

Conversational Assistant with Large Language Model Agent for Natural Interaction in Home Automation Environments

Ricardo Yauri¹, Rafael Espino²

¹Facultad de Ingeniería y Gestión, Universidad Nacional Tecnológica de Lima Sur, Lima, Perú

²Facultad de Ingeniería, Universidad Tecnológica del Perú, Lima, Perú

Article Info

Article history:

Received Jul 28, 2025

Revised Dec 7, 2025

Accepted Dec 11, 2025

Keywords:

OpenAI

Foundational models

RAG

Distributed processing

LangChain

ABSTRACT

Digital transformation has driven the development of smart home automation systems for healthcare and remote assistance, which can integrate large language models, foundational models, and Retrieval-Augmented Generation techniques to design conversational assistants accessible to people with reduced mobility. In these cases, it is necessary to overcome usability barriers to control home devices because traditional home automation interfaces are not adapted to their specific needs. Previous research has identified systems such as voice control, gestures, and conversational agents based on generative artificial intelligence to facilitate assisted interaction in the home through home automation systems. This article describes a conversational agent, aimed at people with reduced mobility, to facilitate natural interaction with home automation environments, using technologies such as OpenAI, Langchain, and LlamaIndex. The results demonstrate the successful deployment of a web-based telecare platform based on conversational agents capable of retrieving technical documents through embedding, generating SQL queries, and MQTT topics for integration into real-world monitoring and assistance environments for people with reduced mobility.

Copyright © 2025 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Ricardo Yauri

Facultad de Ingeniería

Universidad Nacional Tecnológica de Lima Sur

Av. Bolivar S/N, sector 3 grupo 1, mz. A, subote 3 - Villa El Salvador, Lima, Perú

Email: ryauri@untels.edu.pe, boncer99@gmail.com

1. INTRODUCTION

Digital transformation allows the development of smart home automation systems to contribute to improving people's quality of life, in contexts of health [1] and remote assistance [2]. However, these solutions are difficult to use for people with reduced mobility [3], due to unintuitive interfaces [4]. In this context, the emergence of wide language models (LLM) [5] (such as OpenAI's GPT), Vision and Language Models (VLM) or ALM (Audio-Language Model) have allowed the generation of alternatives to create conversational assistants and interact naturally [6], understand questions and adapt to the user's context [7]. These techniques, integrated with Generation Augmented Recovery (GAR) tools and their integration into cyber-physical architectures [8], allow the creation of adaptive and intelligent assistance systems [9].

In relation to the problem related to the control of devices at home, these face usability barriers, since traditional home automation interfaces [10] are not designed for the specific needs of people with reduced mobility [11]. Many of these systems lack tools to know what functions available, technical information and active devices are. Previous research describes assisted home interaction through voice recognition or gesture control. For example, the use of MediaPipe and OpenCV for hand recognition was used in gesture control prototypes to turn on lights [12] or activate devices [13]. Other projects have implemented systems based on Raspberry PI devices or sensor networks (WSN) to send commands to physical actuators using text, video or voice control tools [14] in home automation control systems [15]. In the field of generative artificial

intelligence applied to home automation [16], natural language models have begun to be integrated to facilitate the interaction between the user and the system with conversational assistance agents [17].

The literature review focused on identifying research related to assistive technologies for people with reduced mobility and the use of LLM in healthcare and telecare settings, following the Kitchenham methodology, the PICO (Population, Intervention, Comparison, Outcome) approach, and the PRISMA technique for document selection. Several studies describe the design of assistance technologies in smart homes for people with reduced mobility [18], describing a gesture control platform for activating devices [19]. Similarly, the use of voice control systems based on free hardware modules for turning on lights or opening doors is also described [20, 21]. In addition, the use of motion sensors has also been described, highlighting the adaptability to the user for pattern recognition.

Regarding the use of artificial intelligence to improve accessibility in smart homes, proposals have been identified that focus on neural networks, fuzzy logic models, and supervised learning, which personalize the behavior of home devices based on user patterns. For example, techniques based on convolutional neural networks can be used to detect the presence and health status of people, automatically adjusting their use in the context of smart homes [22, 23]. In addition, deep learning-based architectures are also used to predict usage patterns of household appliances or similar devices [24, 25], which requires extensive datasets and does not always contemplate direct interaction between the user and the system.

On the other hand, in recent years there has been considerable progress in the use of LLM for home automation control. Research shows the integration of these models with home automation systems to interpret commands and transform them into actions for device control [26, 27]. On the other hand, the development of applications with LLM models embedded in microcomputers such as Raspberry Pi is also described in the context of smart environment applications [28, 29]. In the case of the use of LLMs for healthcare and telecare, this is important as a tool to improve remote care. Research has shown the integration of LLM models with healthcare platforms, allowing access to medical information [30, 31]. Although these systems need to be clinically validated, preliminary results indicate a positive impact on user treatment, considering LLMs as a technology with potential for application in telecare.

For these reasons, this paper presents the design of an assistance system using agents with RAG capabilities for natural interaction in home automation environments, targeting people with reduced mobility. The system integrates a real-time control and monitoring stage (MQTT, PostgreSQL) (using the tools LangChain and LlamaIndex) with an LLM agent to interpret and respond to natural language queries. The specific objectives are: defining the characteristics of MQTT databases and topics for control and technical information; designing an intelligent agent with augmented recovery capabilities (RAG); integrating tools for accessing technical documents and dynamic records; and deploying the system in the cloud. Although the system was evaluated in Spanish, the solution is language independent because the agent uses the LLM model, making it possible to adapt it to other languages by changing the description of the prompts used to give instructions to the agent, thus allowing it to be adapted to other contexts in different languages.

Various approaches have been found in different articles that use different objectives and scope. In this research, the main objective is not to comparatively evaluate the performance of various techniques, but rather to demonstrate the development of a specific architecture to prove its viability in IoT applications. The contribution lies in the integration of generative intelligence technologies into a cyber-physical system. The integration of an agent, which combines technical information and real-time sensor data, represents a contribution to healthcare (eHealth/telecare) through a home automation assistance system that facilitates the autonomy of people with reduced mobility. Although the system does not replace direct clinical functions (such as cardiac monitoring or medication administration), it supports well-being and health through assistive technology.

2. RESEARCH METHOD

An intelligent conversational agent is integrated into a home automation system, enabling natural interaction with the environment through natural language queries. To this end, an agent is developed using the LangChain framework and LlamaIndex (for file storage). In addition, retrieval-augmented generation (RAG) techniques are used to answer user questions by accessing technical documentation embedded in vectors (stored in Elasticsearch) or dynamic information from a PostgreSQL database. Using a ReAct-like conversational flow, the agent decides whether to retrieve static data or execute SQL queries to respond to the status or control of home devices. It also incorporates intent detection to generate control commands for home automation components such as lights, wind-controlled motors, roller blinds, or household appliances. All of this is managed by a web application built with Flask.

The conversational agent uses tools defined within the LangChain framework to process user queries. These tools allow the agent to decide whether to retrieve technical information from documents, query a database, or generate instructions for device control (Table 1).

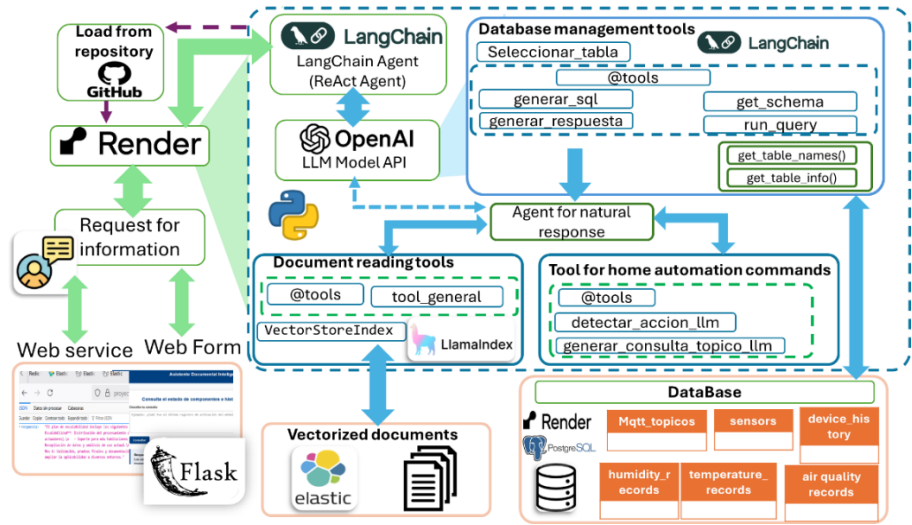


Figure 1. System architecture

Table 1. Role of each tool with respect to its functionality

Group	Role	Tools	Data Source
technical documents database	Answering questions about the system	tool_general, LlamaIndex, ElasticsearchStore, OpenAIEmbedding	Embedded Documents
home automation control topics	Retrieve dynamic data (sensor status, history, etc.)	get_schema, generar_sql, run_query, generar_respuesta, seleccionar_tabla_v2	PostgreSQL Database
	Interpret intents and generate control commands	normalizar_texto, generar_consulta_topico_llm, detectar_accion_llm, ejecutar_consulta	User Input Text

2.1. Characteristics of information sources and control topics

A PostgreSQL database hosted on Render's online service was implemented, with tables intended to store historical records of environmental sensors (temperature, motion), actions performed on devices, system configurations, and users. In parallel, the MQTT topic structure used for publish/subscribe communication with physical devices was defined. Additionally, technical documents (manuals, user guides, specifications) were selected and organized, processed, and converted into semantic embeddings for indexing in an Elasticsearch-based vector store (Figure 2).

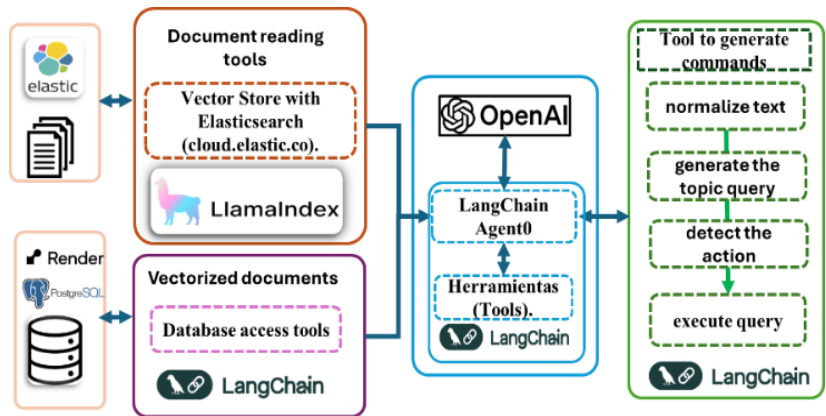


Figure 2. Information storage and retrieval components

2.2. Intelligent agent with Retrieval-Augmented Generation (RAG)

The core of the system is a conversational agent developed with LangChain, which uses the GPT-4 model and combines logical reasoning with automated actions using tools, allowing access to both static (technical documents) and dynamic (sensor and device database) information. The agent chooses among

available tools based on the type of question. If the query requires general knowledge, it accesses indexed technical documents; if it requires updated data, it generates and executes SQL queries (Figure 3). Although the first implementation was designed for Spanish-speaking users, the agent architecture is not tied to a specific language because the semantic retrieval tools using LangChain can be adapted by adding a language indication to the agent configuration prompt done in Python, allowing the same workflow to work in other languages. Although the GPT-4 language model was used, the agent's tool organization is model-independent and can be used with other tools such as Google Gemini or Llama 3. In this case, LangChain generates interfaces for different LLM models.

2.3. Technical documents access tool

To enable the agent to access documents, a semantic retrieval tool was designed that integrates LlamaIndex with ElasticsearchStore. This tool converts PDF documents into embeddings using OpenAI Embeddings and stores them in a vector index. Upon receiving a question requiring an explanation of how it works (e.g., "What gestures are available?"), the agent activates this tool, which performs a semantic similarity search and returns the most relevant text fragments (Figure 4).

2.4. Access to databases

A set of tools was developed for the agent to interact with the PostgreSQL database and retrieve information about the state of the home automation system. For example, when receiving the question "Which devices are on now?", the agent triggers this sequence and responds with the current state. The `get_schema(question)` function retrieves the database schema from the user's question. Then, with `generate_sql(question)`, a SQL query is generated. The `run_query(query)` function then executes this query directly against the database. Finally, the `generate_response` function uses the original question, the SQL query, and the resulting result to construct a natural language response (Figure 5).

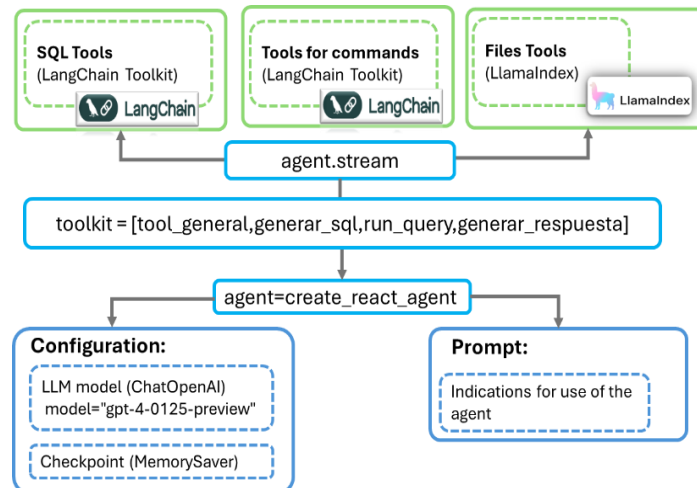


Figure 3. Intelligent agent interaction scheme

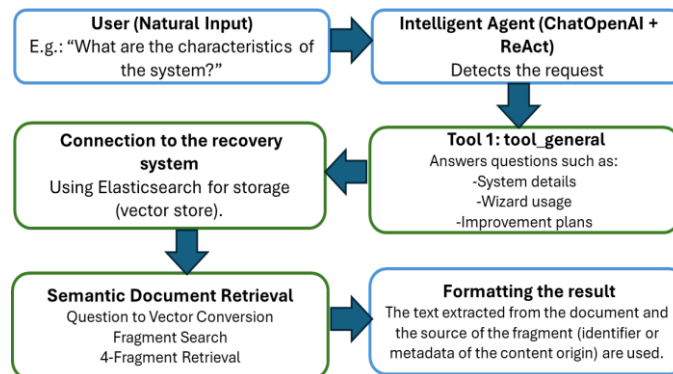


Figure 4. Interaction of document information retrieval tools

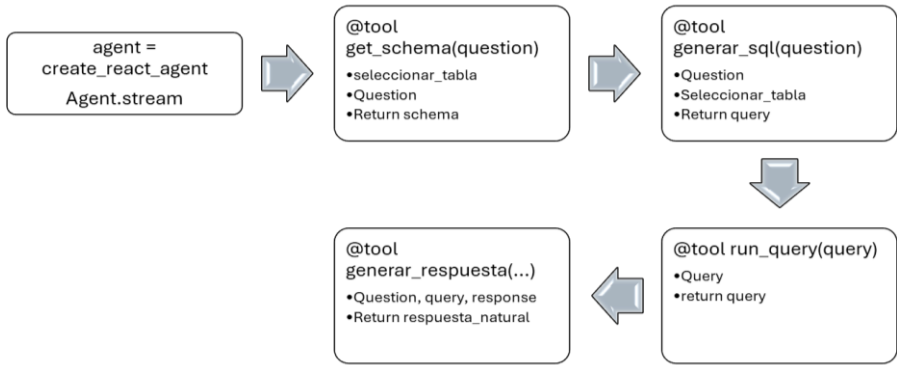


Figure 5. SQL query generation process without topics

2.5. Generation of home automation control topics

The agent manages physical devices in a home automation environment using tools that interpret requests such as "Turn on the living room light" and translate them into actionable commands. The process includes text normalization (removing accents and capitalization and originally in Spanish), intent detection (using a specialized prompt), and transforming the query into a response with a valid MQTT command. This final message can be sent to the home automation system, enabling voice or text control of connected devices (Table 2). As shown in Figure 6, the request to turn on the living room light (originally in Spanish) is used as an example. The result is the topic "luz/sala/set/0" (turn on the living room light with set equal to 1) with text in Spanish, since this is how the MQTT topics were configured and the examples appear in spanish because the system was originally trained in that language. The intent detection modules dynamically configure MQTT themes and SQL statements that are language-independent, so adapting the system to another language only requires redefining the prompt instructions used by the agent tools without modifying the main workflow.

Table 2. Functions for generating control topics

Function	Main use	Description
normalizar_texto(text) (In english normalize_text)	Text preprocessing	Convert text for easy analysis.
generar_consulta_topico_llm (text). (In english generate_topic_query_llm)	Intent classification	Detects the “topic” of the message.
detectar_accion_llm(text). Use of LLM. (In English detect_action_llm)	Action/intent detection	Returns whether there is a recognized action and g topic.
ejecutar_consulta(text) (In English execute_query)	Structured response generation	Returns a response that can be interpreted by the control system.

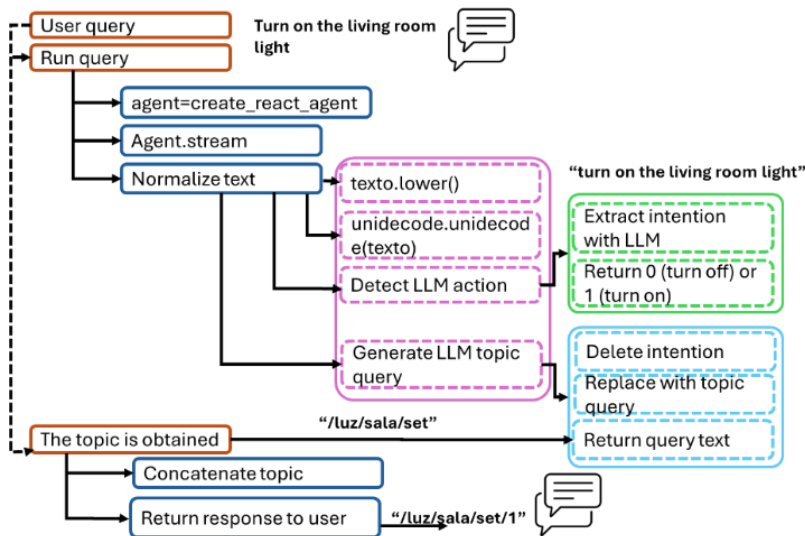


Figure 6. Command generation process for the MQTT topic

2.6. Web Interface Deployment

The agent and its tools were deployed in the cloud using the Render platform, adding a Flask web application, which exposes both graphical and programmatic interfaces (API) for interacting with the agent. Users can submit questions through a web form or using API endpoints (e.g., /info or /query_docs). The web application provides access routes for responding to HTTP POST requests. Two main routes are defined using Flask's `@app.route()` decorator, where "/info" allows a user to submit a query from an HTML form, process it with a function called `run_query`, and display the result on a web page (`info.html`). There is also a "/query_docs" route, which offers a GET API for quick queries via the `q` parameter in the URL (Figure 7).

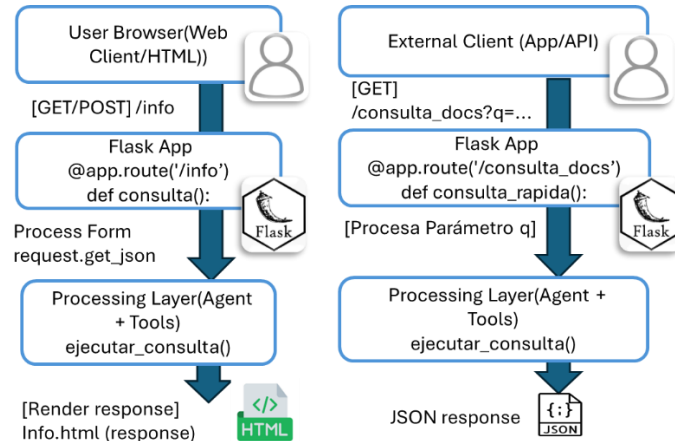


Figure 7. API and Web application deployment diagram

3. RESULTS AND DISCUSSION

3.1. Web Interface Deployment

To deploy the web application, the GitHub service was used to upload the project. Then, using Render's Online platform, a new project was created by selecting the "Deploy from GitHub repo" option. The deployment and component installation process were automated, considering the Python libraries used. In this case, Render hosted the system's backend, the PostgreSQL database, and Flask Python scripts for basic chat interaction. It also encapsulates all the agent logic, tools, and web paths (`/info`, `/consult_docs`) (Figure 8).



Figure 8. Installation and deployment of the application

The results demonstrate the deployment of the web interface using services such as GitHub and Render to establish suitable backend architecture, integrating Flask, PostgreSQL, and agent routes. This implementation offered natural interaction through a web environment, which fulfills the objective of implementing a telecare platform based on conversational agents.

3.2. Document access tool

The documents generated were accessed and synchronized with the cloud using the Elastic Cloud service, verifying that the extracted information was organized in the repositories using vectorization (Figure 9). When a user asks a question in natural language, such as "information about system components?", the

agent decides to use the tool `general` tool to access previously uploaded documents. The process continues with a connection to an Elasticsearch information retrieval system. Finally, the fragments are formatted and include both the textual content and its source, generating responses in Spanish (Figure 10). Some answers or commands appear in Spanish because they correspond to the original implemented MQTT topic structure.

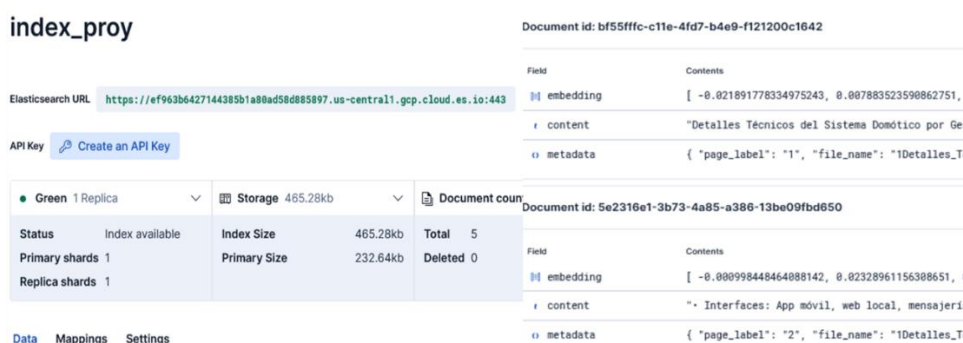


Figure 9. Cloud.elastic storage interface for data vectors

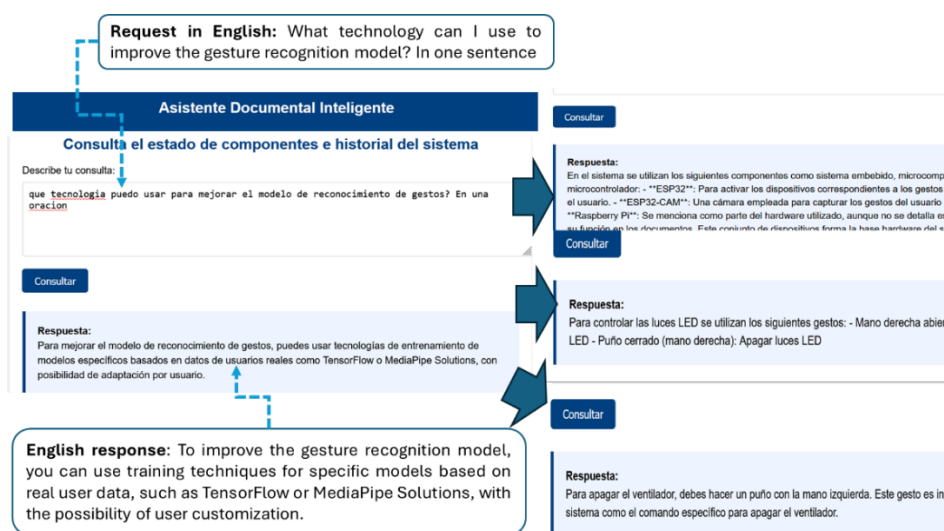


Figure 10. Examples of responses via inquiry form

Regarding access to technical documents, the integration of the LLM agent, the `tool_general` tool, and the Elastic Cloud infrastructure was validated to retrieve contextual information. This confirms that the semantic retrieval system (based on embeddings) generates appropriate natural language responses in the context of home automation applications. In addition, the system produced clear responses in Spanish, demonstrating its application in Spanish-speaking remote support environments.

3.3. Database Access Verification

The process of interacting with the database begins with a natural language question (Figure 11). From this, the intelligent agent, based on the ReAct model and ChatOpenAI, analyzes the query's intent and decides which tools to use. First, `get_schema()` is used to identify the table (e.g., `temperatura_registros`) and retrieve its structure. Then, `generate_sql()` generates a SQL query and transforms the result into an understandable response, such as: "The highest temperature recorded last month was 34.2°C" (In spanish: "El valor máximo de temperatura registrado en el último mes fue 34.2°C").

The agent's access to the assigned tools and data flow management were verified until a response was returned. Figure 12 shows the response to an example request called "Return the fan theme" which was made in Spanish due to the agent's configuration in that language (In spanish "Devuelve el tópico del ventilador"). In this case, the query is performed against the database by understanding the context, returning the topic `"/sala/ventilador/ser"`.

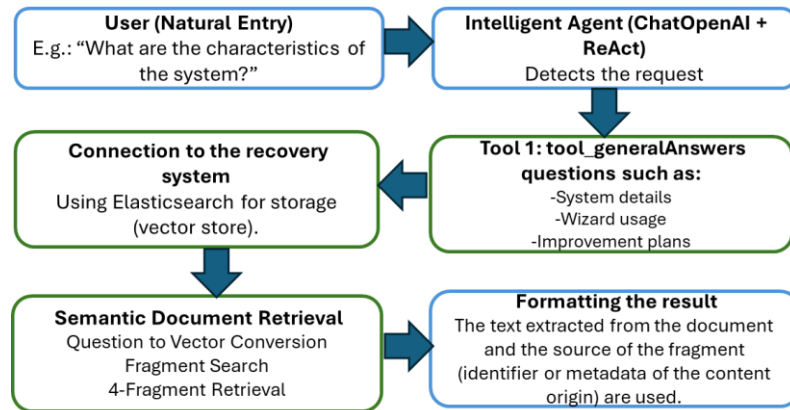


Figure 11. Agent interaction with the database



Figure 12. Function invocation flow for response generation in spanish

Figure 13 shows some examples of queries and responses (in Spanish) accessing information from the home automation system database. In this case, queries and responses are displayed regarding the highest humidity recorded by the sensors, the MQTT topic used to control the living room lights, and the sensors installed in the kitchen. The agent can interpret complex questions and automatically transform them into valid SQL statements, making data querying easier without requiring technical knowledge from the user.

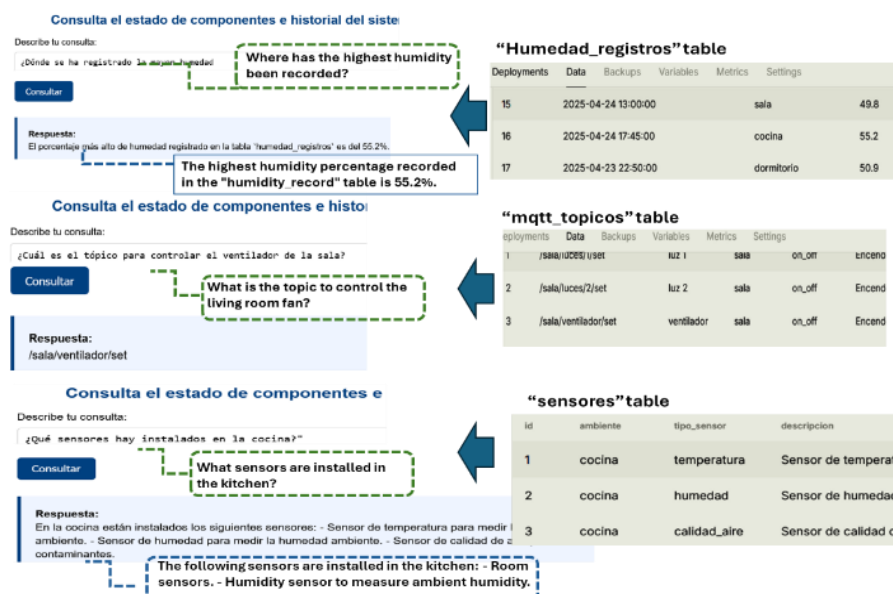


Figure 13. Natural language queries on information in the database

3.4. Generation of home automation control topics

The deployed agent automatically detects the user's intent when a device needs to be controlled. A request for an MQTT topic is automatically generated in the database. The response is then obtained, and the intent code 0 or 1 is added to create the MQTT control topic returned as a response. Token consumption was reduced because only the MQTT topic storage schema is processed (Figure 14).

The generation of the MQTT command managed by the RAG agent was evaluated with several types of queries, detecting the user's intention. Queries such as "activate the coffee maker" ("activa la cafetera" in spanish), "turn off the fan" ("Apaga el ventilador" in spanish), and "open the window" ("abre la ventana" in spanish) were executed through the web interface, generating the concatenated MQTT topic with the code 0 or 1. Since the agent can be used from other devices, a web API ("consulta_docs" route) is available that allows the query to be sent and the information returned in plain text format, for use by home automation hardware devices (Figure 15). Although the tests were performed using GPT-4, the application is not dependent on this specific model. Future experiments will allow for comparisons to quantify the differences in retrieval accuracy or intent detection depending on the model selected.

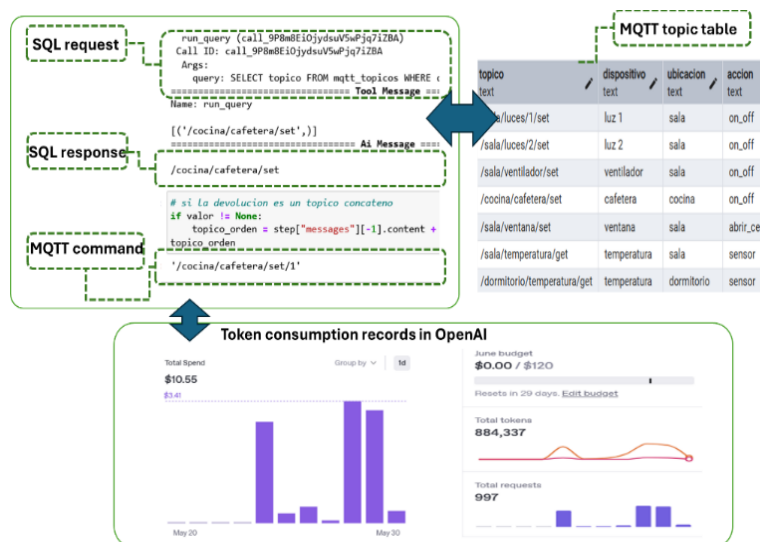


Figure 14. Intention detection process for home automation control

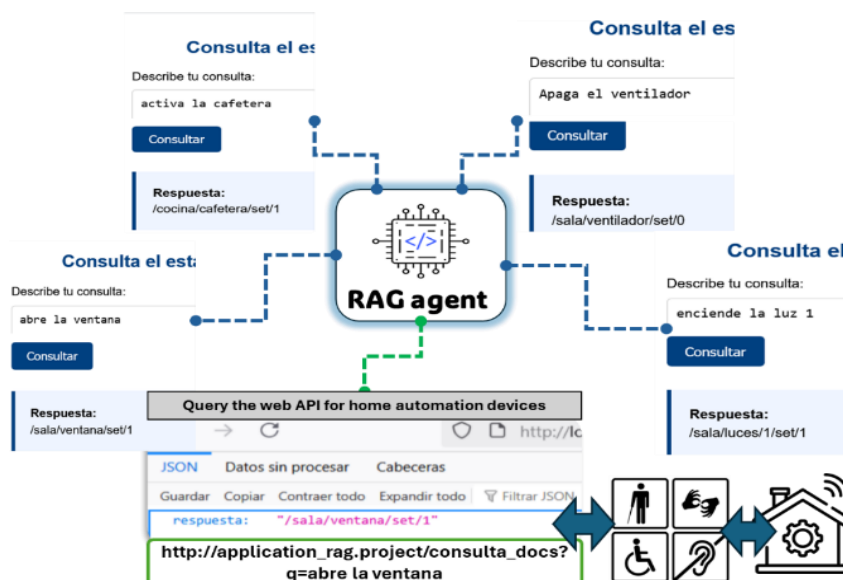


Figure 15. Generating MQTT commands for device control

Furthermore, the agent understands user intent in the context of home automation control and automatically generates MQTT topics, validating its application in home automation and assistance. The

reduction in token consumption was achieved by optimizing the selection of the queried schema, improving computational efficiency, which is important for applications on resource-constrained hardware devices.

Finally, the availability of the web API from external devices extends its use to embedded or mobile devices, allowing its integration into cyber-physical systems. This contributes to the system's scalability and portability for real-world telecare, accessibility, and monitoring environments. Although the research objective is to validate the functionality of the proposed architecture, operational tests were conducted to obtain basic system validation. The solution was implemented on the Render platform, where the service's behavior was evaluated during interactions with the agent. On average, the response time per request—which includes querying the LLM model, retrieving information from the repository, and executing the RAG string—was 2 to 3 seconds, with peak times of 3.4 seconds.

Because the evaluation is performed through interaction and not as a machine learning model, metrics such as accuracy, recall, or F1 score are not considered. However, metrics related to functional errors (failures in query generation or interruptions) were obtained with values less than 3% in 100 consecutive runs. The measurements taken show that the system has stable performance, with low response times, and although comparative metrics were not included, the viability of the architecture was validated in a practical scenario.

4. CONCLUSION

The development of the conversational assistant demonstrated that information retrieval and command generation tools can contribute to remote accessibility and optimize consultation processes in assistance environments.

Using the system with the Render platform enabled the deployment of the agent's backend and services, generating natural language interactions with the user. Additionally, the agent can access information stored in cloud repositories using Elastic Cloud and Elasticsearch, validating the RAG approach for interacting with documentation. The tools used by the agent allowed the following to be achieved: transform natural language queries into SQL queries, validate the agent application to recognize user intent, retrieve the necessary schema, generate the query, and construct a response. In the case of MQTT topic generation, its operation was verified through tests related to the evaluation of device activation or deactivation responses. Although the system was evaluated through interactions in Spanish, the same agent could be used in other languages by configuring prompts that are used internally by the tools in Langchain. While there are different methods that could be used to solve similar problems, this work focuses on validating a solution, rather than making a comparison between architectures because it requires configurations that are not applicable in this research.

As future improvements, we recommend exploring integration with voice assistance platforms to improve accessibility. Additionally, the use of lightweight models for embedded devices could be evaluated to reduce dependence on online services. Although GPT-4 was used as a model in this research, the tools are compatible with other LLM models, allowing their performance to be analyzed and compared in future improvements.

ACKNOWLEDGMENTS

This paper was conducted as part of a project recognized by the Vice-Rectorate for Research at the Universidad Nacional tecnológica de Lima Sur. The authors are grateful for the facilities provided by the laboratories and technological tools for conducting this research.

REFERENCES

- [1] G. Rathee, S. Garg, G. Kaddoum, S. M. Alzanin, A. Gumaei, and M. M. Hassan, "Accessibility and ensured quality of life for disabled people using trusted edge computing," *Alexandria Eng. J.*, vol. 82, no. 1, pp. 35–42, Nov. 2023, doi: 10.1016/j.aej.2023.09.046.
- [2] A. Mahroo, L. Greci, M. Mondellini, and M. Sacco, "Assessment of a mixed reality smart home controller: HoloHome pilot study on healthy adults," *Virtual Real.*, vol. 27, no. 3, pp. 2673–2690, Sep. 2023, doi: 10.1007/s10055-023-00834-8.
- [3] S. Lindsay, K. Fuentes, S. Ragunathan, Y. Li, and T. Ross, "Accessible independent housing for people with disabilities: A scoping review of promising practices, policies and interventions," *PLoS One*, vol. 19, no. 1, p. e0291228, Jan. 2024, doi: 10.1371/journal.pone.0291228.
- [4] M. Periša, P. Teskera, I. Cvitić, and I. Grgurević, "Empowering People with Disabilities in Smart Homes Using Predictive Informing," *Sensors*, vol. 25, no. 1, p. 284, Jan. 2025, doi: 10.3390/s25010284.
- [5] E. Mori, Y. Qiu, H. Kataoka, and Y. Aoki, "A Comprehensive Analysis of a Social Intelligence Dataset and Response Tendencies Between Large Language Models (LLMs) and Humans," *Sensors*, vol. 25, no. 2, p. 477, Jan. 2025, doi: 10.3390/s25020477.
- [6] A. V. F. Porto and M. E. S. Furtado, "Framework to Specify Dialogues for Natural Interaction with Conversational Assistants Applied in Prompt Engineering," in *Lecture Notes in Networks and Systems*, vol. 1068 LNNS, Springer

- Science and Business Media Deutschland GmbH, 2024, pp. 231–253. doi: 10.1007/978-3-031-66336-9_17.
- [7] H. Du *et al.*, “Towards Proactive Interactions for In-Vehicle Conversational Assistants Utilizing Large Language Models,” in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, Aug. 2024, pp. 7850–7858. doi: 10.24963/ijcai.2024/869.
 - [8] B. Saha and U. Saha, “Enhancing International Graduate Student Experience through AI-Driven Support Systems: A LLM and RAG-Based Approach,” in *2024 International Conference on Data Science and Its Applications (ICoDSA)*, Jul. 2024, pp. 300–304. doi: 10.1109/ICoDSA62899.2024.10651944.
 - [9] T. Li, “AI-Driven Multi-Modal Interactive Learning Environment Using Deep Learning and Chain-of-Thought Reasoning,” *Int. J. Pattern Recognit. Artif. Intell.*, vol. 39, no. 03n04, p. 10, Mar. 2025, doi: 10.1142/S0218001425520019.
 - [10] S. K. Oswal and H. K. Oswal, “Examining the Accessibility of Generative AI Website Builder Tools for Blind and Low Vision Users: 21 Best Practices for Designers and Developers,” in *2024 IEEE International Professional Communication Conference (ProComm)*, Jul. 2024, pp. 121–128. doi: 10.1109/ProComm61427.2024.00030.
 - [11] W. Wang, K. Wang, and H. Du, “Design and optimization of human-machine interaction interface for the intelligent Internet of Things based on deep learning and spatial computing,” *Egypt. Informatics J.*, vol. 30, no. 1, p. 100685, Jun. 2025, doi: 10.1016/j.eij.2025.100685.
 - [12] M. M. Linda, K. M. K. M., and U. C., “Gesture Control System Using Computer Vision Techniques,” in *2024 5th International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Aug. 2024, pp. 1402–1406. doi: 10.1109/ICESC60852.2024.10689942.
 - [13] A. Anand, A. Pandey, S. Mehndiratta, H. Goyal, P. Kaushik, and R. Rathore, “Smart AI based Volume Control System: Gesture Recognition with OpenCV & Mediapipe Integration,” in *2023 International Conference on Smart Devices (ICSD)*, May 2024, pp. 1–6. doi: 10.1109/ICSD60021.2024.10751362.
 - [14] A. Taqwa, Indra Griha Tofik Isa, and Indri Ariyanti, “Designing A WSNs-based Smart Home Monitoring System through Deep Reinforcement Learning,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 7, no. 5, pp. 1224–1232, Oct. 2023, doi: 10.29207/resti.v7i5.5143.
 - [15] R. Fitriyan and S. Syafii, “Development design of an IoT-based smart home monitoring system with security features,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 34, no. 2, p. 788, May 2024, doi: 10.11591/ijeecs.v34.i2.pp788-794.
 - [16] S. Albert, M. Hamann, and E. Stokoe, “Conversational User Interfaces in Smart Homecare Interactions: A Conversation Analytic Case Study,” in *Proceedings of the 5th International Conference on Conversational User Interfaces*, Jul. 2023, pp. 1–12. doi: 10.1145/3571884.3597140.
 - [17] D. Suhas, O. Rana, S. Lannon, and C. Perera, “Talking Buildings: Interactive Human-Building Smart-Bot for Smart Buildings,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 15436 LNCS, Springer Science and Business Media Deutschland GmbH, 2025, pp. 399–415. doi: 10.1007/978-981-96-0579-8_28.
 - [18] W. Guo, S. Yamagishi, and L. Jing, “Human Activity Recognition via Wi-Fi and Inertial Sensors With Machine Learning,” *IEEE Access*, vol. 12, no. 1, pp. 18821–18836, 2024, doi: 10.1109/ACCESS.2024.3360490.
 - [19] C. Panagiotou, E. Faliagka, C. P. Antonopoulos, and N. Voros, “Multidisciplinary ML Techniques on Gesture Recognition for People with Disabilities in a Smart Home Environment,” *AI*, vol. 6, no. 1, p. 17, Jan. 2025, doi: 10.3390/ai6010017.
 - [20] Y. Chen, H. Zhang, and S. Zhong, “Design and implementation of smart home system based on IoT,” *Results Eng.*, vol. 24, no. 1, p. 103410, Dec. 2024, doi: 10.1016/j.rineng.2024.103410.
 - [21] N. Nasaruddin, M. A. P. Pratama Tresma, M. Khamdan Muchamad, and Z. Fuadi, “Voice Frequency-Based Gender Classification Using Convolutional Neural Network for Smart Home,” *IEEE Access*, vol. 12, no. 1, pp. 104190–104203, 2024, doi: 10.1109/ACCESS.2024.3434547.
 - [22] M. Javeed *et al.*, “Intelligent ADL Recognition via IoT-Based Multimodal Deep Learning Framework,” *Sensors*, vol. 23, no. 18, p. 7927, Sep. 2023, doi: 10.3390/s23187927.
 - [23] J. Kim and M. Sohn, “Graph Representation Learning-Based Early Depression Detection Framework in Smart Home Environments,” *Sensors*, vol. 22, no. 4, p. 1545, Feb. 2022, doi: 10.3390/s22041545.
 - [24] Z. Severiche-Maury *et al.*, “LSTM Networks for Home Energy Efficiency,” *Designs*, vol. 8, no. 4, p. 78, Aug. 2024, doi: 10.3390/designs8040078.
 - [25] M. Nutakki and S. Mandava, “Resilient data-driven non-intrusive load monitoring for efficient energy management using machine learning techniques,” *EURASIP J. Adv. Signal Process.*, vol. 2024, no. 1, p. 62, May 2024, doi: 10.1186/s13634-024-01157-9.
 - [26] E. King, H. Yu, S. Lee, and C. Julien, “Sasha,” *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol.*, vol. 8, no. 1, pp. 1–38, Mar. 2024, doi: 10.1145/3643505.
 - [27] T. Calò and L. De Russis, “Enhancing smart home interaction through multimodal command disambiguation,” *Pers. Ubiquitous Comput.*, vol. 28, no. 6, pp. 985–1000, Dec. 2024, doi: 10.1007/S00779-024-01827-3/TABLES/2.
 - [28] A. A. Ali, M. Mashhour, A. S. Salama, R. Shoitani, and H. Shaban, “Development of an Intelligent Personal Assistant System Based on IoT for People with Disabilities,” *Sustainability*, vol. 15, no. 6, p. 5166, Mar. 2023, doi: 10.3390/su15065166.
 - [29] A. A. H. Al-Shateri, R. A. Rashid, A. Aburaya, and N. A. Muhammad, “Luna: A Benchmark Project in the Convergence of Artificial Intelligence and Internet of Things for Home Automation,” in *2024 IEEE International Conference on Advanced Telecommunication and Networking Technologies (ATNT)*, Sep. 2024, pp. 1–4. doi: 10.1109/ATNT61688.2024.10719226.

- [30] I. A. Pap and S. Oniga, "eHealth Assistant AI Chatbot Using a Large Language Model to Provide Personalized Answers through Secure Decentralized Communication," *Sensors* 2024, Vol. 24, Page 6140, vol. 24, no. 18, p. 6140, Sep. 2024, doi: 10.3390/S24186140.
- [31] J. Lee *et al.*, "Development of AI-generated medical responses using the ChatGPT for cancer patients," *Comput. Methods Programs Biomed.*, vol. 254, no. 1, p. 108302, Sep. 2024, doi: 10.1016/j.cmpb.2024.108302.

BIOGRAPHY OF AUTHORS



Ricardo Yauri    is a Master of Science in Electronic Engineering with a Mention in Biomedical. He is associate professor at the Universidad Nacional Mayor de San Marcos (UNMSM) and Ph.D. student in Systems Engineering. He is professor at Universidad Tecnológica del Perú and Universidad Peruana del Norte. He has participated as a teacher in courses oriented to the Internet of things and applications in home automation and the Cisco academy for IoT. He was a researcher at INICTEL-UNI in the Embedded Systems and Internet of Things research group. He developed research projects on the implementation of low consumption IoT devices that involve inference techniques, machine learning algorithms. He can be contacted at email: ryauri@untels.edu.pe, ryaurir@unmsm.edu.pe.



Rafael Espino    . He is a graduate of the Faculty of Electronics of the National Universidad Nacional de Ingeniería, Perú. He is Specialist in the development of electronic systems with solid knowledge in photovoltaic systems and experience in implementing embedded firmware in electronic systems. He is currently a professor at the Universidad Tecnológica del Perú and at the Universidad de Ingeniería y Tecnología. He was a researcher at INICTEL-UNI in the Embedded Systems research group. With aptitudes for teaching, research, and project management. He can be contacted at C22165@utp.edu.pe