European Journal of Prosthodontics and Restorative Dentistry (2026) 34(4S), 475–483

IoT System for Monitoring and Electrical Automation using Edge Architecture with AI for Energy Savings in Local Networks

Keywords
Internet of Things (IoT), Edge Computing, local Artificial Intelligence, Offline-First home automation, energy efficiency, MQTT, FastAPI, Flutter.

Authors

[1]**Sergio Suarez Barajas**
Unidades Tecnológicas de Santander, Colombia, ssuarez@correo.uts.edu.co
https://orcid.org/0000-0002-5074-5997

[2]**Leydi Johana Polo Amador**
Unidades Tecnológicas de Santander, Colombia, lpolo@correo.uts.edu.co
https://orcid.org/0000-0002-5335-2814

[3]**Javier Mendoza Paredes**
Unidades Tecnológicas de Santander, Colombia, jmendoza@correo.uts.edu.co
https://orcid.org/0000-0001-6883-398X

[4]**Alejandro Bianchá Hernández**
Unidades Tecnológicas de Santander, Colombia, abiancha@correo.uts.edu.co
https://orcid.org/0000-0003-3248-1198

***Correspondence to**
Sergio Suarez Barajas
Unidades Tecnológicas de Santander, Colombia, ssuarez@correo.uts.edu.co
https://orcid.org/0000-0002-5074-5997

Received-14-05-2026
Revised-18-06-2026
Accepted-23-06-2026
Doi:10.1922/ejprd.v34i4s.1483

Abstract — Commercial home automation systems present a critical dependency on internet connectivity and cloud infrastructure, generating operational vulnerabilities and user data privacy risks. This paper proposes the design and development of EcoMind IoT, a residential energy management prototype based on an Edge Computing architecture with event-driven processing, integrating strictly local artificial intelligence. The system captures real-time electrical telemetry via Shelly 1PM Mini smart switches through the MQTT protocol (Mosquitto), processes data on a central node implemented with FastAPI and Python, stores consumption histories in PostgreSQL, and employs a lightweight local language model (local LLM) to analyze electricity bill tariff information and make autonomous actuator control decisions. The user interface was developed in Flutter under an Offline-First approach, ensuring operational continuity without cloud dependency. Results conceptually validate the proposed architecture, establish the complete system data flow, and demonstrate the technical feasibility of integrating local AI in residential IoT solutions oriented toward energy savings with full data sovereignty.

I. INTRODUCTION

Today, home automation and the Internet of Things (IoT) have profoundly transformed the way users interact with their residential and commercial environments. According to recent estimates, the number of connected IoT devices worldwide will exceed 30 billion by 2030 (International Data Corporation, 2023), which is evidence of the massification of these technologies. However, the vast majority of available business solutions share a fundamental structural problem: their near-total reliance on cloud servers to run automation logic.

This centralized cloud architecture creates at least two critical vulnerabilities. The first is operational disruption: when the internet connection fails, the home automation system immediately loses its ability to automate, leaving the user without intelligent control over their devices (Shi et al., 2016). The second is privacy risk: data on consumption habits, hours of presence, and household routines are transmitted and processed on third-party servers, exposing them to security breaches and information monetization models (Atlam & Wills, 2019).

In parallel, energy efficiency is emerging as a global priority. The residential sector accounts for approximately 27% of global energy consumption (International Energy Agency, 2022), and smart energy management systems have the potential to reduce this consumption by 15% to 30% (Moreno et al., 2014). However, current commercial solutions such as Sonoff, Tuya Smart or Sense are mostly limited to remote switching on/off of devices without offering real optimization strategies based on the variable cost of electricity or the user's historical consumption patterns.

In response to this problem, this article proposes EcoMind IoT, a system that integrates Edge Computing, event-oriented processing and local artificial intelligence to offer autonomous, private and resilient energy management.

The system was designed to operate on a local area network (LAN) without ever depending on external services, guaranteeing service continuity even in the face of total loss of internet connectivity.
The article is structured as follows: Section II presents the statement of the problem and the justification; Section III describes the state of the art; Section IV details the theoretical framework; Section V explains the design and implementation of the system; Section VI presents the results; Section VII discusses the implications; and Section VIII sets out the conclusions and future work.

II. PROBLEM STATEMENT AND JUSTIFICATION

A. Problem Statement

The central problem that motivated this work can be formulated through the following research question: How to design an IoT infrastructure that allows autonomous decision-making for energy savings without depending on internet connectivity?
Today's commercial home automation systems have three fundamental limitations. First, **dependence on the cloud**: platforms such as eWeLink (Sonoff), Tuya Cloud or Amazon Ring servers require a permanent internet connection to execute any automation, which generates a single point of critical failure (Atlam & Wills, 2019). Second, the **absence of smart savings strategies**: most smart devices only allow remote manual control, without integrating optimization models that consider variables such as the hourly electricity rate or the user's consumption habits (Moreno et al., 2014). Third, **privacy risks**: by processing consumption data and household routines on external servers, sensitive information about the user's daily life is exposed (Sicari et al., 2015).

B. Justification

The justification for this project operates in three dimensions. From a **technological perspective**, the implementation of an architecture based on Edge Computing directly solves the vulnerabilities of cloud systems by reducing latency in automatic decisions and eliminating dependence on external services (Shi et al., 2016). From an **economic perspective**, the use of AI to evaluate the electricity rate against household habits impacts energy savings, eliminating vampire consumption (stand-by power) that represents up to 10% of the residential electricity bill (U.S. Department of Energy, 2021). From a **social and ethical perspective**, an Offline-First approach ensures that sensitive information about household routines remains safeguarded in the local network (Sicari et al., 2015).

III. STATE OF THE ART

The convergence between Edge Computing and IoT has been the subject of active research over the past decade. Shi et al. (2016) introduced the concept of Edge Computing as a complementary paradigm to Cloud Computing, arguing that processing at the edge of the network reduces latency, bandwidth consumption and privacy risks, characteristics particularly relevant for real-time IoT applications.
In the field of residential energy management, Moreno et al. (2014) demonstrated that demand-responsive building energy management systems (BEMS) can reduce electricity consumption by 15% to 30%, provided they have real-time consumption data and optimization algorithms. This line of research has evolved towards the integration of machine learning techniques: Li et al. (2021) proposed an energy consumption prediction system based on LSTM networks that operates locally, reaching prediction errors of less than 5%.
Regarding the MQTT protocol, Banks and Gupta (2014) documented its efficiency in low-latency networks, highlighting its suitability for bandwidth-constrained IoT applications. Subsequent studies have confirmed that MQTT outperforms alternative protocols such as HTTP and CoAP in high-frequency message scenarios with resource-limited devices (Al-Fuqaha et al., 2015).
Regarding lightweight language models for local execution (LLM at the edge), Xu et al. (2023) explored the feasibility of running quantized models on low-power hardware, demonstrating that models with 4-bit quantization can run on devices with 4-8 GB of RAM with acceptable response times for automation applications. This finding is central to EcoMind IoT's proposal, which requires local inference without reliance on external APIs.
In the field of Shelly devices, previous research has validated their integration with open-source platforms such as Home Assistant through local MQTT, confirming communication latencies of less than 100 ms in standard LAN networks (Kamilaris & Prenafeta-Boldú, 2018). The Home Assistant platform has been recognized as the most comprehensive open-source solution for on-premises orchestration of IoT devices, with support for more than 2,000 integrations (Home Assistant, 2023).
The fundamental difference between EcoMind IoT and the state of the art lies in the synergistic integration of all these components – Edge Computing, Local MQTT, Lightweight LLM, PostgreSQL and Flutter – under an Offline-First approach that privileges data sovereignty and operational resilience over cloud connectivity.

IV. THEORETICAL FRAMEWORK

A. Edge Computing

Edge Computing is a distributed computing paradigm that brings data processing closer to the source that generates it, as opposed to the centralized model in the cloud. In the IoT context, the Edge Node (Edge Node or Edge Gateway) acts as an intelligent intermediary between the physical device layer and the higher application layers. Shi et al. (2016) define Edge Computing as computing that is performed at the edge of the network, prioritizing real-time response, latency reduction, and data privacy.

B. MQTT Protocol and Mosquitto Broker

MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol based on the publisher/subscriber (Pub/Sub) pattern, designed for environments with limited bandwidth and high latency. Over TCP/IP, it uses a three-level QoS model (QoS 0, 1, and 2) that allows you to adjust the balance between performance and reliability. The Mosquitto broker is the open-source reference implementation of the MQTT standard, being widely used in local IoT installations due to its lightness and stability (Banks & Gupta, 2014).

C. Local Artificial Intelligence and Lightweight LLM

Locally executed Large Language Models (LLMs) represent an emerging trend that seeks to democratize artificial intelligence capabilities without relying on cloud APIs. Models such as Phi-2 (Microsoft), Gemma (Google) or Mistral 7B, quantized using techniques such as GGUF, can be executed on ordinary consumer hardware while maintaining reasonable context understanding and response generation capabilities (Xu et al., 2023). In EcoMind IoT, the local LLM acts as an intelligent rules engine that processes electricity bill tariff information and consumption histories to generate control instructions on the actuators.

D. FastAPI and Microservices Architecture

FastAPI is a high-performance web framework for Python, based on the ASGI (Asynchronous Server Gateway Interface) standard, which allows the development of asynchronous REST APIs with automatic type validation using Pydantic. Its asynchronous design is particularly suitable for EcoMind IoT's Edge Gateway, which must simultaneously handle multiple connections: MQTT messages from devices, mobile application requests via WebSockets, and inferences from the local AI model.

E. Flutter and Offline-First Approach

Flutter is Google's open-source user interface (UI) development kit, based on the Dart language and compiled to native code for multiple platforms. The Offline-First approach is a design strategy that prioritizes offline operation, synchronizing state when connectivity is available. In EcoMind IoT, this approach is implemented by direct communication of the Flutter application with the Edge Gateway via WebSockets or REST APIs within the same LAN network, without external routing.

V. SYSTEM DESIGN AND IMPLEMENTATION

A. System Objectives

The general objective of the project is to develop a prototype of residential energy management using an edge architecture (Edge Gateway) and local artificial intelligence, in order to automate the control of devices and optimize electricity consumption autonomously, guaranteeing operability without dependence on cloud connectivity.

Specific objectives:

1. Design the hardware infrastructure and local communication network using the MQTT protocol and a central node (Edge Gateway), ensuring low-latency interoperability between sensors and actuators in disconnected environments.
2. Integrate a local execution language model in the central node for the processing of consumption parameters and business rules, in order to optimize the control of actuators under different tariff conditions.
3. Validate the efficiency and autonomy of the prototype through tests in scenarios of loss of external connectivity, guaranteeing the continuity of energy saving strategies.

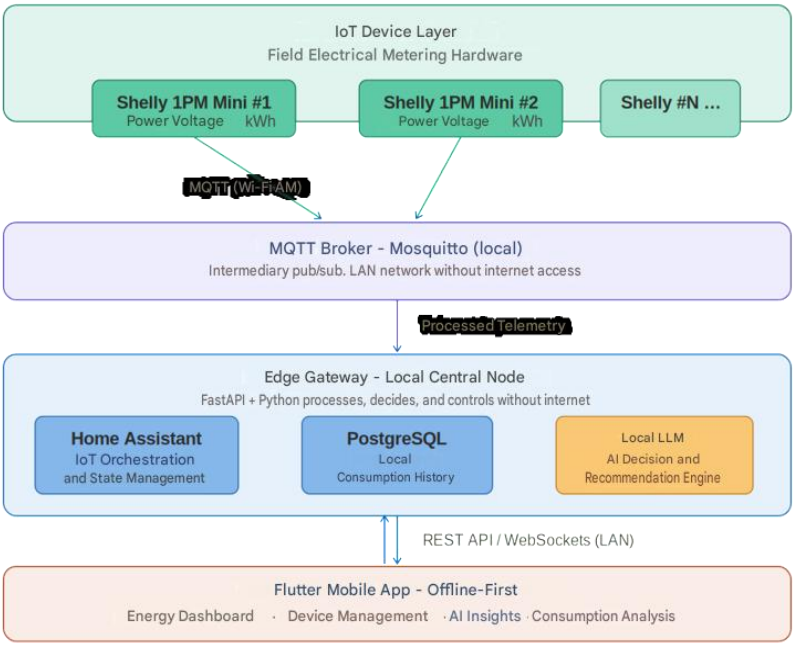
B. System Architecture

EcoMind IoT adopts a **hybrid architecture based on Edge Computing with event-oriented flow**. This architecture allows the system to operate autonomously within a local network without depending on an internet connection, responding to the need to guarantee data privacy and rapid response in decision-making.

The architecture is based on the core non-functional requirements of the system: data sovereignty (RNF01, RNF02), Offline-First operation (RNF01), low latency (RNF03) and total privacy (RNF02). By moving business logic to the edge of the network, you eliminate the dependency on cloud services critical point of failure in commercial ecosystems such as Tuya or Sonoff and ensure that energy-saving rules are executed 100% of the time, regardless of the state of the internet connection.

Figure 1 illustrates the overall architecture of the EcoMind IoT system, showing the four main layers and the bidirectional information flows between them.

Figura 1. Overall EcoMind IoT System Architecture



Source: Authorship

C. Stack Technological

The selection of the technological stack responds to criteria of efficiency, maturity of the ecosystem and suitability for the Offline-First approach:

Table 1. EcoMind IoT Technology Stack

Component	**Technology**	**Technical justification**
Backend language	Python 3.x	Complete ecosystem for AI, data processing, and IoT management
Framework API	FastAPI	High-concurrency asynchronous processing for multiple simultaneous connections
IoT Messaging	MQTT / Mosquitto	Lightweight protocol for local networks; latency < 100 ms; 100% local operation
IoT Orchestration	Home Assistant	Open source platform with support for 2,000+ on-premises integrations
Database	PostgreSQL	Robust RDBMS for historical consumption; Complex analytical queries
Artificial intelligence	LLM local (Python)	Quantized lightweight model for local inference without external APIs
User Interface	Flutter / Dart	Cross-platform native build; Offline-First approach via LAN
IoT Devices	Shelly 1PM Mini	Smart switch with power meter and native MQTT support

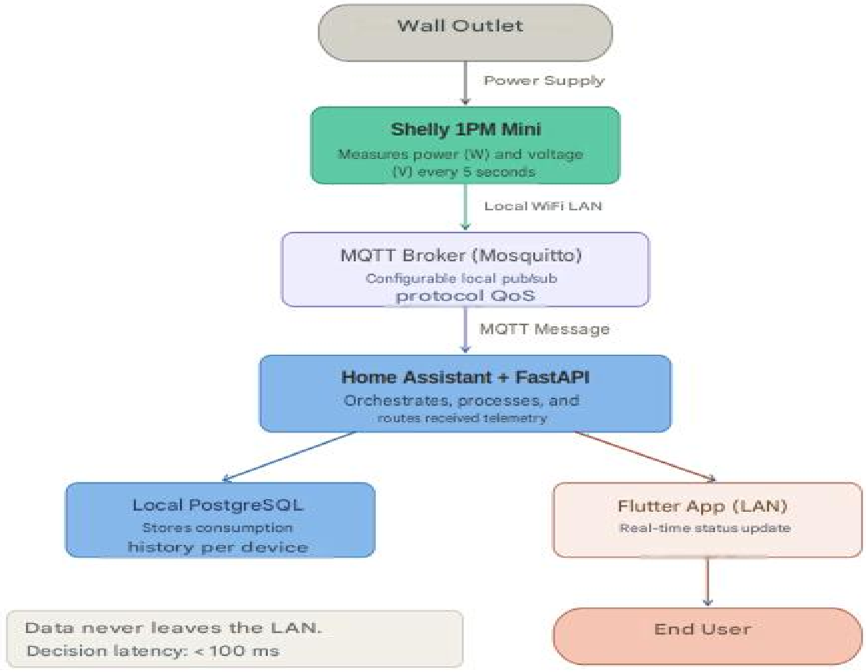
Source: Authorship

D. System Data Flow

EcoMind IoT's data flow follows a layered architecture with event-oriented processing. Shelly 1PM Mini devices publish electrical telemetry (active power in W, voltage in V, and cumulative energy in kWh) via MQTT to the Mosquitto broker installed on the Edge Gateway every 5 seconds. The Home Assistant receives and orchestrates these events, while the FastAPI/Python Gateway persists the data in PostgreSQL and runs the rules engine with local AI. The Flutter application communicates directly with the Gateway via WebSockets or REST within the LAN, ensuring instant control without internet.

Figure 2 shows the complete data capture and telemetry flowchart, describing the data lifecycle from its physical origin in the outlet to its display in the mobile app.

Figure 2. Data Capture and Telemetry Flowchart



Source: Authorship

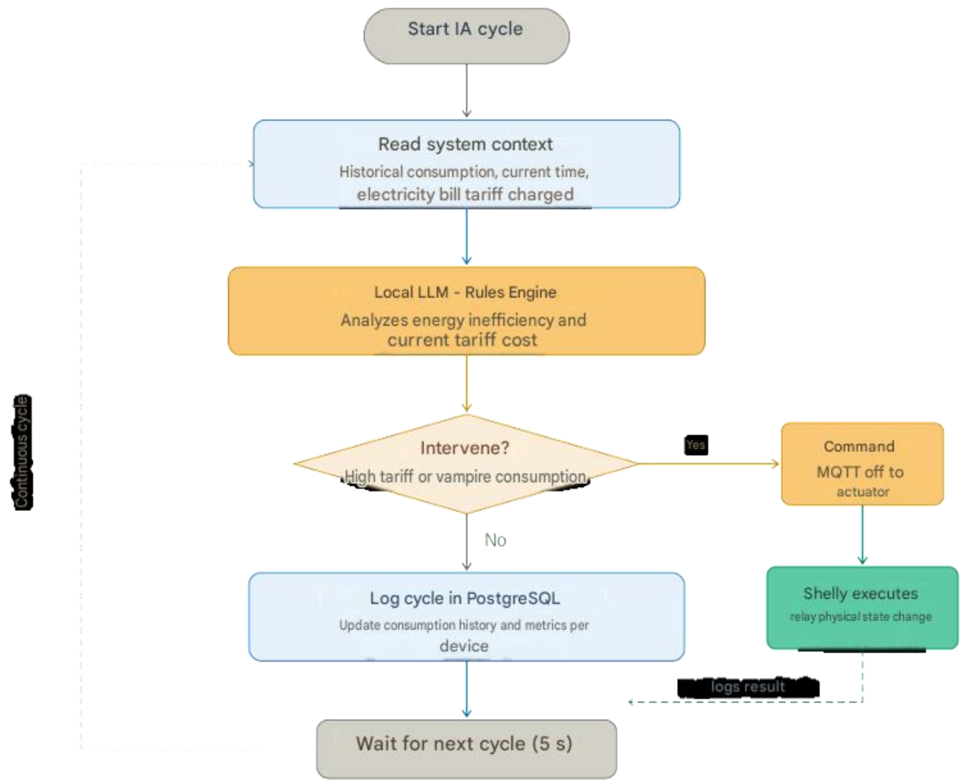
E. Decision Logic with Local Artificial Intelligence

The most differentiating component of EcoMind IoT is its decision engine based on local AI. The autonomous decision-making process operates using a rules engine that combines deterministic logic with inference from the local LLM:

1. **Context ingestion:** The system consolidates historical consumption of the recent period by device, current time, current rate extracted from the locally processed electricity bill, and user-configured thresholds.
2. **Inefficiency detection:** The engine identifies patterns of vampire consumption (stand-by power), devices turned on at high fare times and significant deviations from the historical average.
3. **Instruction generation:** When the model determines that energy savings warrant an intervention, it automatically generates a shutdown command that is sent to the actuator via MQTT.
4. **Feedback:** Executed actions and their results are stored in PostgreSQL, enriching the context available for future model decisions.

Figure 3 illustrates the flow chart of the decision logic and processing with local AI, including the continuous feedback loop.

Figure 3. Decision logic and processing with local AI



Source: Authorship

F. Offline Resiliency and Manual Control
A central feature of the design is the system's ability to operate with full functionality in the absence of an internet connection. This is achieved through two complementary mechanisms. The first is **local rule persistence**: all automation rules and model parameters are stored in local PostgreSQL, so that the decision engine can continue to operate indefinitely without updating configurations from the cloud. The second is **direct LAN routing**: the Flutter app connects directly to the Edge Gateway using its local IP address, with no dependency on external DNS services, ensuring that the user can manually control their devices with latencies of less than 50ms even with completely disrupted internet.

G. System Class Model
The object-oriented design of the EcoMind IoT system identifies the following main classes at the application layer: **User** (main interactor), **AppMovil** (Flutter interface), **EdgeGateway** (FastAPI central node with ipLocalLAN attributes and receiveAppRequest methods(), manageControlManual()), **AImodel** (local LLM with generateRecommendations() methods, analyzeInefficiencies()), and **IoT Device** (Shelly switch with On status and runChangePhysicalState()). In the database layer, the main entities are IoT Device, EdgeGateway, Database (PostgreSQL), ConsumptionRecord, and Electricity Bill.

H. System Requirements
Table 2 consolidates the main functional and non-functional requirements of the EcoMind IoT system.

Table 2. Functional and non-functional system requirements

Code	Description	Responsible	Priority	Type
RF01	Monitor and capture power consumption (power and voltage) in real-time using smart switches	IoT Architect	High	RF
RF02	Upload and process electricity bill information locally on the Edge Gateway	AI Engineer	High	RF
RF03	Analyze consumption and automatically disconnect equipment at high-rate times using local AI	AI Engineer	High	RF
RF04	Individual manual control of IoT devices from the Flutter app	IoT Architect	High	RF
RF05	Store consumption history in local database to identify vampire patterns and consumptions	Data Analyst	Media	RF

Code	Description	Responsible	Priority	Type
RF06	Generate and display automatic energy optimization recommendations (AI Insights) using local LLMs	AI Engineer	High	RF
RNF01	The system must operate under Offline-First architecture without dependence on cloud services	IoT Architect	High	RNF
RNF02	Ensure complete privacy: no data leaves the user's LAN	IoT Architect	High	RNF
RNF03	Low-latency MQTT communication (< 100 ms) via local Mosquitto broker	IoT Architect	High	RNF

Fuente: Autoría propia

I. User Interface — Prototype Views

The EcoMind IoT application was designed with four main screens developed as prototypes in Figma: (1) **Energy Dashboard**: real-time view of total consumption, Gateway connection status and key electrical metrics; (2) **Device Management**: individual management of connected IoTs with manual on/off control; (3) **AI Insights**: intelligent analysis generated by the local LLM with recommendations for optimization and charging of electric bills; and (4) **Consumption Analysis**: historical visualization with daily, weekly and monthly trends identifying usage patterns.

VI. RESULTS

A. Architectural Validation

The development of the EcoMind IoT project allowed the definition and structuring of a complete technological architecture oriented to residential energy management based on Edge Computing. The complete data flow of the system was established, from capturing electrical information using Shelly 1PM Mini devices to local processing using the AI engine and generating actions on the actuators. The hybrid Edge Computing architecture with event-oriented processing proved to be technically feasible to meet the requirements of privacy, autonomy and low latency.

A comparison with existing commercial systems shows the differential advantages of the proposed solution:

Table 3. Comparison of EcoMind IoT with commercial solutions

Feature	EcoMind IoT	Sonoff / Tuya	Sense	Home Assistant
Operation without internet	✓ Total	✗ Limited	✗ No	✓ Partial
Local AI Savings	✓ LLM local	✗ No	✓ ML cloud	✗ Non-native
Data Privacy	✓ 100% LAN	✗ Own cloud	✗ Own cloud	✓ Local
Fare optimization	✓ Automatic	✗ Manual	○ Informative	○ With addons
Active Control Actuators	✓ Self-employed	✓ Manual	✗ Read only	✓ Scripted

Fuente: Autoría propia

B. Design Results

The complete UML models of the system were designed and documented, including: three activity diagrams (data capture flow, AI decision logic, and manual control management), the main use case diagram, the IoT system interaction diagram, the energy-saving automation diagram, the application class diagram, and the database entity-relationship diagram. Likewise, the functional (RF01-RF08), non-functional (RNF01-RNF07) and interface (RI01-RI06) requirements were developed, providing a formal technical basis for the subsequent implementation of the prototype.

C. User Interface Prototype

The full prototype of the user interface was developed in Figma, with the four main screens of the Flutter app. The design validates the feasibility of presenting complex energy information in an accessible and intuitive way, with special emphasis on the AI Insights screen that facilitates user interaction with the local AI model without requiring technical knowledge.

VII. DISCUSSION

Relevance of the Offline-First approach in the Latin American context. In countries such as Colombia, where high-speed internet penetration in rural and peri-urban areas is limited with coverage rates that do not reach 60% in small municipalities according to DANE (2022), the ability to operate without a continuous connection is not an optional feature but an essential requirement. EcoMind IoT addresses this reality head-on, significantly expanding its adoption potential in markets where imported commercial solutions are inadequate.

Privacy as a competitive advantage. The global regulatory trend embodied in the GDPR in Europe, Law 1581 of 2012 in Colombia and similar regulations in other countries introduces increasing pressure on companies that process personal data in the cloud. By keeping all data within the user's network, EcoMind IoT eliminates regulatory compliance risks at the root, turning privacy into a tangible competitive differentiator.

Limitations of the current prototype. The system presented corresponds to a conceptual design validated by software artifacts (UML diagrams, requirements, interface prototypes) but has not been implemented and tested in a real physical environment. The main constraints identified are: (1) dependence on the central node hardware for the capacity of the local LLM; (2) the initial setup curve for non-technical users; and (3) the need to validate performance under real load with multiple simultaneous IoT devices.

Future work. The lines of research derived from this work include: (1) implementation and evaluation of the prototype in a real residential environment with measurable energy savings metrics; (2) quantitative comparison of on-premises LLM performance versus cloud models; (3) incorporation of redundancy mechanisms for the central node; and (4) extension of the system to applications in SMEs and coworking spaces.

VIII. CONCLUSIONS

This article proposes and documents EcoMind IoT, a residential energy management system based on Edge Computing with local artificial intelligence, which establishes a technically feasible solution for the fundamental limitations of today's commercial home automation systems.

The main contributions of the work are three. First, a hybrid Edge Computing architecture with event-oriented processing is proposed that guarantees the autonomous operation of the system without dependence on the cloud, solving the problem of operational interruption due to the loss of internet connectivity. Second, a mechanism is designed to integrate a local LLM into the IoT control flow that allows intelligent energy saving decisions based on the user's real tariff data, without exposing sensitive information to external servers. Third, a formal technical basis is established through UML diagrams, functional and non-functional requirements, and interface prototypes that allow the future construction of a fully functional physical prototype.

The incorporation of local artificial intelligence proposed at the design level shows the potential of implementing energy optimization strategies based on the analysis of consumption data and electricity tariffs, allowing automated decision-making aimed at savings that goes beyond the capacity of current commercial systems.

Although the system has not been physically implemented at this stage, the work developed establishes a solid technical base that allows its future construction, validation and scalability towards a functional prototype. The proposed architecture has the potential to impact both the residential sector and small and medium-sized enterprises, opening a line of applied research of high social and economic value in the Latin American context.

Declaration of conflict of interest. The authors declare that there are no conflicts of interest regarding the research, authorship and/or publication of this article.

REFERENCES

1. Agencia Internacional de Energía. (2022). *World Energy Outlook 2022*. IEA Publications. https://www.iea.org/reports/world-energy-outlook-2022
2. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, *17*(4), 2347–2376. https://doi.org/10.1109/COMST.2015.2444095
3. Atlam, H. F., & Wills, G. B. (2019). IoT security, privacy, safety and ethics. In M. Farsi et al. (Eds.), *Digital Twin Technologies and Smart Cities* (pp. 123–149). Springer. https://doi.org/10.1007/978-3-030-18732-3_8
4. Banks, A., & Gupta, R. (2014). *MQTT Version 3.1.1. OASIS Standard*. OASIS. http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html
5. Departamento Administrativo Nacional de Estadística (DANE). (2022). *Encuesta de calidad de vida 2022*. https://www.dane.gov.co/index.php/estadisticas-por-tema/tecnologia-e-innovacion/encuesta-de-calidad-de-vida-ecv
6. Flutter Development Team. (2023). *Flutter: Build apps for any screen*. Google. https://flutter.dev/docs
7. Gómez, J., & Suárez, A. (2019). *Internet de las cosas (IoT): Aplicaciones y tecnologías*. Editorial RA-MA.
8. Home Assistant. (2023). *Home Assistant Documentation*. https://www.home-assistant.io/docs/
9. International Data Corporation. (2023). *IoT growth demands rethinking long-term storage strategies*. IDC. https://www.idc.com/getdoc.jsp?containerId=prUS49199723
10. Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, *147*, 70–90. https://doi.org/10.1016/j.compag.2018.02.016
11. Li, X., Wen, J., & Smith, A. (2021). Improving forecasting accuracy for building energy systems using LSTM deep learning on edge nodes. *Energy and Buildings*, *250*, 111330. https://doi.org/10.1016/j.enbuild.2021.111330
12. Moreno, M. V., Zamora, M. A., & Skarmeta, A. F. (2014). User-centric smart buildings for energy sustainable smart cities. *Transactions on Emerging*

Telecommunications Technologies, *25*(1), 41–55. https://doi.org/10.1002/ett.2771

13. PostgreSQL Global Development Group. (2023). *PostgreSQL 16 Documentation*. https://www.postgresql.org/docs/16/
14. Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, *3*(5), 637–646. https://doi.org/10.1109/JIOT.2016.2579198
15. Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. *Computer Networks*, *76*, 146–164. https://doi.org/10.1016/j.comnet.2014.11.008
16. Torres, J. (2021). *Domótica y automatización del hogar inteligente*. Editorial Alfaomega.
17. U.S. Department of Energy. (2021). *Standby power: Overview*. https://www.energy.gov/energysaver/articles/energy-vampires-are-attacking-your-home-heres-how-stop-them
18. Xu, R., Baracaldo, N., & Joshi, J. (2023). Privacy-preserving machine learning: Methods, challenges and directions. *arXiv preprint arXiv:2108.04417v3*
19. . https://arxiv.org/abs/2108.04417

ejprd.org- Published by Riset Publishing Services LLC.