

DISEÑO DE UN NODO DE REGISTRO DE ACCESO RFID BASADO EN ARQUITECTURA PUBLICADOR-SUSCRIPTOR EN AL RED IOT DEL INSTITUTO TECNOLÓGICO DE CHIHUAHUA

Edgar Trujillo Preciado, José Alberto Domínguez Terrazas, Oscar Beltrán Gómez,
Luis Adrian Anchondo Carreón, Ana Paola Cardona Valenzuela, Emiliano Pérez Dyc

Instituto Tecnológico de Chihuahua

Departamento de Eléctrica Electrónica, Departamento de Ciencias Básicas

Av. Tecnológico 2909

Col. 10 de mayo, Chihuahua, Chih. CP 31310

Tel. (614) 201 2000

{ edgar.tp, jose.dt, oscar.bg, l22061088, l22061031, l22061061 } @ chihuahua.tecnm.mx

RESUMEN

El presente artículo describe el diseño de un nodo de registro de acceso a espacios físicos utilizando una tarjeta de desarrollo ESP32 cuyo objetivo es mejorar la trazabilidad y la gestión del registro de usuarios a las instalaciones del Instituto Tecnológico de Chihuahua, una arquitectura de comunicación de red tipo publicador-suscriptor. Implementando estrategias que proponen dar respuesta a la necesidad de contar con un sistema confiable, adaptable y funcional que responda incluso en condiciones de conectividad intermitente. Para ello, se implementa un mecanismo de autenticación dual junto con la utilización de métodos para la sincronización de tiempo, con la finalidad de mantener la precisión temporal de los registros. Se sigue un diseño modular y replicable del nodo para integrarlo de manera sencilla en futuros puntos de control dentro de una red IoT Institucional.

Palabras Clave: RFID, ESP32, publicador-suscriptor, nodo, Wiegand

ABSTRACT

This article describes the design of a registration node for access to physical spaces using an ESP32 development card whose objective is to improve the traceability and management of user registration to the facilities of the Technological Institute of Chihuahua, a network communication architecture publisher-subscriber type. Implementing strategies that propose responding to the need to have a reliable system. adaptable and functional that responds even in conditions of intermittent connectivity. To this end, a dual authentication mechanism is implemented along with the use of methods for time synchronization, in order to maintain the temporal precision of the records. A design is modular and replicable of the node to easily integrate it into future control points within an Institutional IoT network.

Keywords: RFID, ESP32, publisher-subscriber, node, Wiegand

1. INTRODUCCIÓN

El uso de tecnologías de Internet de las Cosas (IoT) para el control de acceso a espacios se ha vuelto relevante en los últimos años, específicamente en instituciones educativas donde los recursos son compartidos por varios usuarios, por lo que su administración requiere eficiencia y seguridad. En el Instituto Tecnológico de Chihuahua se tiene el desafío de dirigir de forma adecuada el uso de sus instalaciones, garantizando un acceso controlado y

documentado que fomente el orden y la recopilación de información pertinente.

En los últimos años, se tiene el creciente desarrollo de soluciones basadas en RFID, pero, muchos de estos sistemas presentan inconvenientes relacionados con la dependencia constante de conectividad, baja capacidad de adaptación a entornos cambiantes y la carencia de mecanismos que garanticen la integridad de los datos cuando el nodo pierde conectividad temporalmente [1,2]. De igual manera, pocos de estos sistemas contemplan un diseño modular y escalable que permita su replicación en diferentes puntos dentro de una misma red de acceso.

El presente trabajo propone el desarrollo de un nodo de registro de acceso dentro de una arquitectura publicador-suscriptor, implementado en una placa ESP32, que incorpora un sistema de autenticación dual (tarjetas RFID y códigos por teclado físico Wiegand), junto con un esquema de sincronización temporal eficiente. A diferencia de otros enfoques, este nodo realiza la sincronización de tiempo mediante NTP al arranque y luego mantiene la referencia con un RTC local, permitiendo operar de forma autónoma ante pérdidas de conectividad sin comprometer la precisión de los registros [3].

1.1 Estado del Arte

El uso del ESP32 combinado con lectores RFID ha sido ampliamente explorado en aplicaciones de control de acceso y gestión de asistencia en entornos IoT debido a su bajo costo, versatilidad y escalabilidad. Ejemplos recientes incluyen sistemas de registro estudiantil con integración en Google Sheets [4], control automático de portones con conectividad Wi-Fi y operación offline [5], y cerraduras inteligentes con autenticación dual mediante RFID y huella digital [6]. Estas soluciones reflejan la tendencia hacia mecanismos multifactoriales que incrementan la seguridad y la adaptabilidad en contextos educativos e industriales.

En paralelo, diversos estudios han señalado vulnerabilidades inherentes a la tecnología RFID. [7] documentan más de veinte

vectores de ataque comunes y sugieren medidas de mitigación, mientras que [8] proponen un protocolo ligero de autenticación validado formalmente para mejorar la confiabilidad de sistemas distribuidos. Finalmente, investigaciones recientes amplían las capacidades del ESP32 hacia la localización precisa con el protocolo FTM (IEEE 802.11mc), validando su potencial en aplicaciones híbridas de control y posicionamiento [9].

2. DESARROLLO

2.1. Funcionalidad del nodo

Se realizó la implementación de un nodo basado en una placa ESP32 con el propósito de registrar el acceso de personas a distintas áreas utilizando un sistema de identificación electrónica (RFID), con la capacidad de enviar la información recabada hacia un servidor central, donde se almacenan, gestionan y procesan los registros de forma remota.

Se eligió la tarjeta de desarrollo, Expressif ESP32 Dev Module, ya que presenta ventajas en múltiples aspectos, primordialmente la variedad de métodos para conectividad que posee este dispositivo, pues cuenta con módulos de comunicación WIFI, Bluetooth y otros protocolos de comunicación que facilitan su uso en diferentes aplicaciones en el campo del IoT. Otros aspectos de gran importancia en la elección de este elemento recaen en su bajo consumo energético, lo que resulta adecuado para el sistema, cuenta con un amplio soporte de múltiples periféricos o componentes externos. Además, cuenta con un módulo de aceleración criptográfico por hardware, pudiendo manejar encriptación con SHA, RSA, AES y RNG, de hasta 4096 bits, proporcionando la seguridad necesaria en la transferencia de datos; además admite cifrado y descifrado de código dinámico, lo que hace que disminuya la probabilidad de la información sea vulnerable [10]; aunado a esto, el bajo costo del microcontrolador comparado con otros sistemas de esta índole [11].

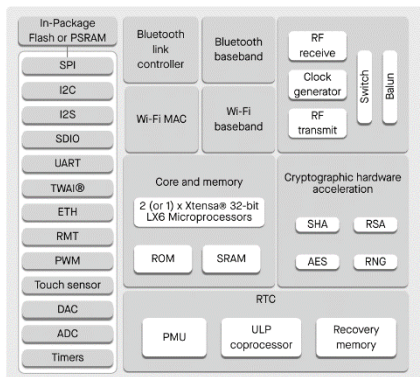


Figura 1. Diagrama de bloques funcional ESP32 [11]

El sistema de identificación empleado combina un lector RFID con teclado físico incorporado, el cual opera bajo el protocolo Wiegand. Esta configuración admite dos formas de autenticación:

mediante tarjetas con tecnología RFID, y mediante la introducción de un código numérico a través del teclado. Gracias a esta doble funcionalidad, se ofrece una mayor flexibilidad al momento de registrar los accesos, ya sea con credenciales físicas o con un número de control previamente asociado al número de tarjeta obtenido del RFID de la credencial.

Una de las características clave del nodo es su capacidad de comunicación unidireccional con el servidor central, ya que actualmente solo está diseñado para enviar datos relacionados con los accesos registrados. No obstante, durante la primera conexión, el nodo realiza una petición de sincronía en la fecha y hora actuales a un servidor de tiempo (NTP) con el fin de configurar su reloj en tiempo real (RTC) como se ve en la figura 1. Esta sincronización se realiza una única vez al inicio y tiene como objetivo asegurar que, en caso de pérdida temporal de conectividad, el dispositivo pueda seguir operando de manera autónoma sin comprometer la precisión temporal de los eventos registrados [3].

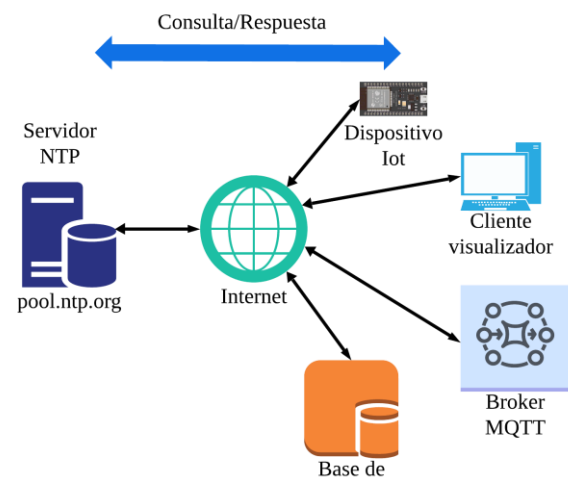


Figura 2. Sincronización de relojes de los dispositivos

Este mecanismo de respaldo evita la dependencia continua del servidor para el control del tiempo y garantiza la integridad de los datos recopilados, incluso en condiciones de red inestable. Al contar con el RTC y memoria local, el nodo puede seguir funcionando sin interrupciones, y cuando la conexión se restablece, reanuda el envío de información al servidor sin pérdida de datos.

La información enviada por el nodo incluye tres elementos principales: el número de tarjeta leído por el sensor RFID o el código ingresado a través del teclado, un identificador del edificio o área donde se encuentra el dispositivo, y la fecha y hora del evento. Esta información es transmitida al servidor central, el cual no la almacena directamente, sino que actúa como intermediario para redirigirla a una base de datos conectada al sistema. Allí es

donde los datos quedan registrados de forma permanente para su posterior consulta, análisis o integración.

El diseño implementado permite que este nodo sea fácilmente replicable en distintas ubicaciones, manteniendo un comportamiento consistente y confiable. Esto facilita su implementación en diferentes puntos de control de acceso sin necesidad de modificaciones complejas, ya que cada nodo conserva una estructura de operación uniforme y puede integrarse rápidamente a la red existente mediante simples configuraciones de identificación y conexión.

2.2. Estructura lógica del código

La implementación del código responsable del manejo del nodo es bajo el paradigma de programación estructurada siguiendo un diseño modular el cual divide las responsabilidades del sistema en diferentes etapas delimitadas [12]. Si bien el lenguaje de programación utilizado es C++ con un enfoque para la programación de microcontroladores, la lógica de programación se puede explicar de manera general por 4 etapas concretas.

a) Inicialización y sincronización temporal del sistema

En el arranque del nodo se hace una inicialización de los periféricos involucrados en el funcionamiento, tanto la interfaz Wiegand para el registro de los datos censados como la comunicación con el módulo RTC para la referencia de tiempo de registro de datos. El paso siguiente inmediato es la conexión a la red por medio de Wi-Fi y se hace un acceso a un servidor NTP de manera temporal para el ajuste de la fecha y hora de referencia para garantizar precisión temporal de los eventos publicados.

b) Lectura de eventos de entrada

A modo de mejorar la robustez y versatilidad del proceso de registro de acceso se implementan 2 métodos distintos de ingreso de datos.

Durante la operación continua, el nodo se mantiene en espera de entradas provenientes del lector Wiegand. Dos tipos de eventos son considerados:

- Lectura de tarjeta RFID: Cuando el dispositivo detecta una tarjeta, se extrae directamente el identificador codificado.
- Ingreso por teclado numérico: Cada tecla presionada se interpreta y almacena en una variable acumuladora. Al presionar la tecla de confirmación (por ejemplo, #), se valida si la secuencia numérica tiene la longitud adecuada (8 dígitos). Si es así, el evento se considera válido para su publicación.

La incorporación de un método alternativo (como el ingreso manual de un número de control por teclado) permite mantener la

operatividad del sistema aun cuando una de las interfaces esté inhabilitada o no disponible. Esta estrategia permite al usuario registrar acceso en escenarios donde el portador no cuenta con su tarjeta institucional, pero sí cuenta con su número de control de identificación. Esto en anteposición a las desventajas de almacenar estos datos en la memoria del microcontrolador como fallos del controlador o la memoria del mismo. [13]

La autenticación y validación de los datos ingresados no se procesa en el nodo, si no en el cliente suscrito a la información publicada por el nodo

c) Generación y publicación del mensaje

Una vez detectado un evento válido por cualquiera de los 2 medios establecidos se construye un mensaje en formato estructurado con los siguientes campos:

- Código (tarjeta o teclado)
- Identificador de edificio
- Fecha y hora del evento

Este mensaje se prepara como cadena de texto (un avance podría ser un envío de XML o json) separada por comas y, en una implementación completa, se publica hacia el tópico correspondiente. En la etapa de pruebas, el envío fue simulado mediante impresión por consola serial.

No es necesario publicar la identidad del nodo, simplemente publica los datos, ya que el medio de comunicación diseñado utiliza un intermediario o broker encargado de direccionar la información al canal de comunicación suscrito a los tópicos enviados por el publicador [14] [15].

d) Respuesta y control de salidas

El sistema permite verificar la correcta captura y transmisión de datos mediante indicadores físicos integrados en el módulo del sensor RFID, como lo son un zumbador y un LED, que proporcionan una retroalimentación visual y sonora al usuario.

De manera general el funcionamiento del sistema del nodo puede ser descrito mediante el diagrama de flujo en la figura 2, en la que podemos dividir la operación del sistema en dos partes, la inicialización y sincronización temporal para el RTC y después la ejecución continua de la espera de los datos para su envío al broker MQTT.

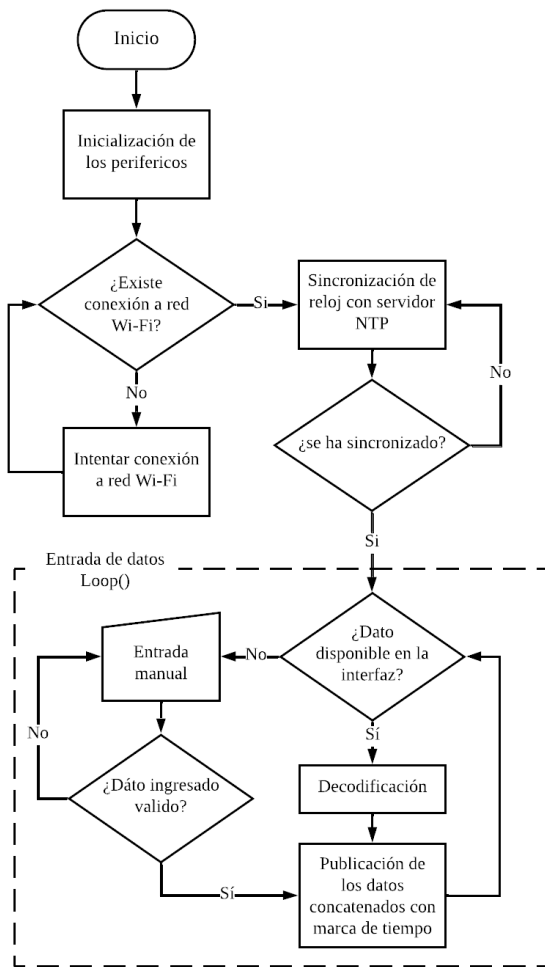


Figura 3. Diagrama de flujo del sistema.

2.3. Diseño físico del nodo

El trazado del nodo se presenta mediante el siguiente diagrama de bloques funcional, donde se ilustran los principales elementos del dispositivo y sus interconexiones. La figura 3 permite la visualización de la estructura general del sistema, incluyendo el elemento de control (ESP32), los elementos de entrada (credenciales e inserción de caracteres propios del usuario), su correspondiente sensor (lector RFID y teclado numérico), así como las salidas del sistema, que comprende indicadores visuales y sonoros (LED/LCD y zumbador) e interfaz de comunicación con el broker.

Adicionalmente, la figura 4 muestra un esquemático aproximado a las conexiones que posee el nodo físicamente, detallando la asignación de pines del microcontrolador a cada módulo periférico. Para fines de prueba y haciendo uso de software de simulación limitado (WOKWI [16]), se implementó un teclado

matricial y una LCD como entrada y salida del sistema, respectivamente.

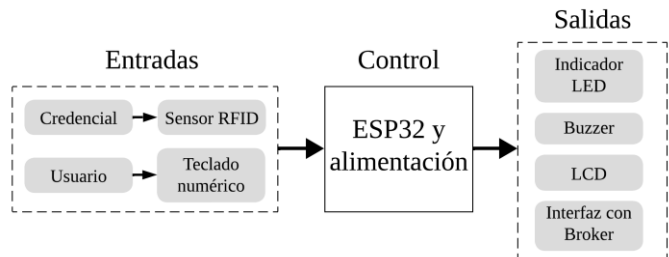


Figura 4. Diagrama de bloques del Nodo

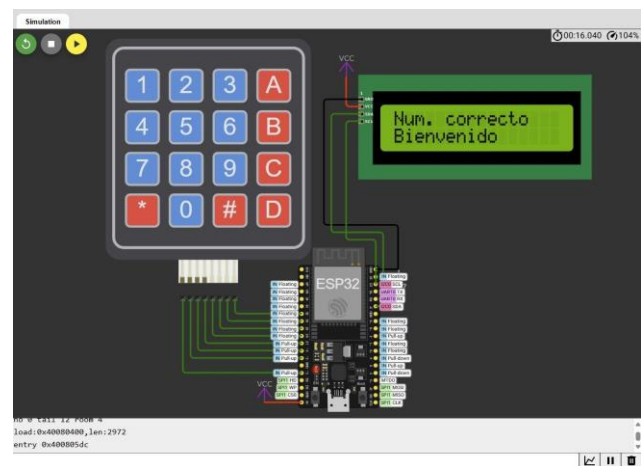


Figura 5. Diagrama de conexionado y simulación por medio de WOKWI [16]

2.4. Arquitectura del sistema general

La figura 5 ilustra el ecosistema al cual se conecta el nodo ESP32, estaciones de consulta, clientes de celular o Raspberry Pi; donde se muestra cómo se lleva a cabo la comunicación con el broker rabbitMQ, presentando una descripción general de la arquitectura publicador-suscriptor, destacando el flujo de datos entre los distintos componentes del sistema con una comunicación bidireccional entre los elementos.

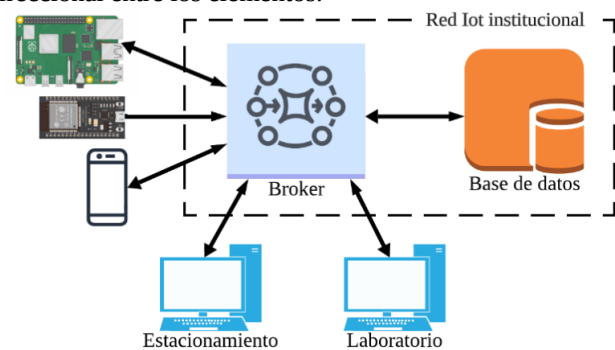


Figura 6. Diagrama de la arquitectura

2.5. Interfaz con el servidor y base de datos

En la arquitectura de conexión de red propuesta, la interfaz entre el nodo de control de acceso y la base de datos está mediada por el broker. Este componente permite desacoplar la adquisición de datos del procesamiento final, lo que facilita tener una arquitectura más flexible y escalable. Esta característica de funcionamiento es particularmente útil en aplicaciones como la que se presenta debido a que permite la comunicación remota con dispositivos de bajo procesamiento.

Esto da la libertad de implementar un sistema de base de datos escalable, seguro, interoperable que permita el manejo de la información generada por los dispositivos IoT que se pueden agregar al sistema en tiempo real. [17]

2.6. Diseño e implementación del hardware

La implementación del hardware de este nodo sigue una estructura muy similar a la propuesta realizada anteriormente en simulación. Consiste en un lector de proximidad RFID con teclado, una pantalla de cristal líquido, y un módulo RTC 1302 conectados a una tarjeta ESP32.

2.7. Resultados

Una infraestructura desacoplada y elástica entre brokers rabbitMQ, permite a productores y consumidores el soporte de varios protocolos (MQTT, AMQP y HTTP/WebSockets) con elasticidad ya que pueden escalar horizontalmente (clustering) al distribuir la carga entre varias sedes (federation/shovel), consumidores y productores según se presente la demanda.

El bajo acoplamiento se fundamenta en una arquitectura dirigida por eventos, Fowler explica que el event-driven [18] desacopla productor/consumidor en tiempo y espacio, facilitando la evolución independiente y composición de las pipelines o colas de trabajo.

En el proyecto descrito en este artículo, estas ideas se ven reflejadas al considerar que los ESP32 y aplicaciones con el cliente de la base de datos pueden variar en número sin afectar la carga de forma negativa, por el contrario, se podrían añadir más elementos si es necesario conservando la forma básica del sistema.

Bajo esta estrategia los consumidores se vuelven reactivos, es decir, reaccionan según el evento que recibe con un comportamiento propio, independiente y aislado.

Por otra parte, Fourati [19] considera que la elasticidad en los esquemas de mensajería también se logra con estrategias de

escalado de consumidores por cola según la demanda, es decir, con la implementación del patrón “work queue”; la cual es una práctica común en el broker, lo cual es coherente con la definición de elasticidad en microservicios que describe la idea de “escalar recursos automáticamente para ajustar costo vs. carga”.



Figura 7. Implementación de hardware del nodo

2.8. Trabajos a futuro

Para desarrollo futuro, se buscará que el nodo no solo funcione como publicador de información, sino que también tenga la capacidad de operar como suscriptor dentro de la red IoT.

En la etapa inicial el nodo está limitado a un canal de comunicación unidireccional, del nodo al servidor; se plantea que la comunicación sea bidireccional, dando la capacidad al nodo de recibir instrucciones o configuraciones del servidor. De esta manera, se convierte al nodo en un suscriptor de un tópico específico, el cual le proporciona información necesaria para restringir físicamente el acceso a usuarios no autorizados a determinadas instalaciones mediante el uso de elementos electromecánicos, como cerraduras magnéticas u otros dispositivos.

Asimismo, se plantea incorporar el uso del formato JSON para la estructuración y envío de los datos, con el fin de mejorar la flexibilidad, compatibilidad e integración con otras plataformas o sistemas de gestión de información.

2.9. Referencias

- [1] A. N. Banagar y R. Khattar, "IoT based Smart Laboratory System," *International Journal of Engineering Research & Technology (IJERT)*, vol. 9, no. 1, pp. 1–4, 2020. [En línea]. Disponible: https://www.researchgate.net/publication/341874078_IoT_based_Smart_Laboratory_System
- [2] R. M. A. Velasco, "IoT Security Solutions for Students' Laboratory with RFID-based Attendance Authentication," *International Journal in Information Technology in Governance, Education and Business*, vol. 5, no. 1, pp. 1–6, 2023. [En línea]. Disponible: <https://ijitgeb.org/ijitgeb/article/download/106/65>
- [3] Amazon Web Services, *AWS Well-Architected Framework: IoT Lens Checklist*, rev. 2022, Best Practice 12.1, p. 35. [Online]. Disponible en : <https://docs.aws.amazon.com/pdfs/wellarchitected/latest/iot-lens-checklist/wellarchitected-iot-lens-checklist.pdf>
- [4] Yavad, A., Dalvi, P., Juwale, H.. (2025). RFID-based attendance management system using ESP32 and Google Sheets. *The Voice of Creative Research*, 4(2), 45–54. <https://doi.org/10.53032/tvcr/2025.v7n2.51>
- [5] Fadlianda, D., Fikry, M., & Nunsina. (2024). Innovative IoT-based automatic gate system with RFID and electro-magnetic lock for secure access. *Proceedings of Malikussaleh International Conference on Multidisciplinary Studies (MICoMS)*, 4. <https://doi.org/10.29103/micoms.v4i.884>
- [6] Ziad, A., Damila, E., & Kurniawati. (2024). Development and implementation of an ESP32 microcontroller and monitoring system for smart door lock using RFID sensor for E-KTP ID and fingerprint based on the Internet of Things. *Proceedings of MICoMS*, 4. <https://doi.org/10.29103/micoms.v4i.908>
- [7] Gavoni, L. (2021). *RFID exploitation and countermeasures*. arXiv. <https://arxiv.org/abs/2110.0009>
- [8] Mansoor, K., Ghani, A., Chaudhry, S. A., Shamshirband, S., Ghayyur, S. A. K., & Mosavi, A. (2019). Securing IoT-based RFID systems: A robust authentication protocol using symmetric cryptography. *Sensors*, 19(21), 4752. <https://doi.org/10.3390/s19214752>
- [9] Barral Vales, V., Fernández, O. C., Domínguez-Bolaño, T., Escudero, C. J., & García-Naya, J. A. (2022). Fine time measurement for the Internet of Things: A practical approach using ESP32. *IEEE Internet of Things Journal*, 9(19), 18305–18318. <https://doi.org/10.1109/JIOT.2022.3158701>
- [10] Espressif Systems "ESP32 Series Datasheet Version 5.0". [En línea]. Disponible en: https://www.espressif.com/en/support/documents/technical-documents?keys=&field_type_tid%5B%5D=492 [Accesado: 05-Sep-2025].
- [11] Espressif Systems, "ESP32-DevKitC V4 User Guide – Overview," esp-dev-kits documentation, s.f. [En línea]. Disponible en: https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32/esp32-devkitc/user_guide.html#overview [Accesado: 25-Jun-2025].
- [12] J. W. Valvano, *Complete Embedded System*, Univ. of Texas at Austin, Chapter 15.2: Modular Design, 2025. [En línea]. Disponible en: https://users.ece.utexas.edu/~valvano/Volume1/E-Book/C15_CompleteEmbeddedSystem.htm
- [13] Babii, A, Samila, A. (2023). Dual Authentication Technique for RFID Access Control Systems with Increased Level of Protection. *Security of Infocommunication Systems and Internet of Things*. 01011. <https://dx.doi.org/10.31861/sisiot2023.1.01011>
- [14] Sagarkar, M., Jain, V., Thakur, R., Jana, P., Kumar, R., Sen, A.. (2021). A Study of Distributed Event Streaming & Publish-Subscribe Systems. <http://dx.doi.org/10.13140/RG.2.2.28795.8016>
- [15] Gopikrishnan S. (2023), "Securing Communication between ESP32-based IoT Devices and Remote Servers: An Analysis of SSL Certificate Verification Methods," *International Journal of Scientific Development and Research (IJS DR)*, vol. 8, no. 5, p. 114, May 2023. [Online]. Disponible en: <https://www.ijedr.org/papers/IJS DR2305015.pdf>
- [16] WOKWI, "Wokwi – World's most advanced ESP32 Simulator," CodeMagic LTD, 2019–2025. [En línea]. Disponible en <https://wokwi.com/>. [Último acceso: jun. 24, 2025].
- [17] Ahmadi, A. F., Hadi, E., Karimi, S., Ahmadi, R., & Hassani, F. (2024). Database management challenges and solutions in IoT environments: A systematic literature review. *Electrical and Computer Experiences*, 2(1), 14–22. <https://doi.org/10.59535/jece.v2i1.257>
- [18] Fowler, M. (2017, 7 de febrero). What do you mean by "Event-Driven"? Retrieved from <https://martinfowler.com/articles/201701-event-driven.html>
- [19] Fourati, Mohamed & Marzouk, Soumaya & Jmaiel, Mohamed. (2024). Cloud Elasticity of Microservices-based Applications: A Survey. <https://doi.org/10.21203/rs.3.rs-3925329/v1>