-300 rounded-lg w-full max-w-xs text-sm"
  />
  <div className="grid gap-4">
    {registers
      .filter(register =>
        filtroProducto.trim() === "" ||
        (register.producto_id && register.producto_id.toLowerCase().includes(filtroProducto.toLowerCas
      )
      .map((register) => {
        const moduloInfo = modulos.find((m) => m.id === register.modulo);
        const isFinalizado = moduloInfo && moduloInfo.id === 8;
        return (
```

Ilustración 7. Estructura para los filtros y funcionalidades del sistema SCPP.

La imagen numero 7 presenta el filtro de productos en el sistema mediante el cual el usuario puede buscar registros de manera dinamica utilizando el identificador de producto. La función onChange se implementa para capturar la entrada del usuario y actualizar el estado, lo que facilita la visualización organizada de los registros en la interfaz. Este fragmento muestra la preocupación por la usabilidad y el acceso rápido a la información.

4.4. Desarrollo del Sistema

El sistema fue desarrolla do utilizando el metodo agil de Scrum, con el trabajo organizado en sprints de dos semanas para permitir una adición incremental de valor. Durante el primer sprint, se crearon los entornos, se modeló la base de datos y se desarrollaron los primeros endpoints de la API. Para el segundo sprint, trabajamos en el panel principal y el desarrollo del formulario de registro inicial. Despues, en el sprint 3, se realizaron validaciones de módulos, se hicieron informes internos y se mejoraron las consultas. Finalmente, el sprint 4 fue la prueba de integración, la reestructuracion de la interfaz y el despliegue en el entorno de producción. Se utilizaron herramientas como Git para el control de versiones y la gestión de horarios con tareas en Excel para mantener una buena colaboración en grupo.

```python
class NotaModulo(models.Model):
    modulo = models.IntegerField(choices=MODULOS)
    usuario = models.ForeignKey(Usuario, on_delete=models.CASCADE)
    texto = models.TextField()
    fecha_hora = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return f"Nota de {self.usuario} en módulo {self.get_modulo_display()} - {self.fecha_hora}"

class RegistroProduccion(models.Model):
    modulo = models.IntegerField(choices=MODULOS)
    producto_id = models.CharField(max_length=100, null=True, blank=True)
    cantidad = models.PositiveIntegerField()
    usuario = models.ForeignKey(Usuario, on_delete=models.CASCADE)
    fecha = models.DateTimeField(auto_now_add=True)
```

Ilustración 8.Modelos de la base de datos En Dajango.

En este fragmento se observa la definición de modelos en Django que permiten estructurar la información relacionada con las notas de los módulos y los registros de producción. La clase NotaModulo guarda las observaciones de un usuario en un módulo específico junto con la fecha, mientras que la clase RegistroProduccion almacena los datos clave de la producción como producto, cantidad y responsable. Este código nos muestra la base lógica que organiza y controla

la información en el sistema de manera correcta.

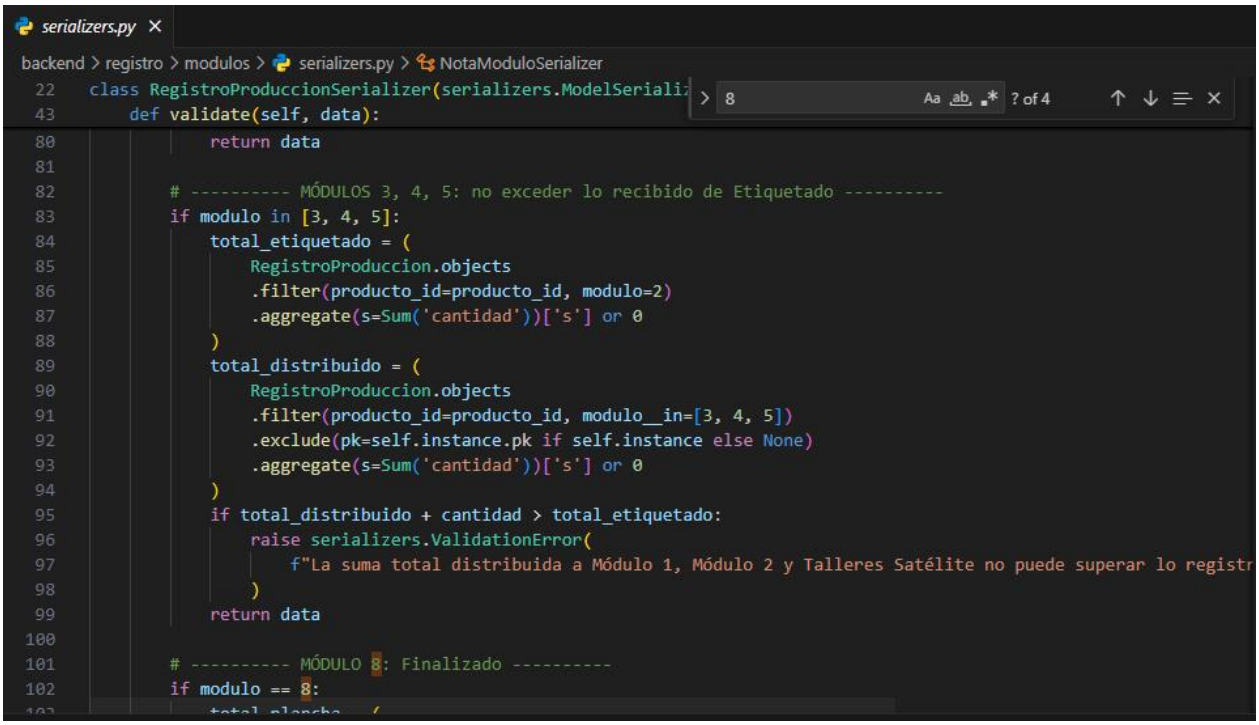


Ilustración 9.Validación de registros en el serializer

La imagen 9 es la validación en el serializador de producción, donde se establecen reglas para evitar que los módulos excedan las cantidades recibidas de la etapa de etiquetado. Este ejemplo muestra lo que se hace en consultas o sumas en la base de datos para comparar los valores registrados y asegurar la consistencia de la información. Este fragmento de codigo refleja la importancia del control interno en el sistema para mantener un flujo de datos confiable.

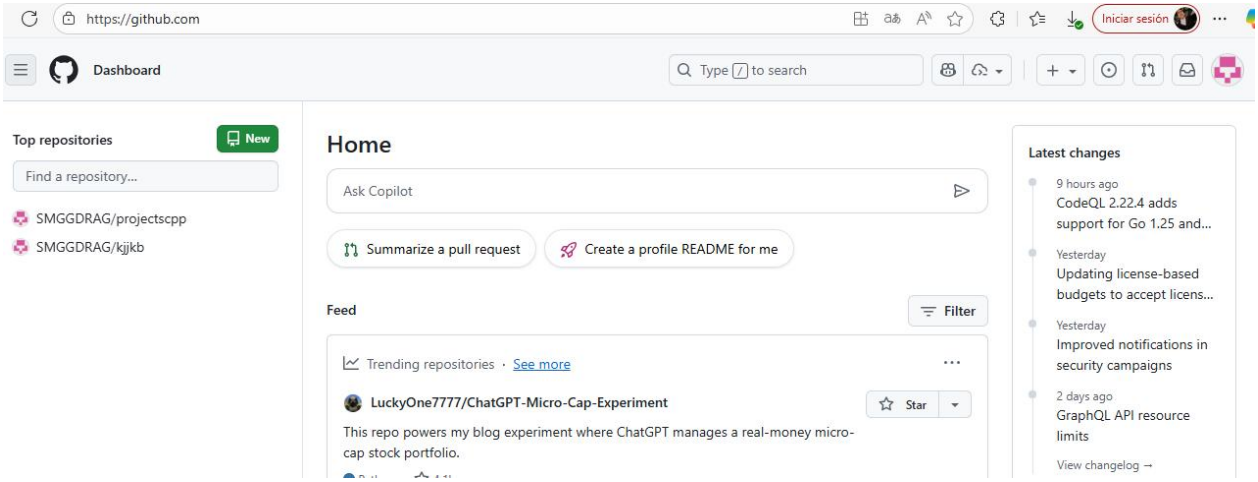


Ilustración 10. Interfaz principal de GitHub con versiones iniciales.

La imagen 10 es una captura de pantalla de GitHub en su totalidad y muestra los principales repositorios que se han subido y opciones para trabajar. En esta ventana se tiene el control sobre la gestión del código y la versión en la nube, donde se mantienen los proyectos organizados y también colaboran como equipo.

4.5. Desarrollo de Módulos Principales

El SCPP fue desarrollado como módulos funcionales, cada uno para diferentes usos y para permitir la gestión de proyectos que añadan gradualmente más funcionalidad, se implento las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) forman la base de la funcionalidad del sistema. Las siguientes funcionalidades fueron implementadas para cada entidad en el modelo: Crear (registrar nuevos datos), Leer (ver información), Actualizar (modificar datos) y Eliminar (borrar registros). Estas funcionalidades pueden demostrarse usando un fragmento de codigo de un serializador DRF (el componente responsable de convertir los datos del modelo en un formato que pueda ser manejado por la API), así se logro una conexión exitosa.

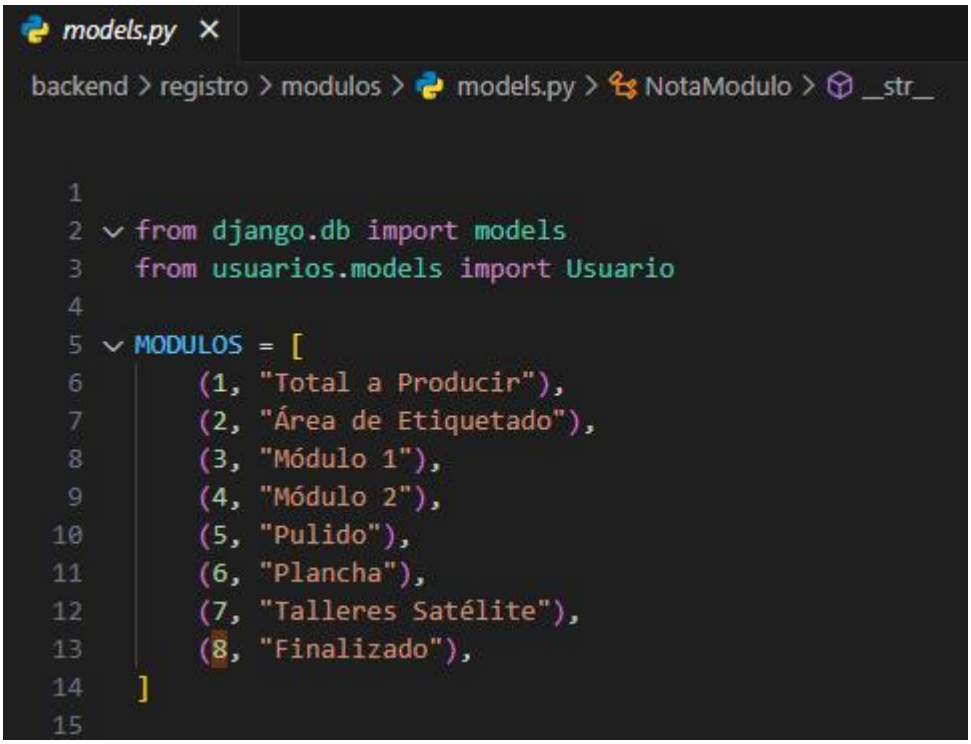


Ilustración 11. Definición de módulos en el sistema.

En esta imagen se observa la configuración de los módulos principales de la planta de producción dentro del archivo models.py. Cada módulo está identificado con un número y un nombre que representan las diferentes etapas del proceso de producción, como el área de etiquetado, pulido, planchado o talleres satélite. Esta estructura permite al sistema gestionar los registros de manera ordenada y facilita el control del flujo de producción.

4.6. Archivos de Migraciones

Los archivos de migración en Django fueron la columna vertebral para gestionar los cambios en el esquema de la base de datos, ya que estos archivos, generados auto maticamente por el

framework, registran de manera incremental todas las modificaciones realizadas a los modelos. Esto permitió un desarrollo colaborativo, ya que cada cambio en la estructura de la base de datos se reflejaba de manera segura y confiable en el entorno de producción. Como ejemplo, se muestra la captura donde se observa la estructura de la carpeta que contiene los archivos de migración.

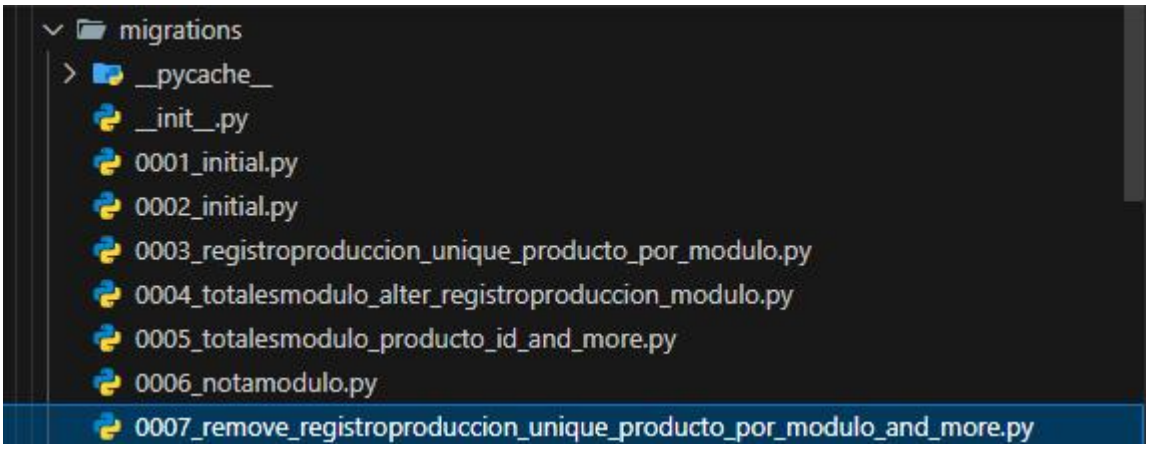


Ilustración 12. Archivos de migración en el sistema.

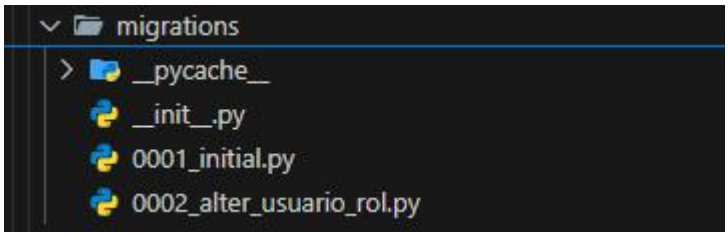


Ilustración 13. Migraciones iniciales de la base de datos.

La carpeta de migraciones se muestra gráficamente en la imagen, específicamente en el módulo de usuarios, donde se registran los cambios aplicados a la estructura de la base de datos, aqui cada archivo corresponde a una version que refleja el proceso iterativo de adaptación del sistema, incluyendo la creación inicial de tablas y modificaciones posteriores como la asignación de roles. Este mecanismo permite un control ordenado de los cambios en los modelos, asegurando consistencia, trazabilidad y la capacidad de reproducir la evolución del sistema en diferentes

entornos de desarrollo.

4.7. Consideraciones de Seguridad

La seguridad fue una consideración esencial en el proceso de diseño que llevó a SCPP, puesto que aqui se tomaron medidas estrictas, incluyendo la autenticación con tokens JWT (para proteger el acceso a la API) y el cifrado de contraseñas con algoritmos de hash para salvaguardar las credenciales de los usuarios, tambien se implementó un control de permisos estricto basado en roles, asegurando que cada usuario solo pudiera acceder a las funcionalidades autorizadas. Por último, se emplearon validaciones de entrada para prevenir vulnerabilidades comunes como la inyección SQL o los ataques XSS, garantizando asi la integridad de los datos.

Ilustración 14. Vista de edición de usuario en Django Admin.

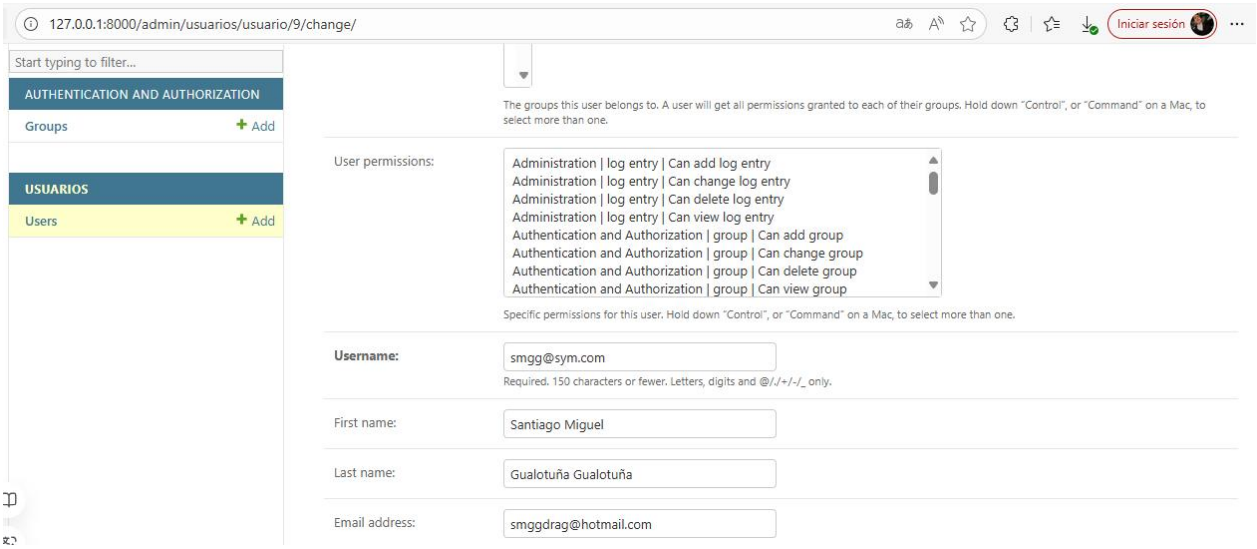


Ilustración 15. Configuración de permisos del usuario.

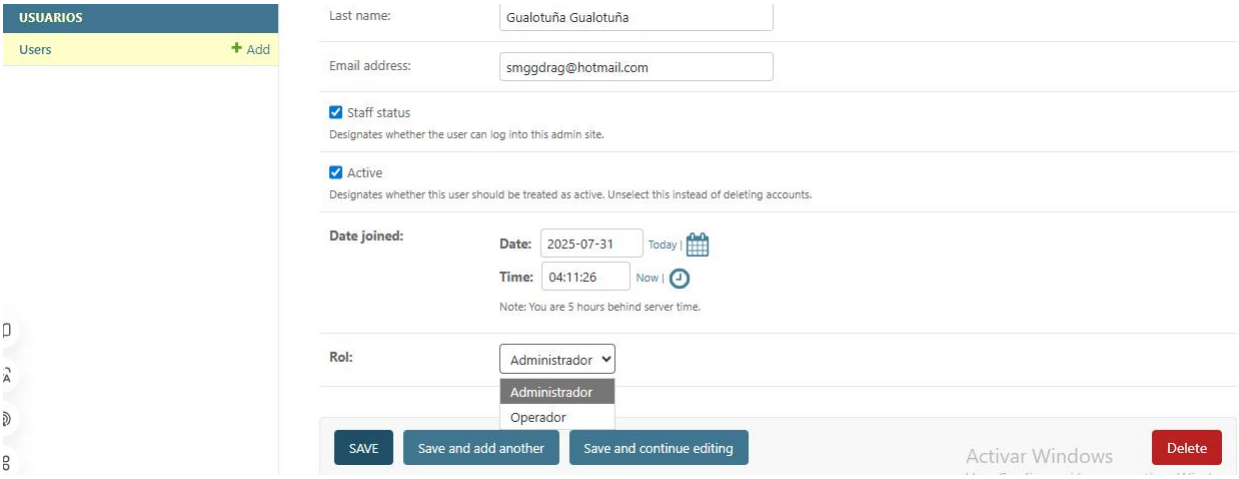


Ilustración 16. Asignación de rol y estado del usuario.

Las imágenes 15 y 16 nos muestran el panel de administración de Django que fue destinado a la gestión de usuarios, donde el administrador puede editar credenciales, asignar permisos específicos y definir roles como administrador u operador, lo que facilita un control seguro y estructurado de los accesos al sistema.

4.8. Despliegue y Capacitación

El sistema se implementó en un entorno de producción local en las instalaciones de Grupo S&M. Esto incluyó la instalación de dependencias, la configuración de la base de datos y el lanzamiento de la aplicación. Se implementó un plan de capacitación en forma de talleres y la entrega de BPMN para ayudarles a entender el flujo del sistema, permitiendo a todo el personal navegar a través de las nuevas herramientas mientras aprenden a usarlas correctamente.

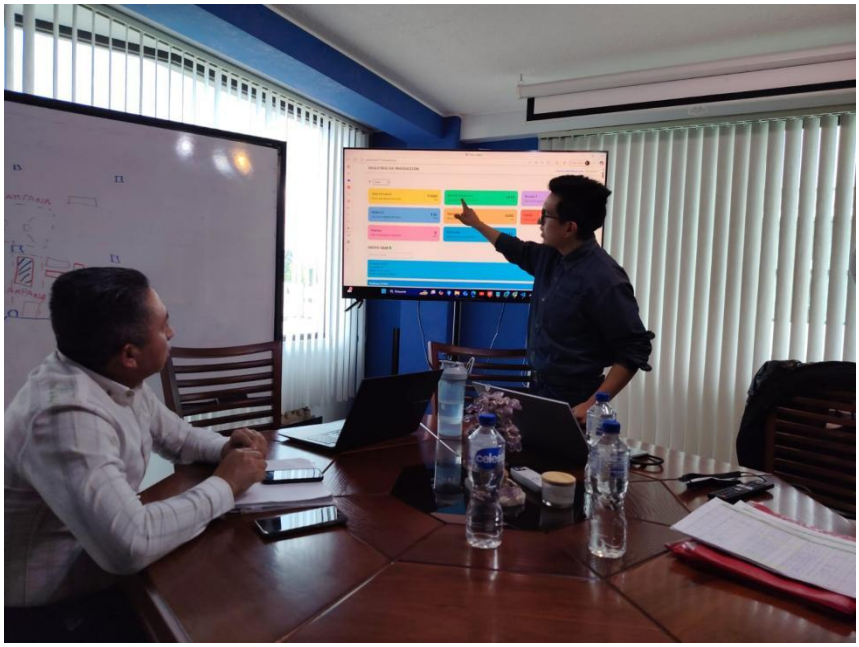


Ilustración 17. Presentación del sistema a la gerencia.

En esta imagen se observa la explicación realizada a la gerencia del Grupo S&M, donde se expone de manera general el funcionamiento del sistema y los beneficios que aporta en la gestión de la producción. La capacitación consistio en mostrar cómo el sistema contribuye al control estratégico de la información y en mostrar la importancia de su implementación dentro de la planta de produccion para la toma de decisiones importantes.

Ilustración 18. Capacitación al personal de producción.

La ilustración visual numero 18 muestra que la capacitacion enfocada en el personal de producción operativa sobre los procesos fundamentales del sistema se está reportando de manera efectiva. El propósito era exponer a los trabajadores a las herramientas de registro y consulta y asegurar el uso adecuado de las herramientas de registro a diario, así como verificar que los datos ingresados reflejaran las condiciones operativas reales del proceso.

4.9. Conclusiones del Capítulo

El SCPP fue diseñado y desarrollado con éxito para que el control, que tiene información escalable y diseño seguro, pueda personalizarse según los requisitos de la organización y obetner mas modulos segun los requerimientos. El diseño cliente-servidor, la interfaz intuitiva y la capacidad de validar datos en tiempo real son las características notables de este capitulo que constituyen el enfoque teórico de la transformación digital de la gestión de producción en Grupo S&M.

CAPÍTULO V: PRUEBAS Y VALIDACIÓN

El proceso de prueba es una de las etapas más cruciales del ciclo de vida del desarrollo de software, en el caso del Sistema de Control y Planificación de Producción (SCPP), la validación se concibió como un proceso sistemático y continuo que acompañó el desarrollo en cada sprint. El objetivo no era solo asegurar la ausencia de errores, sino también verificar que la herramienta cumpliera con los requisitos funcionales, técnicos y de usabilidad establecidos desde el inicio del proyecto. Se realizaron pruebas en diferentes niveles: pruebas unitarias, de integración, del sistema y de aceptación por parte del usuario (UAT), cada una con un enfoque específico que permitía cubrir tanto la robustez técnica como la experiencia de los trabajadores que interactuaran con la solución de este sistema prototipo a implementar.

Ilustración 19. Prueba piloto con el personal de producción.

La imagen 19 n muestra la validación práctica del sistema con las encargadas de cada módulo de producción, fue especificamente este personal quienes realizaron registros de prueba para comprobar la facilidad de uso y el correcto levantamiento de información. Esta fase piloto

permitió a los desarrolladores identificar mejoras y garantizar que el personal se familiarice con el proceso de ingreso de datos, asegurando así la fiabilidad del ingreso de datos al sistema en su aplicación real.

5.1. Estrategia de Pruebas

La estrategia de prueba fue desarrollada sobre la base de la metodología Scrum y los estándares de Aseguramiento de Calidad (QA). Para abordar esto, se establecieron cuatro objetivos principales:

(1) Asegurar la correcta implementación de las reglas de negocio en cada módulo del sistema;

(2) Asegurar la consistencia de los datos almacenados en la base de datos;

(3) Evaluar la comunicación entre la interfaz desarrollada en React y el Back-End implementado en Django REST Framework;

(4) Asegurar que haya aceptación y facilidad de uso por parte del personal de la empresa.

Todos los objetivos se implementaron como una variedad de pruebas, que luego fueron documentadas, ejecutadas y evaluadas en un entorno controlado para garantizar que la solución final fuera confiable y eficiente.

5.2. Pruebas de Usabilidad: Ingreso, Registro y Generación de Reporte

Luego se realizaron pruebas de usabilidad para asegurar que los módulos de entrada de datos, registro, consulta y generación de informes fueran intuitivos y eficientes, estas pruebas se diseñaron en torno a la interfaz web como una utilidad, asegurando que las tareas clave pudieran completarse rapida y fácilmente. En el módulo de entrada, evaluaron la facilidad de registrar nuevos pedidos y asignar tareas a los empleados, confirmando que los formularios fueran claros y

las opciones de selección fueran lógicas.

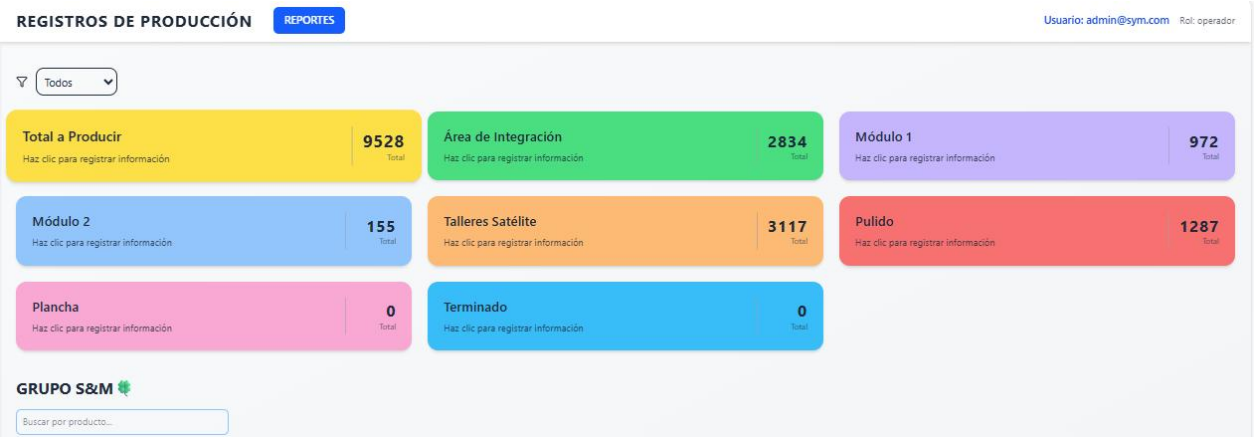


Ilustración 20. Dashboard inicial del sistema de producción.

En la imagen numero 20, es el panel principal del sistema donde se muestran dinámicamente los diferentes módulos de producción, en este panel de control, los usuarios pueden ingresar a cada módulo y registrar información en consecuencia, permitiendo la organización de datos y la generación de informes a través de reportes generados auto máticamente. En esta consulta, se verificó que el panel de control mostraba la información claramente y que los filtros de búsqueda funcionaban correctamente; por lo tanto, los usuarios pudieron encontrar rápidamente lo que buscaban. Por último, se probó la funcionalidad de exportar datos al formato Excel para la generación de informes, asegurándose de que los archivos generados sean precisos y estén bien formateados para un análisis posterior. Los resultados de estas pruebas destacaron que la interfaz de SCPP es fácil de usar y que los flujos de trabajo son comprensibles para el personal, facilitando una rápida adopción dentro del ambiente de producción.

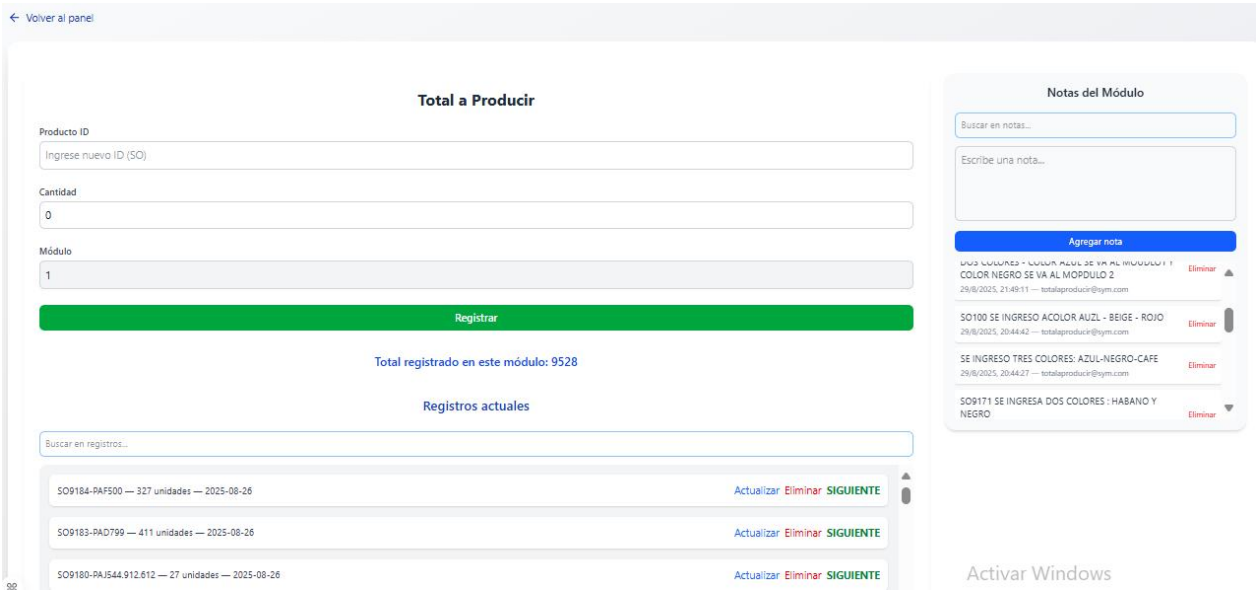


Ilustración 21. Formulario de registro en el módulo "Total a Producir".

En la imagen 21, se puede acceder al formulario para el módulo de Total a Producir y el usuario puede ingresar el identificador del producto, la cantidad y el módulo asignado, tambien se muestran las funcionalidades que son complementarias, como filtros de búsqueda, registro de notas, actualización, eliminación y la opción siguiente para que las cantidades terminadas sean controladas en términos de seguimiento de producción detallado y facil de visualizar.

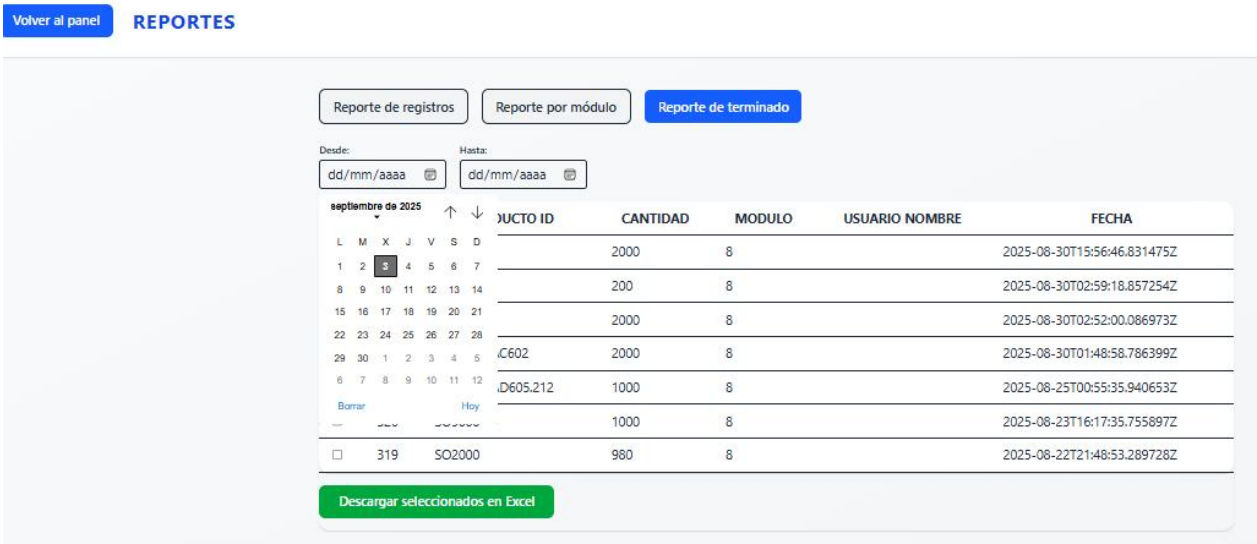


Ilustración 22. Sección de reportes del sistema.

La sección de informes en la Imagen 22 contiene 3 categorías: informes de registro general, informes de módulos e informes de productos terminados, aqui tambien incluimos un filtro de fecha y una opción para exportar su información en formato Excel, simplificando el proceso de análisis y gestión de datos de producción de manera rápida y organizada.

5.3. Pruebas de Usabilidad del Sistema Web Local

La evaluación de funcionalidad y rendimiento en el entorno operativo de Grupo S&M fue el área de enfoque de las pruebas del sistema web local, aqui se evaluo la velocidad de carga de la página, así como la capacidad de respuesta de algunos módulos interactivos y el rendimiento de la aplicación al utilizar el equipo de la empresa, tambien se verificó que el acceso a la base de datos local SQLite fue eficiente y que la comunicación entre el back-end de Django y la API DRF estass funcionaron sin problemas. Estas pruebas se realizaron bajo condiciones de red simuladas

para asegurar que el sistema continuara funcionando incluso después de probar la conectividad. Los resultados mostraron que el SCPP funciona de manera estable y rápida en el entorno de producción. La experiencia del usuario es consistente e ininterrumpida, lo cual es crucial para la continuidad .

Ilustración 23. Visualización de resultados en pantallas de producción.

La etapa de usabilidad del sistema se presenta en la imagen 23, ya que muestra los resultados proyectados en las pantallas instaladas en la planta, en esta etapa el personal también puede acercarse para verificar el progreso de cada módulo en tiempo real, y el gerente de producción puede tomar una decisión más informada basada en datos actualizados y precisos.

5.4. Acceso Usuario Administrador

El rol de Usuario Administrador fue desarrollado con derechos de gestión completos, permitiendo un control total del sistema, al momento de hacer las pruebas para este perfil se centraron en interacciones de alto nivel, como la creación y gestión de cuentas de usuario, la administración de roles y permisos, y la configuraciòn de ajustes de sistema de alto nivel. Se confirmó que el administrador puede crear nuevos usuarios, asignarles roles específicos (administrador, operario.), y editar sus permisos de acceso segun las necesidades organizacionales, tambien verificamos la capacidad de monitorear la actividad de los usuarios y acceder a todos los informes. Se tuvo que comprobar que el perfil tiene cobertura y control total sobre su uso para que el SCPP funcione correctamente para los usuarios y el sistema.

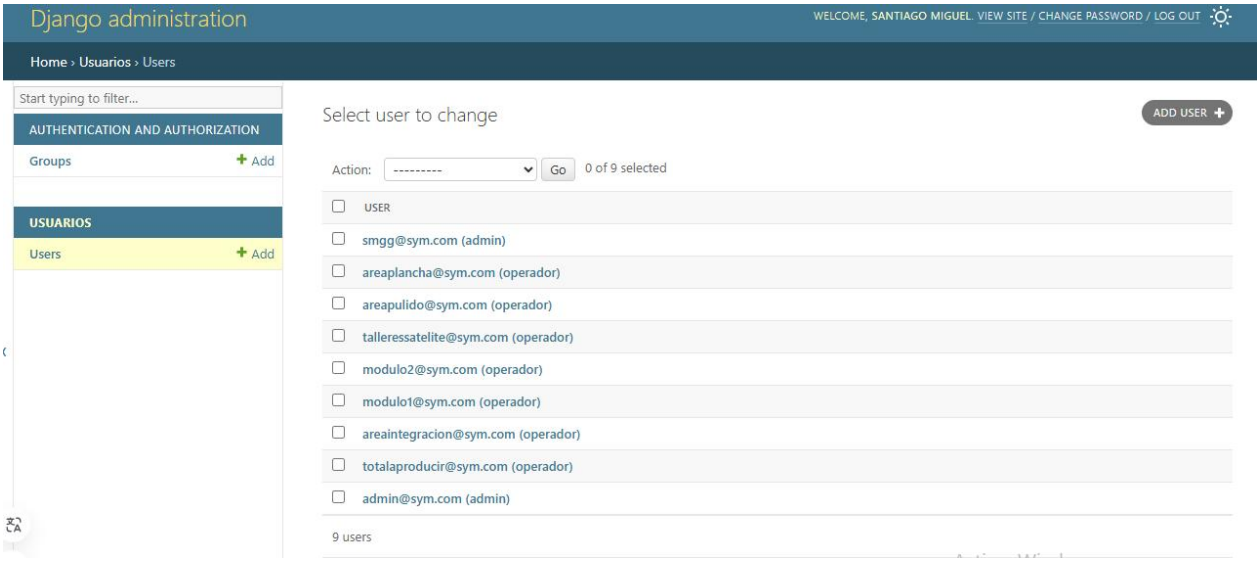


Ilustración 24. Listado de usuarios registrados en el sistema.

La imagen 24 nos da a mostrar el panel de administración de Django en donde se gestionan las cuentas de los usuarios. Desde este entorno se pueden crear perfiles, asignar permisos específicos y mantener un registro organizado de administradores y operadores, lo que asegura un control

centralizado y seguro del acceso al sistema.

5.5. Acceso Usuario de cada Módulo de Planta de Producción

El sistema cuenta con roles de usuario específicos para cada módulo de la planta de producción, lo que garantiza que cada colaborador tenga acceso únicamente a las funcionalidades relevantes para su área de trabajo. Las pruebas para estos perfiles se concentraron en verificar que la interfaz y las opciones de cada módulo (por ejemplo, modulo 1, modulo 2, talleres satelite) estuvieran diseñadas para optimizar el registro de datos de producción de manera eficiente y sin errores. Se validó que el acceso a cada módulo estuviera restringido a los usuarios con los permisos correspondientes y que la información ingresada por un módulo se reflejara correctamente en el dashboard de control general, asegurando así la integridad y la trazabilidad de los datos en toda la cadena de producción. Este enfoque especifico en los permisos minimiza la curva de aprendizaje y el riesgo de errores operativos.

Ilustración 25.Interfaz de inicio de sesión del sistema.

La imagen 25 muestra la pantalla de acceso al sistema, donde se muestran los usuarios previamente registrados con sus credenciales guardadas, en esta interfaz el sistema determina el rol (por ejemplo, administrador u operador) asociado con cada perfil seleccionado y muestra el nombre de usuario activo, permitiendo a los usuarios tomar las medidas necesarias para controlar el acceso al sistema y acceder a las funciones asociadas con ese perfil.

5.6. Discusión del Problema

De acuerdo con el proceso de prueba y los comentarios del usuario, el SCPP resuelve efectivamente el problema de la gestión de producción manual y empírica, porque se ha centralizado la información y democratizado los datos, eliminando la dependencia de hojas de

cálculo, desarrollando esta disponibilidad de visibilidad del flujo de producción en vivo ha permitido a los supervisores detectar y mitigar cuellos de botella de manera preventiva (en comparación con esperar a que los cuellos de botella lleguen a su puerta y actuar). Además, para permitir que la gestión realice análisis de rendimiento, evalúe la productividad y base decisiones estratégicas en evidencia, se hizo posible el proceso de generación de informes históricos. El proyecto no solo ha mejorado la productividad de la organización, sino que también ha establecido un compromiso con la mejora continua en Grupo S&M, y sus resultados productivos.

CAPITULO VI: Conclusiones y Recomendaciones

6.1 Conclusiones:

- El estudio permitió reconocer que los procesos manuales causaban retrasos, errores y falta de trazabilidad en la producción. Después de realizar el diagnóstico, se necesitaba crear un sistema automatizado para centralizar la información y así minimizar los cuellos de botella y maximizar la eficiencia en la toma de decisiones.
- Abordó con el desarrollo del Sistema de Control de Planificación de la Producción (SCPP), que consolida el registro en un formato digital, en lugar de mantenerlo en un diario de papel. Su arquitectura escalable y versátil permite el registro de información en tiempo real, lo que le otorga más control, y también ofrece una base sólida para el crecimiento o la integración con otras plataformas.
- El sistema desarrollado tiene mas herramientas integradas para la planificación anticipada y el monitoreo continuo de los módulos, también mejoró la programación y permitió el seguimiento del rendimiento de las diversas áreas, contribuyendo a la detección temprana de incidentes y a la utilización eficiente de los recursos.
- La interfaz intuitiva tipo tablero de control ofrecía una vista interactiva y dinámica de la información productiva, presentando cantidades inmediatas registradas y cómo progresaba cada módulo. Este enfoque, centrado en la usabilidad, aseguraba que tanto los usuarios como los gerentes pudieran acceder a los datos de manera fluida y precisa.
- Contiene un módulo de informes que permite descargar información en formato Excel para estudiar la producción en diferentes períodos de tiempo con precisión, la funcionalidad ha sido crítica para medir el logro de objetivos, detectar tendencias y proporcionar datos

robustos y relevantes para tomar decisiones estratégicas.

6.2 Recomendaciones:

- Incorporar otros procesos relacionados con la cadena de producción para tener una visión integral que incluya registros actuales, pero también indicadores de eficiencia, control de calidad y gestión de recursos humanos en áreas específicas de la planta.
- Migrar en futuras versiones a una base de datos más robusta como PostgreSQL o SQL Server, especialmente si el volumen de datos crece significativamente, asegurando así una mayor escalabilidad, seguridad y rendimiento en la gestión de la información.
- Implementar un plan de capacitación continua para los trabajadores, con el fin de asegurar el uso adecuado del sistema y fortalecer la cultura organizacional orientada a la toma de decisiones basada en datos. Esto contribuirá a reducir errores en los registros y a incrementar la confianza en los resultados obtenidos.
- Desarrollar nuevas funcionalidades de análisis predictivo basadas en machine learning, lo que permitiría anticipar retrasos en la producción, proyectar escenarios futuros y optimizar la asignación de recursos, impulsando la innovación tecnológica dentro de la empresa.
- Mantener un esquema de retroalimentación periódica con los usuarios del sistema, fomentando un proceso de mejora continua que garantice que la herramienta se adapte a las necesidades cambiantes de la empresa y evolucione en concordancia con su crecimiento.

6.3 Bibliografia

Pérez Polo, S. P., & Peña Castrillo, S. P. (2024). Intraemprendimiento, una respuesta a la necesidad de las empresas en el sector textil de diversificar sus estrategias de producción para adaptarse rápidamente a las tendencias del consumidor actual (Bachelor's thesis, Escuela de Economía, Administración y Negocios).

Calva, H. C. G., García, J. V., & Herrera, R. A. (2017). Determinantes de la quiebra empresarial en las empresas ecuatorianas en el año 2016. Revista publicando, 4(13 (1)), 108-126.

Boston Consulting Group. (2024). Beyond productivity: Textiles in the era of Industry 4.0.

Calva, A., Bravo, P., & López, D. (2017). Causas de quiebra empresarial en el Ecuador: Un análisis estadístico. Revista Publicando, 4(13), 398–408.

Davenport, T. H., & Harris, J. G. (2007). Competing on Analytics: The New Science of Winning. Harvard Business School Press.

Drucker, P. F. (1985). Innovation and Entrepreneurship: Practice and Principles. Harper & Row.

Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2018). Fundamentals of Business

Process Management (2nd ed.). Springer.

Few, S. (2006). Information Dashboard Design: The Effective Visual Communication of Data. O'Reilly Media.

Firouzi, F., Maroufi, K., & Sarhadi, M. (2023). Smart manufacturing: A systematic literature review. Journal of Manufacturing Systems, 69, 305–327.

Goldratt, E. M., & Cox, J. (2004). The goal: A process of ongoing improvement (3rd ed.). North River Press.

Imai, M. (1986). Kaizen: The Key to Japan's Competitive Success. McGraw-Hill.

Kagermann, H., Wahlster, W., & Helbig, J. (2013). Recommendations for implementing the strategic initiative INDUSTRIE 4.0.

Ko, M., Lee, C., & Cho, Y. (2022). Design and implementation of a cloud-based collaborative MES in the fashion industry. Applied Sciences, 12(18), 9381.

Laudon, K. C., & Laudon, J. P. (2020). Management Information Systems: Managing the Digital Firm (16th ed.). Pearson.

Microsoft. (2024). Visual Studio Code. Recuperado de

https://code.visualstudio.com/

Nakajima, S. (1989). Introduction to TPM: Total productive maintenance. Productivity Press.

Nielsen, J. (1993). Usability Engineering. Morgan Kaufmann.

Orlicky, J. (1975). Material Requirements Planning: The New Way of Life in Production and Inventory Management. McGraw-Hill.

Orteaga, M. (2013). Análisis de los factores que inciden en el quiebre de empresas del sector Industrias Manufactureras del Ecuador al año 2009. [Tesis de Grado, Universidad Nacional de Loja].

Porter, M. E. (1985). Competitive Advantage: Creating and Sustaining Superior Performance. Free Press.

Slack, N., & Lewis, M. (2020). Operations strategy (6th ed.). Pearson.

Vivanco, M. (2017). El control interno y su incidencia en la gestión administrativa de las pequeñas y medianas empresas del sector comercial del cantón Loja. [Tesis de Grado, Universidad Nacional de Loja].

Womack, J. P., & Jones, D. T. (1996). Lean thinking. Simon & Schuster.

Womack, J. P., Jones, D. T., & Roos, D. (1990). The Machine That Changed the World. HarperCollins.

*Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly, 13*(3), 319–340.*

Gerchev, I. (2022). Tailwind CSS. SitePoint Pty Ltd.

Prescott, P. (2015). HTML 5. Babelcube Inc..

Luna, F. (2019). JavaScript-Aprende a programar en el lenguaje de la web. RedUsers.

Khider Ismail, B. (2025). Designing and developing an online platform for learning react and vite for beginners: cases: user-centered data collection: integrating surveys and user stories in interviews.

HOYOS, G. (2024). DOMINANDO REACT: GUÍA INTRODUCTORIA PARA DESARROLLADORES.

Viejo Pomata, D. (2020). Arquitectura de desarrollo web con Django y apps con Flutter. Universitat Oberta de Catalunya (UOC). https://hdl.handle.net/10609/106467

ANEXOS

BPMN DE PROCESOS

Total Producir

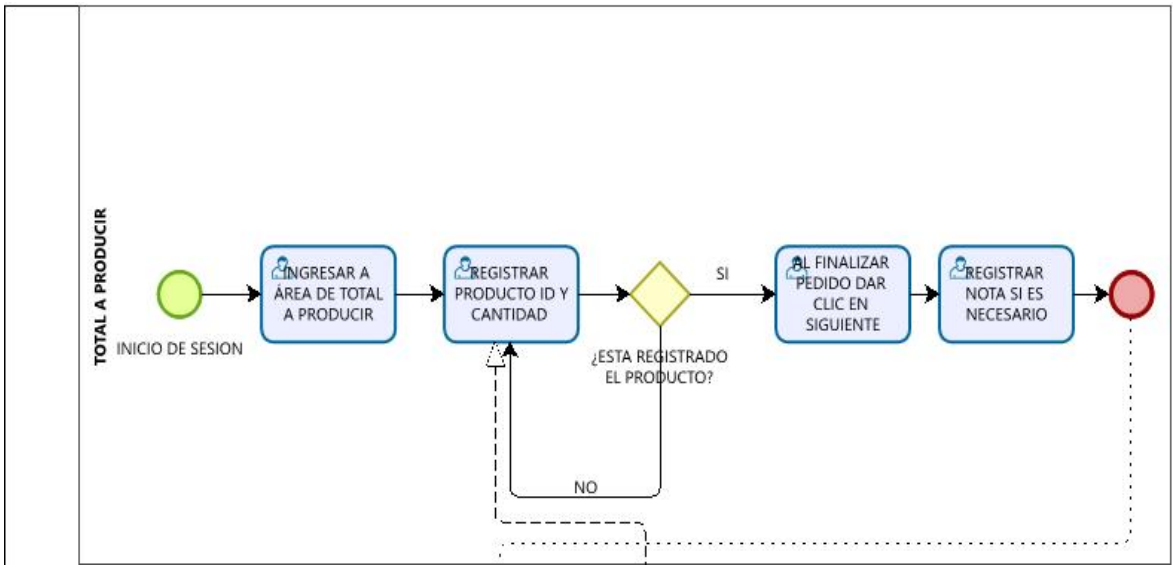


Ilustración 26. Diagrama BPMN del proceso en el módulo Total a Producir.

En la imagen 26 se observa el flujo de actividades del módulo Total a Producir representado mediante un diagrama BPMN, donde se inicia con el ingreso al sistema y la selección del área correspondiente para posteriormente registrar el producto y la cantidad. El diagrama permite visualizar la decisión de validación sobre si el producto está registrado y, en

caso afirmativo, continuar con la finalización del pedido y la opción de añadir notas, lo que facilita comprender de manera clara y ordenada cómo funciona este proceso dentro del sistema.

Área de Integración

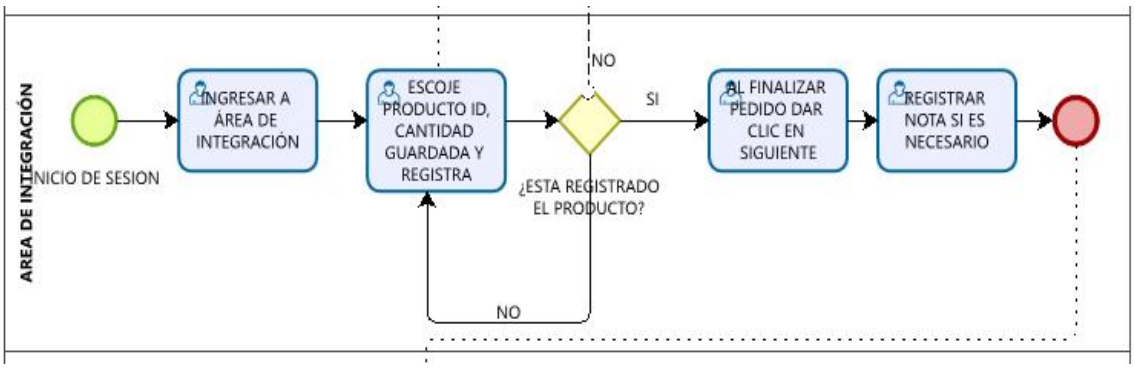


Ilustración 27. Diagrama BPMN del área de integración.

La imagen muestra el flujo del área de integración donde el usuario ingresa al módulo, selecciona el producto con su cantidad y lo registra. El diagrama evidencia la validación del producto y la posibilidad de finalizar el pedido o añadir notas en caso necesario, representando de manera clara la secuencia de este proceso.

Módulo 1

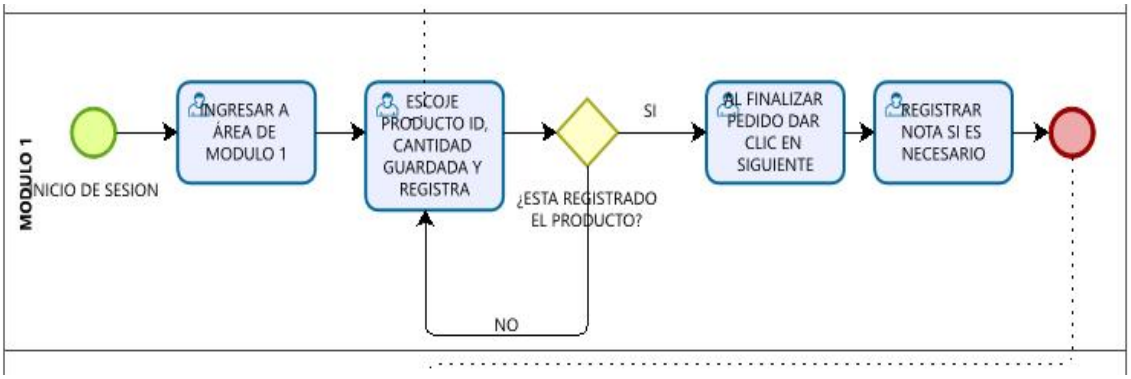


Ilustración 28. Diagrama BPMN del Módulo 1.

Módulo 2

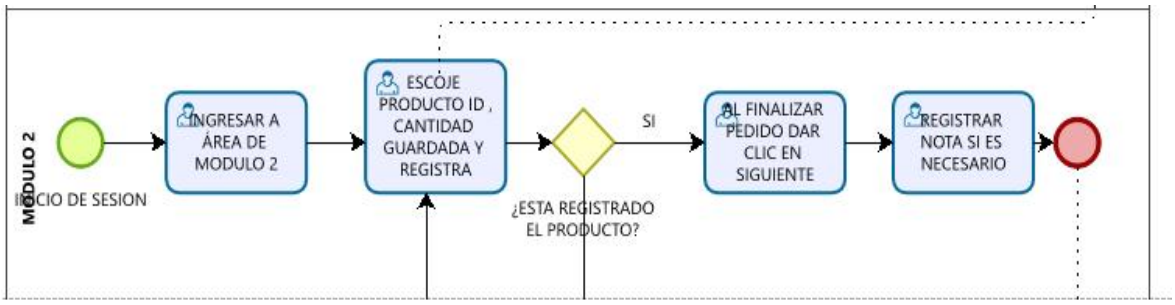


Ilustración 29. Diagrama BPMN del Módulo 2.

Talleres Satelite

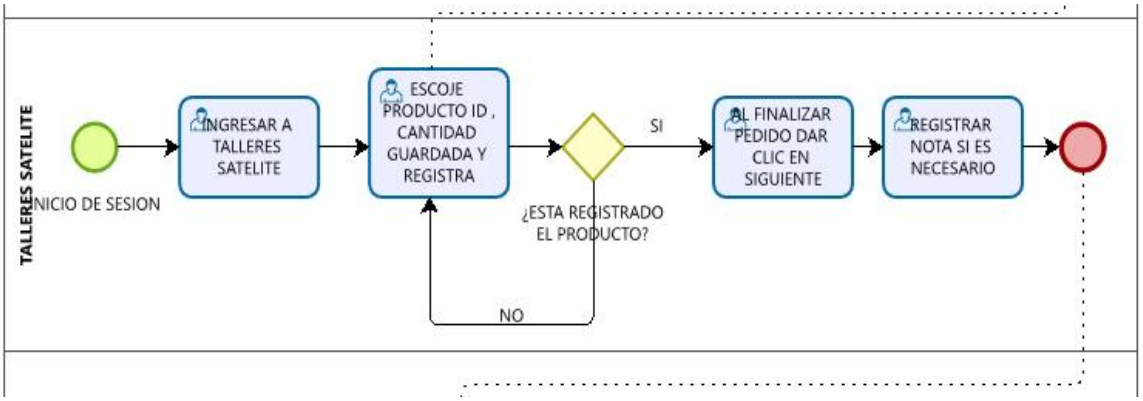


Ilustración 30. Diagrama BPMN de Talleres Satélite.

Las imágenes 28 , 29 ,30 representan el flujo en el que el usuario accede al área correspondiente, selecciona el producto con su cantidad y lo registra. El proceso incluye la validación del producto y concluye con la opción de continuar el pedido o añadir notas si es necesario, mostrando un esquema uniforme en estos módulos del sistema.

Pulido

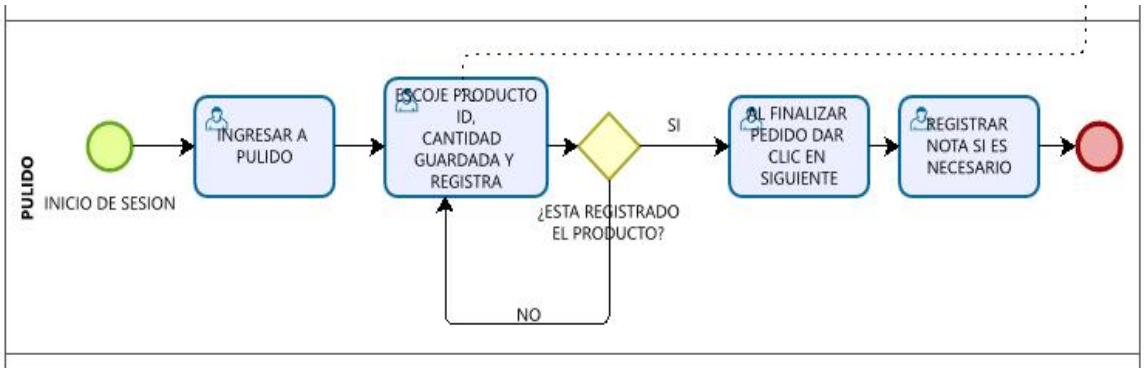


Ilustración 31. Diagrama BPMN del módulo Pulido.

La imagen 31 representa el flujo del módulo Pulido donde el usuario accede al área, selecciona el producto con su cantidad y lo registra. El proceso contempla la validación del producto y concluye con la opción de continuar el pedido o añadir notas si es necesario, manteniendo coherencia con la lógica aplicada en los demás módulos del sistema.

Plancha

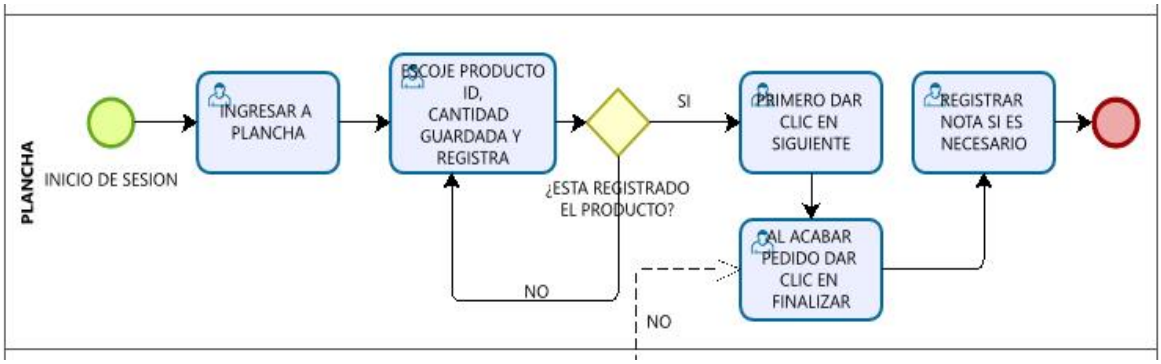


Ilustración 32. Diagrama BPMN del módulo Plancha.

La imagen 32 muestra el flujo del módulo Plancha en el que el usuario ingresa al área, selecciona el producto con su cantidad y lo registra. El proceso contempla la validación del

producto y añade la opción de avanzar con siguiente o finalizar el pedido, además de permitir registrar notas si es necesario, garantizando un control detallado de esta etapa de producción.

Terminado

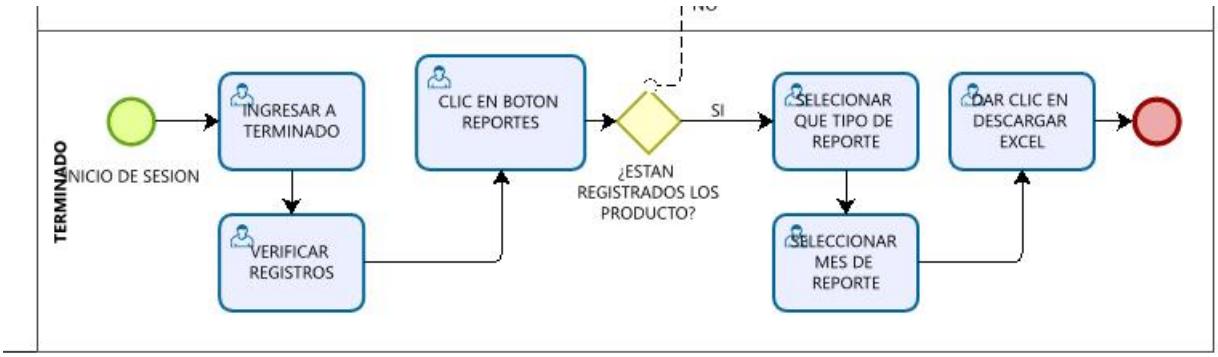


Ilustración 33. Diagrama BPMN del módulo Terminado.

La imagen 33 representa el flujo del módulo Terminado donde el usuario ingresa al área, verifica los registros y accede a la sección de reportes. El proceso incluye la validación de productos y la selección del tipo y mes de reporte, finalizando con la descarga en Excel, lo que permite consolidar y analizar la información productiva de manera eficiente.

MANUAL DE USUARIO

Login de Usuario

Iniciar Sesión por Modulo de planta

Para ingresar al sistema, cada usuario debe contar con un nombre de usuario y una contraseña que previamente le serán proporcionados por el administrador encargado. Este es el único responsable de crear las cuentas y asignarlas a los encargados de cada módulo de producción, siguiendo las normativas internas de la empresa.

Ilustración 34. Interfaz de inico de sesiòn con logos de las empresas que conforman el grupo S&M.

En la pantalla de inicio de sesión, el usuario deberá ingresar correctamente sus credenciales en los campos correspondientes de USUARIO y CONTRASEÑA, y posteriormente hacer clic en el botón INICIAR SESIÓN. Es importante escribir los datos con precisión, ya que cualquier error impedirá el acceso al sistema.

Usuarios creados

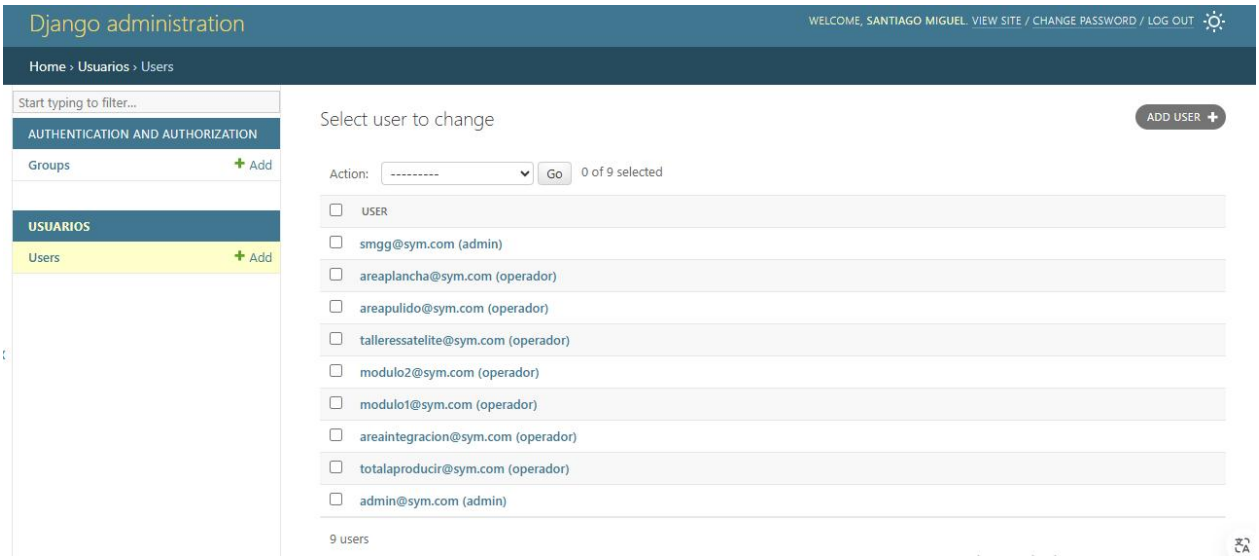


Ilustración 35. Panel de administración de usuarios.

El administrador cuenta con una interfaz de control en Django desde la cual puede crear nuevos usuarios, asignar permisos, modificar roles o eliminar cuentas existentes. Para acceder a este panel es necesario ingresar con las credenciales de superusuario, las cuales son creadas inicialmente por el desarrollador del sistema.

Desde este entorno, el administrador tiene la facultad de asignar los roles de administrador o operador, asegurando que cada usuario disponga únicamente de los accesos y funciones que le corresponden dentro de su módulo de trabajo.

Perfil de usuario

Ilustración 36. Identificación de usuario en el sistema

En la parte superior izquierda de la pantalla el sistema muestra el correo del usuario activo y el rol asignado (administrador u operador). Esta función permite verificar en todo momento con qué tipo de perfil se ha iniciado sesión, lo cual es importante para confirmar los permisos disponibles y las acciones que cada usuario puede realizar dentro del sistema.

LogOut de Usuario

Ilustración 37. Boton para cerrar sesiòn.

El sistema incorpora un botón de cerrar sesión ubicado en la parte inferior izquierda de la pantalla, que permite finalizar de forma segura la sesión activa. Esta opción es fundamental para proteger la información y evitar accesos no autorizados, además mantiene la simplicidad y originalidad del sistema al utilizar un diseño común y de fácil identificación para todos los usuarios.

Pantalla inicial

1. Si el usuario ingresa al sistema, también verá una pantalla principal o tablero en el que se presentan todos los módulos de producción en varios colores para facilitar su identificación. Cada módulo indica la cantidad total acumulada, que denota el número de prendas pendientes de producción en esa etapa. En esta interfaz, el usuario puede seleccionar cualquier módulo para ingresar, así como para registrar información particular dependiendo de su rol y responsabilidades.

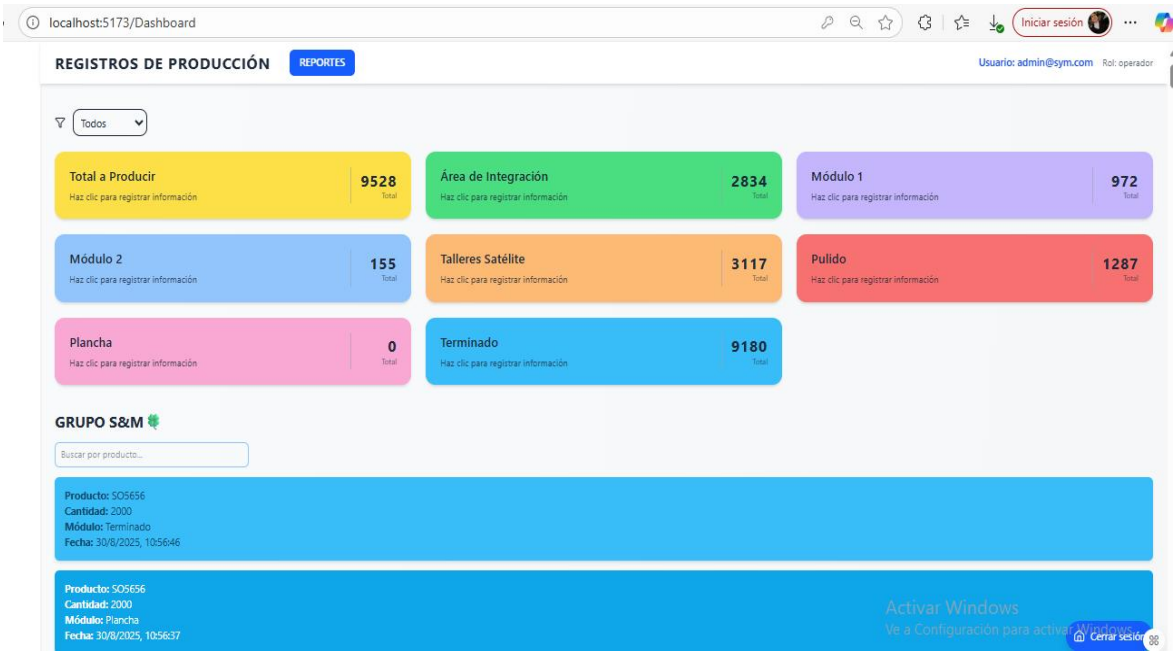


Ilustración 38. Pantalla inicial – Dashboard de producción.

2. En la parte inferior de la pantalla inicial se encuentra la sección de historial de registros, donde se listan las actividades realizadas en todos los módulos del sistema. El usuario puede

realizar búsquedas por ID del producto, fecha o cantidad, lo que permite filtrar información y consultar de manera rápida los datos registrados. Esta funcionalidad es clave para verificar los tiempos de producción y analizar los intervalos de trabajo, facilitando la toma de decisiones y el control del flujo productivo.

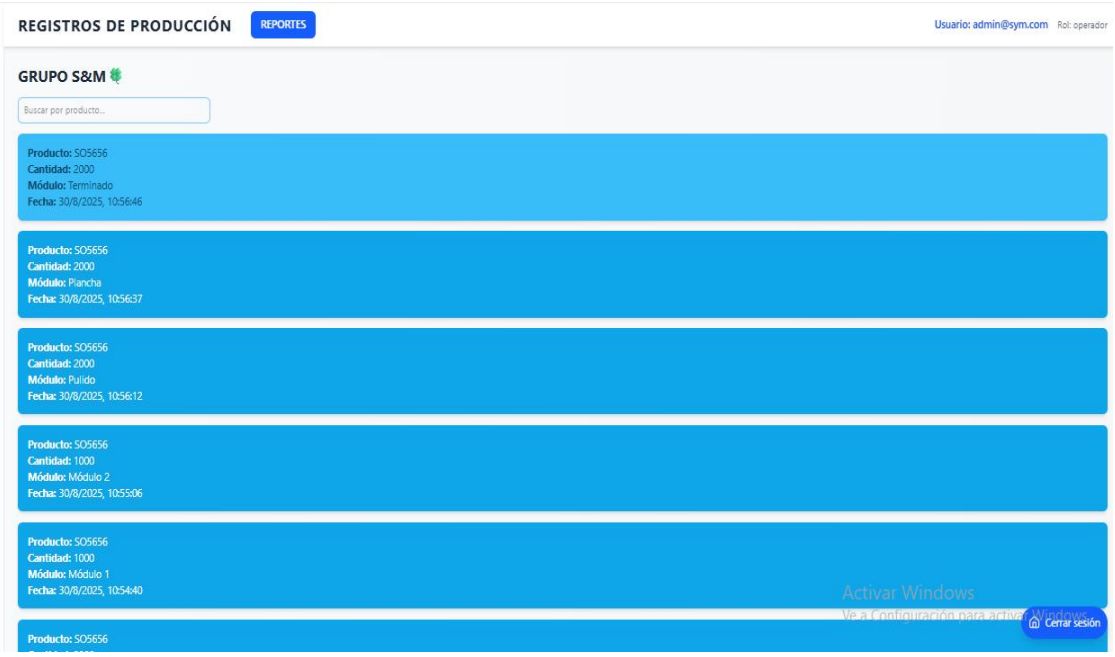


Ilustración 39. Historial de registros.

3. Cuando el usuario ingresa a una tarjeta del dashboard inicial, accede a la interfaz de registro de producción en forma de rectangulos, donde debe ingresar el ID del producto y la cantidad a procesar. El sistema calcula automáticamente la suma acumulada del módulo en tiempo real, mostrando el total registrado hasta el momento.

Adicionalmente, se cuenta con un buscador de registros por ID o cantidad, lo que facilita la localización rápida de información pendiente o procesada.

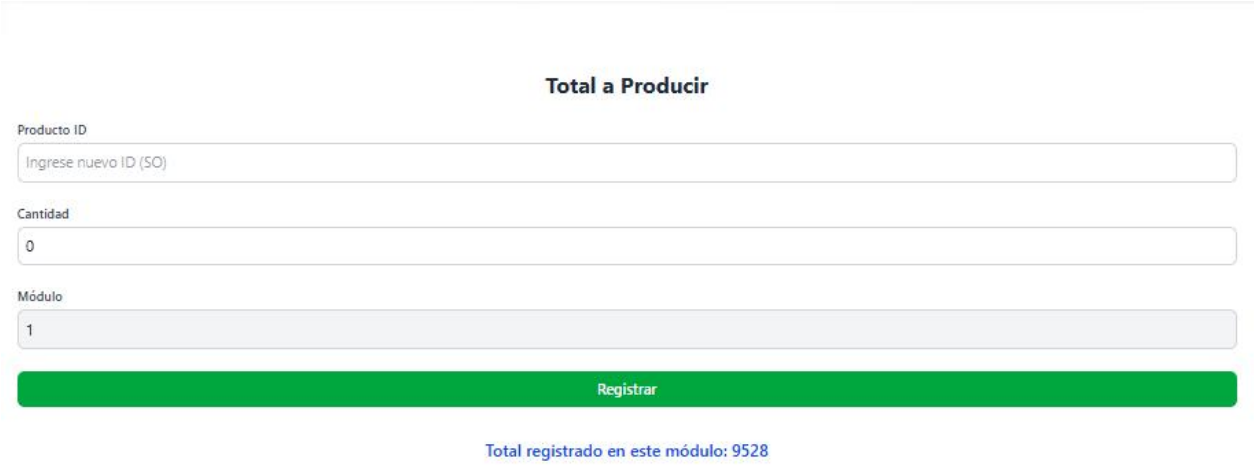


Ilustración 40. Registro de datos en los módulos.

En la parte inferior aparece la lista de registros actuales, cada uno con opciones de Actualizar, Eliminar o Siguiente.

Ilustración 41. Registros guardados de cada modulo.

- La opción Actualizar permite modificar los datos registrados.
- La opción Eliminar borra el registro seleccionado.
- La opción Siguiente descuenta la cantidad del total y guarda el registro como finalizado, apareciendo en la lista con letras opacas para indicar su cierre.

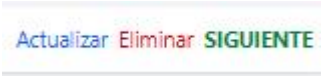


Ilustración 42. Botones con la funcionalidad de actualizar, Eliminar , Siguiente.

En la parte superior derecha se encuentra el apartado de Notas del Módulo, donde los usuarios pueden registrar observaciones relacionadas con la producción. Cada nota queda identificada con el nombre del usuario, fecha y hora. También existe un filtro que permite buscar notas mediante palabras clave, mostrando todas aquellas que compartan el término buscado.

Ilustración 43. Apartado para ingresar notas necesarias de cada pedido.

4. El módulo de Terminado muestra un tablero con tres indicadores principales: Total a Producir, Total Finalizado y Restante. Estos valores permiten al usuario tener una visión clara del estado general de la producción y del progreso realizado en relación con los pedidos. En la parte inferior, hay una lista de todos los registros completados que proporciona el ID, producto, cantidad y fecha de cierre. Esta es información esencial para llevar un seguimiento de los pedidos completados y verificar el logro de objetivos al final de cada operación. También ofrece la capacidad de eliminar registros para facilitar la gestión y limpieza de la información según sea necesario

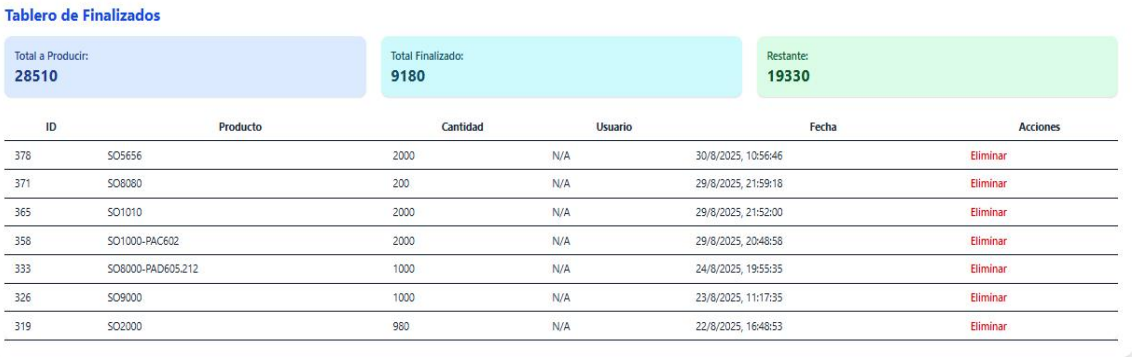


Ilustración 44. Módulo de Terminado.

5. El sistema viene con un módulo de informes que permite consultar la información de producción desde tres puntos de vista diferentes: registros generales, datos filtrados por módulo e informes de productos terminados. Ca da vista incluye filtros basados en el rango de fechas y contiene la información de (ID, producto, cantidad, usuario y fecha). Ademas, los informes se pueden descargar en formato Excel, lo que facilita el análisis de la eficiencia y el seguimiento de los resultados productivos.

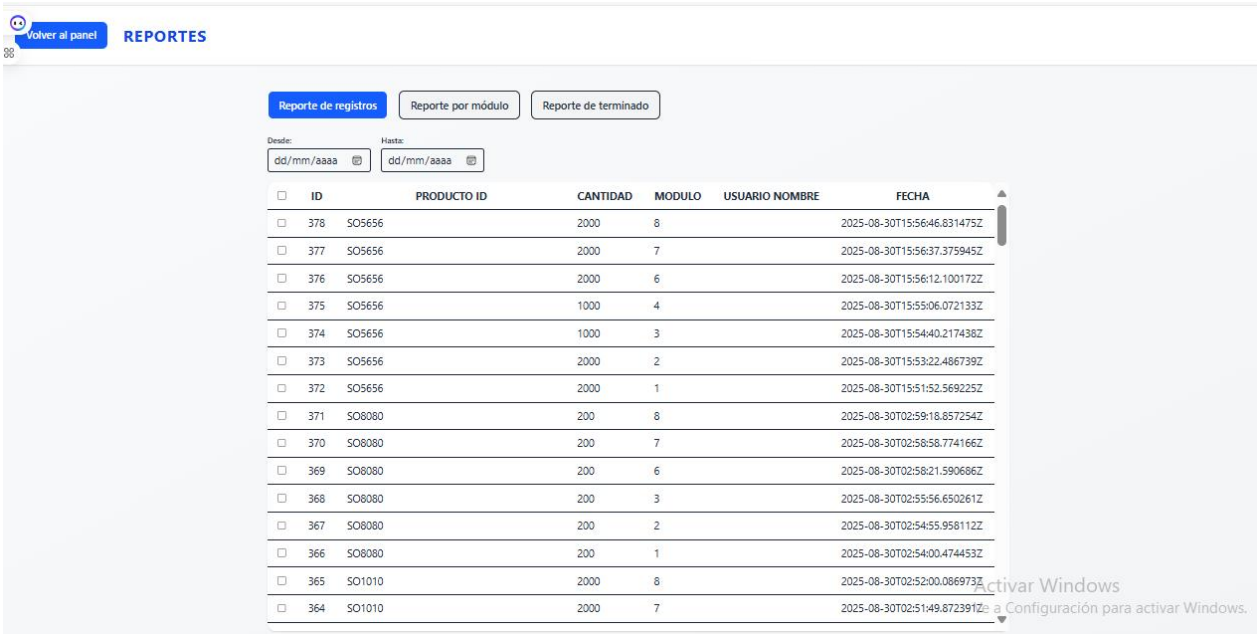


Ilustración 45.Reporte de registros.

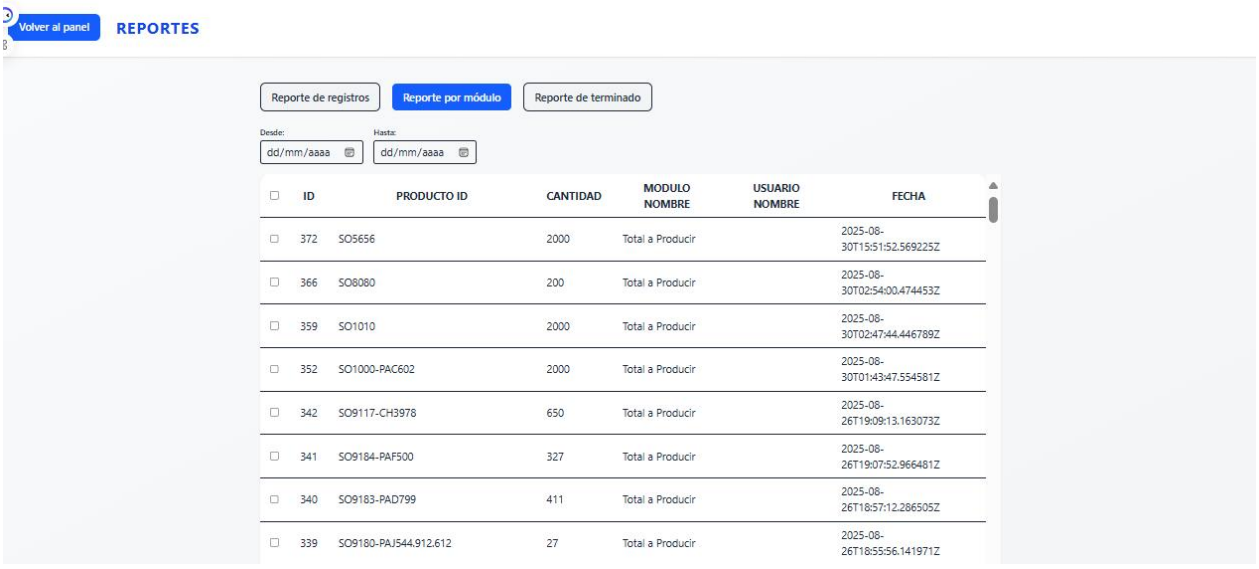


Ilustración 46. Reporte por módulo.

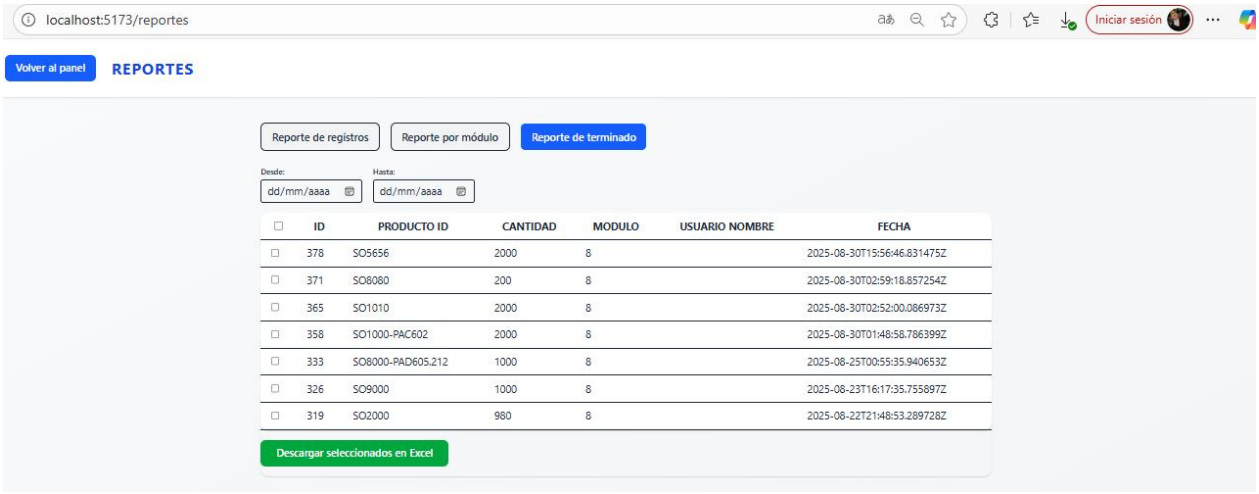


Ilustración 47.Reporte de terminado.

MANUAL DEL PROGRAMADOR

Tecnologías utilizadas

Backend

- Django 5.2.3 (backend/registro/registro/settings.py)

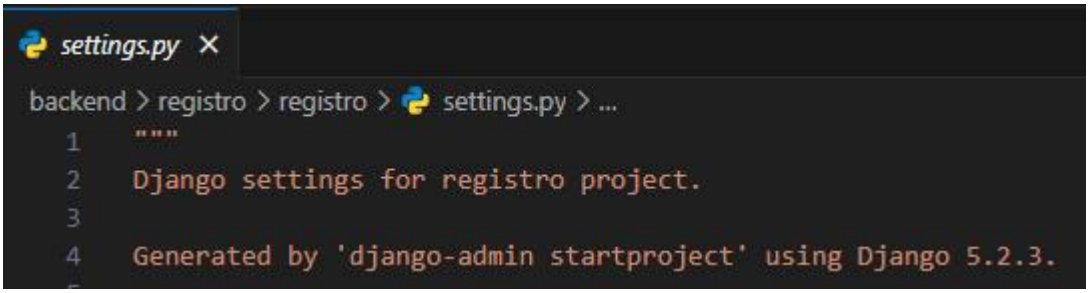


Ilustración 48. Versión Django.

- Django REST Framework

Ilustración 49. Configuración de framework en el proyecto Django.

- SimpleJWT

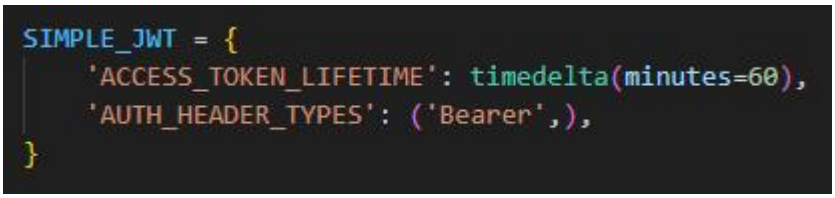


Ilustración 50. Configuraciòn de Simple_jwt para acceso a tokens.

- CORS Headers (mismo archivo, CORS_ALLOW_ALL_ORIGINS y/o CORS_ALLOWED_ORIGINS)

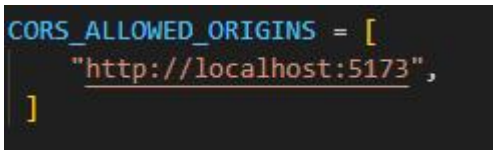


Ilustración 51. Ruta del local Host para ingresar al front end.

- Base de datos: SQLite por defecto (ver DATABASES en settings)

Ilustración 52. Configuraciones para usar la base de datos.

Frontend

- React 18 + Vite (ver frontend/crud-regristro/package.json)
- Axios, Framer Motion, Lucide React, Tailwind (mismo package.json)

Ilustración 53. Configuración de dependencias – archivo package.json

En la imagen se muestra el archivo package.json correspondiente al frontend del sistema, desarrollado con Vite y React. Este archivo define las dependencias principales necesarias para ejecutar el proyecto, entre ellas:

axios, para la comunicación con la API.

tailwindcss y @tailwindcss/vite, que permiten gestionar los estilos de forma eficiente.

framer-motion y lucide-react, utilizadas para animaciones e iconografía.

xlsx, que habilita la exportación de datos a Excel.

Además, el archivo contiene los scripts básicos para levantar el entorno en desarrollo (vite), generar la build de producción y ejecutar análisis de estilo con eslint. Este componente es fundamental para garantizar la correcta instalación y funcionamiento del entorno frontend.

Herramientas

- Node.js, npm
- Git
- Postman/Insomnia para probar el API (opcional)

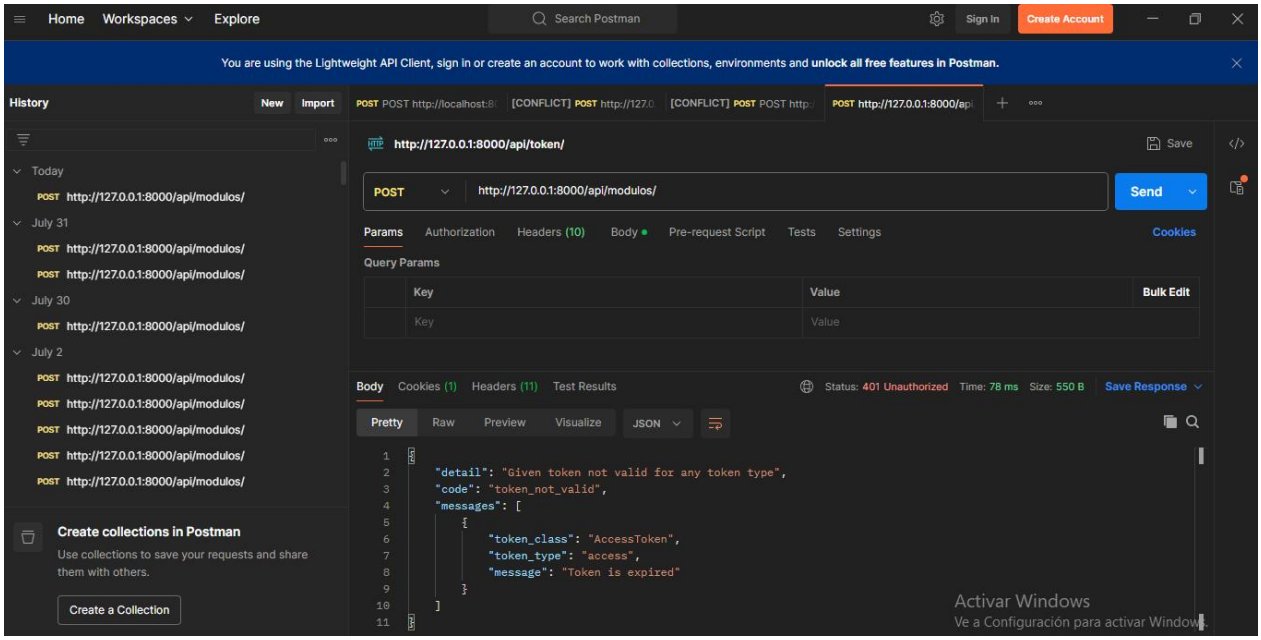


Ilustración 54. Pruebas en Postman.

Dependencias:

Las devDependencies incluyen librerías necesarias para el desarrollo y mantenimiento del proyecto. Entre ellas destacan eslint y sus plugins, que garantizan buenas prácticas de programación en React, @types/react y @types/react-dom para mejorar compatibilidad en entornos con TypeScript, y @vitejs/plugin-react-swc para optimizar la compilación. Asimismo, se utiliza vite como servidor y herramienta principal de construcción. Estas dependencias no se ejecutan en producción, pero son fundamentales para asegurar un desarrollo ágil y con calidad.

Ilustración 55.Dependencias de desarrollo – archivo package.json

Requisitos del sistema

Hardware necesario

- PC con Windows 10/11,
- 8GB RAM mínimo,
- 10GB libres.

Software necesario

- Python 3.13.4
- Node.js v22.16.0
- Git version 2.50.1.windows.1
- (Opcional) Postman
- VS Code

Instalación de software requerido

Instalación de Node.js

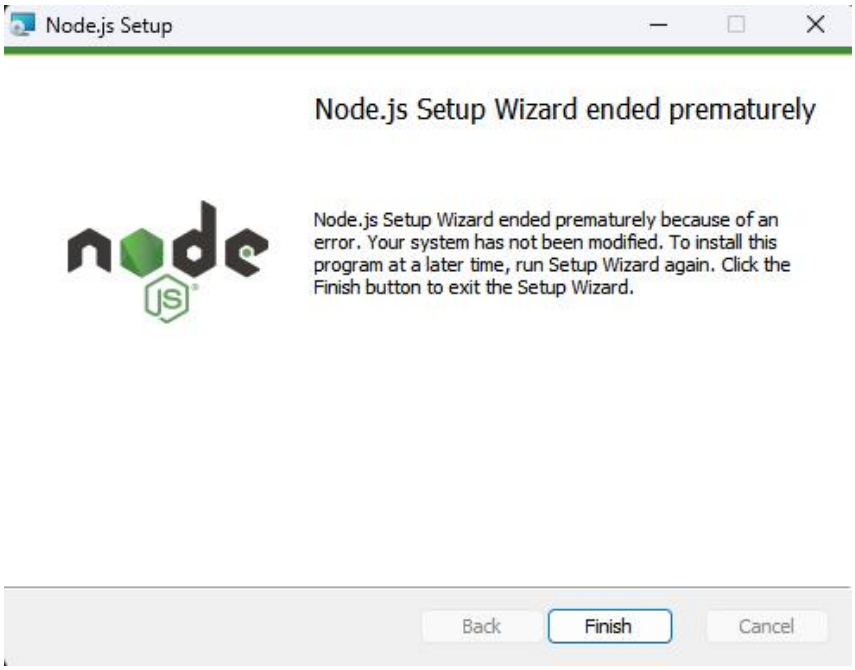


Ilustración 56. Ejecuciòn de Node.js Setup .

Instalación de Python

Ilustración 57.Ejecución de Python 3.11.4 Setup.

Instalación de VS Code:

Ilustración 58. Ejecución de Instalador de VS Code.

Abrir el proyecto con VS Code

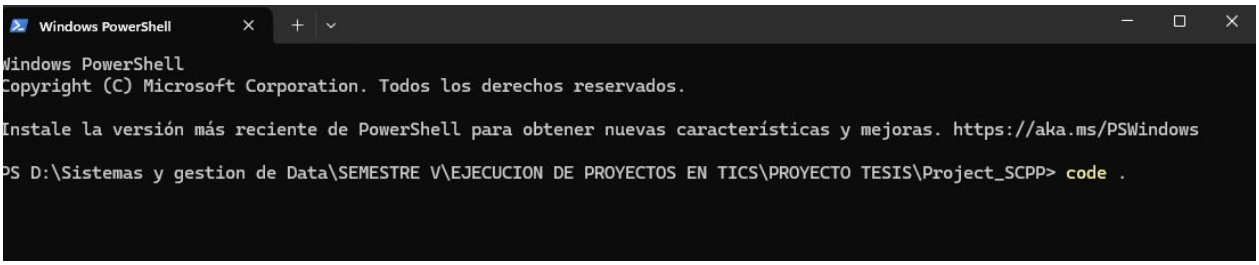


Ilustración 59. Ingreso a proyecto VS Code mediante Consola.

Archivo de configuración

El archivo settings.py define la configuración principal del proyecto Django.

SECRET_KEY: clave de seguridad, debe mantenerse oculta.

DEBUG: en desarrollo se usa True, en producción debe estar en False.

ALLOWED_HOSTS: lista de dominios autorizados para acceder al sistema.

Por defecto el sistema usa SQLite y CORS habilitado para localhost. Como mejora futura se recomienda el uso de un archivo .env para mayor seguridad en producción.

```
# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/5.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-i(4qx3=z+y&7#+j11l-f&sg-otnj)5p!m#oce(!ym1#vr+ga0i'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []
```

Ilustración 60. Archivo de configuración (settings.py)

Agregar variables de entorno

Instalación de dependencias (backend)

- cd Project_SCPP/backend/registro
- python -m venv env
- env\Scripts\activate
- pip install django djangorestframework
- djangorestframework-simplejwt
- django-cors-headers

```
cd Project_SCPP/backend/registro
python -m venv env
env\Scripts\activate
pip install django djangorestframework djangorestframework-simplejwt django-cors-headers
```

Ilustración 61. Instalacion de dependencias dentro del proyecto Django.

Creación de la base de datos

Migraciones y superusuario

- python manage.py makemigrations usuarios modulos
- python manage.py migrate
- python manage.py createsuperuser

```
python manage.py makemigrations usuarios modulos
python manage.py migrate
python manage.py createsuperuser
```

Ilustración 62. Comando para migrar los modelos de la base de datos y comando para crear un Super usuario.

Ejecutar backend

- python manage.py runserver

```
PS D:\Sistemas y gestion de Data\SEMESTRE V\EJECUCION DE PROYECTOS EN TICS\PROYECTO TESIS\Project_SCPP> cd .\backend\
PS D:\Sistemas y gestion de Data\SEMESTRE V\EJECUCION DE PROYECTOS EN TICS\PROYECTO TESIS\Project_SCPP\backend> cd .\registro\
PS D:\Sistemas y gestion de Data\SEMESTRE V\EJECUCION DE PROYECTOS EN TICS\PROYECTO TESIS\Project_SCPP\backend\registro> python man
age.py runserver
```

Ilustración 63. Comando para iniciar el proyecto dentro del servidor.

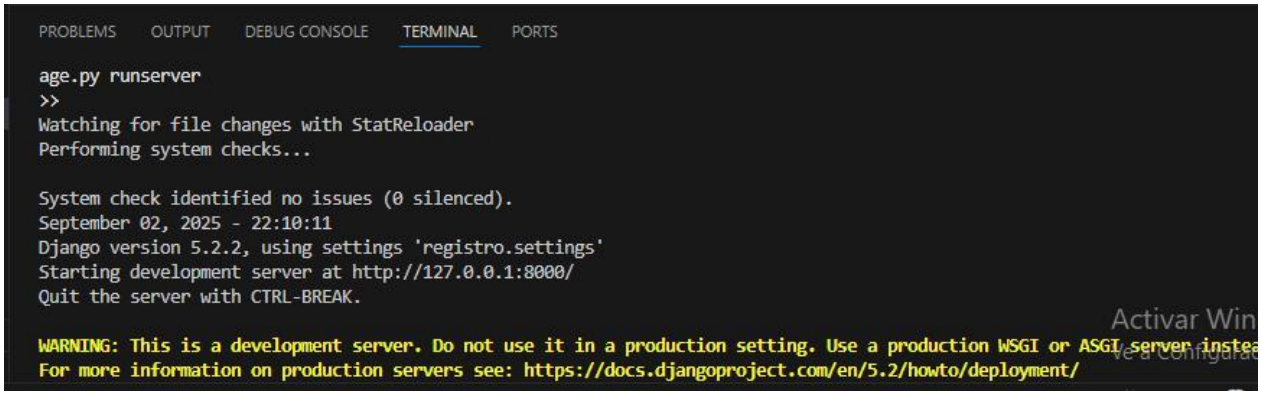


Ilustración 64.Ruta en la cual se puede ingresar al Servidor de Django.

Instalación de dependencias (frontend)

- cd Project_SCPP/frontend/crud-regristro
- npm install
- npm run dev

```
cd Project_SCPP/frontend/crud-regristro
npm install
npm run dev
```

Ilustración 65. Dependencias para utilizar el Front end.

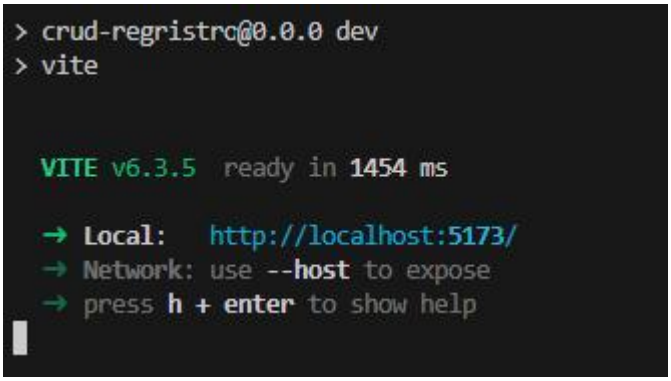


Ilustración 66. Ruta de localHost para ingresar a la vista del Front end.

URLs del API en el frontend

- host hardcodeado como http://localhost:8000.
- src/components/Login.jsx → fetch("http://localhost:8000/api/token/")

Actualmente, la URL del host está hardcodeada en localhost:8000, lo que funciona en desarrollo, pero para producción se recomienda configurarla en un archivo .env o en una variable de entorno, de modo que pueda adaptarse fácilmente al servidor donde se despliegue el sistema.

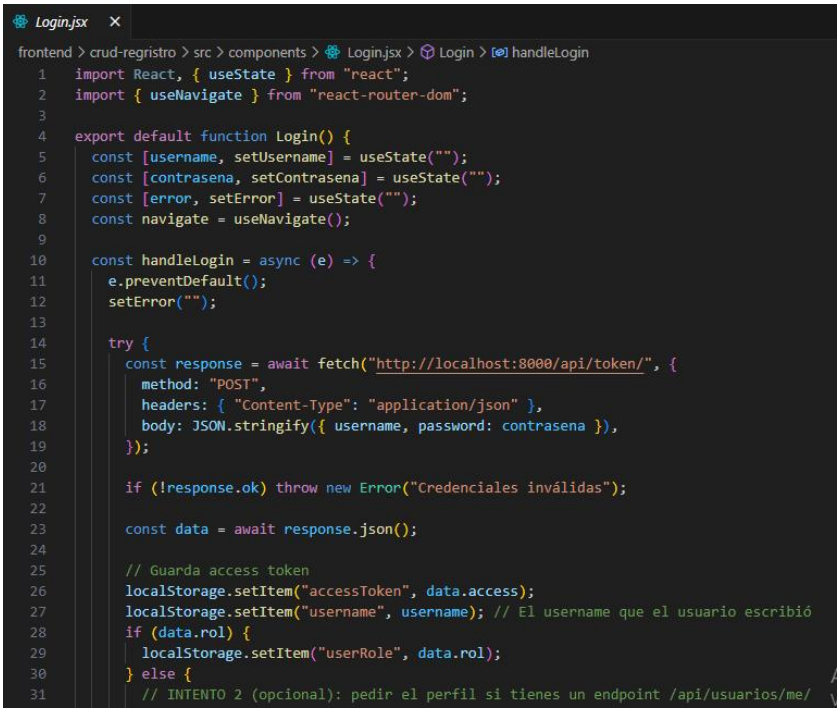


Ilustración 67. Estructura de código donde host está hardcodeada en localhost:8000.

- src/components/ModuloForm.jsx > varias llamadas http://localhost:8000/api/modulos/

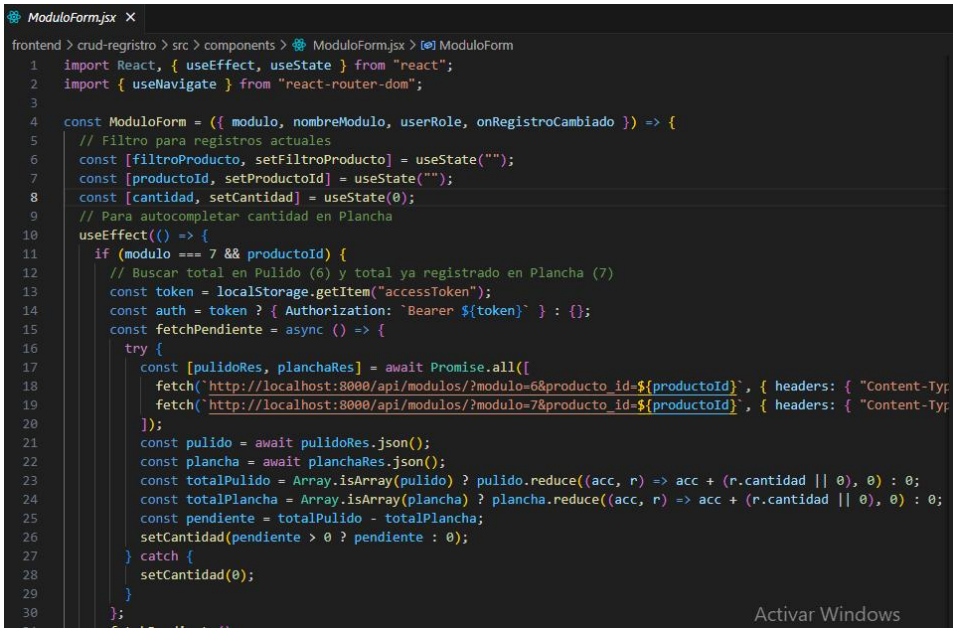


Ilustración 68. Consumo del API en módulos de producción.

En el componente ModuloForm.jsx se realizan llamadas al backend mediante fetch para obtener y registrar datos de producción. Por ejemplo, se consulta la cantidad ya registrada en el módulo de Pulido y Plancha utilizando el endpoint:

fetch("http://localhost:8000/api/modulos/?modulo_id=${productoId}")

La información recuperada permite calcular automáticamente las cantidades pendientes en cada módulo y mostrarlas en la interfaz. Actualmente, la URL del host también está hardcodeada en localhost:8000, por lo que se recomienda migrar esta configuración a variables de entorno para facilitar el despliegue en producción.

- src/components/RegisterList.jsx > getRegisters() pide usuarios

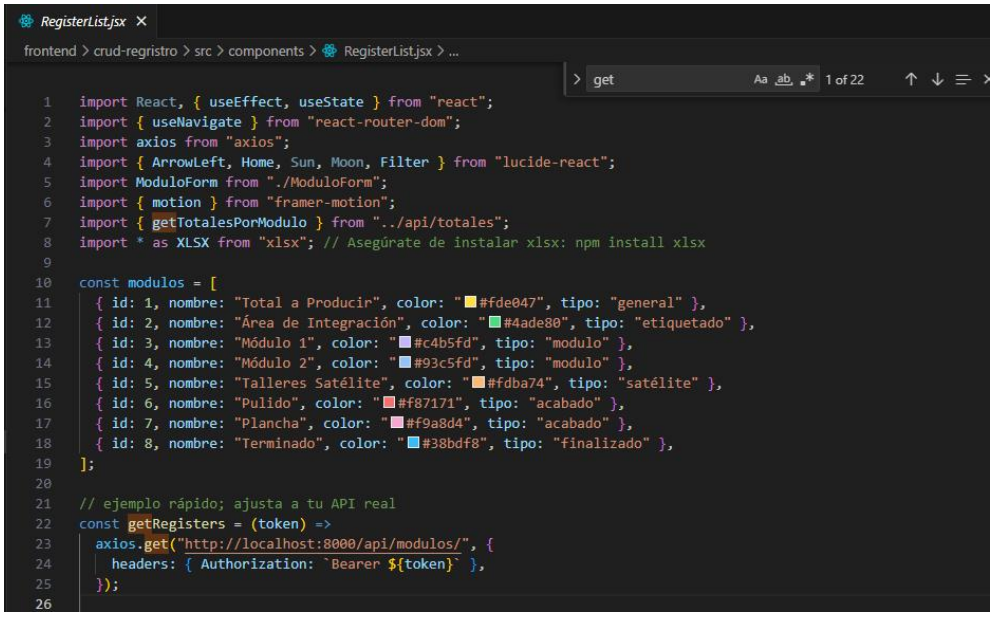


Ilustración 69. Componente RegisterList.jsx – Visualización general de módulos

Este componente define la estructura principal del tablero de producción, donde se muestran todos los módulos con su nombre, color y tipo (general, módulo, satélite, acabado o

finalizado). Cada tarjeta representa un módulo de la planta y permite visualizar las cantidades asociadas.

La conexión con el backend se realiza mediante axios, consumiendo el endpoint http://localhost:8000/api/modulos/, utilizando el token de autenticación almacenado en el navegador. De esta forma, el componente actualiza dinámicamente los datos de producción y asegura que solo los usuarios autorizados puedan acceder a la información.

- src/api/Register.js > baseURL para usuarios

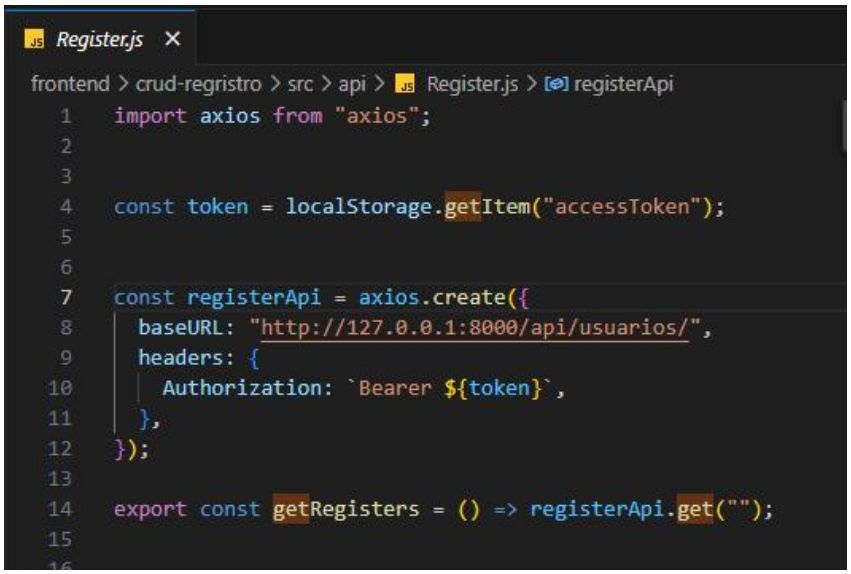


Ilustración 70.Archivo Register.js – Centralización de peticiones.

Este archivo configura axios para conectarse con la API, usando el token guardado en localStorage. Define la baseURL hacia http://127.0.0.1:8000/api/usuarios/ y la función getRegisters para obtener datos.

Se recomienda centralizar la baseURL en un archivo .env para facilitar el despliegue en distintos entornos sin modificar el código.

Creación del administrador del sistema

Para gestionar el sistema desde el Django Admin, es necesario crear un superusuario. Esto se realiza ejecutando el siguiente comando en la terminal dentro de la carpeta del proyecto:

python manage.py createsuperuser

El sistema solicitará un username, email y contraseña. Una vez creado, se podrá iniciar sesión en el panel de administración disponible en http://127.0.0.1:8000/admin/.

El superusuario tendrá permisos completos para crear y administrar usuarios, asignar roles y gestionar todos los módulos del sistema.

createsuperuser y acceso JWT:

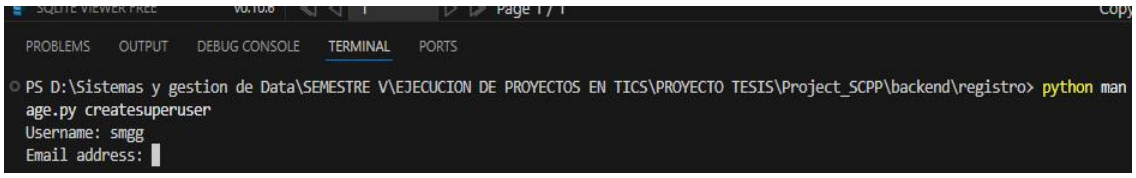


Ilustración 71.Creación del superusuario.

Endpoint login: POST /api/token/ (usuario/clave).

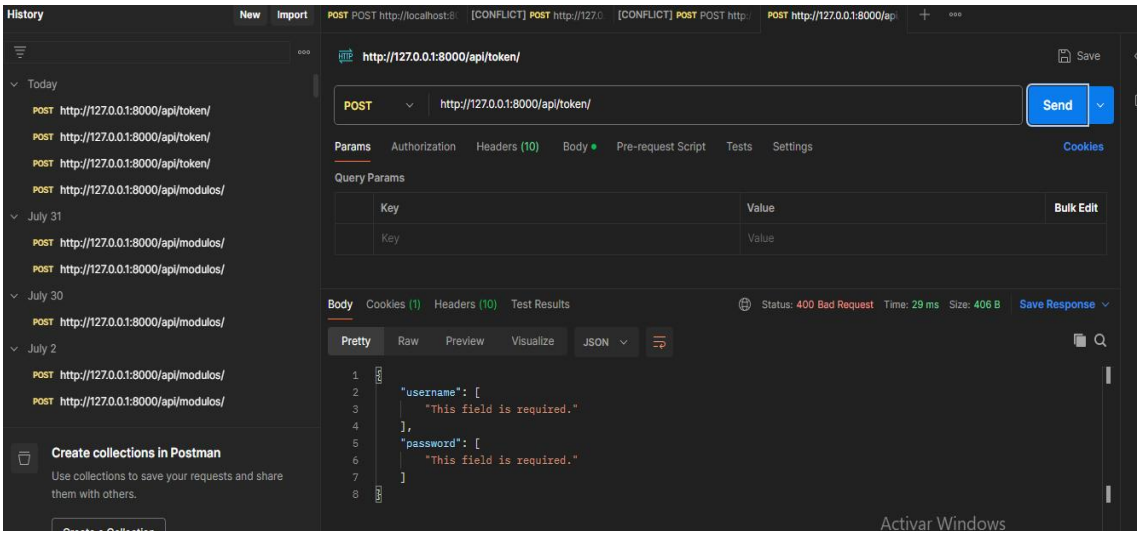


Ilustración 72.Prueba en Postman de POST para el token.

Respuesta con access y refresh. Guarda access en localStorage (src/components/Login.jsx).

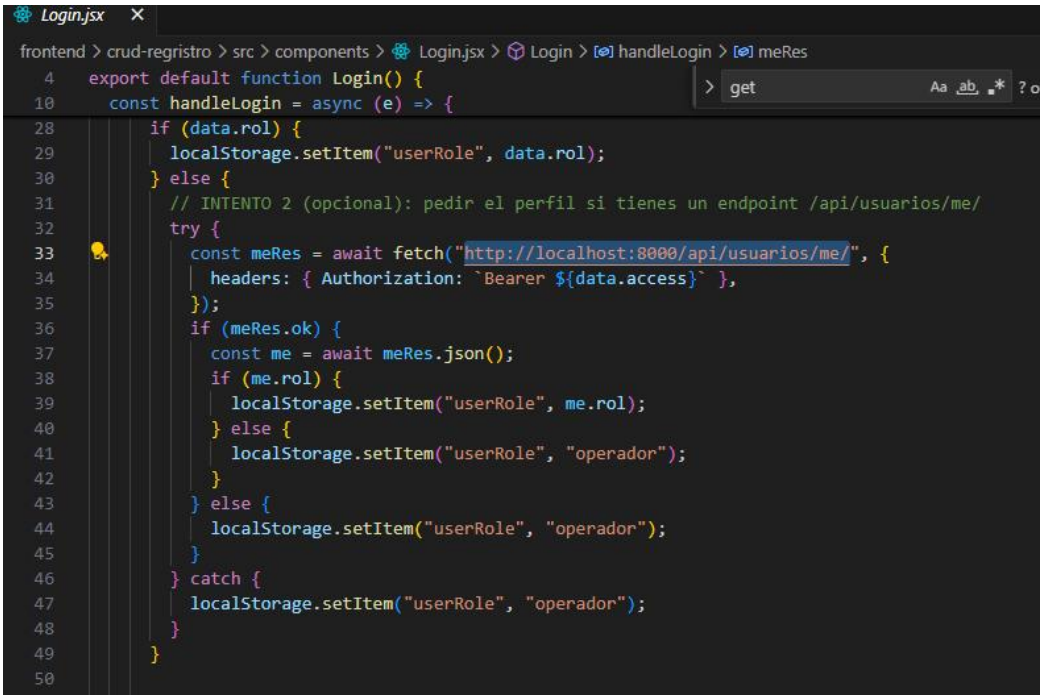


Ilustración 73. Manejo de autenticación y tokens

El sistema guarda el access token en localStorage, junto con el rol del usuario, lo que permite controlar accesos según el perfil (administrador u operador).

En caso de no recibir el rol directamente, se realiza una segunda petición a /api/usuarios/me/ para obtenerlo y almacenarlo. Esto asegura que la sesión mantenga la información necesaria para la autorización en todo el frontend.

Petición en Postman y localStorage con accessToken.

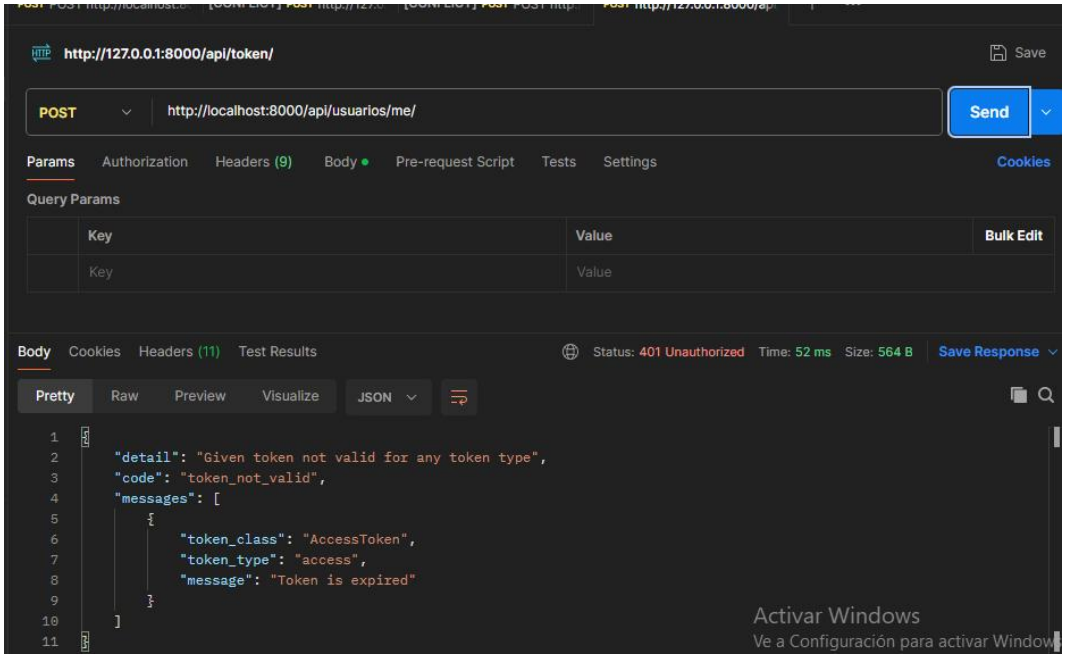


Ilustración 74. Manejo de expiración de tokens.

CÓDIGO FUENTE – MÓDULO PRODUCCIÓN

Interfaz principal (Dashboard)

Archivo: frontend/crud-regristro/src/components/RegisterList.jsx

```jsx
import React, { useEffect, useState } from "react";
import { useNavigate } from "react-router-dom";
import axios from "axios";
import { ArrowLeft, Home, Sun, Moon, Filter } from "lucide-react";
import ModuloForm from "./ModuloForm";
import { motion } from "framer-motion";
import { getTotalesPorModulo } from "../api/totales";
import * as XLSX from "xlsx"; // Asegúrate de instalar xlsx: npm install xlsx

const modulos = [
  { id: 1, nombre: "Total a Producir", color: "#fde047", tipo: "general" },
  { id: 2, nombre: "Área de Integración", color: "#4ade80", tipo: "etiquetado" },
  { id: 3, nombre: "Módulo 1", color: "#c4b5fd", tipo: "modulo" },
  { id: 4, nombre: "Módulo 2", color: "#93c5fd", tipo: "modulo" },
  { id: 5, nombre: "Talleres Satélite", color: "#fdba74", tipo: "satélite" },
  { id: 6, nombre: "Pulido", color: "#f87171", tipo: "acabado" },
  { id: 7, nombre: "Plancha", color: "#f9a8d4", tipo: "acabado" },
  { id: 8, nombre: "Terminado", color: "#38bdf8", tipo: "finalizado" },
];
```

Ilustración 75. Definición de módulos en el frontend

En este fragmento se define el arreglo modulos, donde cada objeto representa un área de producción de la planta. Cada módulo incluye:

- id : identificador único.
- nombre : nombre del módulo (ejemplo: "Total a Producir", "Módulo 1", "Plancha").
- color : color asignado para la interfaz gráfica, lo que facilita la diferenciación visual en el dashboard.

- tipo : clasificación del módulo (general, módulo, satélite, acabado o finalizado).

Estos datos permiten que el sistema construya el tablero principal de forma dinámica, mostrando cada área con un estilo único y manteniendo la consistencia de la información en toda la aplicación.

Render del grid de tarjetas (onClick abre ModuloForm y vuelve al panel).

```
    {/* Botón azul redondeado para volver al panel */}
    <button
      className="mt-6 bg-blue-600 hover:bg-blue-800 text-white font-semibold py-2 px-8 rounded-full shadow
      onClick={() => navigate('/dashboard')}
      type="button"
    >
      Volver al panel
    </button>
  </div>
)}
```

Ilustración 76.Botón de retorno al panel.

Sección “Tablero de Finalizados” (suma totalAProducir, totalFinalizado y “Restante”).

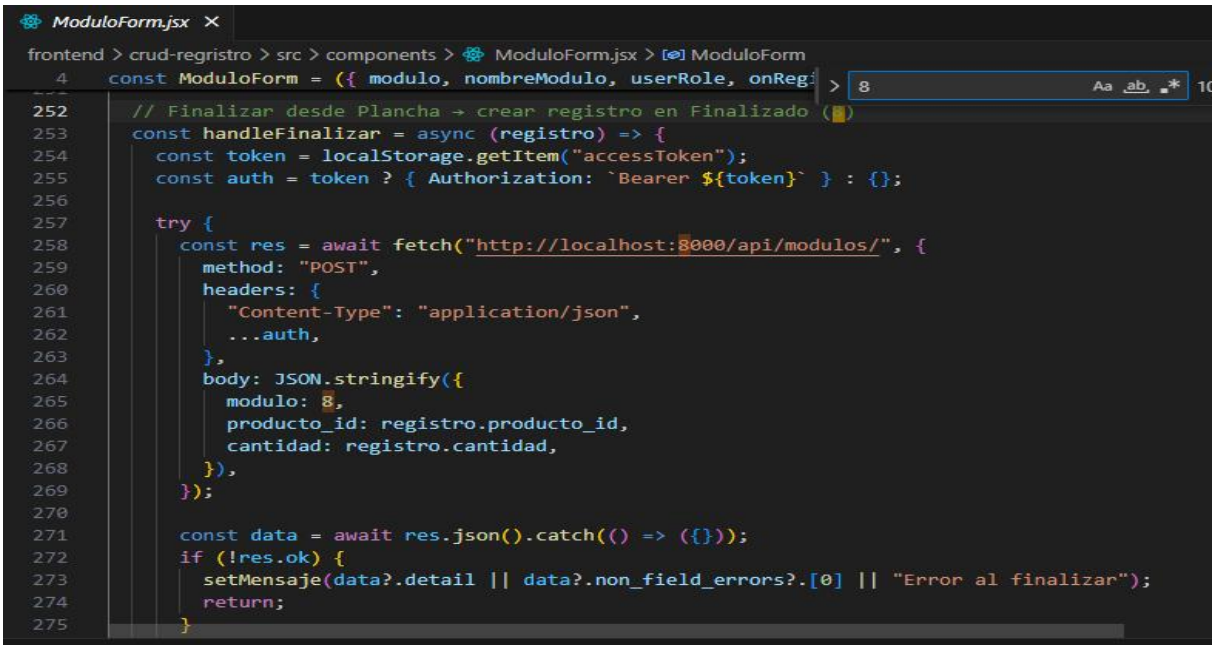


Ilustración 77. Tablero de Finalizados.

Navegación interna.

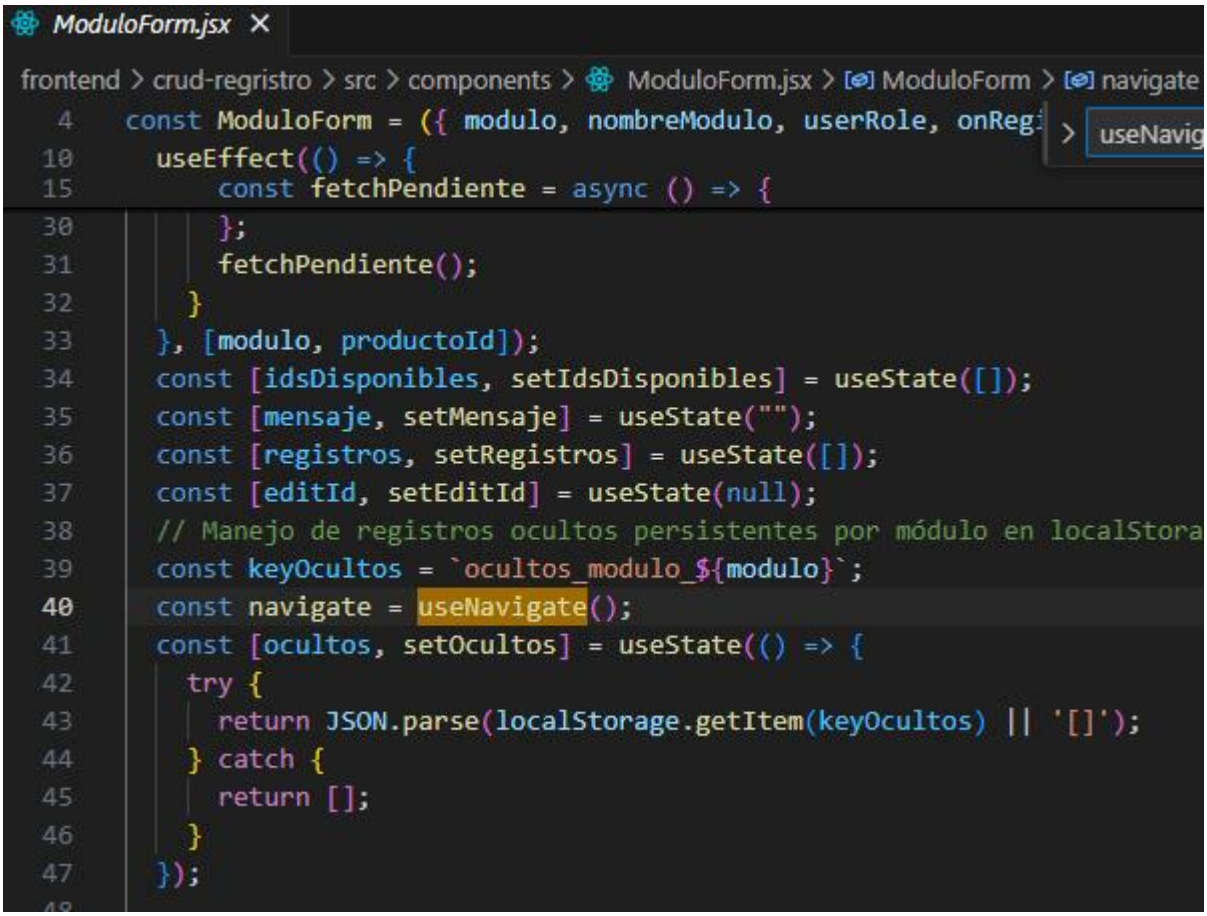


Ilustración 78. Manejo de registros y estados en módulos.

En ModuloForm.jsx se gestionan los estados de cada módulo con useState y useEffect, controlando IDs, registros, mensajes y edición.

Se utiliza localStorage con claves dinámicas para mantener persistencia de registros ocultos aun tras recargas, y la navegación fluida entre vistas se maneja con useNavigate, garantizando usabilidad sin recargar la aplicación.

Cálculo de totales en el frontend.

```
// Totales
const registrosArray = Array.isArray(registros) ? registros : [];
// Filtrar los registros que no están ocultos visualmente
const registrosVisibles = registrosArray.filter(r => !ocultos.includes(r.id));
const sumaTotal = registrosVisibles.reduce((acc, r) => acc + (parseInt(r.cantidad, 10) || 0), 0);
// Guardar el total visual en localStorage para que RegisterList lo use
useEffect(() => {
  try {
    localStorage.setItem(`total_modulo_visual_${modulo}`, sumaTotal.toString());
  } catch {}
}, [sumaTotal, modulo]);
const totalModulo = registrosVisibles.reduce((acc, r) => acc + (r.cantidad || 0), 0);
const totalPorProducto = registrosVisibles
  .filter((r) => r.producto_id === productoId)
  .reduce((acc, r) => acc + (r.cantidad || 0), 0);
```

Ilustración 79. Cálculo de totales por módulo.

En este bloque se calculan los totales de producción visibles filtrando los registros no ocultos.

El valor acumulado se guarda en localStorage para que otras vistas como RegisterList.jsx lo consuman y también se obtiene el total por producto, lo que facilita el seguimiento detallado dentro de cada módulo.

Interfaz crear/editar/eliminar registro

```
const handleSubmit = async (e) => {
  e.preventDefault();
  setMensaje("");
  const token = localStorage.getItem("accessToken");
  const auth = token ? { Authorization: `Bearer ${token}` } : {};

  const body = {
    modulo,
    producto_id: productoId,
    cantidad: parseInt(cantidad, 10),
  };
```

Ilustración 80. Función handleSubmit: Preparación de datos para el registro.

```
// CREAR
const res = await fetch("http://localhost:8000/api/modulos/", {
  method: "POST",
  headers: {
    "Content-Type": "application/json",
    ...auth,
  },
  body: JSON.stringify(body),
});

const data = await res.json().catch(() => ({}));

if (!res.ok) {
  setMensaje(data?.detail || data?.non_field_errors?.[0] || "Error al registrar");
  return;
}

setMensaje("Registro exitoso");
setProductoId("");
setCantidad(0);
setRegistros((prev) => [...prev, data]);
onRegistroCambiado && onRegistroCambiado();
```

Ilustración 81.Petición POST al API: Creación de registros en la base de datos.

La magne 81 muestra secciones del código del flujo completo para registrar datos en el sistema. Primero, la función handleSubmit obtiene el token de seguridad, prepara el cuerpo de la petición con el módulo, producto y cantidad, y limpia mensajes previos. Luego, se envía una petición POST al backend en la ruta /api/modulos/, validando errores y confirmando el éxito con el mensaje correspondiente. De esta manera, se asegura que los registros ingresados se almacenen de forma segura y en tiempo real.

Autocompletado por módulo (IDs previos):

Este fragmento implementa un campo de entrada con lista desplegable dinámica (datalist) que permite al usuario escribir o seleccionar un ID proveniente del módulo anterior. La lista se llena recorriendo el arreglo idsDisponibles, de modo que solo se sugieren opciones válidas registradas previamente. Esta funcionalidad agiliza el ingreso de datos y evita errores al reutilizar IDs ya existentes dentro del flujo de producción.

```
    {/* Autocompletado: escribe y elige IDs provenientes del módulo anterior */}
    <input
      list={`idsDisponiblesList-${modulo}`}
      className="border border-gray-300 rounded-lg p-2 w-full"
      value={productoId}
      onChange={(e) => setProductoId(e.target.value)}
      placeholder="Escriba o seleccione un SO del módulo anterior"
      required
    />
    <datalist id={`idsDisponiblesList-${modulo}`}>
      {idsDisponibles.map((id) => (
        <option key={id} value={id} />
      ))}
    </datalist>
  </>
)}
div>
```

Ilustración 82. Autocompletado de IDs provenientes del módulo anterior.

Registrar/Actualizar:

Ilustración 83.Función para actualizar registros en los módulos.

Este fragmento permite modificar registros ya existentes en la base de datos mediante una petición PUT a la API, identificando el elemento por su editId. Si la respuesta es exitosa, el registro se actualiza en la interfaz y se muestra el mensaje “Registro actualizado”, mientras que en caso de error se notifica al usuario. Esta funcionalidad asegura que los datos puedan corregirse o ajustarse sin necesidad de eliminarlos y volverlos a crear.

Eliminar:

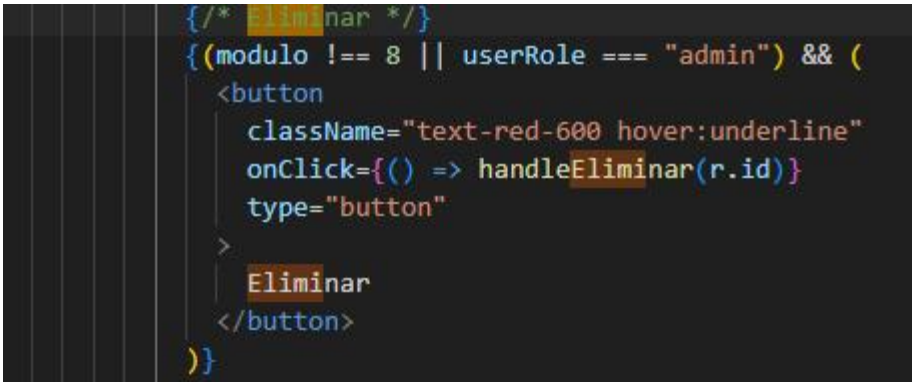


Ilustración 84. Botón para eliminar registros con control de permisos.

Este código implementa un botón que permite eliminar registros, pero solo está disponible para usuarios con rol de administrador o en módulos diferentes al número 8. El evento onClick ejecuta la función handleEliminar con el ID del registro, asegurando que la acción se realice únicamente bajo las condiciones establecidas y reforzando la seguridad del sistema.

Finalizar (desde Plancha):

función handleFinalizar que crea registro con modulo: 8.

```
{/* FINALIZAR solo visible en Plancha (7) */}
{modulo === 7 && (
  <button
    className="text-cyan-700 hover:underline font-bold"
    onClick={() => handleFinalizar(r)}
    type="button"
  >
    FINALIZAR
  </button>
)}
```

Ilustración 85. Botón de finalización exclusivo del módulo Plancha.

Este fragmento define un botón de FINALIZAR que solo se muestra cuando el usuario trabaja en el módulo 7 (Plancha). Al hacer clic, se ejecuta la función handleFinalizar que marca el registro como completado, garantizando que únicamente en este módulo se pueda cerrar el proceso de producción.

Totales del módulo y por producto:

sumaTotal, totalModulo, totalPorProducto.

```
// Totales
const registrosArray = Array.isArray(registros) ? registros : [];
// Filtrar los registros que no están ocultos visualmente
const registrosVisibles = registrosArray.filter(r => !ocultos.includes(r.id));
const sumaTotal = registrosVisibles.reduce((acc, r) => acc + (parseInt(r.cantidad, 10) || 0), 0);
// Guardar el total visual en localStorage para que RegisterList lo use
useEffect(() => {
  try {
    localStorage.setItem(`total_modulo_visual_${modulo}`, sumaTotal.toString());
  } catch {}
}, [sumaTotal, modulo]);
const totalModulo = registrosVisibles.reduce((acc, r) => acc + (r.cantidad || 0), 0);
const totalPorProducto = registrosVisibles
  .filter((r) => r.producto_id === productoId)
  .reduce((acc, r) => acc + (r.cantidad || 0), 0);
```

Ilustración 86.Cálculo de totales y control de registros visibles.

fetch con token: Authorization: Bearer ${token}.

Manejo de errores (res.ok, try/catch) para evitar el clásico “data.map is not a function”.

Interfaz módulo por módulo (capturas)

Módulo 1 – Total a producir: campo de ID libre + cantidad.

```
# ---------- MÓDULO 1: Total a Producir (único) ----------
if modulo == 1:
    existente = RegistroProduccion.objects.filter(producto_id=producto_id, modulo=1)
    if existente.exists() and not self.instance:
        raise serializers.ValidationError("Este ID ya fue registrado en 'Total a Producir'.")
    return data
```

Ilustración 87.Validación de unicidad en el módulo “Total a Producir”

Módulo 2 – Etiquetado: ID desde Módulo 1 (autocompletar) + cantidades acumulables.

```python
# ---------- MÓDULO 2: Etiquetado (no menor a lo ya repartido 3,4,5) ----------
if modulo == 2:
    total_distribuido = (
        RegistroProduccion.objects
        .filter(producto_id=producto_id, modulo__in=[3, 4, 5])
        .aggregate(s=Sum('cantidad'))['s'] or 0
    )
    total_actual_otras_filas = (
        RegistroProduccion.objects
        .filter(producto_id=producto_id, modulo=2)
        .exclude(pk=self.instance.pk if self.instance else None)
        .aggregate(s=Sum('cantidad'))['s'] or 0
    )
    if total_distribuido > total_actual_otras_filas + cantidad:
        raise serializers.ValidationError(
            "La cantidad en Etiquetado no puede ser menor que lo ya distribuido a Módulo 1, 2 y Satélite."
        )
    return data
```

Ilustración 88.Validación en el Módulo 2: Etiquetado

Módulo 3/4/5 – Módulos 1/2/Satélite: reciben de Etiquetado.

```python
# ---------- MÓDULOS 3, 4, 5: no exceder lo recibido de Etiquetado ----------
if modulo in [3, 4, 5]:
    total_etiquetado = (
        RegistroProduccion.objects
        .filter(producto_id=producto_id, modulo=2)
        .aggregate(s=Sum('cantidad'))['s'] or 0
    )
    total_distribuido = (
        RegistroProduccion.objects
        .filter(producto_id=producto_id, modulo__in=[3, 4, 5])
        .exclude(pk=self.instance.pk if self.instance else None)
        .aggregate(s=Sum('cantidad'))['s'] or 0
    )
    if total_distribuido + cantidad > total_etiquetado:
        raise serializers.ValidationError(
            f"La suma total distribuida a Módulo 1, Módulo 2 y Talleres Satélite no puede superar lo r
        )
    return data
```

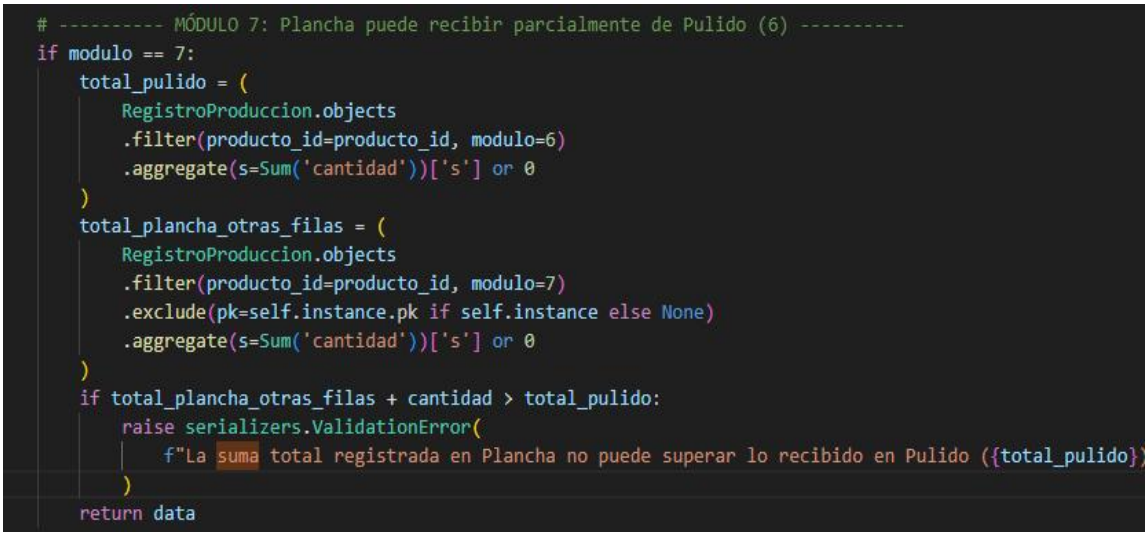
Ilustración 89. Validación en los Módulos 3, 4 y 5: No exceder lo recibido en Etiquetado.

Módulo 6 – Pulido: recibe de 3/4/5.

```
# ---------- MÓDULO 6: Pulido puede recibir parcialmente de 3,4,5 ----------
if modulo == 6:
    total_entrada = (
        RegistroProduccion.objects
        .filter(producto_id=producto_id, modulo__in=[3, 4, 5])
        .aggregate(s=Sum('cantidad'))['s'] or 0
    )
    total_registrado_otras_filas = (
        RegistroProduccion.objects
        .filter(producto_id=producto_id, modulo=6)
        .exclude(pk=self.instance.pk if self.instance else None)
        .aggregate(s=Sum('cantidad'))['s'] or 0
    )
    if total_registrado_otras_filas + cantidad > total_entrada:
        raise serializers.ValidationError(
            f"La cantidad total en Pulido no debe superar lo recibido de Módulo 1, Módulo 2 y Talleres
        )
    return data
```

Ilustración 90. Validación en el Módulo 6: Control de entradas desde los módulos 3, 4 y 5

Módulo 7 – Plancha: recibe de 6 + botón FINALIZAR (crea en 8).



```
# ---------- MÓDULO 7: Plancha puede recibir parcialmente de Pulido (6) ----------
if modulo == 7:
    total_pulido = (
        RegistroProduccion.objects
        .filter(producto_id=producto_id, modulo=6)
        .aggregate(s=Sum('cantidad'))['s'] or 0
    )
    total_plancha_otras_filas = (
        RegistroProduccion.objects
        .filter(producto_id=producto_id, modulo=7)
        .exclude(pk=self.instance.pk if self.instance else None)
        .aggregate(s=Sum('cantidad'))['s'] or 0
    )
    if total_plancha_otras_filas + cantidad > total_pulido:
        raise serializers.ValidationError(
            f"La suma total registrada en Plancha no puede superar lo recibido en Pulido ({total_pulido})
        )
    return data
```

Ilustración 91. Validación en el Módulo 7: Control de registros en Plancha desde Pulido.

Módulo 8 – Finalizado: tabla con totales; botón Eliminar solo si userRole === 'admin' (ya contemplado en JSX: condicional en el listado).

```
# ---------- MÓDULO 8: Finalizado ----------
if modulo == 8:
    total_plancha = (
        RegistroProduccion.objects
        .filter(producto_id=producto_id, modulo=7)
        .aggregate(s=Sum('cantidad'))['s'] or 0
    )
    ya_finalizado_otras_filas = (
        RegistroProduccion.objects
        .filter(producto_id=producto_id, modulo=8)
        .exclude(pk=self.instance.pk if self.instance else None)
        .aggregate(s=Sum('cantidad'))['s'] or 0
    )
    if total_plancha > 0:
        if ya_finalizado_otras_filas + cantidad > total_plancha:
            raise serializers.ValidationError(
                "La cantidad finalizada excede lo recibido de Plancha."
            )
        return data

    total_pulido = (
        RegistroProduccion.objects
        .filter(producto_id=producto_id, modulo=6)
```

Ilustración 92. Validación en el Módulo 8: Control de registros en Terminado.

A continuación en esta parte del código se implementa un condicional que restringe la acción de eliminar registros únicamente a los usuarios con rol admin, validando además que pertenezca al módulo indicado. El parámetro userRole se asigna automáticamente tras el inicio de sesión, ya que el backend devuelve el rol del usuario según lo definido en el modelo usuarios.Usuario. De esta forma, se asegura que solo los administradores tengan el privilegio de ejecutar eliminaciones, manteniendo el control y la integridad de los datos en el sistema.

Ilustración 93. Condicional de eliminación solo para administradores.

El campo rol del modelo usuarios.Usuario en Django define si un usuario es Administrador o Operador. Al iniciar sesión, el backend envía este rol junto con el token y en el frontend se guarda en userRole. Esto permite condicionar las funcionalidades según permisos, por ejemplo que solo los administradores puedan eliminar registros.

```python
# usuarios/models.py
from django.contrib.auth.models import AbstractUser
from django.db import models
from django.utils import timezone
# Custom user model

class Usuario(AbstractUser):
    date_joined = models.DateTimeField(default=timezone.now)
    ROL_CHOICES = (
        ('admin', 'Administrador'),
        ('operador', 'Operador'),
    )

    rol = models.CharField(
        max_length=20,
        choices=ROL_CHOICES,
        default='operador'
    )

    def __str__(self):
        return f"{self.username} ({self.rol})"
```

Ilustración 94. Manejo de roles con userRole.

API/Backend que soporta la interfaz

Rutas: backend/registro/registro/urls.py

En este archivo se definen las rutas principales del backend. Se incluye el panel de administración de Django y las rutas para el manejo de autenticación con JWT (api/token y api/token/refresh). Además, se enlazan los endpoints personalizados para usuarios y módulos, lo que organiza el acceso a cada aplicación del sistema desde una única configuración central.

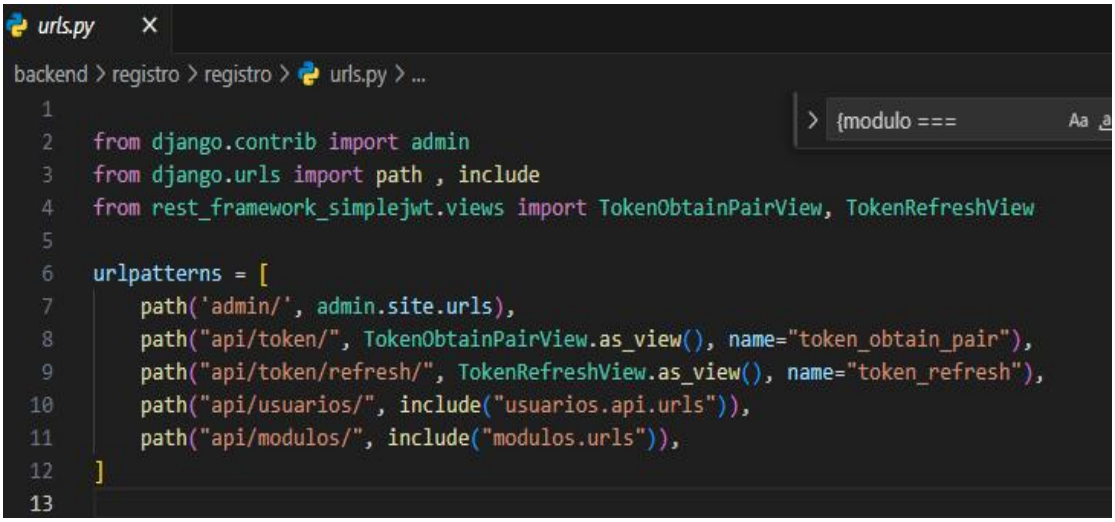


Ilustración 95. Configuración de Rutas (urls.py).

Módulo Producción

Rutas: backend/registro/modulos/views.py

El RegistroProduccionViewSet implementa los métodos CRUD protegidos con JWTAuthentication e IsAuthenticated, permitiendo solo a usuarios logueados acceder al API. El método get_queryset filtra registros por módulo o producto, mientras que perform_create guarda automáticamente el usuario que creó el registro para mantener trazabilidad. Además, se incluye la acción finalizar, que registra productos como completados en el módulo 8 o mediante un endpoint específico.

views.py

backend > registro > modulos > views.py > NotaModuloViewSet > permission_classes

{modulo ===

```
46
47  class RegistroProduccionViewSet(viewsets.ModelViewSet):
48      queryset = RegistroProduccion.objects.all().order_by('-fecha')
49      serializer_class = RegistroProduccionSerializer
50      authentication_classes = [JWTAuthentication]
51      permission_classes = [IsAuthenticated]
52
53      def get_queryset(self):
54          qs = super().get_queryset()
55          modulo = self.request.query_params.get('modulo')
56          producto_id = self.request.query_params.get('producto_id')
57          if modulo:
58              qs = qs.filter(modulo=modulo)
59          if producto_id:
60              qs = qs.filter(producto_id=producto_id)
61          return qs
62
63      def perform_create(self, serializer):
64          serializer.save(usuario=self.request.user)
65
66      @action(detail=False, methods=['post'], url_path='finalizar')
67      def finalizar(self, request):
68          producto_id = request.data.get('producto_id')
69          cantidad_req = request.data.get('cantidad')  # opcional
70
71          if not producto_id:
```

Ilustración 96.Vista de Registros de Producción (views.py).

Rutas: backend/registro/modulos/serializers.py

Propósito: centraliza la validación del flujo de producción: define de qué módulos puede provenir un producto_id y aplica límites de suma para impedir sobre-registro entre módulos (reglas 1>2>3/4/5>6>7>8).

Salida: expone campos id, modulo, producto_id, cantidad, fecha y un total_acumulado vía SerializerMethodField, que calcula la suma registrada del mismo producto dentro del módulo (útil para el front).

Efecto: garantiza consistencia entre lo recibido y lo distribuido en cada etapa y evita duplicidades, devolviendo mensajes claros de ValidationError cuando se violan las reglas.

```python
class RegistroProduccionSerializer(serializers.ModelSerializer):
    # opcional: si quieres devolver el acumulado del módulo en cada respuesta
    total_acumulado = serializers.SerializerMethodField()

    class Meta:
        model = RegistroProduccion
        # Si quieres incluir usuario en la respuesta, añádelo aquí
        fields = ['id', 'modulo', 'producto_id', 'cantidad', 'fecha', 'total_acumulado']
        read_only_fields = ['fecha']  # 'usuario' no está en fields; quítalo de read_only

    def get_total_acumulado(self, obj):
        """
        Suma de cantidades del mismo producto_id dentro del mismo módulo.
        Útil para mostrar 'Total registrado hasta ahora' en el front.
        """
        return (
            RegistroProduccion.objects
```

Ilustración 97.RegistroProduccionSerializer (serializers.py).

TotalesModuloSerializer: para lecturas/acumulados

```python
class TotalesModuloSerializer(serializers.ModelSerializer):
    class Meta:
        model = TotalesModulo
        fields = ['modulo', 'producto_id', 'total_cantidad', 'ultima_actualizacion']
```

Ilustración 98.TotalesModuloSerializer (serializers.py).

Modelos (muestra clases y campos):

usuarios/models.py → Usuario(AbstractUser, rol=…)

El modelo Usuario hereda de AbstractUser y agrega el campo rol, que puede ser Administrador o Operador. Esto permite controlar permisos y vistas según el perfil asignado en el login. Además, sobrescribe el método __str__ para mostrar el nombre de usuario junto con su rol, facilitando la identificación en la administración del sistema y en los reportes.

```python
# usuarios/models.py
from django.contrib.auth.models import AbstractUser
from django.db import models
from django.utils import timezone
# Custom user model

class Usuario(AbstractUser):
    date_joined = models.DateTimeField(default=timezone.now)
    ROL_CHOICES = (
        ('admin', 'Administrador'),
        ('operador', 'Operador'),
    )

    rol = models.CharField(
        max_length=20,
        choices=ROL_CHOICES,
        default='operador'
    )

    def __str__(self):
        return f"{self.username} ({self.rol})"
```

Ilustración 99.Modelos principales — modulos/models.py

Reporte de Finalizados:

En este componente se implementa el Tablero de Finalizados, que presenta de forma visual tres métricas clave: Total a Producir, Total Finalizado y Restante. Los valores se calculan dinámicamente a partir de los registros y permiten monitorear el avance global de la producción. Con esta sección, el usuario puede evaluar en tiempo real cuánto se ha completado y cuánto queda pendiente, facilitando la gestión y control del proceso.

```
/* TABLERO FINALIZADO */}
mostrarSoloFinalizado ? (
 <div className="bg-white dark:bg-gray-800 rounded-2xl shadow-lg p-6">
  <h2 className="text-2xl font-bold mb-4 text-blue-700 dark:text-blue-300">Tablero de Finalizados</h2>
  <div className="flex flex-col md:flex-row gap-6 mb-6">
   <div className="flex-1 bg-blue-100 dark:bg-blue-900 rounded-xl p-4 text-blue-900 dark:text-blue-200 shadow">
    <p className="font-semibold">Total a Producir:</p>
    <p className="text-2xl font-bold">{totalAProducir}</p>
   </div>
   <div className="flex-1 bg-cyan-100 dark:bg-cyan-900 rounded-xl p-4 text-cyan-900 dark:text-cyan-200 shadow">
    <p className="font-semibold">Total Finalizado:</p>
    <p className="text-2xl font-bold">{totalFinalizado}</p>
   </div>
   <div className="flex-1 bg-green-100 dark:bg-green-900 rounded-xl p-4 text-green-900 dark:text-green-200 shado
    <p className="font-semibold">Restante:</p>
    <p className="text-2xl font-bold">{totalAProducir - totalFinalizado}</p>
   </div>
  </div>
  <div className="overflow-x-auto">
   <table className="min-w-full bg-white dark:bg-gray-800 rounded-xl shadow">
    <thead>
     <tr>
```

Ilustración 100. Resumen — Tablero de Finalizados (RegisterList.jsx).

frontend/crud-regristro/src/api/Register.js

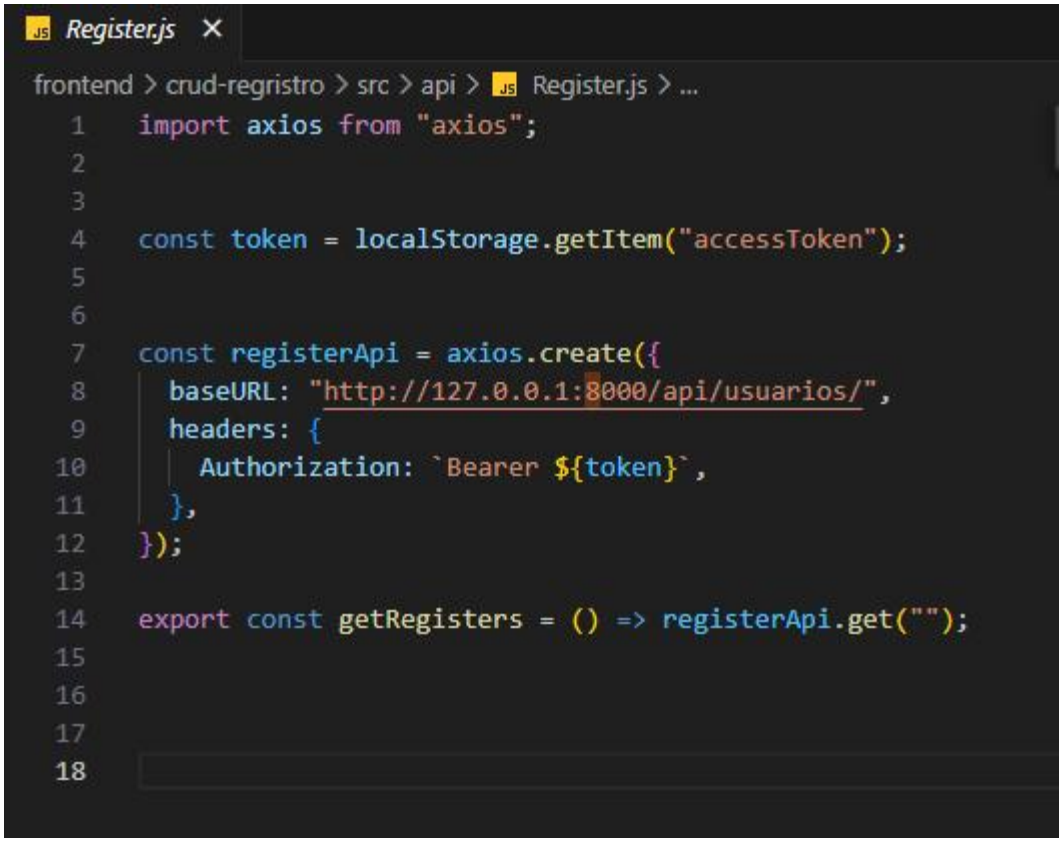


Ilustración 101. Resumen — Register.js y mejora propuesta.

En este archivo se define la conexión al endpoint /api/usuarios/ mediante Axios, incluyendo el token almacenado en localStorage para la autenticación. Esto permite obtener la lista de usuarios registrados en el sistema.

Mejora recomendada: en lugar de repetir la configuración de Axios en cada archivo, es conveniente crear un api.js centralizado, donde se defina el baseURL y se configure un interceptor que inyecte automáticamente el token en cada petición. Esto facilita el mantenimiento,

evita duplicación de código y asegura que todas las peticiones estén autenticadas de forma consistente.