



WebSockets para lectura rápida de etiquetas RFID de Ultra Alta Frecuencia

Durán Bayona, José Daniel

Universidad Francisco de Paula Santander
Cúcuta, Colombia
0000-0003-2968-3901

Quintero Ayala, Sergio Iván

Universidad Francisco de Paula Santander
Cúcuta, Colombia
0000-0001-7968-3665

Castro Casadiego, Sergio Alexander

Universidad Francisco de Paula Santander
Cúcuta, Colombia
0000-0003-0962-9916

Niño Rondón, Carlos Vicente

Universidad Francisco de Paula Santander
Cúcuta, Colombia
0000-0002-3781-4564

Sandoval Martínez, Gloria Esmeralda

Universidad Francisco de Paula Santander
Cúcuta, Colombia
0000-0002-8644-3835

Abstract

RFID readings must be obtained in the shortest possible time. The method for achieving reads and validations, including tagging, programming logic and data collection process, is presented. Key characteristics of the chosen technologies are presented in the results, and it was concluded that Javascript is the ideal programming language due to its asynchronous capacity, together with WebSockets as a communication protocol complies with fast readings.

Keywords: RFID, UHF, IoT, WebSockets, Javascript.

Resumen

Las lecturas RFID deben ser obtenidas en el menor tiempo posible. Se presenta el método para lograr lecturas y validaciones, incluyendo el etiquetado, la lógica de programación y el proceso de tomas de datos. En los resultados se presentan características claves de las tecnologías elegidas, y se concluyó que Javascript es el lenguaje de programación ideal debido a su capacidad asíncrona, junto a WebSockets como protocolo de comunicación cumplen con realizar lecturas rápidas.

Palabras claves: Identificación por Radio Frecuencia, Ultra Alta Frecuencia, Internet de las Cosas, WebSockets, Javascript.

1. INTRODUCCIÓN

La identificación por radiofrecuencia (*Radio Frequency IDentification*) es una de las tecnologías usadas para controlar la movilidad y el acceso del personal de una empresa, y se ha convertido en el estándar para los sistemas de administración de recursos, siendo empleada en el inventario de contenedores de carga, bodegas [1], cadenas de reparto [2], almacenes y bibliotecas [3], en donde el uso de etiquetas *RFID* pasivas en un entorno controlado permite monitorear el flujo de entrada y salida de los recursos.

Las lecturas de las etiquetas de identificación por radiofrecuencia deben ser realizadas en el menor tiempo posible para que los sistemas de gestión de recursos en los que se implementan puedan responder a tiempo, con un sistema de alerta o con información precisa para quienes administran los recursos, garantizando el control del flujo donde sea necesario y previendo movimientos o retiros no autorizados de los mismos, siendo esta información validada con una de base de datos que permite determinar qué tipo de lectura fue la que se realizó y en qué parte del lugar se hizo tal lectura. Es importante que estas lecturas se hagan en el menor tiempo posible, con el fin de que la validación se dé cuando aún sea relevante para el flujo de información del sistema.

Por tanto, en este documento analizamos cómo se pueden lograr lecturas y validaciones de la manera más rápida posible, empleando tecnologías de código abierto para presentar adicionalmente un factor especial en el tema económico.

2. MÉTODO

A. Etiquetado del material

Para el etiquetado del material se implementaron etiquetas *RFID* de Ultra Alta Frecuencia (*Ultra High Frequency*) en 10 libros, 5 de ellos ubicados en una habitación denominada Estantería A y 5 de ellos en otra habitación denominada Estantería B, con un lector *UHF RFID* ubicado entre ellas, con el fin de hacer una analogía a dos salas de una biblioteca, en donde se controlan qué libros pertenecen a cada sala y se necesita saber cuándo es retirado de una sala e ingresa a otra.

B. Programación

Siguiendo las recomendaciones del fabricante del lector *UHF RFID*, se programó el dispositivo en lenguaje C++, desde el cual se envían a través del puerto USB con comunicación serial las lecturas obtenidas, y desde este punto se puede implementar el lenguaje de programación deseado para interpretar la información del puerto serial y realizar los procesos pertinentes al sistema.

Se tomó la decisión de emplear y programar las pruebas de lectura con Python y Javascript (a través del framework NodeJS), ambos de código abierto, debido a que son 2 de los lenguajes que tienen mayor cantidad de librerías, complementos, ejemplos y soporte de la comunidad (ayuda no oficial).

La programación a implementar seguía la siguiente lógica procedimental, la cual tendría que realizar de manera repetitiva o continuamente:

1. Esperar un nuevo dato en el puerto serial.
2. Lectura del puerto serial.
3. Validación o interpretación de la lectura (cantidad de caracteres que correspondan al estándar *RFID EPC* [4]).
4. Envío de la lectura a la base de datos del sistema.
5. Interpretar la respuesta de la base de datos.
6. Presentar el resultado al usuario o al dispositivo de alerta.
7. Comenzar nuevamente.

Esta lógica, de manera complementaria y deseada, tendría que poderse realizar para cada nueva etiqueta leída, sin importar en qué parte de la lógica se encontrara actualmente la anterior lectura.

C. Protocolos de comunicación

Con el fin de enviar la información obtenida con la programación a los demás puntos del sistema se hizo necesario evaluar tecnologías de Internet de las Cosas (*IoT*) [5], de las cuales se eligieron dos en específico para su estudio, MQTT, el cual trabaja en la capa 4 del

modelo OSI [6], y WebSockets, el cual trabaja en la capa 7 del modelo [7]. Para evaluar los protocolos de comunicación se realizaron pruebas de transferencia con poca y con gran ocupación del tráfico de la red (creado de manera artificial), con el fin de obtener dos características fundamentales: confiabilidad de la información (datos veraces o completos), y la velocidad de transmisión.

D. Tomas de datos

En la toma de datos se realizaron movimientos de libros etiquetados de manera individual (uno a uno) y en conjunto (grupos de 2 a 5 libros al tiempo), con el fin de comprobar que el flujo de información se cumpliera sin importar la cantidad de recursos que se intentaran leer en simultáneo.

Este proceso se repitió 15 veces por cada movimiento y los valores mínimos obtenidos en estas tomas de datos se presentan en los resultados.

3. RESULTADOS

A continuación, se presentan los resultados obtenidos siguiendo el método planteado, con los cuales se procede a seleccionar las tecnologías más acordes a las necesidades.

A. Lenguajes de programación

Las tomas de datos realizadas permiten comparar algunas características fundamentales entre los lenguajes de programación de Javascript (NodeJS) y Python, dicha comparativa se evidencia en la Tabla 1. La tabla refleja los valores mínimos globales obtenidos luego de las tomas de datos en las diferentes etapas de las pruebas.

Tabla 1. Comparativa de características de Javascript y Python.

Característica	Javascript	Python
Tiempo de respuesta en las lecturas	52 ms	518 ms
Capacidad para realizar lecturas simultáneas	Sí	No
Uso de recursos del sistema	Código asíncrono 84 MB	Código síncrono 23 MB

B. Protocolos de comunicación

Las características principales analizadas con los protocolos de comunicación de MQTT y WebSockets se ven comparadas en la Tabla 2. La tabla refleja tanto la interpretación de la documentación propia del protocolo como los valores mínimos globales obtenidos en las tomas de datos.

Tabla 2. Comparativa de características de MQTT y WebSockets.

Característica	MQTT	WebSockets
Reliabilidad	Mayor	Menor
(Integridad de los datos)	(Garantizada)	(Pérdidas de conexión)
Velocidad de transmisión	102 ms	23 ms

4. CONCLUSIONES

Debido a que logra realizar lecturas en simultáneo con un menor tiempo de respuesta, se eligió a Javascript (NodeJS) como lenguaje de programación a emplear, siendo hasta 10 veces más rápido que el tiempo obtenido con Python.

Se eligieron los WebSockets como protocolo de comunicación a usar debido a su velocidad para transmitir la información a los dispositivos, presentando velocidades hasta 4 veces mayores que MQTT.

El lenguaje de programación y el protocolo de comunicación seleccionado satisfacen la necesidad de conseguir lecturas rápidas de identificación por radiofrecuencia y de validar los valores leídos con un sistema de gestión de recursos, ofreciendo además una alternativa de bajo costo ya que emplea tecnologías y licencias de software libre y abierto (FOSS – Free Open Source Software).

Referencias

- [1] S. F. Pane, R. M. Awangga, and B. R. Azhari, "Qualitative evaluation of RFID implementation on warehouse management system," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 16, no. 3, pp. 1303–1308, 2018, doi: 10.12928/TELKOMNIKA.v16i3.8400.
- [2] A. K. Biswal, M. Jenamani, and S. K. Kumar, "Warehouse efficiency improvement using RFID in a humanitarian supply chain: Implications for Indian food security system," *Transp. Res. Part E Logist. Transp. Rev.*, vol. 109, no. June 2017, pp. 205–224, 2018, doi: 10.1016/j.tre.2017.11.010.
- [3] F. Nisha, "Implementation of RFID technology at defence science library, DESIDOC: A case study," *DESIDOC J. Libr. Inf. Technol.*, vol. 38, no. 1, pp. 27–33, 2018, doi: 10.14429/djlit.38.1.12351.
- [4] G. EPCglobal, "EPC Tag Data Standard Gen 2 RFID Tags," pp. 1-200., 2017, [Online]. Available: https://www.gs1.org/sites/default/files/docs/epc/GS1_EPC_TDS_i1_10.pdf.
- [5] P. Kayal and H. Perros, "A Comparison of IoT application layer protocols through a smart parking implementation," *2017 20th Conf. Innov. Clouds, Internet Networks*, pp. 331–336, 2020.
- [6] R. Denenberg, "Open Systems Interconnection," *Libr. Hi Tech*, vol. 3, no. 1, pp. 15–26, 1985, doi: 10.1108/eb047578.
- [7] G. M. B. Oliveira *et al.*, "Comparison between MQTT and WebSocket Protocols for IoT Applications Using ESP8266," *2018 Work. Metrol. Ind. 4.0 IoT, MetroInd 4.0 IoT 2018 - Proc.*, pp. 236–241, 2018, doi: 10.1109/METROI4.2018.8428348.

Autorización y Licencia CC

Los autores autorizan a APANAC XVIII a publicar el artículo en las actas de la conferencia en Acceso Abierto (Open Access) en diversos formatos digitales (PDF, HTML, EPUB) e integrarlos en diversas plataformas online como repositorios y bases de datos bajo la licencia CC:

Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) <https://creativecommons.org/licenses/by-nc-sa/4.0/>.

Ni APANAC XVIII ni los editores son responsables ni del contenido ni de las implicaciones de lo expresado en el artículo.