Universidad Nacional de Cuyo

Facultad de Ingeniería

Lic. en Ciencias de la Computación

Introducción a la Electrónica

Informe Final

Cairo, Lucía - 13030

Corral, Rocío - 13199

Sanchez Lanza, Agustina - 11549

Noviembre 2024

1. Resumen ejecutivo

Este proyecto consiste en el diseño y desarrollo de un sistema de control de una estufa eléctrica utilizando una placa ESP32. El sistema integra un sensor que detecta la temperatura ambiente y calidad del aire, y actúa en consecuencia para activar o desactivar una estufa. Además, se integra con una plataforma web que permite controlar la estufa y leer las mediciones de forma remota, ofreciendo mayor comodidad y seguridad al usuario.

1.1. Objetivo del Sistema

El objetivo principal del sistema es crear un ambiente con una temperatura estable de forma artificial, mediante el uso de sensores y una estufa, que garanticen un espacio seguro y controlado para el usuario.

El objetivo técnico es que una estufa eléctrica se active automáticamente cuando un sensor de temperatura detecte un descenso de la temperatura por debajo de un umbral predefinido, para luego apagarse automáticamente al alcanzar la temperatura ambiente deseada. A su vez, el sistema desarrollado debe ser controlable de forma remota desde cualquier lugar con acceso a internet, para supervisar las lecturas del sensor y prender o apagar la estufa manualmente, brindando control remoto para situaciones imprevistas.

1.2. Componentes Principales

- Microcontrolador ESP32: es el microcontrolador principal encargado de gestionar la lectura de los sensores, el control de activación de la estufa y la comunicación con la plataforma web. La conectividad WiFi del ESP32 permite la comunicación con la web y la operación remota.

- Sensor BME680: es un sensor 4 en 1 que mide humedad, presión, temperatura y calidad del aire con un consumo de energía bajo y estabilidad a largo plazo. Se utiliza para medir la temperatura ambiente para determinar la activación de la estufa.

- Estufa eléctrica: es el dispositivo de calefacción que se activa o desactiva según las lecturas del sensor de temperatura.

- Plataforma web: es la interfaz de usuario que permite la supervisión y control del sistema de forma remota. A través de la plataforma, los usuarios pueden visualizar en tiempo real las mediciones del sensor, ajustar los umbrales de temperatura y activar o desactivar la estufa según sea necesario.

2. Antecedentes

La necesidad de mejorar la eficiencia energética y la seguridad en el hogar ha impulsado el desarrollo de sistemas inteligentes de control de dispositivos eléctricos. Entre estos, las estufas eléctricas representan un desafío particular debido a la naturaleza de su funcionamiento y el riesgo asociado al control manual. En sistemas de calefacción tradicionales, el ajuste manual a menudo genera un uso ineficiente de energía, además de aumentar los riesgos al olvidar apagar el dispositivo en momentos innecesarios. Ante esta problemática, el uso de Internet de las Cosas (IoT) permite desarrollar sistemas automatizados que no solo controlen estos dispositivos de forma autónoma, sino que además faciliten el monitoreo y control remoto.

La integración de sensores ambientales en sistemas de IoT ha avanzado considerablemente, permitiendo medir variables clave como temperatura, humedad y calidad del aire. Estos sensores, en combinación con microcontroladores económicos pero prácticos como el ESP32, posibilitan el desarrollo de sistemas de control precisos, adaptables a diversos entornos y capaces de optimizar el uso de dispositivos eléctricos. El ESP32, al contar con conectividad WiFi y múltiples pines de entrada y salida, es una opción ideal para integrar sensores y actuadores en un sistema autónomo.

Asimismo, la tecnología blockchain, reconocida principalmente por su aplicación en criptomonedas, ha encontrado una utilidad emergente en el control y trazabilidad de datos en sistemas de IoT. Gracias a su estructura distribuida e inmutable, blockchain puede garantizar la integridad y trazabilidad de la información almacenada, lo que permite confiar en que los datos y comandos del sistema no sean manipulados de forma indebida. Al combinar IoT con blockchain, es posible asegurar que los dispositivos respondan a condiciones predefinidas de manera segura, y que toda acción quede registrada para su verificación posterior.

A su vez, en la actualidad es posible encontrar múltiples herramientas low-code para facilitar la integración de sistemas de monitoreo y control web con dispositivos IoT sin necesidad de escribir grandes cantidades de código de bajo nivel para este fin.

Este proyecto integra IoT y una plataforma cloud para crear un sistema de control de estufa eléctrica que ofrece autonomía y seguridad.

3. Objetivos

Como se mencionó anteriormente, el objetivo principal del proyecto fue desarrollar un sistema de control automatizado y remoto para una estufa eléctrica, usando un microcontrolador ESP32 junto con un sensor de calidad del aire y temperatura (BME680). La finalidad fue mantener la temperatura ambiente estable y segura en espacios interiores (principalmente en una habitación de un hogar), activando la estufa cuando la temperatura desciende por debajo de un umbral predefinido.

En este contexto, el sistema creado no solo permite la automatización del encendido y apagado de la estufa mediante un relé, sino que también habilita su control remoto a través de una aplicación de la plataforma Blynk. A través de esta aplicación, el usuario puede monitorear en tiempo real los niveles de temperatura y calidad del aire y también puede ajustar el funcionamiento de la estufa desde cualquier lugar con acceso a Internet. Esto último, permite una solución cómoda para el usuario, maximizando la eficiencia energética y disminuyendo riesgos con el control manual en escenarios inesperados o atípicos.

A su vez, el proyecto contempla la integración de indicadores LED para brindar retroalimentación visual sobre el estado de la conexión WiFi y el funcionamiento de la estufa.

En resumen, los objetivos específicos son:

- Automatizar el encendido y apagado de la estufa basado en la temperatura ambiente medida por el sensor BME680.
- Evaluar continuamente la temperatura y la calidad del aire.
- Permitir al usuario el control y monitoreo remoto del sistema mediante la plataforma Blynk.
- Ofrecer un sistema confiable y seguro mediante el uso de indicadores visuales.

4. Descripción de la solución

La solución propuesta consiste en un sistema embebido, cuya integración se explica en esta sección. En la Figura 1 se puede visualizar la arquitectura de componentes de la solución y en las siguientes subsecciones se detalla la integración de componentes de hardware y software que permiten medir, comunicar y responder a condiciones del entorno de forma dinámica para la implementación de todo el sistema.

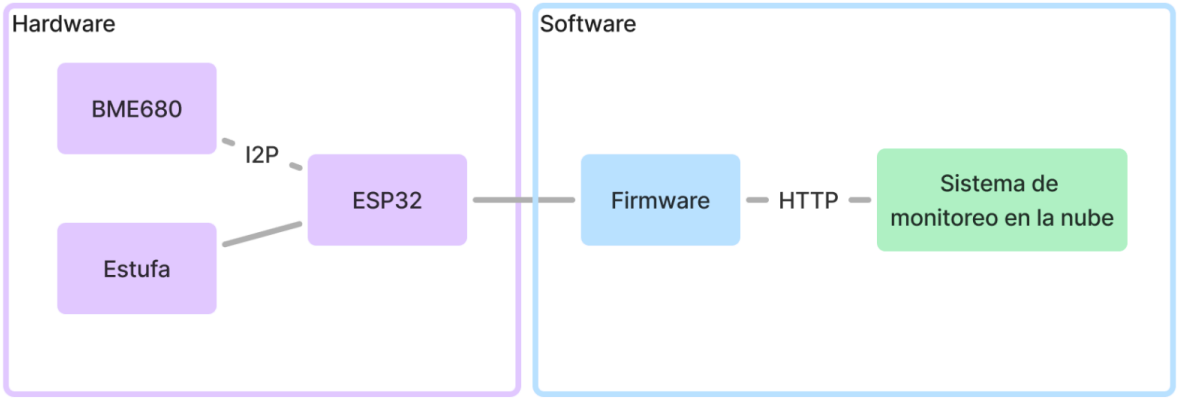

Figura 1. Arquitectura del sistema

4.1. Hardware

El componente central de procesamiento del sistema es una placa ESP32, que tiene la capacidad de manejar las lecturas que provee el sensor ambiental, controlar el dispositivo

externo (la estufa) a través de un relé, conectarse a la red WiFi y enviar datos a la plataforma externa en la nube. Todo esto conforma el sistema que permite la supervisión y el control de la estufa. A continuación, se detallan estos componentes de hardware de forma específica.

Microcontrolador ESP32

Como el microcontrolador es el núcleo del sistema, se seleccionó la placa ESP32 debido a su capacidad para manejar varias funciones esenciales de manera simultánea. Este microcontrolador dispone de un procesador dual-core, múltiples pines GPIO, y conectividad WiFi, lo que permite una comunicación eficiente tanto con los sensores y leds, como con la red. Otra característica clave del ESP32 es su bajo consumo de energía que es crucial para el uso continuo del sistema sin requerir grandes fuentes de alimentación.

La configuración y programación del ESP32 se realizó a través de Arduino IDE, utilizando las bibliotecas específicas para la comunicación con el sensor y la conexión WiFi.

Sensor BME680

BME680 es un componente avanzado que permite monitorear cuatro parámetros ambientales en simultáneo: temperatura, humedad, presión atmosférica, y calidad del aire (VOCs).

En el sistema, el BME680 es el encargado de enviar datos que el ESP32 interpreta para activar o desactivar la estufa. El sensor se conecta al ESP32 a través del protocolo I2C que permite una conexión sencilla y confiable, requiriendo solo dos cables de datos (SDA y SCL) además de la alimentación y tierra, lo cual es ideal para ahorrar espacio y pines GPIO en el microcontrolador.

Relé para Control de la Estufa Eléctrica

El relé funciona como un interruptor controlado eléctricamente que permite activar y desactivar el flujo de corriente hacia la estufa. El ESP32 controla el relé enviándole una señal cuando la temperatura medida cae por debajo del umbral establecido.

Este componente es capaz de manejar la corriente y el voltaje necesarios para alimentar la estufa y proporciona seguridad al usuario al aislar el circuito de alta potencia de la estufa del circuito de baja potencia del microcontrolador.

La conexión del relé al ESP32 se realiza utilizando uno de los pines GPIO del ESP32 para activar o desactivar el relé, y asegurando que la estufa funcione de acuerdo con las condiciones de temperatura deseadas.

LEDs Indicadores

Para complementar el sistema, los LEDs ofrecen una retroalimentación visual sobre el estado de este. Esto permite al usuario tener una referencia rápida sobre el funcionamiento

del sistema sin necesidad de monitorear la app en todo momento. Se utilizaron dos pares de LEDs de doble color (verde y rojo):

- LED de Conexión a WiFi: Este LED indica el estado de la conexión WiFi del sistema. Si la conexión es exitosa, el LED se ilumina en verde; si hay problemas de conexión, el LED cambia a rojo.
- LED de Estado de la Estufa: Muestra si la estufa está encendida o apagada, proporcionando una indicación visual local de su estado. Cuando el sistema activa la estufa, el LED verde se enciende; cuando la estufa está apagada, se enciende el LED rojo.

Estos LEDs están conectados a diferentes pines GPIO del ESP32 y se controlan en función de las señales enviadas desde el microcontrolador en base a las condiciones ambientales detectadas.

Fuente de Alimentación y Reguladores de Voltaje

La alimentación del sistema es elemental para su funcionamiento. El ESP32 y los componentes conectados requieren una fuente de alimentación estable. Se utiliza una fuente de 5V que se regula para proporcionar los 3.3V necesarios para el ESP32 y el sensor BME680. Los reguladores de voltaje garantizan que cada componente recibe el voltaje adecuado y previenen daños en el sistema debido a variaciones de corriente.

En conjunto, la elección de estos componentes permite un sistema eficiente, seguro y accesible para el usuario, que optimiza el control de la estufa eléctrica y proporciona un entorno seguro, cómodo y monitoreado.

La estufa eléctrica está conectada directamente a la red eléctrica y su activación está mediada por el relé.

4.1.1 Diagrama de bloque

En la Figura 2 se muestra un diagrama de bloque que muestra la interconexión de los componentes mencionados:

- El sensor BME680 conectado al ESP32 a través del protocolo I2C.
- El relé controlando la alimentación de la estufa.
- LEDs indicadores conectados al ESP32 para proporcionar retroalimentación visual.
- La conexión del ESP32 al WiFi y la interacción con la aplicación cloud.

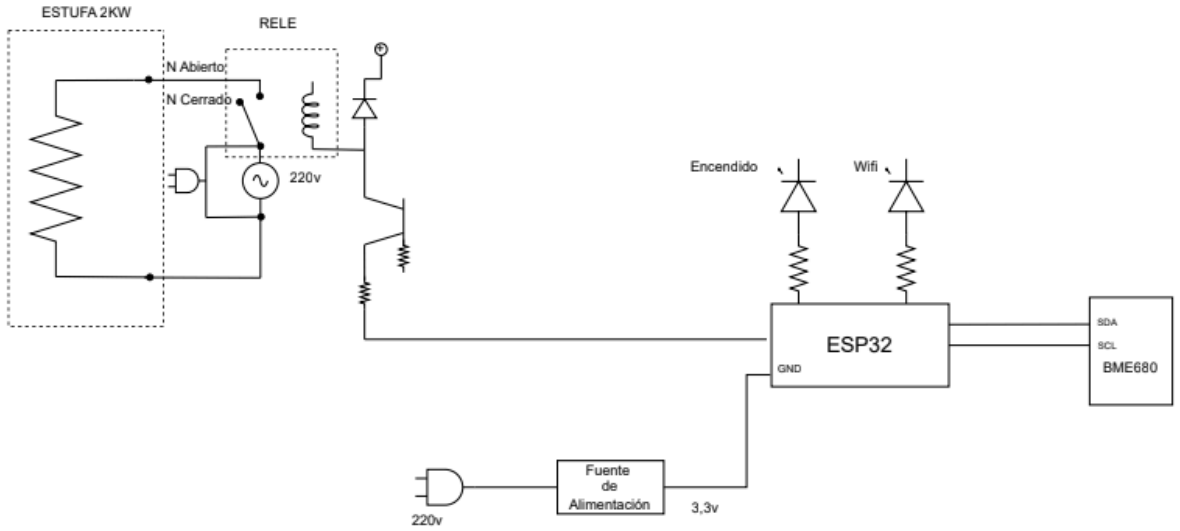

Figura 2. Diagrama de bloque del hardware del sistema

4.2. Firmware

El firmware es la pieza de software que permite el control y funcionamiento del sistema a nivel de hardware, permitiendo que el ESP32 gestione tanto los sensores como los actuadores, la comunicación WiFi y la interacción con el usuario a través de la aplicación cloud mediante el protocolo HTTP. Está desarrollado en C usando el entorno Arduino IDE, y su diseño modular facilita el control sobre cada componente del sistema. A continuación, explicamos la estructura y funcionamiento de los bloques que conforman el firmware.

Inicialización y Configuración del Sistema

En la función setup() preparamos y configuramos el sistema al inicio de su operación, realizando lo siguiente:

- Configuración de Pines GPIO: Definimos los pines GPIO que controlarán los actuadores (relé y LEDs) y configuramos los pines de entrada/salida.

- Conexión a la red WiFi: Utilizando las credenciales correspondientes, el ESP32 intenta conectarse a la red WiFi. Esta conexión es esencial para que el sistema pueda enviar datos a la aplicación cloud y recibir comandos del usuario de forma remota. Mientras se intenta establecer la conexión, el pin de salida se enciende de color rojo, se configuró pin para cambiar el color de LED a verde una vez que la conexión se establece de manera exitosa. En caso de fallo en la conexión, el sistema intenta restablecer la conexión periódicamente.

- Inicialización del Sensor: Haciendo uso de la biblioteca *Zanshin_BME680.h* se maneja el sensor y configuramos sus parámetros de muestreo para obtener datos de temperatura, humedad, presión atmosférica y calidad del aire. También definimos la interfaz de comunicación I2C, que permite la integración con el ESP32.

- Conexión con Blynk: La función Blynk.begin() inicializa la conexión con la plataforma en la nube Blynk. Si la conexión es exitosa, permite que el sistema envíe y reciba datos de la aplicación en tiempo real, permitiendo al usuario monitorear el sistema y realizar ajustes.

- Configuración de Temporizadores (Timers): A través de BlynkTimer, se configuran intervalos de tiempo específicos que determinan la frecuencia con la que se envían datos a Blynk y se verifica el estado de conexión WiFi. Esto asegura que los datos se sincronicen en la nube periódicamente sin necesidad de intervención manual. A su vez, se inicializan las rutinas de sincronización periódica con la blockchain, una base de datos distribuida para controlar las automatizaciones de forma alternativa y confiable.

Gestión de Sensores

En la función loop() se encuentra el código que permite a la ESP32 realizar las siguientes tareas:

- Lectura de Sensores: Se encuentra el método *getSensorData()* donde el ESP32 recoge los valores de temperatura, humedad, presión y calidad del aire desde el sensor BME680. Estas lecturas se convierten y normalizan para almacenarse en variables globales que se usan tanto para la activación del relé como para el envío a la plataforma cloud.

- Control de Temperatura: Los valores de temperatura se comparan constantemente con los umbrales de activación y desactivación de la estufa predefinidos. Cuando la temperatura cae por debajo de un nivel específico, el dispositivo recibe desde el cloud la orden de activar la estufa, entonces el relé se activa a través de un pin de salida para encender la estufa, y cuando la temperatura supera este umbral, el sistema cloud envía la orden de apagar la estufa, generando que el relé se desactive y en consecuencia se apague la estufa.

Control de Actuadores (Relé y LEDs)

El control del relé y los LEDs es una de las tareas principales del firmware, permitiendo controlar la estufa en función de los datos de los sensores enviados a la nube y brindar una retroalimentación visual:

- Control del Relé: El ESP32 utiliza un pin GPIO para activar o desactivar el relé. El relé se activa si la temperatura es baja y se apaga cuando el ambiente alcanza el nivel deseado.

- Indicadores LED: Como se mencionó anteriormente, el sistema cuenta con dos leds indicadores, el LED de conexión wifi que Indica si el sistema está conectado a la red WiFi, encendiéndose en verde cuando la conexión es exitosa y en rojo si la conexión falla o está intentado establecerse; y el LED de estado de la estufa que muestra el estado actual de la estufa, encendiéndose en verde cuando está activa y en rojo

cuando está apagada. Esto permite al usuario verificar el estado de la estufa de forma rápida y visual.

Comunicación con la Plataforma Blynk

El sistema también permite la interacción del usuario a través de la plataforma Blynk, permitiendo el control remoto y la visualización de datos en tiempo real desde cualquier dispositivo conectado a internet. En el firmware se configuraron las siguientes funciones de control:

- Envío de Datos: se implementó la función reportDataTimer() que envía los valores de temperatura, humedad, presión y calidad del aire a Blynk cada 30 segundos. Estos datos se pueden visualizar en la app web de Blynk, permitiendo al usuario monitorear el estado del ambiente y la estufa desde su computadora.

- Recepción de Comandos: Con la función predefinida BLYNK_WRITE(), el programa recibe comandos desde Blynk para encender o apagar la estufa. Esto brinda al sistema la capacidad de controlar la estufa manualmente en situaciones específicas.

Comunicación con la Plataforma Blockchain

A su vez, el firmware también se configuró para conectarse con una base de datos distribuida en blockchain que almacene de forma confiable el estado del sistema y emita la orden de prender o apagar la estufa en base a este estado. Se configuraron los siguientes timers para esta integración:

- Envío de Datos: se creó un timer que envía a la blockchain, mediante un request POST HTTP, la última lectura de los sensores periódicamente cada 30 segundos. El usuario puede acceder a estos datos en la plataforma web de visualización de estado de la blockchain desde su dispositivo móvil o computadora.
- Recepción de Comandos: se creó un nuevo timer que lee de la blockchain el comando de encender o apagar la estufa periódicamente cada 50 segundos. El usuario puede consultar el último valor enviado en este comando desde la plataforma web de visualización de la blockchain, pero no puede modificarlo manualmente.

4.3. Software

Finalmente, el software del sistema complementa el hardware al proporcionar una interfaz gráfica y comunicación en tiempo real. Para este trabajo implementamos dos alternativas compatibles para este sistema con distinta tecnología y características. La primera implementación utiliza la aplicación Blynk, una plataforma IoT cloud low-code enfocada en la experiencia de usuario. La segunda implementación utiliza blockchain, una base de datos distribuida y confiable junto con un contrato inteligente para guardar el estado del sistema y accionar actuadores.

Ambos sistemas desempeñan la misma función y son independientes entre sí. En un proyecto productivo, deberá elegirse una de las dos alternativas para su uso definitivo. A continuación entraremos en detalle en ambas alternativas,

4.3.1 Blynk.io

Blynk.io, es una plataforma IoT que ofrece una interfaz visual para gestionar dispositivos conectados y flujos de datos. Esta plataforma provee una librería de código firmware que se carga en el dispositivo y gestiona automáticamente la conexión del dispositivo con la plataforma cloud, la sincronización y los reportes.

Al ser una plataforma low-code, no requiere que se escriba código para la visualización de los datos y automatizaciones de control del dispositivo, sino que esto es configurable desde una plataforma web. La aplicación web de Blynk nos proporciona control y monitoreo remoto con las siguientes funcionalidades:

- Configuración de Flujos de Datos: permite configurar flujos de datos desde y hacia el dispositivo, permitiendo disparar acciones en base a los valores de estos datos.
- Visualización en Tiempo Real: muestra datos en tiempo real sobre temperatura, humedad y calidad del aire, actualizándose cada vez que el ESP32 envía nuevos datos. Permite configurar un dashboard personalizado para visualizar estos datos de forma amigable y funcional.

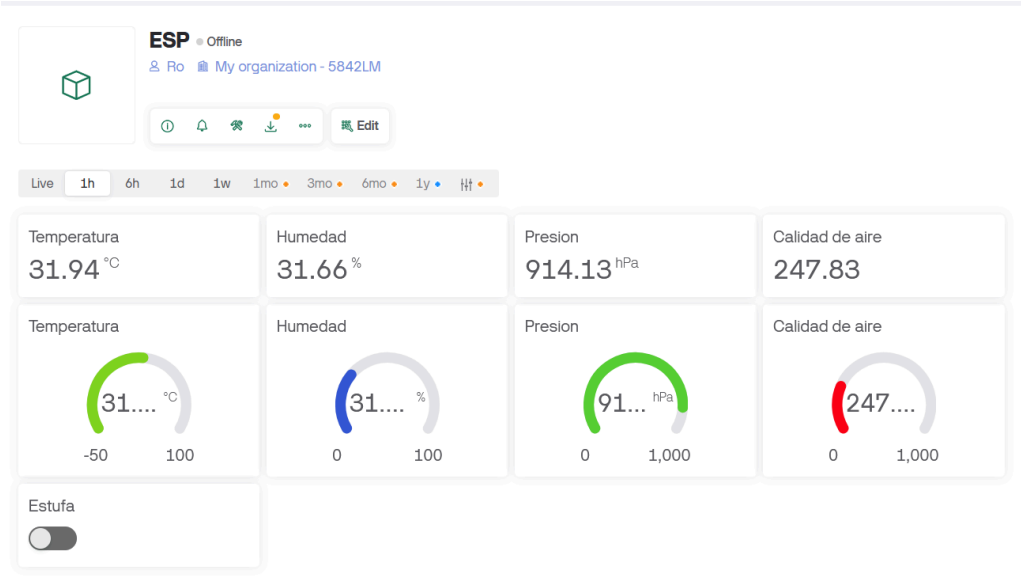

Figura 3: Dashboard de personalizado de flujo de datos de Blynk

- Control de la Estufa Automatizado: permite configurar automatizaciones para enviar al dispositivo la orden de encender o apagar la estufa cuando la temperatura se encuentre en cierto rango. Se pueden establecer los valores de temperatura umbrales para el encendido y apagado, personalizando así según preferencias y las condiciones del entorno. En este caso, se configuró la temperatura de encendido en

un valor más bajo que la temperatura de apagado, para evitar el encendido y apago intermitente de la estufa cerca del valor umbral.

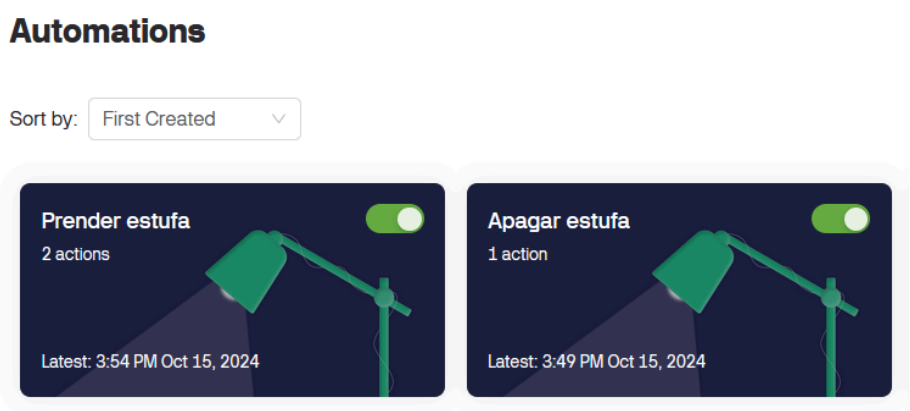

Figura 4: Lista de automatizaciones de Blynk

- Control Manual de la Estufa: el dashboard cuenta con botones que permiten al usuario encender o apagar la estufa manualmente, independientemente de los umbrales de temperatura. Esto es útil en situaciones en las que el usuario necesita ajustar manualmente la calefacción sin interferir con el sistema automatizado.

4.3.2 Blockchain

Blockchain es una tecnología de base de datos distribuida que almacena la información en forma de transacciones agrupadas en bloques interconectados mediante criptografía. En este sistema, la información de la base de datos únicamente es actualizable mediante contratos inteligentes, que son programas computacionales guardados en la misma base de datos que se ejecutan en un entorno seguro y distribuido. A su vez, estos contratos permiten codificar lógica de negocios ejecutable cuando la información es actualizada. De esta manera, los sistemas que utilizan blockchain como bases de datos tienen una fuente de la verdad confiable que no puede ser alterada por ningún medio que no haya sido predefinido en el contrato inteligente, brindando a su vez transparencia y trazabilidad. Estas características son deseables en proyectos donde participen distintos actores con intereses diversos y haya poca o baja confianza entre ellos, de manera que ningún actor puede alterar la información para su beneficio sin que la otra parte lo pueda detectar en el sistema instantáneamente. Con este enfoque, los datos son inmutables y verificables, y cada cambio es completamente rastreable, generando una base confiable para todas las partes interesadas y también útil para la toma de decisiones y el control automatizado.

En este proyecto, se implementa una solución de blockchain para asegurar la integridad de los datos generados por los sensores y para controlar la estufa mediante reglas de código confiables e inmutables.

La implementación se realizó en una red blockchain de pruebas local, aunque el sistema ha sido diseñado para poder funcionar de manera equivalente en un entorno productivo. Para la construcción del sistema, utilizamos la herramienta *scaffold-eth-2*, la cual permite el

desarrollo y despliegue de contratos inteligentes en Solidity en redes compatibles con Ethereum. *Scaffold-eth-2* integra el framework *Hardhat*, que facilita el despliegue de los contratos, y cuenta con un frontend en Next.js que se conecta directamente al contrato inteligente. Esta interfaz es amigable y se adapta automáticamente al contrato, lo que permite que el usuario visualice el estado del contrato y acceda a sus métodos de forma intuitiva.

Además, el sistema incluye una API en el servidor de Next.js, que permite la comunicación entre los dispositivos IoT y la blockchain. El dispositivo ESP32 reporta datos a la blockchain mediante un POST HTTP a la API Rest, y cada cierto tiempo el mismo dispositivo consulta a la misma API mediante un GET HTTP el estado objetivo de la estufa, ya sea encendido o apagado, y activa su actuador (la estufa) en base a este parámetro. Este valor se almacena y actualiza en la blockchain según las lecturas de los sensores y los umbrales de temperatura previamente configurados en el contrato inteligente. De este modo, el contrato controla de manera automática el estado de la estufa: cuando el dispositivo IoT consulta el valor almacenado en la blockchain, actúa en consecuencia, encendiendo o apagando la estufa según lo indicado por el contrato. En la Figura 5 se puede visualizar esta interacción.

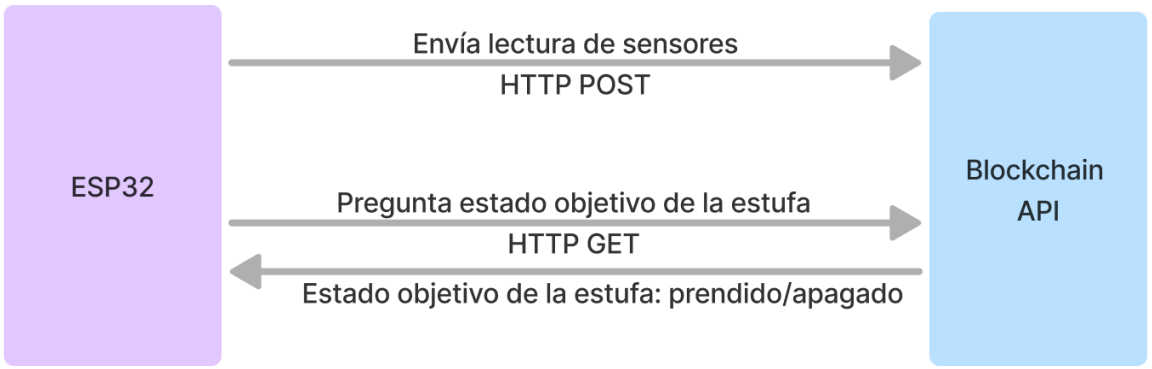

Figura 5. Interacción entre el dispositivo y la blockchain

Una funcionalidad adicional es la actualización de los umbrales de temperatura a través del contrato, lo que permite ajustar el sistema sin intervención directa sobre el firmware del dispositivo. Los sensores envían lecturas al contrato, y en función de los umbrales configurados, el contrato determina y registra la acción deseada (encender o apagar la estufa) en la blockchain. Es importante señalar que el valor de la estufa no se puede controlar manualmente en este sistema, ya que toda decisión se basa en la lógica del contrato inteligente, lo cual evita la manipulación externa y contribuye a la seguridad del sistema, aunque conlleva el riesgo de seguridad de no poder controlar manualmente el sistema en caso de emergencia en un escenario atípico.

Para mejorar la seguridad en futuras implementaciones, se recomienda agregar autenticación al sistema, de modo que solo dispositivos autorizados puedan interactuar con el contrato y realizar llamadas a sus métodos. Esta capa de seguridad adicional permitirá restringir el acceso a los métodos críticos del contrato, asegurando que únicamente

dispositivos verificados puedan influir en las decisiones de control, como la modificación de umbrales, y almacenar datos en la blockchain.

5. Conclusiones

En conclusión, este prototipo ha demostrado ser un precedente positivo para el desarrollo de sistemas de control automatizado en IoT, haciendo uso de blockchain para garantizar la confiabilidad y robustez de los datos sin incrementar significativamente la complejidad del firmware. Mediante el uso de una blockchain y contratos inteligentes, se ha logrado un sistema seguro y confiable que permite el control automatizado de dispositivos IoT, ofreciendo una plataforma escalable y trazable para aplicaciones futuras. A su vez, mediante el uso de Blynk....

El desarrollo de este sistema de control automatizado para una estufa eléctrica, que combina IoT y tanto blockchain como Blynk.io, ha demostrado ser una solución confiable y funcional para el control seguro y eficiente de dispositivos eléctricos en el hogar.

Particularmente al utilizar una red blockchain de pruebas para almacenar y gestionar los estados de la estufa en función de los datos del sensor, se logró un sistema que no solo ofrece automatización, sino que además asegura la integridad de la información y la trazabilidad de cada acción. Esto permite al usuario confiar en el sistema y en las decisiones tomadas de manera automática según los parámetros definidos. Este esquema probó ser escalable a otros tipos de sensores y dispositivos que puedan surgir para diversos casos de uso.

A su vez, la plataforma Blynk demostró ser efectiva y fácil de usar con un nivel mínimo de configuración, permitiendo configurar tanto flujos de datos, como automatizaciones y control manual. En sistemas de mayor escala, este sistema también prueba ser efectivo, pero se debe adquirir un plan pago en la plataforma.

En futuros desarrollos, se sugiere implementar autenticación para que solo dispositivos autorizados puedan interactuar con los contratos inteligentes. Este paso adicional fortalecería la seguridad y garantizaría que las modificaciones y comandos en el sistema provengan únicamente de fuentes confiables. A su vez, se propone poner a prueba el sistema blockchain con múltiples dispositivos y para casos de uso con múltiples actores con intereses diversos. En conclusión, este prototipo abre la puerta a una implementación práctica de blockchain en IoT, validando que esta tecnología puede contribuir significativamente a la creación de entornos controlados, seguros y altamente automatizados sin complejidad adicional en el firmware del dispositivo.

6. Referencias

- Alldatasheet. (s. f.). ESP8266 datasheet. https://www.alldatasheet.com/view.jsp?Searchword=Esp8266
- Amazon. (s. f.). ESP8266 Wi-Fi module. https://www.amazon.com/s?k=esp8266+wi-fi+module
- Wikipedia. (s. f.). ESP8266. https://es.wikipedia.org/wiki/ESP8266
- Naylamp Mechatronics. (s. f.). Usando ESP8266 con el IDE de Arduino. https://naylampmechatronics.com/blog/56_usando-esp8266-con-el-ide-de-arduino.htm
- Todomicro. (s. f.). Módulo ESP12F NodeMCU D1 Mini ESP8266. https://www.todomicro.com.ar/NODEMCU/377-modulo-esp12f-nodemcu-d1-mini-esp8266.html
- Espressif. (s. f.). ESP8266 overview. https://espressif.com/en/products/socs/esp8266
- Prometec. (s. f.). Subir valores con ESP8266. https://www.prometec.net/esp8266-subir-valores/4
- Sigma Electrónica. (s. f.). ESP8266 4MB. https://www.sigmaelectronica.net/producto/esp8266-4mb/
- Alibaba. (s. f.). ESP8266 Wi-Fi module. https://www.alibaba.com/premium/esp8266_wifi_module.html
- Spluss. (s. f.). Sensor de calidad del aire AERASGARD RCO2 Modbus. https://www.spluss.de/es/sensor-de-la-calidad-del-aire-para-interiores-aerasgard-rco2-modbus-modbus-d501-61b0-6000-000-p6890
- Amazon. (s. f.). Onyehn Temperature and Humidity Sensor. https://www.amazon.com/Onyehn-Temperature-Humidity-Barometric-Pressure/dp/B07KR24P6P
- Naylamp Mechatronics. (s. f.). Sensor BME680 presión, temperatura, humedad y gas. https://naylampmechatronics.com/sensores-posicion-inerciales-gps/650-sensor-bme680-presion-temperatura-humedad-y-gas.html
- Mercado Libre. (s. f.). Sensor de calidad de aire. https://listado.mercadolibre.com.ar/sensor-calidad-aire
- Bosch Sensortec. (s. f.). BME680. https://www.bosch-sensortec.com/products/environmental-sensors/gas-sensors/bme680/
- Uelectronics. (s. f.). BME680 Sensor. https://uelectronics.com/producto/bme680-sensor-de-presion-temperatura-y-humedad/
- Octopart. (s. f.). BME680 datasheet. https://octopart.com/es/bme680-bosch+tools-84623084
- Datasheets.com. (s. f.). BME680 datasheet. https://www.datasheets.com/part-details/bme680-bosch-93559225
- Programo Ergo Sum. (s. f.). Configurar ESP8266 con Arduino IDE. https://www.programoergosum.es/tutoriales/configurar-esp8266-con-arduino-ide/
- Luis Llamas. (s. f.). Programar ESP8266 con el IDE de Arduino. https://www.luisllamas.es/programar-esp8266-con-el-ide-de-arduino/

- YouTube. (2018). Tutorial ESP8266 [Video]. https://www.youtube.com/watch?v=9JzKR8EDSN0
- YouTube. (2017). Cómo programar ESP8266 [Video]. https://www.youtube.com/watch?v=5vr4W_Qe_v4
- YouTube. (2018). Configurar ESP8266 [Video]. https://www.youtube.com/watch?v=AGmUqL_xxU8
- YouTube. (2019). Programar con Arduino y ESP8266 [Video]. https://www.youtube.com/watch?v=0g7sazWXfEI
- Instructables. (s. f.). Adafruit BME680 & Arduino UNO tutorial. https://www.instructables.com/Adafruit-BME680-Arduino-UNO-a-Tutorial/
- Luis Llamas. (s. f.). Sensor de calidad ambiental con Arduino y BME680. https://www.luisllamas.es/sensor-de-calidad-ambiental-con-arduino-y-bme680/