



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO
CARRERA DE INGENIERÍA ELECTRÓNICA**

**SISTEMA DE CONTROL EN RED INALÁMBRICO (WNCS) PARA EL
CONTROL AUTOMÁTICO DE UN
MOTOR DC MEDIANTE DIFERENTES PROTOCOLOS DE RED**

Trabajo de titulación previo a la obtención
del Título de Ingeniero Electrónico

AUTORES: José Orlando Oña Suintaxi

Noe Misael Vargas Palacios

TUTOR: Carlos Germán Pillajo Angos

Quito-Ecuador

2024

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Noe Misael Vargas Palacios con documento de identificación N° 1719859405 y José Orlando Oña Sntaxi con documento de identificación N° 1723037667 manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 01 de agosto del 2024

Atentamente,



Noe Misael Vargas Palacios

1719859405



José Orlando Oña Sntaxi

1723037667

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Noe Misael Vargas Palacios con documento de identificación N° 1719859405 y José Orlando Oña Suntaxi con documento de identificación N° 1727037767 expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos los autores del artículo académico: “Sistema de control en red inalámbrico (WNCS) para el control automático de un motor DC mediante diferentes protocolos de red”, el cual ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 01 de agosto del 2024

Atentamente,



Noe Misael Vargas Palacios

1719859405



José Orlando Oña Suntaxi

1723037667

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Carlos Germán Pillajo Angos con documento de identificación N° 1709255119 docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: SISTEMA DE CONTROL EN RED INALÁMBRICO (WNCS) PARA EL CONTROL AUTOMÁTICO DE UN MOTOR DC MEDIANTE DIFERENTES PROTOCOLOS DE RED, realizado por Noe Misael Vargas Palacios con documento de identificación N° 1719859405 y José Orlando Oña Suintaxi con documento de identificación N° 1723037667 , obteniendo como resultado final el trabajo de titulación bajo la opción de artículo académico, que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 01 de agosto del 2024

Atentamente,

A handwritten signature in blue ink, consisting of a large, stylized 'C' followed by several loops and a long horizontal stroke extending to the right.

Ing. Carlos Germán Pillajo Angos PhD
1709255119

SISTEMA DE CONTROL EN RED INALÁMBRICO (WNCS) PARA EL CONTROL AUTOMÁTICO DE UN MOTOR DC MEDIANTE DIFERENTES PROTOCOLOS DE RED

¹Noe M Vargas, ²José O Oña

¹Universidad Politécnica Salesiana, Av. Rumichaca Ñan, y Av. Moran Valverde, Quito.
nvargasp@est.ups.edu.ec
jonas@est.ups.edu.ec

Resumen: El presente estudio integra los protocolos MQTT, CoAP y HTTP para implementar un sistema de control PID en un motor de corriente continua (DC). Empleando el protocolo MQTT, se puede establecer una comunicación eficiente, bidireccional y escalable entre dispositivos IoT, de esta manera permite la monitorización y ajustes remotos de parámetros de control. Este protocolo es ideal debido a su capacidad para gestionar mensajes de manera ligera y eficiente. El protocolo CoAP se emplea para la transferencia de datos en redes con recursos limitados, optimiza el intercambio de información al minimizar el overhead de los mensajes y adaptarse a las limitaciones de ancho de banda y energía de los dispositivos conectados, optimizando la comunicación entre sensores, actuadores y a su vez el controlador PID. Por último, HTTP facilita una interfaz de usuario accesible a través de la web o dispositivos móviles, proporcionando visualizaciones en tiempo real y herramientas de configuración para mejorar la eficiencia y precisión del control de velocidad del motor DC. El corazón del proyecto es el control PID, un algoritmo frecuentemente utilizado de control de retroalimentación que ajusta continuamente la señal de control enviada al motor DC en función de la diferencia entre la velocidad deseada y la velocidad medida. Este algoritmo garantiza que el motor DC opere de manera óptima y estable, incluso frente a variaciones en la carga o condiciones externas cambiantes.

Keywords: coap, http, IoT, protocolo, mqtt.

Abstract: The present study integrates MQTT, CoAP, and HTTP protocols to implement a PID control system in a direct current (DC) motor. By using the MQTT protocol, efficient, bidirectional, and scalable communication can be established between IoT devices, allowing for remote monitoring and adjustments of control parameters. This protocol is ideal due to its ability to manage messages in a lightweight and efficient manner. The CoAP protocol is used for data transfer in resource-constrained networks, optimizing information exchange by minimizing message overhead and adapting to the bandwidth and energy limitations of connected devices, thereby enhancing communication between sensors, actuators, and the PID controller. Finally, HTTP provides a user interface accessible via the web or mobile devices, offering real-time

visualizations and configuration tools to improve the efficiency and accuracy of DC motor speed control. The core of the project is the PID control, a commonly used feedback control algorithm that continuously adjusts the control signal sent to the DC motor based on the difference between the desired and measured speed. This algorithm ensures that the DC motor operates optimally and stably, even in the face of load variations or changing external conditions.

Keywords: coap, http, IoT, protocolo, mqtt.

1. Introducción

El Internet de las Cosas (IoT), es una red gigante de dispositivos interconectados entre sí y que a su vez pueden comunicarse, además de compartir información a través del Internet [1]. Hoy en día ya no solo se trata de ordenadores o dispositivos inteligentes, que se pueden controlar, sino que todo tipo de objetos cotidianos son controlados por usuarios remotos, gracias a los sensores que poseen cada uno [2]. IoT ha permitido cambiar la forma en que se administran los recursos, tales como la salud, un reloj inteligente puede predecir un posible infarto, e incluso si las personas de la tercera edad llegasen a perderse, se podría ubicarlos a través de los mismos [3]. Por esta razón desarrollar un sistema de control en red inalámbrico (WNCS) para el control automático de un motor DC, mediante el análisis de métricas de tres diferentes protocolos de red HTTP, CoAP y MQTT es el objetivo principal del presente proyecto.

El crecimiento de IoT no presenta signos de desaceleración. El crecimiento se debe a que cada vez existen más dispositivos que son capaces de conectarse al internet y compartir datos. El principal objetivo de las empresas es desarrollar productos innovadores que utilicen IoT con la finalidad de que estos productos mejoren muchos aspectos de la vida cotidiana [4]. Por lo cual desarrollar un sistema automático para el control de velocidad de un motor DC didáctico, mediante un sistema de control en red inalámbrico WNCS, implica un avance importante que podría ser empleado en la industria sin ningún problema.

Ahora bien, para que todo funcione tal como se tiene planteado se necesita que los dispositivos se comuniquen entre sí. Es por ello que para el desarrollo de este proyecto en primer lugar se debe investigar sobre los sistemas WNCS, protocolos IoT y programación basada en flujo de datos, para la actualización de conocimientos utilizando base de datos de artículos científicos.

De este modo es como aparecen los protocolos de comunicación que se van a implementar HTTP, CoAP y MQTT en WNCS. Aquellos protocolos de comunicación se conocen como el lenguaje del IoT, ya que permiten el envío y la recepción de datos [6]. También pueden ser considerados como traductores, ya que aseguran que a pesar de la diferencia de tamaños de los dispositivos inteligentes estos puedan entablar una

comunicación satisfactoria [7]. En conclusión, los protocolos son importantes para que se genere la conexión entre dispositivos IoT y los sistemas que administran y procesan los datos que son generados [8].

Para emplear los protocolos previamente se toma en cuenta las mejores cualidades de cada uno. Para comprobar que el proyecto funcione correctamente el último objetivo es analizar las métricas de rendimiento de los tres protocolos (HTTP, CoAP y MQTT) implementados en un motor DC para su comparación mediante pruebas en entornos controlados. Los clientes MQTT son pequeños por lo cual requieren recursos mínimos para su correcto funcionamiento. De este modo se pueden utilizar en pequeños microcontroladores, los cuales son de fácil acceso [9]. Los encabezados que se emplean en los mensajes MQTT tienen un tamaño reducido lo cual hace que el ancho de banda necesario para la transmisión de los mismos no necesariamente sea robusto. La comunicación bidireccional se usa dentro del desarrollo del protocolo MQTT. Lo cual permite que los mensajes se envíen desde el dispositivo a la nube y también de la nube hacia el dispositivo [10]. Esta comunicación facilita el envío de datos y mensajes a varios dispositivos inteligentes o IoT. Por último, este protocolo facilita el cifrado de mensajes a través de TLS y la autenticación de clientes mediante protocolos modernos, por ejemplo: OAuth [11].

El protocolo CoAP al igual que el protocolo MQTT puede ejecutarse en hardware simple, como sensores de baja potencia, microcontroladores y a su vez en dispositivos que no pueden ser ejecutados con el protocolo HTTP, por lo que permite que los dispositivos se comuniquen a través de una red. La comunicación que presenta este protocolo es similar al modelo cliente – servidor de HTTP [12]. No obstante, existen ocasiones en este protocolo que puede actuar como servidor y como cliente. Cuando el servidor CoAP recibe una solicitud o request, la procesa y posterior a ello, envía una respuesta al cliente con un código de respuesta y una imagen del contenido en caso de que corresponda [13]. Los distintos tipos de mensajes que se logran enviar son identificados con un ID que poseen una calidad de servicio asociada. Para que el protocolo CoAP funcione adecuadamente utiliza dos capas. Una primera capa que se denomina Message y es transaccional es utilizada para trabajar con UDP, la segunda capa que se emplea es Request/Response la cual se encarga de las interacciones empleando distintos métodos [14]. UDP es un protocolo simple y sin conexión, lo que significa que no establece una conexión antes de enviar datos y no garantiza la entrega ni el orden de los mismos.

Este artículo implementa tres alternativas o soluciones eficientes y que pueden ser empleadas en un ámbito industrial sin ningún problema. Una parte fundamental de esta implementación es un microcontrolador capaz de comunicarse con los protocolos MQTT, CoAP y HTTP.

El documento presente se encuentra constituido en la sección I Introducción, en la sección II Metodología, sección III Análisis de Resultados, sección IV Discusión y finalmente las Conclusiones en la sección V.

2. Metodología

Puesto a que el mundo del IoT ha crecido proporcionalmente y no deja de hacerlo, se requiere que los protocolos de comunicación sean capaces de emitir y recibir mensajes en periodos de tiempo muy cortos y utilizando la mínima cantidad de recursos. Se propuso emplear en el presente proyecto 3 protocolos MQTT, CoAP y HTTP para controlar un motor DC.

Para iniciar con el desarrollo del presente proyecto se realiza un diagrama de bloques del control WNCs en lazo cerrado de la planta, tal como se puede observar en la Figura 1, en la misma se describe los elementos que se van a utilizar.

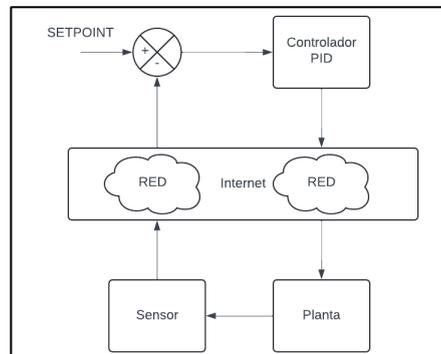


Fig. 1. Diagrama de bloques del control WNCs en lazo cerrado de la planta.

Empleando el comando IDENT del software Matlab se obtiene la función de transferencia del sistema. En este caso se estiman valores de acuerdo a la necesidad del usuario.

En la Figura 2, se presenta la respuesta que se obtiene de la función de transferencia, como se puede observar presenta un sobre impulso que rápidamente se estabiliza y llega a la respuesta deseada.

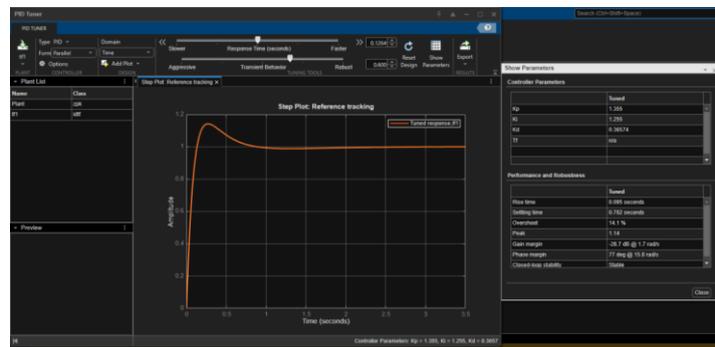


Fig. 2. Respuesta de la función de transferencia.

Función de transferencia del sistema.

El control del motor que se emplea en el desarrollo del presente proyecto es un PID, por lo cual se requiere una función de transferencia. Función de transferencia que está diseñada para controlar la planta y se presenta a continuación:

$$FT = \frac{42.54}{s^2 + 0.106 s + 0.7751}$$

Ecuación. 1. Función de Transferencia del sistema.

En la Figura 3 se presenta la respuesta del control PID con la función de transferencia que se obtuvo, misma que se acerca en gran manera al Set Point ingresado.



Fig. 3. Respuesta de la Función de Transferencia.

Protocolos de comunicación MQTT, HTTP y CoAP.

Para el desarrollo de este proyecto se tuvo en cuenta los aspectos más relevantes como la velocidad de respuesta de cada protocolo, las similitudes que tienen entre sí y el tiempo de procesamiento. A partir de ahí se desarrollan los diagramas de bloques con los que se va a trabajar. Los cuales se pueden observar en la Figura 4, en ella se observa el procedimiento paso a paso desarrollado en la implementación del protocolo MQTT. Similar a los demás protocolos que se detallan más adelante, en primera instancia se declaran e incluyen las librerías propias del protocolo. Lo siguiente que procede es

declarar y definir las constantes, objetos y las variables necesarias. Un punto importante es la configuración de pines, para el motor y también para el LCD que permite visualizar los datos. A partir de aquí, se diferencia de los demás protocolos, puesto a que en este paso se configura el cliente MQTT. EL cliente MQTT es quien publica los mensajes ya que contiene los datos en formato de bytes, al igual que determina el formato de los datos, pueden ser datos tipo texto, datos binarios, archivos XML o JSON [16]. Suscribirse al Topic que contiene el dato de la salida PID. Los Bróker MQTT trabajan filtrando los mensajes que son recibidos mediante los publicadores, este filtro es lo que se conoce como Topic y consiste en una cadena de texto UTF-8, la misma que tiene una longitud máxima de 65536 caracteres, sin embargo, puede ser menor y a su vez se distingue entre mayúsculas y minúsculas. En la siguiente sección del diagrama de flujo se puede observar un cuadro de decisión, “¿Existe comunicación MQTT?”, en caso de que la respuesta sea “NO” el programa envía nuevamente el dato después de 5 segundos, mientras que la respuesta sea “SI”, el programa procede a leer el dato de salida del PID que proviene de NODE RED [17]. A partir de este paso inicia el trabajo del motor ya que se asigna el valor leído al PWM del motor. A través de un encoder se obtiene la velocidad a la cual se encuentra girando el motor. Finalmente se publica a NODE RED el valor de la velocidad obtenida.

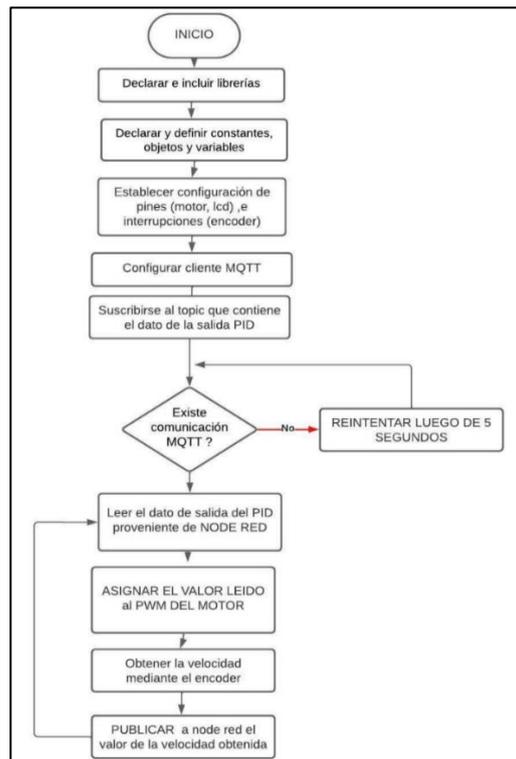


Fig. 4. Arquitectura del protocolo MQTT.

El siguiente protocolo empleado para el desarrollo de este proyecto es HTTP, como ya se indicó anteriormente los primeros pasos son similares. En el protocolo HTTP primero se inicia estableciendo la conexión a la red, posterior a ello se tiene un cuadro de decisión con la pregunta “¿Existe conexión a Wifi?”, si existe continúa trabajando y enviando datos, sin embargo, sino tiene conexión el bucle se repite hasta el momento que el protocolo establezca conexión. Una vez que se ha establecido la conexión se configura el protocolo HTTP. Una vez que se haya obtenido la salida PID desde el NODE RED, se envía el dato del PWM al motor, de igual manera que en el anterior protocolo a través del encoder, finalmente se envía las RPM del motor hacia NODE RED [11].

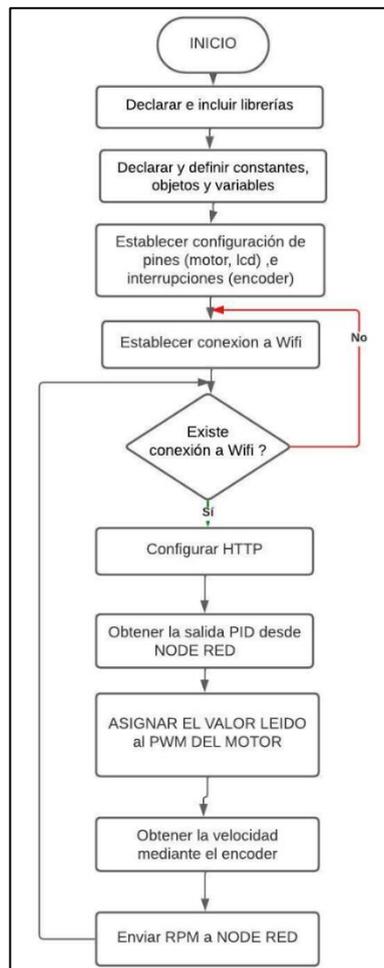


Fig. 5. Arquitectura del protocolo HTTP.

El protocolo CoAP y el protocolo HTTP son muy similares si se los compara en relación al diagrama de flujo, únicamente después de configurar el protocolo se debe iniciar.

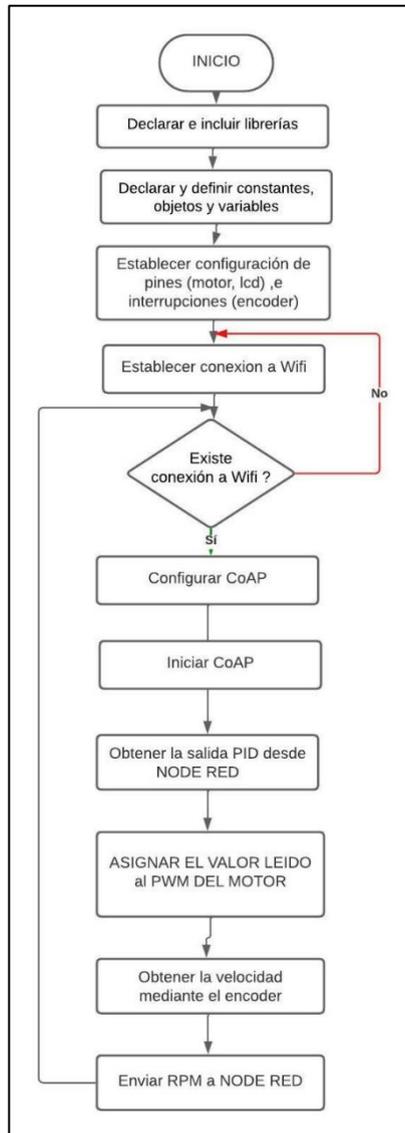


Fig. 6. Arquitectura del protocolo CoAP.

En la Figura 7, se tiene el diagrama de flujo que abarca a los 3 protocolos, en este caso se encuentran integrados en la plataforma gratuita NODE RED. Una vez que el usuario elije el set point que se requiere que trabaje se inicia con la ejecución de los

protocolos. Se tiene que tomar en cuenta que cada protocolo funciona por separado. En la sección de Resultados, se describe con mayor detalle el tiempo de procesamiento y de respuesta de cada uno de los protocolos.

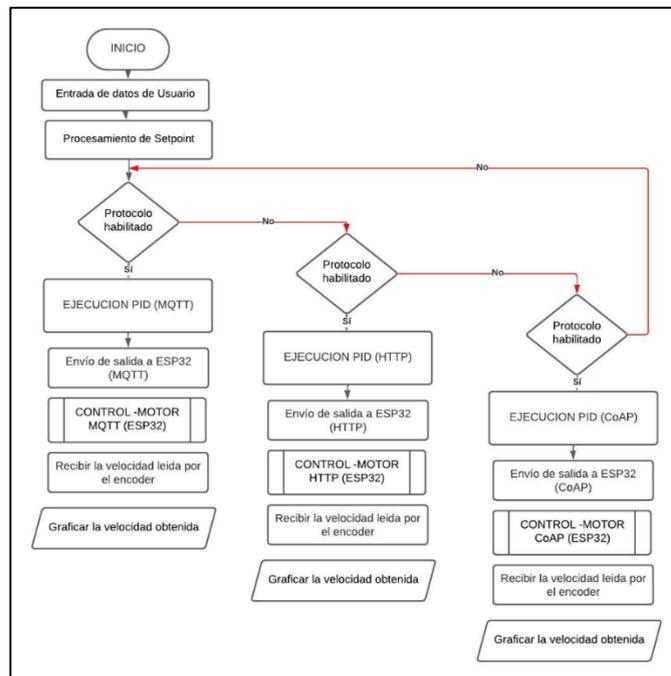


Fig. 7. Arquitectura de integración de los protocolos MQTT, HTTP y CoAP en Node-RED.

En la Figura 8 se presenta la maqueta ensamblada del presente proyecto.



Fig. 8. Maqueta del proyecto.

3. Análisis de resultados

En esta sección se analizan los resultados obtenidos al ejecutar cada protocolo previamente descrito. Después de ejecutar un protocolo se coloca un Set Point al cual debe alcanzar la respuesta del sistema en la menor cantidad de tiempo posible.

A continuación, se detallan las pruebas realizadas al protocolo MQTT.

Para obtener el resultado de la primera prueba con el protocolo MQTT se colocó un Set Point de 280, que en este caso la Figura 9 presenta la velocidad deseada. Después de 4.4 segundos el motor se encuentra estable y alcanza la velocidad deseada. El error que se tiene para esta prueba es de 0.11 %. En la parte inferior izquierda se puede observar la respuesta del control PID.

Desde la Figura 9 hasta la Figura 15 se presentan las pruebas realizadas al protocolo MQTT, esto con la finalidad de encontrar errores tal como: pérdidas en el envío y recepción de datos, que no se alcance la velocidad deseada, sin embargo, el error porcentual más alto es de 0.35 % que se llegó a obtener a la velocidad de 300 RPM, esto podría suceder ya que se bajó la velocidad del motor de 350 RPM a 300 RPM. De igual manera, es un error aceptable que no afecta en el desempeño del protocolo.

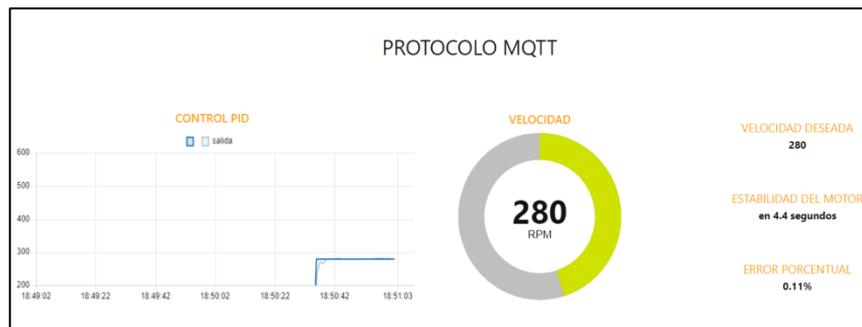


Fig. 9. Prueba de Protocolo MQTT (Set Point 280).

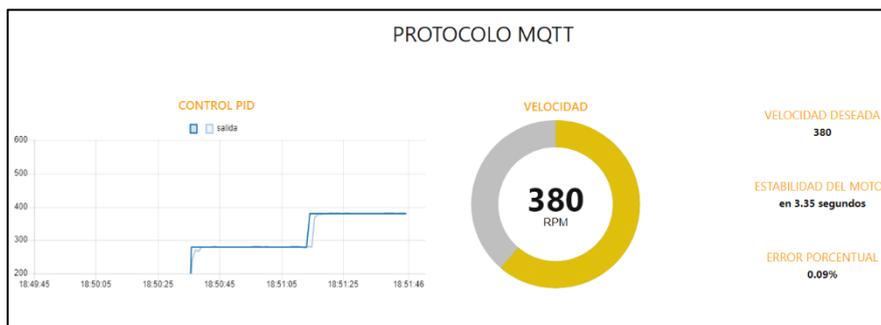


Fig. 10. Prueba de Protocolo MQTT (Set Point 380).

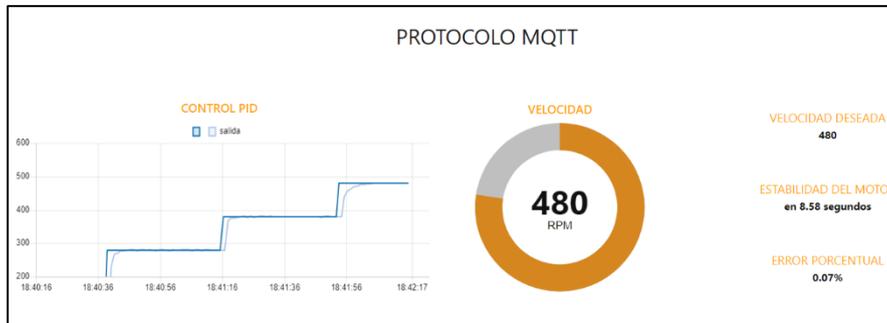


Fig. 11. Prueba de Protocolo MQTT (Set Point 480).

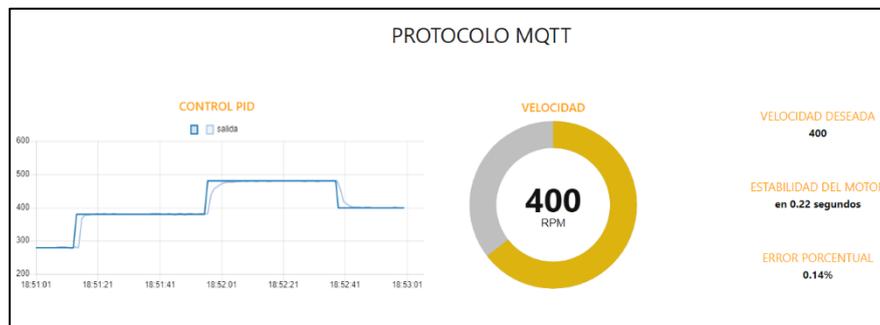


Fig. 12. Prueba de Protocolo MQTT (Set Point 400).

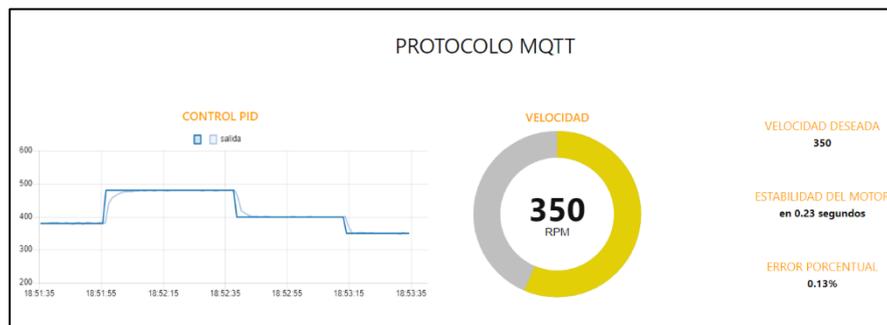


Fig. 13. Prueba de Protocolo MQTT (Set Point 350).

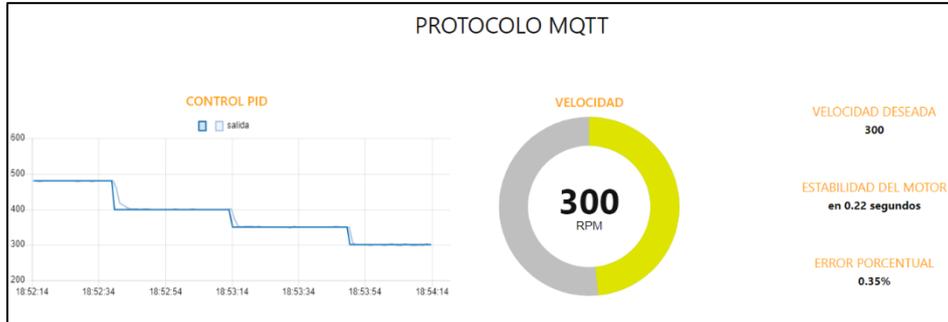


Fig. 14. Prueba de Protocolo MQTT (Set Point 300).

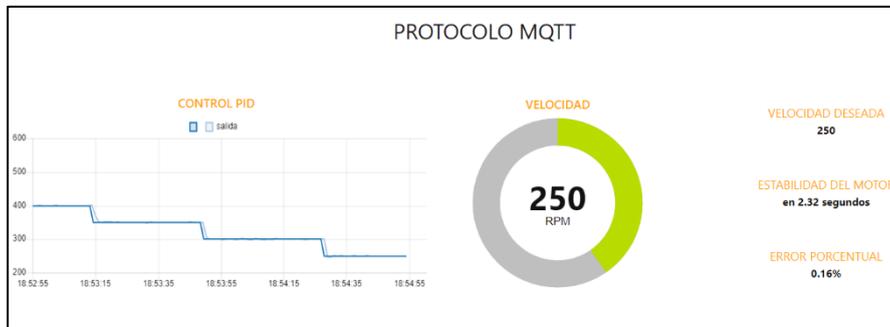


Fig. 15. Prueba de Protocolo MQTT (Set Point 250).

En la Tabla 1., se tiene un consolidado de las pruebas del Protocolo MQTT, el tiempo que se demora en estabilizarse es muy pequeño. Si se observa con mayor detenimiento la tabla se encuentra que al momento de subir las RPM de 380 a 480 RPM el tiempo que tarda en estabilizarse el motor es de 8.58 s. Aún con ese tiempo el controlador se encuentra trabajando sin problemas y la respuesta sigue siendo buena.

PROTOCOLO MQTT		
SET POINT (RPM)	ESTABILIDAD DEL MOTOR (segundos)	Error porcentual
280	4.4	0.11 %
380	3.35	0.09 %
480	8.58	0.07 %
400	0.22	0.14 %
350	0.23	0.13 %
300	0.22	0.35 %
250	2.32	0.16 %

Tabla 1. Pruebas protocolo MQTT.

Desde la Figura 16 hasta la Figura 22, se presentan las pruebas realizadas al protocolo CoAP, cabe recalcar que de igual manera presenta una buena respuesta al alcanzar la velocidad deseada en tiempos muy cortos como se puede observar.

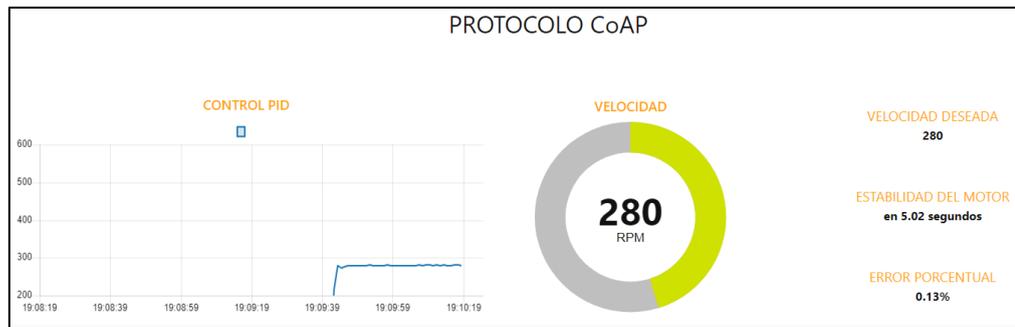


Fig. 16. Prueba de Protocolo CoAP (Set Point 280).

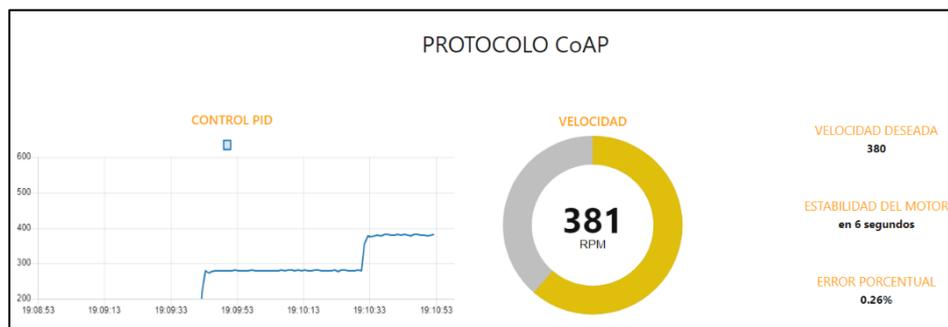


Fig. 17. Prueba de Protocolo CoAP (Set Point 381).

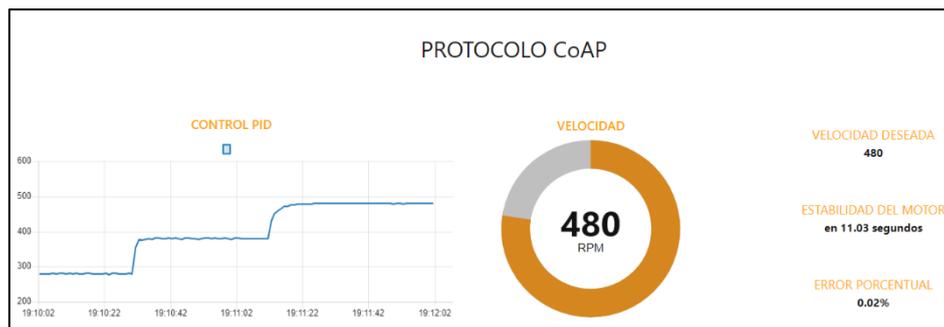


Fig. 18. Prueba de Protocolo CoAP (Set Point 480).

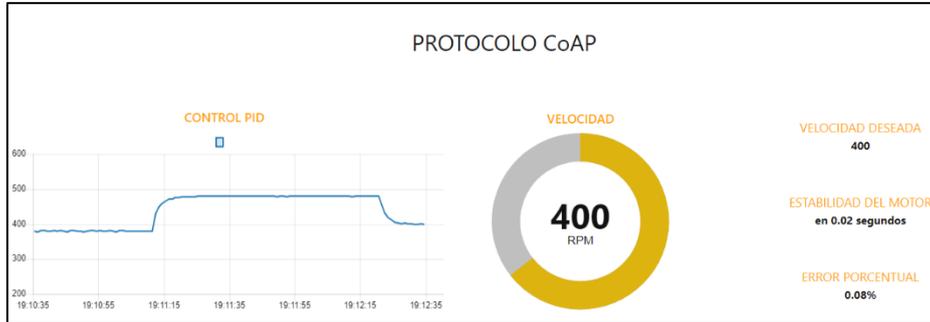


Fig. 19. Prueba de Protocolo CoAP (Set Point 400).

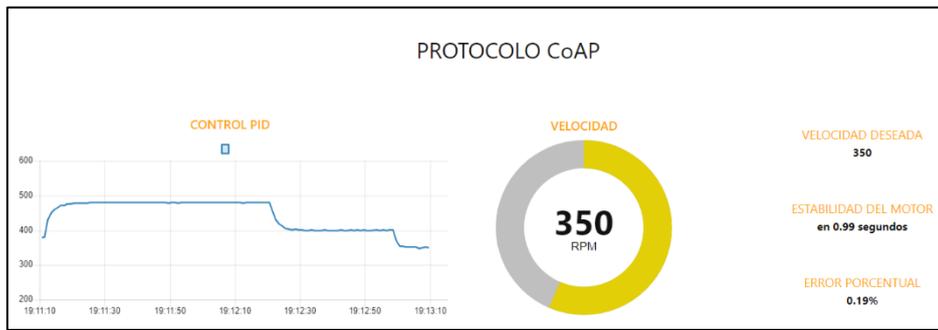


Fig. 20. Prueba de Protocolo CoAP (Set Point 350).

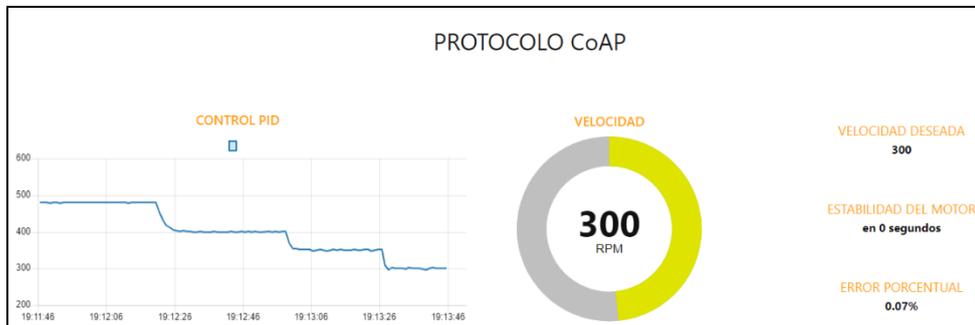


Fig. 21. Prueba de Protocolo CoAP (Set Point 300).

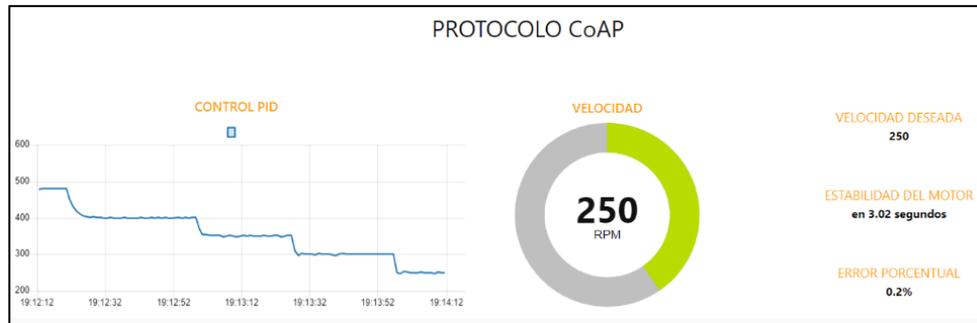


Fig. 22. Prueba de Protocolo CoAP (Set Point 250).

En la Tabla 2., se presenta el consolidado de las pruebas realizadas al protocolo CoAP, similar al protocolo MQTT, una vez que se sube las RPM de 381 a 480 el tiempo de respuesta tarda 5 segundos aproximadamente más, que en los otros cambios de velocidad y el error porcentual sigue siendo bajo.

PROTOCOLO CoAP		
SET POINT (RPM)	ESTABILIDAD DEL MOTOR (segundos)	Error porcentual
280	5.02	0.13 %
381	6	0.26 %
480	11.03	0.02 %
400	0.02	0.08 %
350	0.99	0.19 %
300	0	0.07 %
250	3.02	0.2 %

Tabla 2. Pruebas protocolo CoAP.

Por último, se tienen las pruebas realizadas al protocolo HTTP, desde la Figura 23 hasta la Figura 29. Se debe tomar en cuenta que se está trabajando con una ventana de histéresis de medición dentro del rango -2 hasta +2. En este caso para que el sistema determine si está estable, durante 5 segundos la velocidad puede variar entre 278 y 282, es por ello que el error que se tiene es menor a 5, a diferencia de la última prueba que el error porcentual sobrepasa el 5%.

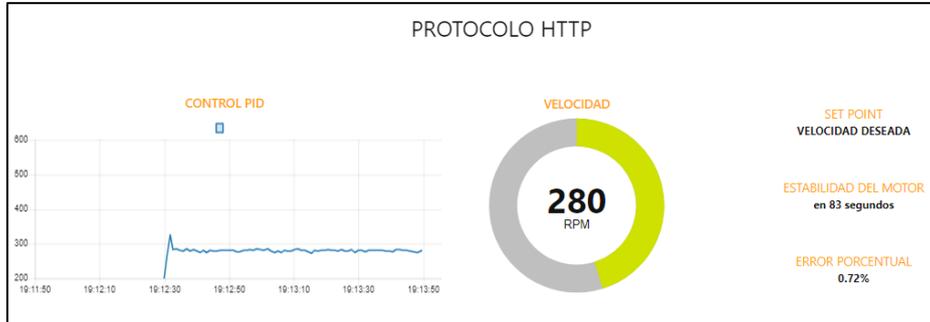


Fig. 23. Prueba de Protocolo HTTP (Set Point 280).

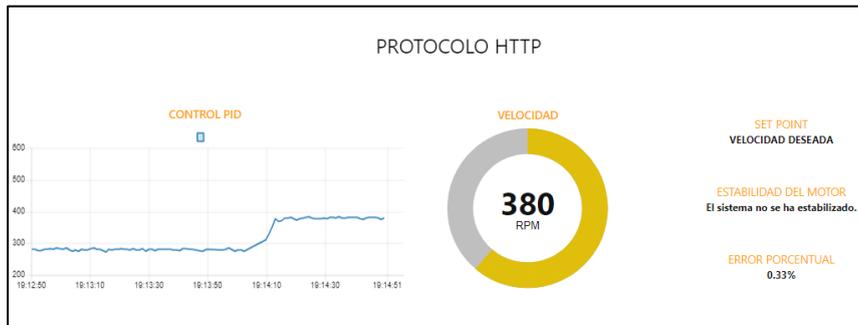


Fig. 24. Prueba de Protocolo HTTP (Set Point 380).

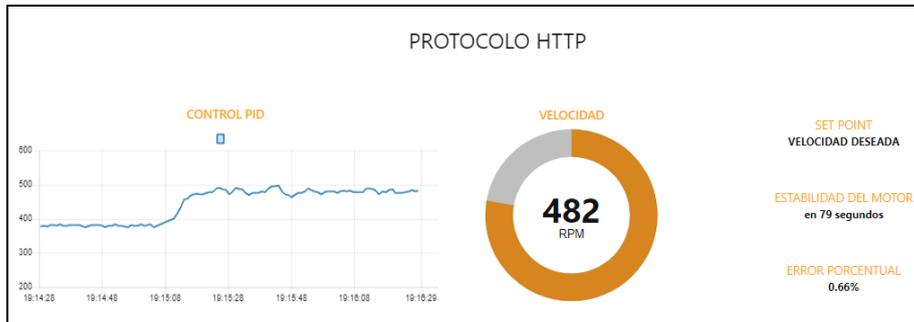


Fig. 25. Prueba de Protocolo HTTP (Set Point 482).

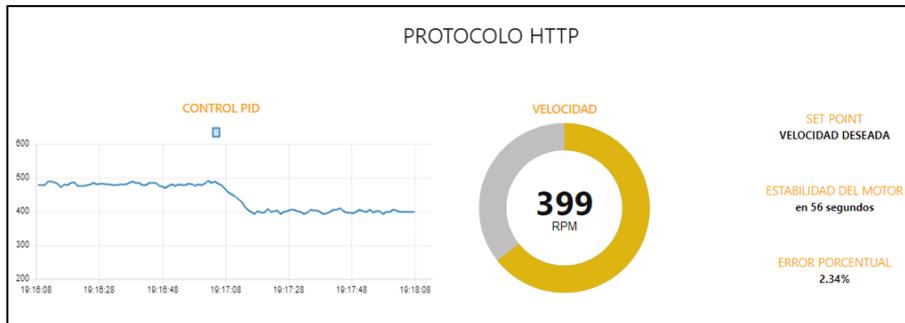


Fig. 26. Prueba de Protocolo HTTP (Set Point 399).

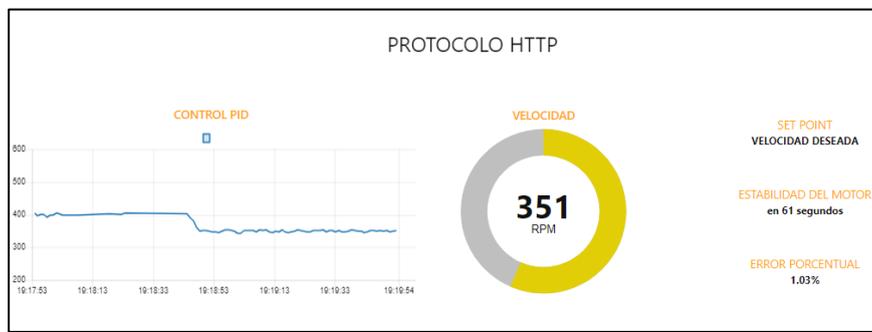


Fig. 27. Prueba de Protocolo HTTP (Set Point 351).

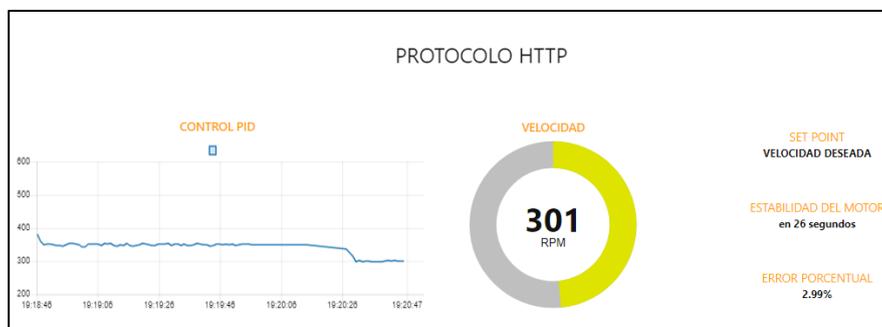


Fig. 28. Prueba de Protocolo HTTP (Set Point 301).

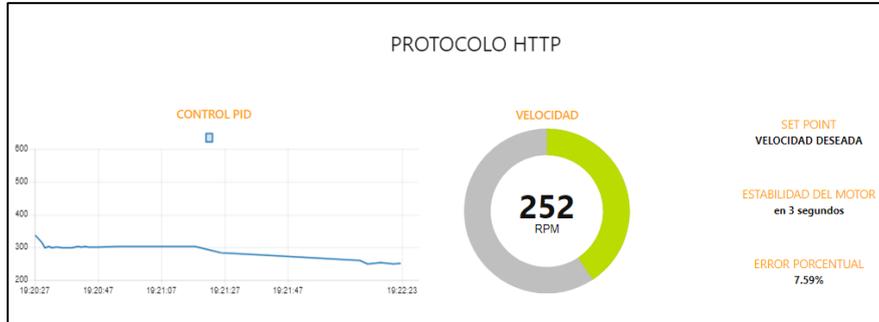


Fig. 29. Prueba de Protocolo HTTP (Set Point 252).

En la **Tabla 3.**, se presenta el último consolidado que se realizó en esta sección de pruebas. Después de analizar la tabla se presta más atención en el caso del Set Point de 380 RPM no se estabiliza, pero el error porcentual es de 0.33, como ya se mencionó esto sucede debido a la ventana de histéresis.

PROTOCOLO HTTP		
SET POINT (RPM)	ESTABILIDAD DEL MOTOR (segundos)	Error porcentual
280	83	0.72 %
380	-	0.33 %
482	79	0.66 %
399	56	2.34 %
351	61	1.03 %
301	26	2.99 %
252	3	7.59 %

Tabla 3. Pruebas protocolo HTTP.

En la Figura 30 se presenta un gráfico comparativo de los 3 protocolos empleados para el control de un motor de DC. Para poder realizar la comparación se obtuvo el promedio que se demora en estabilizar cada protocolo. El mejor promedio obtenido es de 2.76 segundos que corresponde al protocolo MQTT, el siguiente con mejor promedio es el Protocolo CoAP con 3.72 segundos, mientras que el Protocolo HTTP tiene un promedio de estabilización de 51.33 segundos. Sin embargo, también se toma en cuenta el error porcentual ya que a pesar de que el Protocolo MQTT se estabiliza en un menor tiempo el Protocolo CoAP tiene un promedio de error porcentual más bajo, 0.136 % vs 0.15 %, por último, se encuentra el protocolo HTTP con un error porcentual aceptable de 2.237 %.

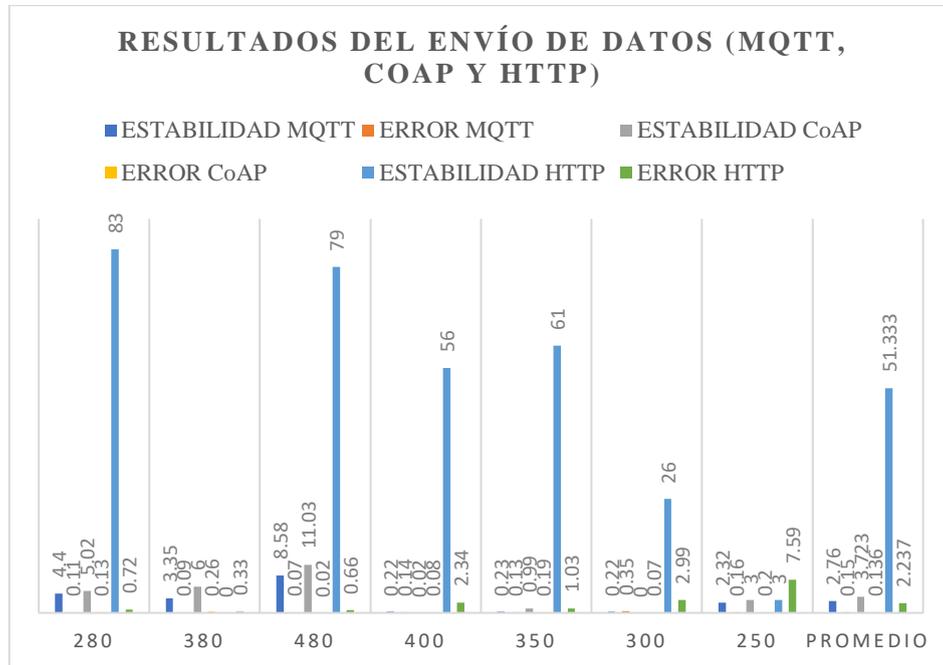


Fig. 30. Comparación de los protocolos MQTT, CoAP y HTTP.

Pruebas con el analizador Wireshark.

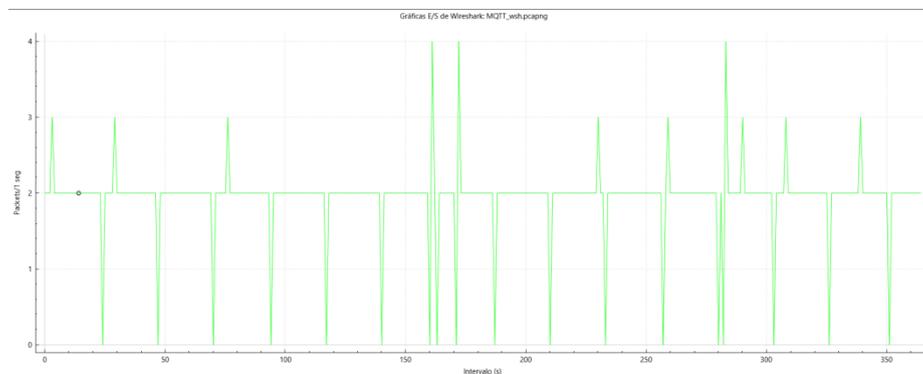


Fig. 31. Prueba con el analizador Wireshark del protocolo MQTT

Para analizar de una mejor manera se emplea el analizador Wireshark, en la Figura 31, se presentan las pruebas para el protocolo MQTT en el cual los intervalos de tiempo son más anchos entre los picos de los paquetes, indicando que el protocolo envía datos en menos paquetes a intervalos más largos.

Por otra parte, la cantidad de paquetes oscila entre 2, 3 y hasta 4 veces. Esto refleja la eficiencia de MQTT en la transmisión de datos ocupando una menor cantidad de recursos de la red.

De igual manera se realiza la misma prueba al protocolo CoAP, el resultado de la misma se presenta en la Figura 32, en este caso los intervalos de tiempo son más cortos en comparación con el protocolo MQTT, esto se debe a que CoAP generalmente tiene una comunicación más frecuente.

La cantidad de paquetes puede ser más variable, llegando a veces hasta 5 y 6, lo cual indica una interacción más intensiva en términos de solicitud y respuesta que MQTT.

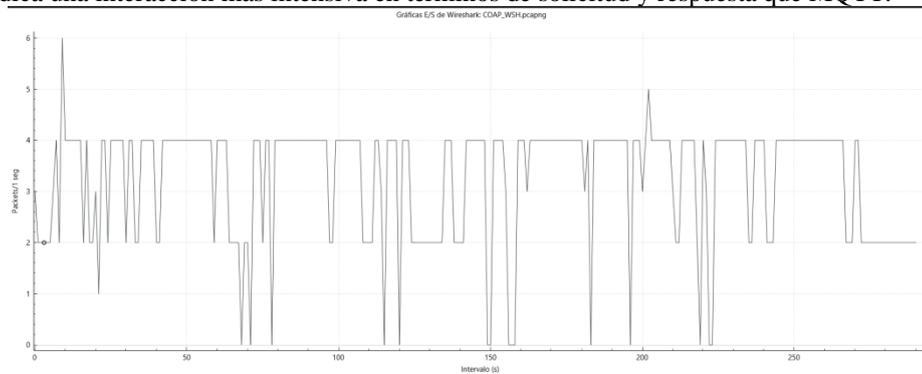


Fig. 32. Prueba con el analizador Wireshark del protocolo CoAP.

Por último, en la Figura 33, se presentan los resultados de la prueba realizada al protocolo HTTP, en la cual se puede observar que los intervalos de envío y recepción de datos son más cortos y con una cantidad de paquetes más alta en comparación con MQTT y CoAP.

Además, se observan picos más altos en la cantidad de paquetes, llegando más veces a 6 en algunos intervalos de tiempo, lo cual refleja la naturaleza más intensiva en datos de HTTP.

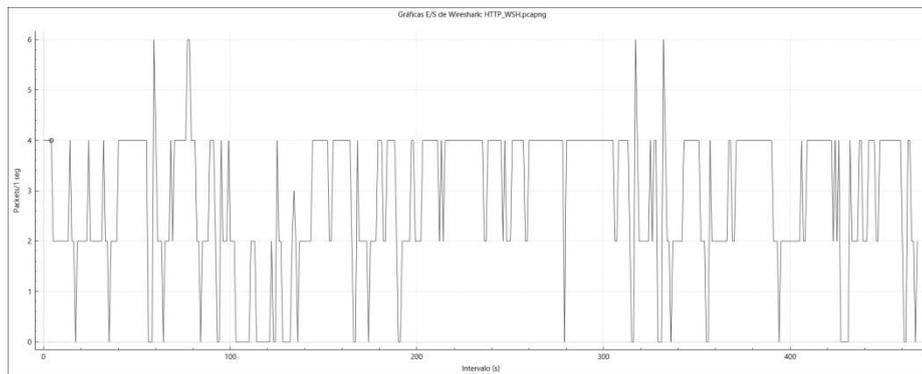


Fig. 33. Prueba con el analizador Wireshark del protocolo HTTP.

4. Discusión

Dentro de la implementación física se tiene un sistema que se podría emplear a gran escala o industrialmente sin problema alguno. Claro está que la limitante dentro de todo el mundo del IoT es la energía y el internet. Últimamente en Ecuador se han presentado problemas en cuanto a la generación y distribución eléctrica, por lo cual inevitablemente se pierde la conexión al tener ausencia de energía.

MQTT utiliza un modelo de publicación/suscripción donde los dispositivos (clientes) pueden publicar mensajes en temas (topics) y otros dispositivos interesados (suscriptores) pueden suscribirse a estos temas para recibir los mensajes relevantes. Este modelo asincrónico permite una comunicación eficiente y distribuida entre los dispositivos conectados.

MQTT es ampliamente utilizado en una variedad de aplicaciones IoT, como monitoreo ambiental, gestión de edificios inteligentes, agricultura de precisión, seguimiento de activos, y más. Su capacidad para manejar datos en tiempo real con baja latencia lo hace adecuado para aplicaciones donde la respuesta rápida y la fiabilidad son críticas.

Inicialmente, MQTT carece de seguridad integrada, pero puede implementarse sobre TLS (Transport Layer Security) para cifrar y autenticar las comunicaciones, asegurando así la confidencialidad e integridad de los datos transmitidos. Esto es crucial para proteger la información sensible en entornos IoT.

Comparado con protocolos como HTTP y CoAP, MQTT se destaca por su eficiencia en el uso de la red y la capacidad de mantener conexiones persistentes de manera eficiente. Especialmente en entornos donde los dispositivos pueden estar inactivos durante períodos prolongados, MQTT minimiza el overhead de la conexión y facilita una comunicación fluida entre dispositivos.

5. Conclusiones

Se desarrolló un sistema de control en red inalámbrico (WNCS) para el control automático de un motor DC, implementando un control PID y realizando pruebas exhaustivas de estabilidad y error. La investigación y actualización de conocimientos en sistemas WNCS, protocolos IoT y programación basada en flujo de datos fueron fundamentales para el éxito del proyecto.

Acorde a la Figura. 30, se puede observar que la implementación de los protocolos de comunicación empleados en el presente proyecto cumple con el objetivo de controlar un motor DC en red inalámbrica con tiempos de respuesta y errores porcentuales aceptables. El error porcentual del protocolo MQTT es del 0.15%, el del protocolo

CoAP es del 0.136% y el del protocolo HTTP es del 2.237%. Aunque el error porcentual del protocolo HTTP es mayor en comparación con MQTT y CoAP, el sistema aún logra estabilizarse y mantener un control efectivo del motor.

En cuanto a la estabilidad del sistema, el protocolo MQTT es el más rápido en alcanzar la estabilidad, con un tiempo promedio de 2.76 segundos, seguido de CoAP con 3.73 segundos y HTTP con 51.33 segundos. Esta diferencia se debe a la eficiencia de MQTT en la gestión de paquetes y la optimización del ancho de banda. Estos resultados son cruciales para aplicaciones que requieren respuestas rápidas y ajustes dinámicos en tiempo real.

En general se observa que cuando el sistema disminuye la velocidad desde un valor alto, tarda menos tiempo en estabilizarse en comparación con aumentos de velocidad. Este comportamiento es consistente en los tres protocolos y es importante para diseñar sistemas de control que respondan eficientemente a variaciones rápidas en la carga.

El análisis de las gráficas de Wireshark revela que MQTT y CoAP utilizan menos paquetes y añaden menos datos adicionales por mensaje (sobrecarga) en comparación con HTTP, (como se observa en las figuras 31,32 y 33), lo que contribuye a la menor latencia y mayor eficiencia en la comunicación. Esto es especialmente relevante en aplicaciones con restricciones de ancho de banda y energía.

Este estudio proporciona una evaluación exhaustiva de la eficiencia y adecuación de los protocolos HTTP, CoAP y MQTT en aplicaciones específicas de control de motores. Los hallazgos destacan la importancia de elegir el protocolo adecuado según las necesidades de la aplicación, considerando factores críticos como tiempo de respuesta, latencia, consumo de ancho de banda y fiabilidad de la comunicación.

En resumen, el presente estudio demuestra que la implementación de los protocolos MQTT, CoAP y HTTP en un sistema de control de motor DC permite alcanzar un control eficiente y estable, con errores porcentuales aceptables y tiempos de respuesta adecuados. Los resultados obtenidos validan la aplicabilidad de estos protocolos en diversas aplicaciones industriales y educativas, resaltando la importancia de la investigación y la innovación en el campo de IoT y control de sistemas.

6. Referencias

1. mqtt-v3.1.1. (2014). MQTT Version 3.1.1. Retrieved January 6, 2018, from <http://docs.oasisopen.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>
2. About the Apache HTTP Server Project - The Apache HTTP Server Project. (2018). Retrieved September 1, 2018, from http://httpd.apache.org/ABOUT_APACHE.html

3. Baheti, R., & Gill, H. (2011). Cyber-physical Systems. The Impact of Control Technology, (1). <http://doi.org/10.1145/1795194.1795205>
4. Blanco, R., Fontrodona, J., & Poveda, C. (2017). La industria 4.0: el estado de la cuestión. *Economía Industrial*.
5. Internet Society, I. (2015). the Internet of Things :, (October). Retrieved from www.internetsociety.org
6. Pickdata. (21 de Octubre de 2019). Pickdata. Obtenido de <https://www.pickdata.net/es/noticias/mqtt-vs-coap-mejor-protocolo-iot>
7. Shelby, Z. (28 de Junio de 2013). Datatracker. Obtenido de <https://datatracker.ietf.org/doc/html/draft-ietf-core-coap-18#page-9>
8. Silva, André Luis de Souza y Regina Célia Marques Freitas Silva. "PROTOCOLO HTTP X PROTOCOLO HTTPS". *Nucleus* 6, n.º 1 (20 de abril de 2009). <http://dx.doi.org/10.3738/1982.2278.146>.
9. Huang, Xinyue, Anmin Zhou, Peng Jia, Luping Liu y Liang Liu. "Fuzzing the Android Applications With HTTP/HTTPS Network Data". *IEEE Access* 7 (2019): 59951–62. <http://dx.doi.org/10.1109/access.2019.2915339>.
10. Baccelli, François, Augustin Chaintreau, Danny De Vleeschauwer y David R. McDonald. "HTTP turbulence". *Networks & Heterogeneous Media* 1, n.º 1 (2006): 1–40. <http://dx.doi.org/10.3934/nhm.2006.1.1>.
11. B. M. Biswajeetan Mishra, "Evaluating and analyzing mqtt brokers with stress-testing," Article, 2020.
12. F. M. Cerdà, "Demostrador arquitectura publish/subscribe con mqtt," Tesis final de grado-UPC, 2018.
13. L. Llamas. (2019) ¿qué es mqtt? su importancia como protocolo iot. Last visited: 2020-11-1. [Online]. Available: <https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot/>
14. M. Serozhenko. (2017) Mqtt vs. http: which one is the best for iot? Last visited: 2020-10-30. [Online]. Available: <https://medium.com/mqtt-buddy/mqtt-vs-http-which-one-is-the-best-for-iot-c868169b3105#:~:text=MQTT%20is%20data%20centric%20whereas,always%20optimized%20for%20mobile%20devices.&text=Besides%2C%20publish%2Fsubscribe%20model%20provides,relability%20of%20the%20whole%20system.>
15. Chapter 2, Block Diagram of EM Systems, pp 21, 43(23) (Fuchs E.F., Masoum M.A.S. (2011) Block Diagrams of Electromechanical Systems. In: Power Conversion of Renewable Energy Systems. Springer, Boston, MA)
16. Kamp, Poul-Henning. "HTTP/2.0". *Communications of the ACM* 58, n.º 3 (23 de febrero de 2015): 40–42. <http://dx.doi.org/10.1145/2717515>.
17. M. Calabretta, "MQTT-AUTH: A TOKEN-BASED SOLUTION TO ENDOW MQTT," *JOURNAL OF COMMUNICATIONS SOFTWARE AND SYSTEMS* , vol. 12, Dec. 2018.
18. B. Canet, "Fundamentos de seguridad de MQTT en la automatización industrial ," Jul. 14, 2020.
19. L. del Valle Fernández, "Introducción a Node-Red,"

<https://programarfacil.com/blog/raspberry-pi/introduccion-node-red-raspberrypi/>,
2019.

20. I. Porro Sáez, "IoT: protocolos de comunicación, ataques y recomendaciones,"
[https://www.incibe-cert.es/blog/iot-protocolos-comunicacion-ataques-](https://www.incibe-cert.es/blog/iot-protocolos-comunicacion-ataques-y-recomendaciones)
[yrecomendaciones](https://www.incibe-cert.es/blog/iot-protocolos-comunicacion-ataques-y-recomendaciones), Feb. 07, 2019.