

Unidad Profesional Interdisciplinaria de Ingeniería Campus Zacatecas



Secadora de filamento con sistema RTOS y base de datos

Julio Cesar Núñez Quiñones

Cesar Eduardo Montiel Torres

RTOS (Real Time Operating System)

Zacatecas, Zac. a 12 de enero del 2026.

Introducción

En la impresión 3D por deposición fundida (FDM), la calidad de las piezas finales depende directamente del estado del material. La mayoría de los filamentos técnicos (como el Nylon, TPU, PETG e incluso el PLA) son higroscópicos, lo que significa que absorben humedad del ambiente con facilidad. Esta humedad atrapada se expande y se convierte en vapor durante la extrusión, provocando defectos superficiales, baja adherencia entre capas, burbujas y fallos estructurales en la pieza.

El presente proyecto propone el desarrollo de una Secadora de Filamento Inteligente, la cual no solo regula la temperatura y la humedad de forma activa, sino que gestiona estas tareas mediante un RTOS (Real-Time Operating System). La implementación de un RTOS permite un control determinista de las tareas críticas (como el control PID de temperatura) mientras se ejecutan procesos secundarios. Además, la integración de una base de datos permite el almacenamiento de perfiles de secado específicos para diferentes materiales y el registro histórico del estado del filamento, optimizando el flujo de trabajo en entornos de fabricación digital.

Objetivos

Objetivo General

Diseñar e implementar un sistema avanzado de secado de filamento para impresión 3D, controlado por un Sistema Operativo en Tiempo Real (RTOS) y respaldado por una base de datos para la gestión eficiente de parámetros térmicos y materiales.

Objetivos Específicos

Implementar una arquitectura de software basada en RTOS para gestionar de forma simultánea el control de temperatura, la lectura de sensores y la interfaz de usuario sin retardos críticos.

Desarrollar un sistema de control de lazo cerrado (PID) para mantener la temperatura constante y estable según el tipo de material.

Diseñar una base de datos embebida o local para almacenar y consultar perfiles preconfigurados (PLA, ABS, PETG, etc.) y registrar logs de operación.

Reducir el porcentaje de humedad relativa dentro de la cámara de secado a niveles óptimos (generalmente <20%) mediante la gestión de flujo de aire y calor.

Marco teórico

Este proyecto se sustenta en la deshidratación de polímeros mediante calor controlado, atacando directamente la humedad retenida en sus cadenas moleculares. Para lograr una ejecución eficiente, se implementa un sistema operativo de tiempo real (RTOS) que organiza los procesos por prioridad, garantizando que el monitoreo de seguridad sea constante y preciso. La regulación térmica se confía a un algoritmo PID que minimiza el error de temperatura para proteger el filamento de posibles sobrecalentamientos. Por otro lado, una base de datos integrada permite automatizar el proceso mediante perfiles de materiales preconfigurados, apoyándose en la transferencia de calor por convección forzada para asegurar que el flujo de aire seco alcance todo el material de manera uniforme y efectiva.

Desarrollo

El desarrollo de este proyecto se centró en la creación de una secadora de filamento inteligente materiales como Nylon, TPU, PETG y PLA, los cuales absorben humedad del entorno afectando la calidad de la impresión 3D. El sistema se basa en la aplicación de calor controlado para romper los enlaces de hidrógeno y facilitar la evaporación del agua atrapada. Para gestionar este proceso con precisión y permitir la ejecución de múltiples tareas sin retardos, se implementó una arquitectura de software basada en un Sistema Operativo en Tiempo Real (RTOS).

- Implementación del Controlador Difuso: El núcleo lógico del sistema utiliza un controlador difuso para gestionar la respuesta térmica. Para la etapa de fusificación, se definió como variable de entrada el "Error de Temperatura" en un rango de -45 a 45. Se establecieron siete funciones de membresía para esta entrada: muy negativo, medio negativo, poco negativo, cero, poco positivo, medio positivo y muy positivo, utilizando tipos de funciones en L, triangulares, de impulso y en R.

- Gestión de Salidas y Reglas: Para la defusificación, se estableció como salida el "Voltaje del ventilador" en un rango de 0 a 12 voltios. Esta salida cuenta con cinco funciones de membresía: apagado, lento, medio, rápido y full. El comportamiento del sistema se rige por siete reglas de inferencia; por ejemplo, si el error de temperatura es "Cero", el voltaje del ventilador se sitúa en "Lento", mientras que si el error es "Muy positivo", el ventilador opera en modo "Full".
- Conectividad y Base de Datos: El proyecto integra una base de datos en la nube mediante Firebase Realtime Database para el monitoreo y almacenamiento de perfiles. Durante las pruebas, se validó la recepción de datos en tiempo real, registrando variables como una humedad del 36% y una temperatura de 18.4°C. La base de datos permite manejar distintos tipos de datos, incluyendo valores flotantes, enteros y cadenas de texto para identificar los estados del sistema.
- Hardware y Montaje Físico: El control principal recae en un microcontrolador ESP32. El equipo utilizado incluye una resistencia calefactora, un ventilador para la circulación de aire, un teclado matricial para la interacción del usuario y una pantalla LCD para la visualización de datos. La estructura física se construyó sobre una base de madera que contiene la cámara de secado y el soporte para los componentes electrónicos.
- Diagrama de Conexiones e Integración: La implementación electrónica unifica todos los componentes mediante un esquema que conecta el ESP32 a sensores de temperatura y humedad, un teclado matricial, una pantalla LCD y actuadores. Se utiliza un puente H para gestionar el ventilador y la calefacción, alimentado por una fuente de poder de 12V a 24V. Un relevador también forma parte del circuito para el control de la potencia de calefacción

Controlador difuso:

Funciones de membresía

Fusificación				
Entrada(s)				
Error de Temperatura. [-45,45]		ET		
No. FP	Etiqueta	Dominio	Tipo de FP	Parametros
1	Muy negativo	[-45,-21]	L	[-45,-45,-40,-21]
2	medio negativo	[-30,-12]	Triangular	[-30,-21,-12]
3	poco negativo	[-17,-1]	Triangular	[-17,-10,-1]
4	Cero	[0]	Impulso	[0,0,0]
5	poco positivo	[1,24]	Triangular	[1,17,24]
6	medio positivo	[20,47]	Triangular	[20,35,39]
7	Muy positivo	[39,60]	R	[32,40,45,45]

Figura 1. tabla de entradas.

Salida(s)				
Voltaje del ventilador [0,12]		VV		
No. FP	Etiqueta	Dominio	Tipo de FP	Parametros
1	Apagado	[0]	L	[0,0,5,6.5]
2	Lento	[2.4,4.6]	Triangular	[6.8,2.9.5]
3	Medio	[4,9]	Triangular	[8.8,10,11]
4	Rapido	[7,11.9]	R	[10,10.8,11.9,11.9]
5	Full	[12]	Impulso	[12,12,12]

Figura 2. tabla de salidas.

Reglas

No. Regla		Variable de entrada [ET]		Antecedente		Variable de salida		Consecuencia
1	IF	Error de temperatura	ES	Muy negativo	THEN	Voltaje ventilador VV	ES	Apagado
2	IF	Error de temperatura	ES	medio negativo	THEN	Voltaje ventilador VV	ES	Apagado
3	IF	Error de temperatura	ES	poco negativo	THEN	Voltaje ventilador VV	ES	Apagado
4	IF	Error de temperatura	ES	Cero	THEN	Voltaje ventilador VV	ES	Lento
5	IF	Error de temperatura	ES	poco positivo	THEN	Voltaje ventilador VV	ES	Medio
6	IF	Error de temperatura	ES	medio positivo	THEN	Voltaje ventilador VV	ES	Rapido
7	IF	Error de temperatura	ES	Muypositivo	THEN	Voltaje ventilador VV	ES	Full

Figura 3. tabla de reglas.

Gráfica de las funciones de entrada

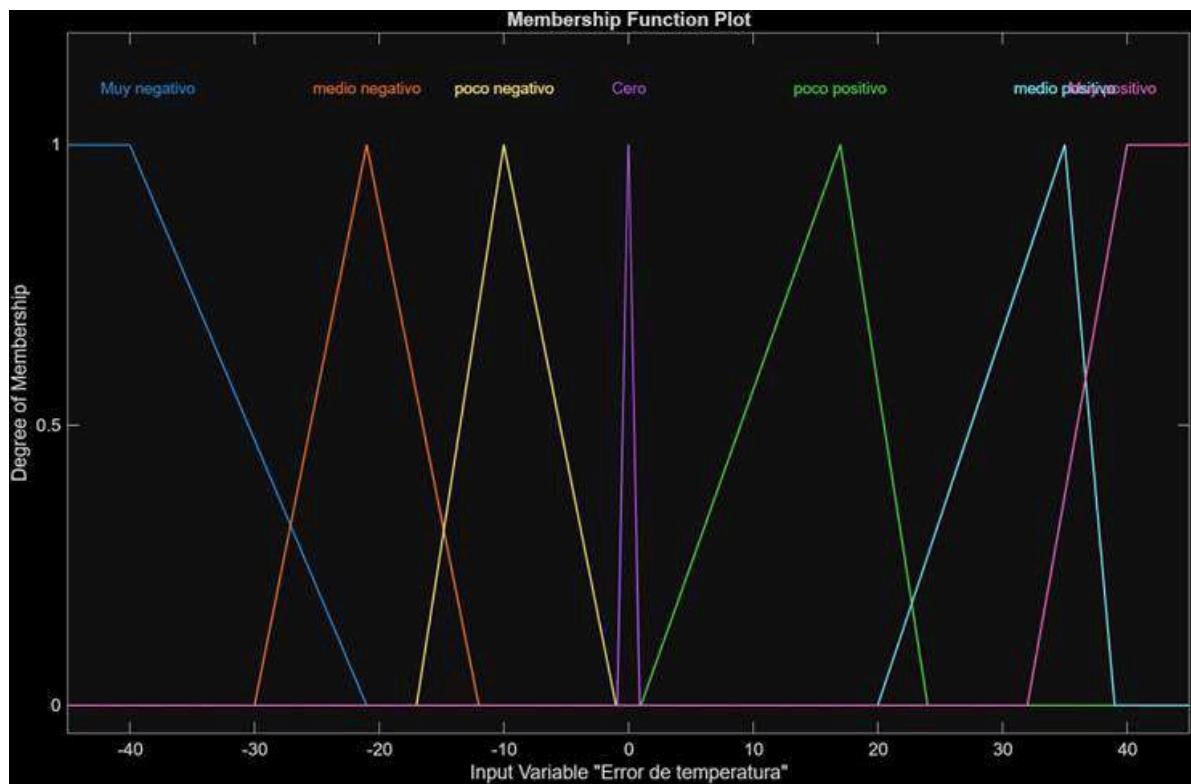


Figura 4. Gráfica de las funciones de entrada.

Gráfica de las funciones de salida

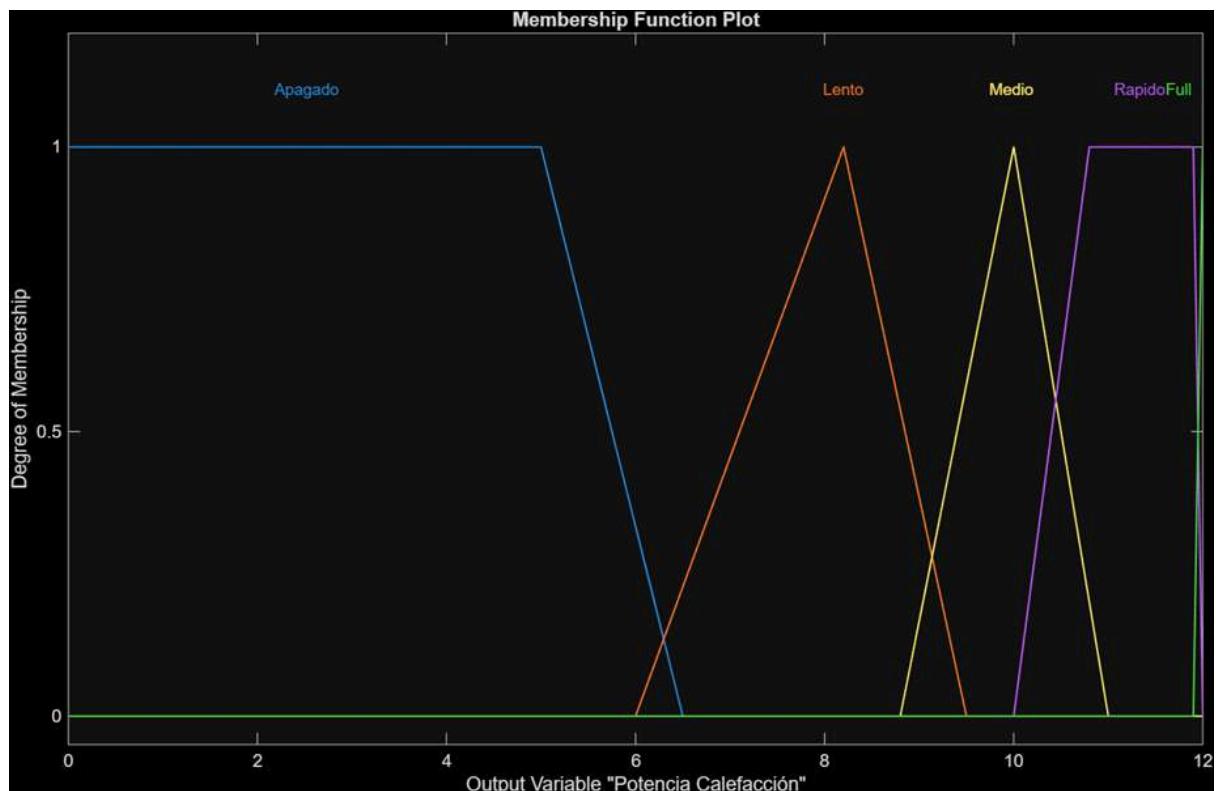


Figura 5. Gráfica de las funciones de salida.

Implementación:



Figura 6. Material y equipo que se ocupó



Figura 7. Base del secador



Figura 8. Conexiones iniciales.

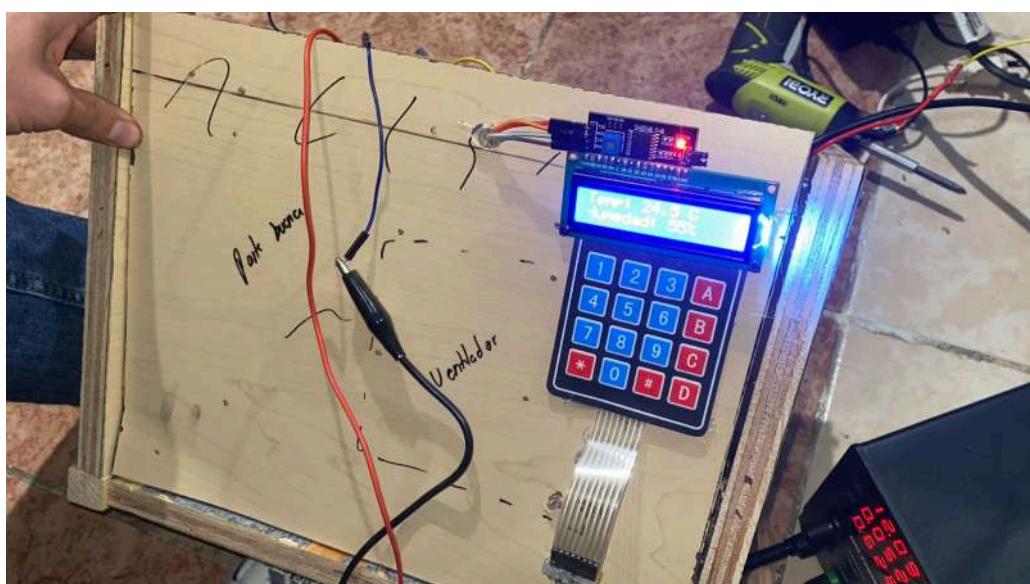


Figura 9. conexiones al teclado matricial y a la pantalla

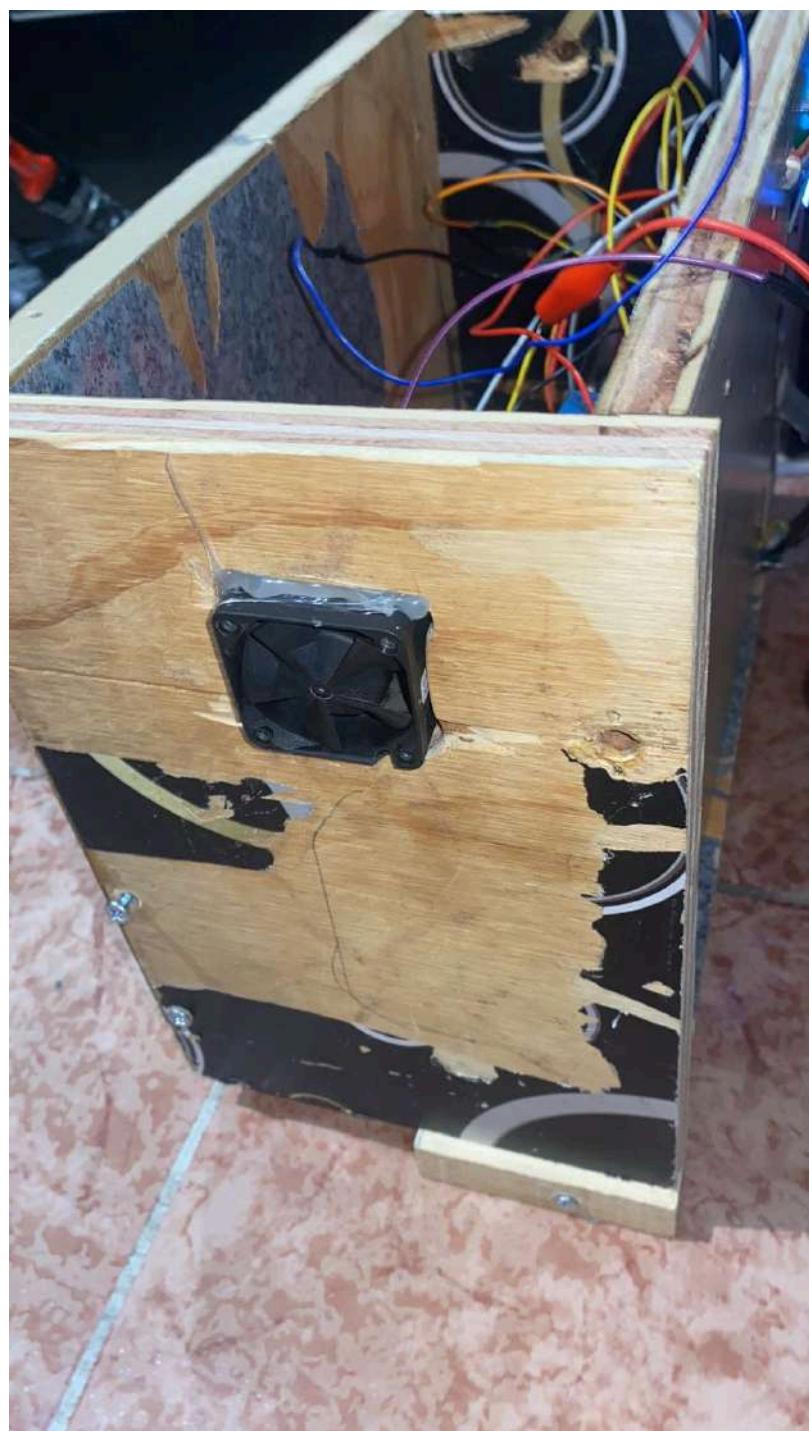


Figura 10. Ventilador de secador.



Figura 11. Resistencia.

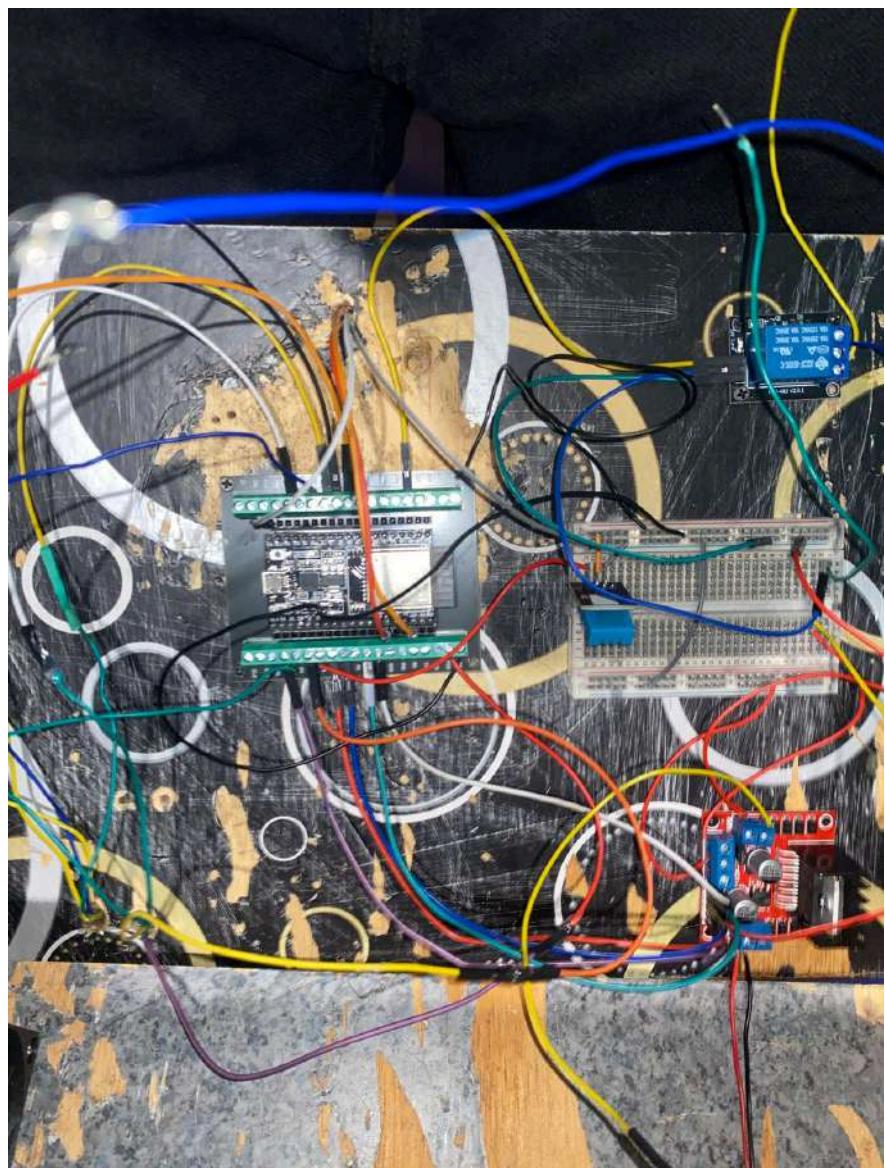


Figura 12. Conexiones finales.

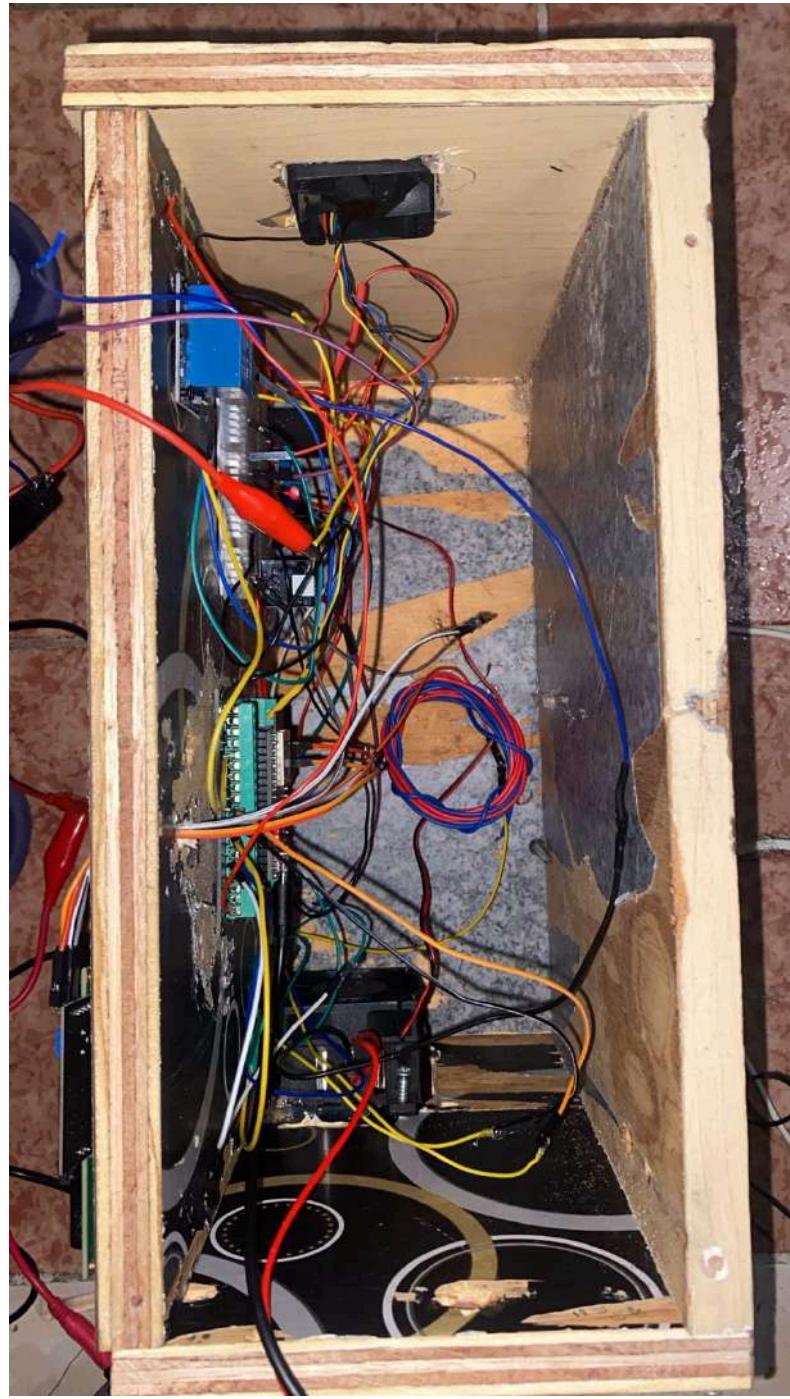


Figura 13. Conexiones ya en prototipo

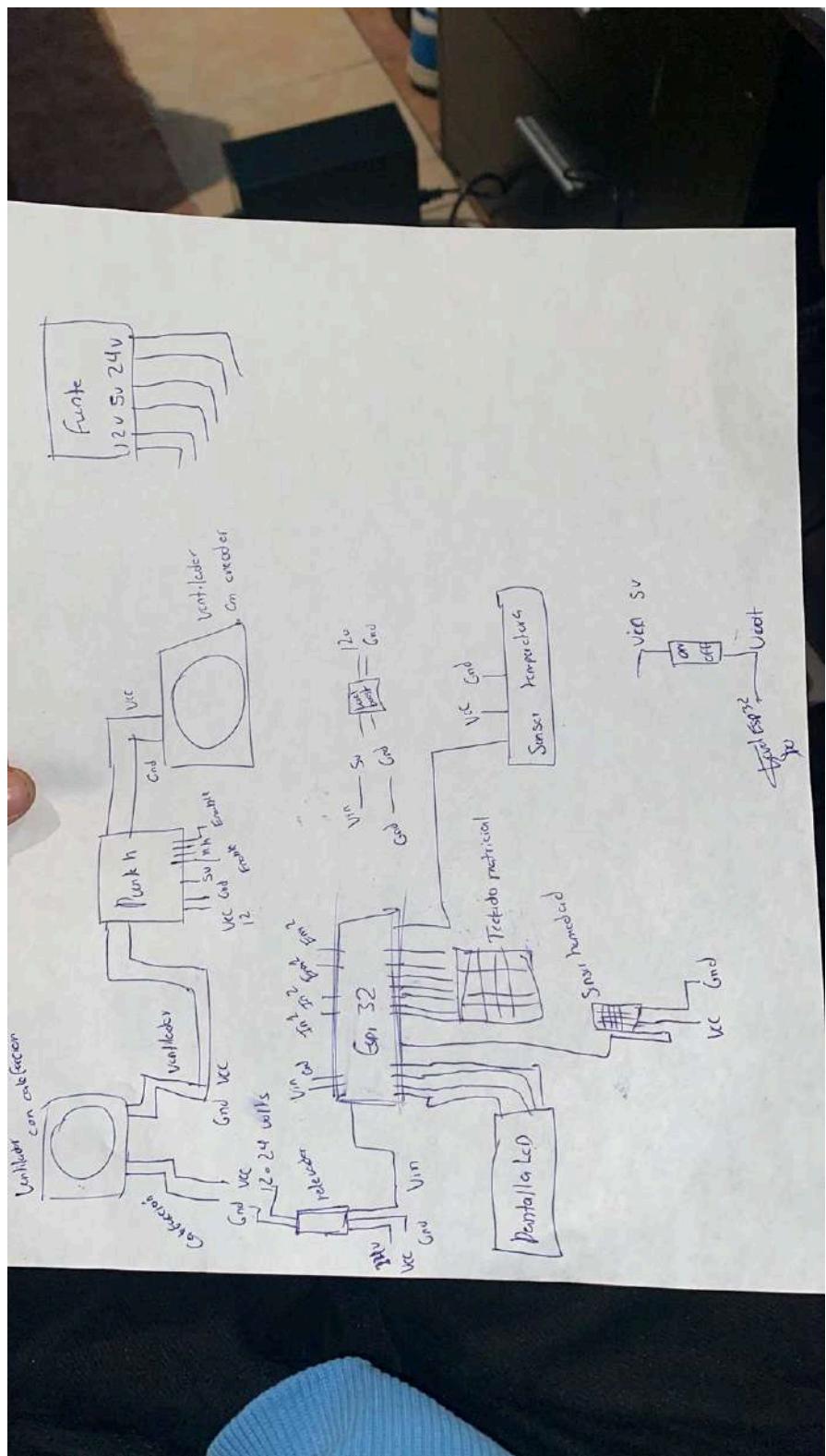
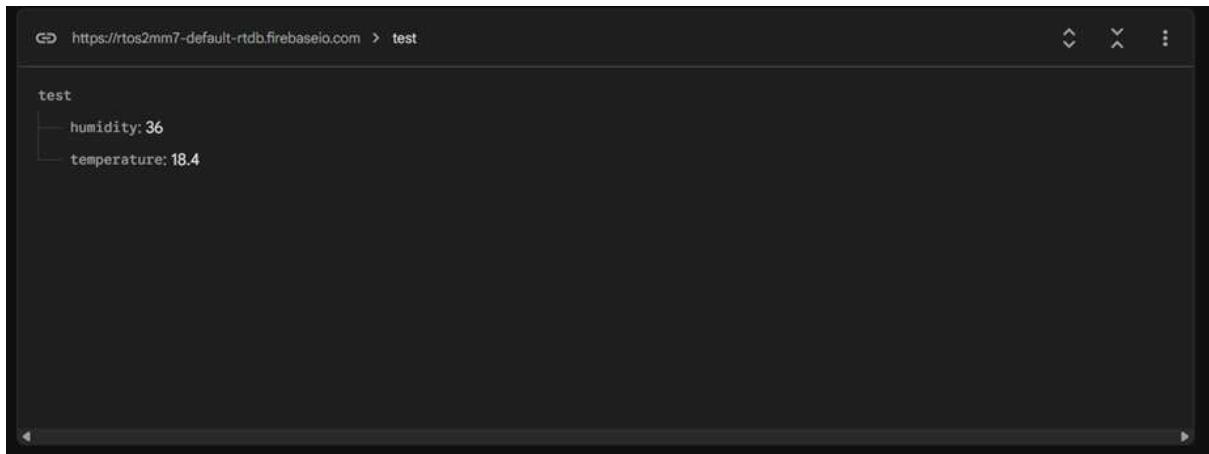


Figura 14. Diagrama de implementación

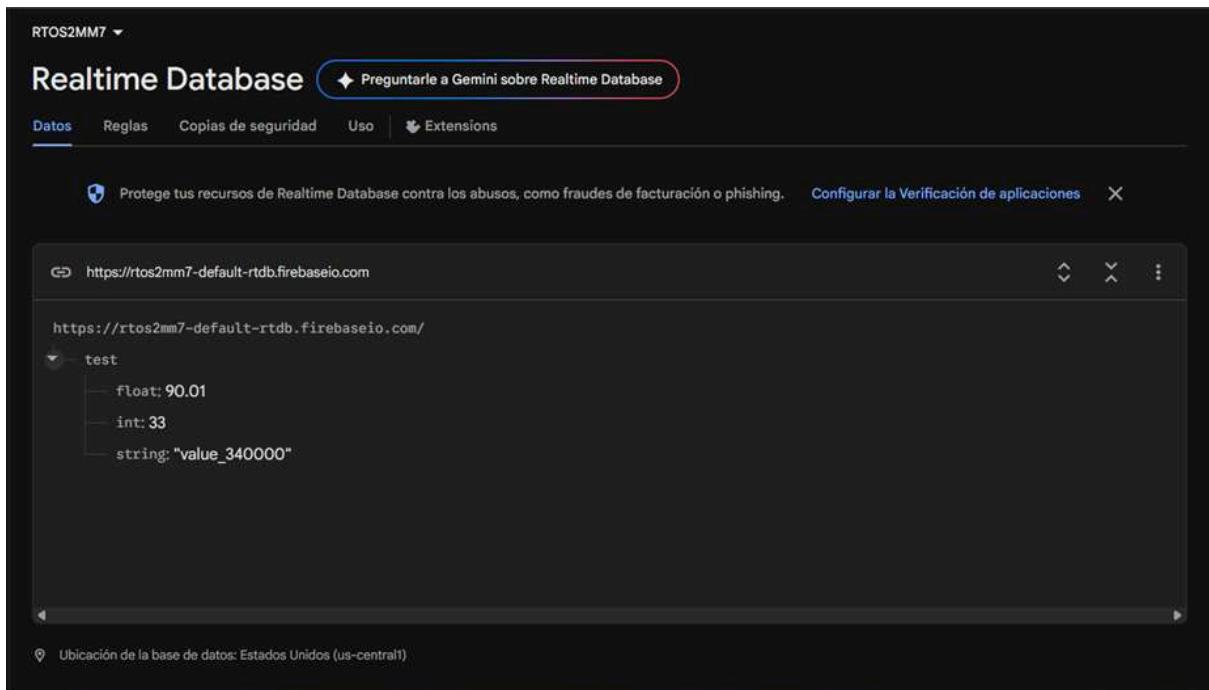
Resultados

Datos recibidos en la base de datos:



```
test
  humidity: 36
  temperature: 18.4
```

Figura 15. datos recibidos en la base



RTOS2MM7 ▾

Realtime Database

◆ Preguntarle a Gemini sobre Realtime Database

Datos Reglas Copias de seguridad Uso Extensions

🛡 Protege tus recursos de Realtime Database contra los abusos, como fraudes de facturación o phishing. Configurar la Verificación de aplicaciones X

https://rtos2mm7-default-rtbd.firebaseio.com/

test

- float: 90.01
- int: 33
- string: "value_340000"

Ubicación de la base de datos: Estados Unidos (us-central1)

Figura 16. datos recibidos en la base

Los resultados obtenidos durante la fase de experimentación y pruebas finales del prototipo revelaron desafíos técnicos significativos que impidieron alcanzar la sinergia total entre los módulos del sistema. Aunque de manera individual se lograron hitos importantes, la integración simultánea de la conectividad a la nube y el algoritmo de control no operó de la forma esperada bajo el entorno de tiempo real.

Análisis de la Integración Firebase-Control: A pesar de que la Realtime Database de Firebase mostró una recepción exitosa de variables aisladas como humedad y temperatura , se presentó una falla crítica al intentar ejecutar el controlador difuso de manera concurrente. La latencia generada por las peticiones de red hacia el servidor de Firebase en Estados Unidos parece haber interferido con los tiempos deterministas exigidos por el RTOS para el cálculo de las reglas de inferencia. Esto provocó que el procesador priorizara la comunicación sobre la actualización del voltaje del ventilador, resultando en una inestabilidad térmica que invalidó la precisión del control PID y difuso diseñado originalmente.

Evaluación de las Funciones de Control: El diseño de las funciones de membresía para el error de temperatura y el voltaje del ventilador fue técnicamente robusto en el papel. Sin embargo, al momento de la implementación física en el ESP32, el sistema no logró realizar la transición suave entre etiquetas como "medio positivo" y "muy positivo" de manera estable mientras la tarea de conectividad estaba activa. La lógica de las siete reglas programadas para el voltaje del ventilador se vio interrumpida por desbordamientos de memoria o conflictos de prioridad en el programador de tareas del RTOS, lo que impidió que el actuador respondiera en tiempo real a las fluctuaciones del sensor.

Estado del Hardware y Conexiones: Desde una perspectiva objetiva, el montaje físico y la infraestructura de hardware se completaron satisfactoriamente según los planos de implementación. Se logró la correcta conexión del teclado matricial, la pantalla LCD y los actuadores de potencia a través del puente H. No obstante, la complejidad del cableado final y la falta de un filtrado de ruido más estricto en la alimentación de 12V-24V pudieron haber contribuido a lecturas erráticas que dificultaron aún más la estabilización del lazo de control en condiciones de carga completa.

Posibles Soluciones y Optimización: Para solventar la falta de sincronización entre el control y la base de datos, una solución técnica viable sería la implementación de

un buffer de datos local que almacene las lecturas y las envíe en ráfagas durante los tiempos muertos del procesador, reduciendo la carga sobre la tarea de control crítico. Asimismo, es imperativo realizar un ajuste fino en las prioridades de las tareas del RTOS, otorgando al algoritmo de control una prioridad absoluta y delegando el stack de red de Firebase a un núcleo secundario del ESP32 para evitar colisiones de tiempo. Finalmente, la migración de un motor de base de datos pesado a un protocolo más ligero como MQTT podría disminuir el overhead de procesamiento, permitiendo que el sistema mantenga el determinismo necesario para proteger la integridad del filamento

Conclusiones

Las conclusiones derivadas de la ejecución de este proyecto permiten realizar una evaluación crítica sobre la complejidad de integrar sistemas de control de lazo cerrado con servicios en la nube bajo un entorno de tiempo real. El desarrollo cumplió satisfactoriamente con la construcción física y la programación individual de los módulos, pero enfrentó limitaciones fundamentales durante la etapa de integración sistémica.

Se concluye que el diseño del hardware y la lógica del controlador difuso fueron técnicamente sólidos, logrando una implementación física funcional que incluía la correcta operación del teclado matricial, la pantalla LCD y la etapa de potencia. La estructura del gabinete de madera y la disposición de los componentes permitieron validar la viabilidad mecánica de la cámara de secado.

Una apreciación fundamental es que el determinismo del RTOS se vio comprometido por la naturaleza no determinista de las peticiones de red hacia Firebase. La comunicación con servidores remotos interfirió con los ciclos críticos del algoritmo de control, lo que demuestra que en sistemas embebidos de alto rendimiento, las tareas de conectividad deben estar rígidamente aisladas de las tareas de control térmico para evitar la inestabilidad del sistema.

Se observó que el uso simultáneo de un stack de red pesado y un controlador de lógica difusa complejo puede derivar en conflictos de prioridad y desbordamientos

de memoria en el ESP32. Esto impidió que el actuador respondiera con la suavidad esperada a las variaciones de los sensores, invalidando la precisión del control térmico cuando la transmisión de datos estaba activa.

Como conclusión final, el proyecto identifica que el éxito de una secadora inteligente depende de una gestión de recursos más eficiente, sugiriendo la migración a protocolos más ligeros y el uso de procesamiento de doble núcleo para separar las tareas críticas de las de monitoreo.

Anexos

Video:

<https://youtube.com/shorts/jkL0ZLf1e3E?si=tCU5yq7q-WjTt53q>

Firebase :

<https://console.firebaseio.google.com/project/rtos2mm7/database/rtos2mm7-default-rtdb/data>

GitHub :

<https://github.com/JulioCesarNunez/RTOS>

Referencias

<https://www.mdpi.com/2227-7080/13/8/376>

<https://pmc.ncbi.nlm.nih.gov/articles/PMC11442157/>

<https://www.mdpi.com/2227-7080/13/2/83>

https://www.researchgate.net/publication/397214496_IOT-BASED_ENVIRONMENTAL_CONTROL_IN_3D_PRINTER_ENCLOSURES

<https://www.inspiro.nl/en/how-to-implement-real-time-operating-system-rtos-in-embedded-projects/>

http://www.ijmrset.com/upload/67_REAL-TIME%20BI-DIRECTIONAL%20CLOUD-SYNCED%20RELAY%20CONTROL.pdf