



UNIVERSIDAD DE BUENOS AIRES

Facultad de Ingeniería

86.65 Sistemas Embebidos

Memoria del Trabajo Final:

**Sistema de Control de Servicio
(Cinta Transportadora de Paquetes)**

Autores y Legajos:

Ignacio Botbol - 109834

Sergio Nahuel Chaparro - 109752

Josefina Maria Duvidovich - 107467

Santiago Fontela - 109921

*Este trabajo fue realizado en las Ciudad Autónoma de Buenos Aires,
entre marzo y agosto de 2025.*

RESUMEN

En este trabajo final se desarrolló un sistema embebido para controlar una cinta transportadora de paquetes, diseñado e implementado sobre una plataforma con microcontrolador de la familia STM32. El sistema permite seleccionar entre modo de operación normal o setup, modificar la velocidad de la cinta en tres niveles y gestionar una condición de falla simulada. Las acciones del usuario se registran a través de cuatro botones físicos, y el estado del sistema se presenta mediante un display de caracteres y un conjunto de indicadores luminosos. Además, se incorporó un sensor de temperatura, y se implementó almacenamiento no volátil para conservar la configuración del sistema tras reinicios.

El desarrollo del sistema requirió integrar varios elementos clave del diseño de sistemas embebidos, como el manejo de entradas digitales con filtrado por software para evitar rebotes, el control de salidas a través de máquinas de estados finitas, la comunicación con memorias externas mediante el protocolo I2C, y la lectura de señales analógicas desde sensores. La arquitectura del software se organizó de forma modular, con tareas que se ejecutan de manera periódica, lo cual facilita el mantenimiento y la ampliación del sistema. Este enfoque permitió aplicar de forma práctica los conocimientos adquiridos durante el curso, incluyendo programación estructurada, diseño de firmware, uso de periféricos y técnicas de depuración. En esta memoria se detalla cada etapa del desarrollo, desde el diseño inicial hasta la implementación final y las pruebas de funcionamiento.

El sistema implementado permite al usuario interactuar de forma intuitiva con la cinta, cambiar su velocidad de operación, alternar entre modos de control y visualizar el estado general mediante indicadores claros y persistentes. Su funcionamiento refleja una integración equilibrada entre hardware y software, orientada a una aplicación real de automatización. La memoria documenta cómo cada módulo contribuye al comportamiento general del sistema, y demuestra que se alcanzó un diseño robusto, funcional y coherente con los objetivos propuestos.

ABSTRACT

In this final project, an embedded system was developed to control a package conveyor belt, designed and implemented on a platform based on a microcontroller from the STM32 family. The system allows switching between normal and setup modes, adjusting the conveyor speed across three levels, and managing a simulated fault condition. User actions are captured through four physical buttons, and the system state is displayed on a character display along with a set of visual indicators. Additionally, a temperature sensor was integrated, and non-volatile storage was implemented to preserve system configuration across restarts.

The development of the system required the integration of several key elements of embedded systems design, such as handling digital inputs with software-based debouncing, controlling outputs through finite state machines, communicating with external memory via the I2C protocol, and reading analog signals from sensors. The software architecture was organized in a modular way, with tasks executed periodically, which facilitates system maintenance and future scalability. This approach enabled the practical application of knowledge acquired throughout the course, including structured programming, firmware design, peripheral management, and debugging techniques. This report details each stage of the project, from initial design through implementation and testing.

The implemented system allows the user to interact intuitively with the package handling system, change its operating speed, switch between control modes, and visualize the overall system status through clear and persistent indicators. Its operation reflects a balanced integration of hardware and software, aimed at a real-world automation application. The memory documents how each module contributes to the system's overall behavior and demonstrates that a robust, functional, and goal-oriented design was achieved.

Índice General

Registro de versiones	5
Introducción general	6
1.1 Descripción del Sistema de Control	7
1.2 Análisis de sistemas similares al desarrollado	11
Introducción específica	10
2.1 Requisitos	10
2.2 Casos de Uso	11
2.3 Descripción de tecnologías y componentes utilizados	13
2.3.1 Microcontrolador STM32F103RB	13
2.3.2 Pantalla LCD (Liquid Crystal Display) 2x16	14
2.3.3 Sensor de temperatura analógico	14
2.3.4 Memoria EEPROM externa AT24C256	15
2.3.5 Plataforma de desarrollo STM32CubeIDE	15
2.3.6 Protocolo de comunicación I2C	15
2.3.7 Periféricos básicos de entrada/salida: botones y LEDs	15
Diseño e implementación	16
3.1 Hardware del Sistema de Control	16
3.2 Firmware del Sistema de Control	17
Conclusiones	18
5.1 Resultados obtenidos	18

Registro de versiones

Revisión	Cambios realizados	Fecha
1.0	Versión Original	06/08/2025
1.1		
1.2		

CAPÍTULO 1

Introducción general

1.1 Descripción del Sistema de Control

El presente trabajo tuvo como objetivo el diseño e implementación de un sistema embebido para el control de una cinta transportadora de paquetes. Se trata de un sistema de automatización que integra hardware y software sobre un microcontrolador STM32, orientado a aplicaciones educativas e industriales.

El sistema permite al usuario operar la cinta en diferentes modos de funcionamiento, modificar parámetros de velocidad, simular condiciones de falla y almacenar configuraciones de manera persistente. Todo ello se realiza mediante una interfaz simple compuesta por botones físicos, indicadores LED y un display LCD de 2 líneas por 16 caracteres. Además, se incorporaron sensores analógicos como un sensor de temperatura, cuya lectura permite monitorear condiciones de operación en tiempo real.

El sistema de control fue concebido bajo una arquitectura modular, que facilita tanto su mantenimiento como su escalabilidad. Se implementó una estructura de software basada en máquinas de estados finitas, con programación tipo bare-metal, ejecución cíclica mediante un super-loop con tick de 1 ms y tareas no bloqueantes. Esta estructura permite gestionar periféricos mediante técnicas eficientes como polling, interrupciones o callbacks, asegurando un comportamiento predecible y de bajo consumo energético.

Este trabajo resultó relevante porque abordó problemáticas típicas de automatización en entornos donde es necesario transportar y clasificar objetos de forma repetitiva y controlada. Su diseño estuvo orientado tanto a la enseñanza de sistemas embebidos como a la aplicación en contextos reales de producción, especialmente en líneas de ensamblaje o clasificación de productos.

La importancia del trabajo radicó en su capacidad para integrar múltiples aspectos técnicos en una plataforma compacta y funcional. El principal desafío consistió en desarrollar una solución robusta que pudiera funcionar en diferentes modos (NORMAL, SET_UP y FALLA), interactuar con periféricos mediante protocolos estándar como I2C, realizar almacenamiento no volátil de parámetros y brindar una interfaz amigable al usuario. Todo ello se logró sin la necesidad de sistemas operativos ni herramientas de

desarrollo complejas, lo que representó una ventaja significativa para la formación técnica y la replicabilidad del sistema.

El presente trabajo se destacó especialmente por incorporar control de modos de operación, almacenamiento persistente de configuraciones, monitoreo de temperatura y una interfaz interactiva en un sistema de bajo costo y fácil implementación. Esto lo diferenció de otros sistemas similares en que no solo realizó el transporte básico de objetos, sino que permitió configurar y monitorear en tiempo real distintas variables del proceso, adaptándose a distintos contextos de uso sin grandes modificaciones estructurales.

A continuación, se presenta en la Figura 1.1 el diagrama de bloques general del sistema de control de la cinta transportadora. Este esquema permite visualizar cómo se estructura el sistema en módulos funcionales que interactúan de forma jerárquica y modular.

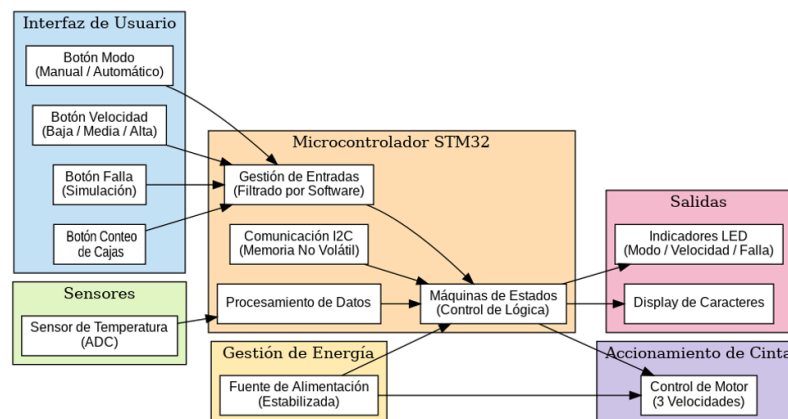


Figura 1.1: Diagrama de bloques del sistema de control

1.2 Análisis de sistemas similares al desarrollado

Para contextualizar el trabajo realizado, se analizaron productos existentes en el mercado que ofrecen funcionalidades similares al sistema de control de cinta transportadora de paquetes desarrollado. Este análisis permitió identificar las características esenciales que estos productos ofrecen, y establecer un conjunto base de funcionalidades alcanzables dentro del marco del trabajo propuesto.

Se seleccionaron dos productos de referencia, ampliamente utilizados con fines didácticos y demostrativos en áreas de automatización y control industrial. Las

características principales de ambos sistemas se resumen en la Tabla 1.1 mostrada a continuación.

Tabla 1.1: Características de productos comerciales similares

Características	Producto A: Cinta Transportadora (Festo, modelo 548705)	Producto B: Mini Cinta Transportadora Trifásica con Sensores Electrónicos (Insur, Argentina)
Control de velocidad	Selección manual mediante botones, múltiples niveles disponibles según configuración didáctica	Motor trifásico 1/8 HP con reductor mecánico y control práctico mediante sensores electrónicos
Modo de operación	Soporta modo automático y manual, pensado para formación técnica	Permite prácticas como arranque directo, inversión de marcha y detención por sensor fotoeléctrico
Interfaz de usuario	Botones físicos, display LCD y LEDs indicadores (didáctico)	Pulsadores, sensores fotoeléctricos y bornes de conexión experimentales
Gestión de fallas	Simulación de errores y detención automática mediante sensores didácticos	Detención mediante sensor reflectivo o de barrera en prácticas reales
Sensor de temperatura	Incluido en el equipo didáctico Festo	No especificado
Almacenamiento persistente	Memoria EEPROM para guardar configuración de usuario	No disponible
Compatibilidad educativa	Diseñado para laboratorios y simulaciones técnicas	Equipamiento didáctico para automatización trifásica
Precio estimado	~100 USD, con enfoque educativo especializado	Precio aproximado en mercado educativo argentino; no especificado en el sitio, supongo al ser menos complejo ~80 USD
Enlace sitio web	https://www.festo.com/ar/es/p/cinta-transportadora-id_PROD_DID_548705/?page=0	https://www.insur.com.ar/equipos/area-electromecanica/cinta-t/216

Ambos productos fueron tomados como referencia para comprender los alcances y limitaciones de soluciones existentes, especialmente en el ámbito formativo. El producto de Festo presenta una solución más completa, con mayor integración y funcionalidades avanzadas, pero con un costo más elevado. Por su parte, la cinta de Insur presenta un

enfoque más accesible y modular, permitiendo realizar prácticas de conexión y control en entornos de enseñanza técnica.

A partir de este análisis, se definió una línea base de funcionalidades a implementar en el sistema propuesto, seleccionando aquellas que fueran relevantes, factibles y alineadas con los objetivos del trabajo, tales como control de velocidad, interfaz de usuario con botones y LEDs, modos de operación diferenciados y almacenamiento de parámetros en memoria no volátil.

El análisis de estos sistemas también sirvió para identificar oportunidades de mejora y diferenciación, las cuales serán retomadas más adelante en el Capítulo 4, donde se realiza una comparación directa entre el trabajo desarrollado y estos sistemas comerciales.

CAPÍTULO 2

Introducción específica

2.1 Requisitos

El sistema de control de la cinta transportadora fue especificado a partir de un análisis de soluciones similares y de los objetivos establecidos para el desarrollo de un prototipo funcional. Para guiar este proceso se adoptaron los criterios SMART (*Specific, Measurable, Achievable, Relevant, Time-bounded*), que permitieron definir requisitos específicos, medibles y alcanzables dentro del plazo de desarrollo estipulado.

Los requisitos se clasificaron en grupos funcionales, y se presentan en la Tabla 2.1.1 que se muestra a continuación.

Tabla 2.1.1: Requisitos del sistema de control de cinta transportadora

Grupo de Requisitos	ID	Descripción
Interfaz	1.1	El sistema debía contar con cuatro botones físicos para controlar modo, velocidad, cantidad de paquetes y simular fallas.
	1.2	El sistema debía mostrar información del estado en un display de caracteres.
	1.3	Se debían incluir indicadores LED para modo y velocidad seleccionada.
Modos de operación	2.1	El sistema debía operar en modo NORMAL y modo SET_UP seleccionables por el usuario.
	2.2	En modo SET_UP el usuario debe poder iniciar/detener la cinta y configurar parámetros.
	2.3	En modo NORMAL la cinta debía detenerse ante una falla simulada.

Velocidad	3.1	El sistema debía permitir seleccionar entre tres niveles de velocidad.
Sensores	4.1	El sistema debía medir la temperatura ambiente mediante un sensor analógico.
Almacenamiento	5.1	La última configuración debía guardarse en memoria no volátil.
Robustez	6.1	Las entradas digitales debían ser filtradas por software para evitar rebotes.
Comunicación	7.1	El sistema debía comunicarse con una memoria externa mediante el protocolo I2C.
Tiempo de Desarrollo	8.1	El sistema debía completarse en un período de una semana.
Costos	9.1	El costo total del prototipo no debía superar los 40.000 pesos argentinos.

Estos requisitos constituyen la base para las decisiones de diseño adoptadas en los capítulos posteriores. Su cumplimiento fue evaluado y documentado en el Capítulo 5.

2.2 Casos de uso

A partir de los requisitos definidos, se identificaron los principales escenarios de interacción entre el usuario y el sistema, formalizados mediante casos de uso. Estos casos describen acciones típicas esperadas por parte del usuario, las condiciones previas necesarias, el comportamiento del sistema y posibles variantes en la operación.

Se definieron tres casos de uso representativos, presentados en las Tablas 2.2.1, 2.2.2 y 2.2.3.

Tabla 2.2.1: Caso de uso #1 – Activación manual de la cinta

Elemento del caso de uso	Descripción
Disparador (Trigger)	El usuario desea iniciar el movimiento de la cinta manualmente.

Precondición	El sistema debía estar encendido y configurado en modo manual.
Flujo básico	El usuario presiona el botón de inicio. La cinta se movía a la velocidad configurada.
Flujo alternativo	1a: El usuario presiona el botón de parada, y la cinta se detiene inmediatamente.

Tabla 2.2.2: Caso de uso #2 – Detección de falla en modo automático

Elemento del caso de uso	Descripción
Disparador (Trigger)	El usuario activa el modo automático.
Precondición	El sistema está encendido y funcionando normalmente.
Flujo básico	El usuario selecciona modo automático. La cinta funciona hasta que se simula una falla.
Flujo alternativo	2a: El sistema detecta una falla simulada → se detiene y activa la señalización.

Tabla 2.2.3: Caso de uso #3 – Cambio de velocidad

Elemento del caso de uso	Descripción
Disparador (Trigger)	El usuario desea cambiar la velocidad de la cinta.
Precondición	El sistema debe estar encendido.
Flujo básico	El usuario presiona el botón de velocidad → el sistema alterna entre tres niveles posibles.

Flujo alternativo

3a: El sistema vuelve al primer nivel al llegar al máximo (ciclo circular).

Estos casos de uso contribuyeron a validar los requisitos funcionales y sirvieron como guía para el desarrollo del firmware y la interfaz de usuario, los cuales se detallan en el Capítulo 3.

2.3 Descripción de tecnologías y componentes utilizados

El trabajo realizado integró diversos componentes de hardware y herramientas de software que resultaron fundamentales para el desarrollo del sistema, pero que no fueron diseñados ni implementados por el equipo. A continuación se describen brevemente aquellos elementos que tuvieron un rol central en el funcionamiento del sistema, con el objetivo de ofrecer al lector el contexto necesario para comprender las decisiones de diseño adoptadas.

2.3.1 Microcontrolador STM32F103RB

El sistema fue desarrollado sobre el microcontrolador STM32F103RB, basado en la arquitectura ARM Cortex-M3, el cual se muestra en la Figura 2.3.1.1. Este dispositivo forma parte de la serie STM32F1 de STMicroelectronics, y se caracteriza por su bajo consumo, alta eficiencia y disponibilidad de periféricos integrados. Posee conversores analógico-digital (ADC), temporizadores, interfaces de comunicación I2C (*Inter-Integrated Circuit*), SPI (*Serial Peripheral Interface*) y USART (*Universal Synchronous/Asynchronous Receiver-Transmitter*), y pines GPIO (*General-Purpose Input/Output*) configurables como entrada o salida.

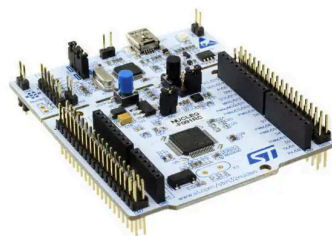


Figura 2.3.1.1: Microcontrolador STM32F103RB

2.3.2 Pantalla LCD (*Liquid Crystal Display*) 2x16

Se utilizó una pantalla alfanumérica LCD de 2 líneas por 16 caracteres, basada en el controlador HD44780 presente en la Figura 2.3.2.1. Este módulo se comunicó con el microcontrolador mediante un bus paralelo de 4 bits, utilizando pines de propósito general (GPIO). Para el control del LCD (*Liquid Crystal Display*) se utilizó una biblioteca de código en C desarrollada por terceros, basada en el controlador HD44780. Esta biblioteca permite operar el display en modo de 4 bits utilizando pines *GPIO*, y proporciona funciones básicas como inicialización, escritura de caracteres, control de cursor y limpieza de pantalla. La comunicación se realizó por interfaz paralela directa y no requirió el uso de periféricos dedicados del microcontrolador. Esta pantalla fue utilizada para mostrar el modo de operación, la velocidad de la cinta, mensajes de advertencia y otros datos relevantes para el usuario.



Figura 2.3.2.1: LCD 2x16

2.3.3 Memoria EEPROM externa AT24C256

Se utilizó un módulo de memoria EEPROM externo AT24C256, el cual se muestra en la Figura 2.3.4.1, que se comunica con el microcontrolador mediante el protocolo I2C (Inter-Integrated Circuit). Esta memoria, de tipo serial, cuenta con una capacidad de 256 kilobits (32 kilobytes) y permite almacenar datos de forma persistente, incluso después de apagar el sistema.

El módulo se conectó a través de las líneas SDA (datos) y SCL (reloj), y permitió guardar parámetros configurables por el usuario, como el modo de operación y la velocidad de la cinta transportadora. Para acceder a esta memoria, se utilizaron funciones implementadas desde el firmware, aprovechando la interfaz I2C del microcontrolador STM32F103RB.

La elección del módulo AT24C256 se debió a su bajo costo, compatibilidad con sistemas de 3.3 V, amplia disponibilidad en el mercado y facilidad de integración con

plataformas de desarrollo como STM32. Además, su capacidad fue suficiente para el tipo y volumen de datos requeridos por el sistema. Este componente es ampliamente utilizado en sistemas embebidos y cuenta con buena documentación y soporte en comunidades de desarrollo.

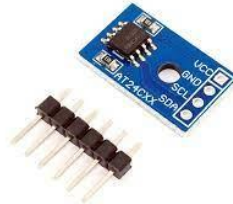


Figura 2.3.3.1: Memoria EEPROM externa AT24C256

2.3.4 Plataforma de desarrollo STM32CubeIDE

La herramienta STM32CubeIDE fue utilizada como entorno de desarrollo para programar y depurar el *firmware* del sistema. Esta plataforma incluyó funciones de generación de código automático para la inicialización de periféricos, así como utilidades de simulación, análisis de uso de CPU y depuración en tiempo real. La posibilidad de configurar periféricos desde una interfaz gráfica facilitó la implementación modular del sistema.

2.3.5 Protocolo de comunicación I2C

El protocolo I2C (Inter-Integrated Circuit) fue empleado para establecer la comunicación entre el microcontrolador y la memoria EEPROM externa. Este protocolo permite la transferencia de datos en forma serial utilizando solo dos líneas de comunicación (SDA(*Serial Data Line*) y SCL(*Serial Clock Line*)), lo cual simplificó el cableado del sistema. Se utilizó la implementación HAL (*Hardware Abstraction Layer*) proporcionada por STM32CubeIDE, facilitando así el control de la comunicación sin necesidad de programar a bajo nivel.

2.3.6 Periféricos básicos de entrada/salida: botones y LEDs

Se utilizaron botones pulsadores tipo *push-button* mostrado en la Figura 2.3.7.1, conectados a entradas digitales del microcontrolador para permitir la interacción del usuario con el sistema. Estos botones fueron empleados para cambiar el modo de operación, ajustar la velocidad, simular fallas y navegar por el menú interactivo. Para evitar lecturas erróneas, las entradas fueron filtradas por software utilizando técnicas de "*debouncing*".

También se incorporaron LEDs de color para indicar visualmente el estado del sistema: modo de operación activo, nivel de velocidad seleccionado y condición de falla. Los LEDs, mostrados en la Figura 2.3.7.2, fueron conectados a salidas digitales del microcontrolador y actualizados periódicamente desde el *firmware* como parte del bucle principal de ejecución.



Figura 2.3.6.1: Boton pulsador



Figura 2.3.6.2: Indicadores LEDs

CAPÍTULO 3

Diseño e implementación

3.1 Hardware del Sistema de Control

Se espera que se incluya un desarrollo similar al presentado en el Example 12-3 del libro. A modo de ejemplo se sugiere considerar la Figura 12.1 y el análisis que se realiza en el libro a partir de ella. Este análisis incluye por ejemplo a las Tablas 12.10 y 12.11 y a la Figura 12.2.

Es deseable encontrar también un diagrama como el de la Figura 12.3 y un conjunto de tablas similar al de las Tablas 12.2 a 12.7.

También se considera importante incluir una tabla similar a la Tabla 12.8.

En definitiva, la idea de esta sección es resaltar los problemas encontrados, los criterios utilizados y la justificación de las decisiones.

3.2 Firmware del Sistema de Control

Se espera que se incluya un desarrollo similar al presentado en el Example 12-4 del libro. A modo de ejemplo se sugiere considerar las Figuras 12.4, 12.5 y 12.6, y el análisis que se realiza en el libro a partir de ellas, lo que incluye por ejemplo una tabla similar a la Tabla 12.19.

A partir de esto sería bueno incluir un detalle similar al presentado en las Tablas 12.20 a 12.32.

También se espera encontrar diagramas de las máquinas de estado en forma similar a la Figura 12.7, junto con la explicación correspondiente, tal como se hace en el libro.

Si se realizó el diseño de pantallas se recomienda incluir el detalle correspondiente, tal como en las Figuras 12.8 a 12.11.

La idea de esta sección es resaltar los problemas encontrados, los criterios utilizados y la justificación de las decisiones.

Es esperable que haya fragmentos del código que sean relevantes. Para eso se recomienda utilizar un formato como el del Código 12.1, con las líneas numeradas y con una explicación en donde se haga uso de la numeración de las líneas. Por ejemplo: “en las líneas 1 a 3 tal cosa, mientras que en las líneas 5 a 9 tal otra”. Es importante que quede claro que no es necesario presentar en este documento todas las líneas de código, sino solamente aquellas que sea de interés explicar por alguna razón en particular.

Finalmente, se recomienda incluir el link al repositorio del proyecto, en el que se espera que haya un archivo Readme.md en el que se incluyan algunos fragmentos de este informe tal que permitan al usuario del repositorio entender en qué consiste el proyecto. Incluso se puede dejar este informe completo como anexo en el repositorio.

CAPÍTULO 5

Conclusiones

5.1 Resultados obtenidos

El desarrollo del sistema permitió poner en práctica de forma integral los conceptos aprendidos en la materia, consolidando tanto habilidades de diseño como de implementación. El resultado final fue un sistema eficiente y funcional, capaz de operar de manera segura y flexible en modos Normal y Set Up, lo que facilita su adaptación a diferentes escenarios de trabajo en la cinta transportadora.

Durante el proyecto, se abordaron con éxito desafíos técnicos relevantes, entre ellos la integración de múltiples dispositivos y la representación precisa de las tareas mediante diagramas de estado, lo que permitió obtener un diseño estructurado y optimizado. Un hito destacado fue la correcta programación y gestión de la memoria EEPROM, superando las dificultades propias de la comunicación y el almacenamiento persistente de datos, e implementando adecuadamente los protocolos requeridos.

Asimismo, se realizaron cálculos detallados, como la estimación del consumo eléctrico y el Worst-Case Execution Time (WCET), que brindaron un respaldo técnico sólido para validar el rendimiento del sistema bajo condiciones de operación normales. Estos análisis garantizaron que el sistema cumpla con los criterios de eficiencia, confiabilidad y seguridad establecidos al inicio del proyecto.

En síntesis, el trabajo desarrollado no solo cumplió los objetivos planteados, sino que también aportó una experiencia valiosa en el diseño y validación de sistemas embebidos, reforzando competencias clave para el ejercicio profesional.