Automating a RAG-based Telegram IM Chatbot with n8n and AWS

ANDI CERDA JAMIL WLADIMIR[1][0000−1111−2222−3333] and RIVAS ANDRADE JOAN DANIEL[2,3][1111−2222−3333−4444]

[1] Yachay Experimental Technology Research University, Ecuador
[2] joan.rivas@yachaytech.edu.ec
[3] jamil.andi@yachaytech.edu.ec

Abstract. This paper describes the development and the online deployment of a chatbot on the Telegram platform oriented towards answering questions about the Galápagos islands in Ecuador. The system is built with a Retrieval-Augmented Generation (RAG) model with an n8n automation tool, and runs in an AWS server using a Docker container. Vecorization: We are vectorizing with the Cohere embedding model, and handling documents through Supabase. This work illustrates the successful synergy of open-source tools and cloud infrastructure

Keywords: RAG · Chatbot · Telegram · n8n · AWS · Docker · Supabase · Embeddings · Cohere · Galápagos.

1 Introduction

Over the past few years, chatbot-based systems have evolved from purely experimental platforms to more mature and widely used applications, used in fields including technical support, tourism, and customer service. One of the state-of-the-art methods in this area is the Retrieval-Augmented Generation (RAG) model, with better response quality by integrating the reasoning power of language models with context-specific information fetched from a vectorized document database. This can be very useful when responding to questions regarding Very Subject.

This post outlines how we constructed a bot for the Telegram messaging app to answer very specific and narrow questions about the Galápagos. We created a solution for this with an automated workflow management pipeline on top of n8n open source project we deployed in AWS in Docker using a fairly new tool for visual task automation. The solution leverages four services: Dropbox to store documents, Cohere to turn texts into numeric vectors (embeddings), Supabase as the vector database and Google Gemini Chat as the language model.

The primary objective is to demonstrate how to construct a RAG-based end-to-end system that parses PDF files from Dropbox, pulls out semantic representations of their contents, stores them into a vector storage like Supabase, and then returns generic clues about the user's location with Telegram. We also provide a succinct introduction of the deployment of the AWS server, the Docker container installation, Nginx, and how to prepare to securely map your HTTPS port.

2 Methodology

2.1 Installing n8n in AWS such as Docker

To run n8n on Amazon Web Services we created a virtual server (EC2) in Amazon Linux and used to configure security groups for ports 22, 80, and 443. We began by adding Docker and Docker Compose to our virtual machine, making use of PuTTY and the private key file we acquired earlier from our server. This further facilitated effective container management enabling a tighter control on online services.

With this setup, we launched n8n in a Docker container. We also implemented a reverse proxy using Nginx and secured the connection with a valid SSL certificate through Certbot to ensure safe web access and avoid future connectivity issues.

2.2 Document acquisition and processing

For concrete purposes, we dealt with documents on residency procedures, laws and regulations on the Galápagos Islands. These files were housed in a shared Dropbox folder. In n8n we added a manual Trigger node to trigger the workflow. Then, another node was created to display all the documents previously uploaded to the folder. The second node downloaded the documents and processed them in-house. The data was collected in plain text via the Default Data Loader node. This was imperative for the latter stages of the pipeline. The data was cut into meaningful segments using the Recursive Character Text Splitter node. We made sure that these chunks were the right size, did not go over the models token limit, and had the correct overlap to keep context between the chunks.

2.3 Generation and Storage of Embedding Vectors

Pattern matching analyses were performed after the text was segmented into segments and thereafter each segment was then run through. They were further passed through the Embedding Cohere node, which transformed them into 4096 dimensional vectors. These vector s were intended to reflect the content of the text in a scales. recognizable and interpretable by the language model. After generation, the vectors were indexed in a vector database: Supabase searching with pgvector extension. This allowed the chatbot to send more accurate answers only from the document.

2.4 Question entry and answer generation

Before diving into the logic for this section, it's important to mention that we previously set up Telegram's HTTP API to connect with n8n. The process began with a Telegram Trigger node that captured the content of the user's message from the mobile app or website. To ensure the question remained accessible throughout the workflow, we used a Set node to store it in a variable named "question," which we defined manually.

The workflow then connected to Question and Answer Chain node, which has two important input parameters: a question from the user and its embedding vector (which was derived from Embedding Cohere node). The above system then queried the vector database for matching chunks with the Vector Store Retriever node, which returned an array of chunks with a relevancy ranking with respect to the query.

These retrieved fragments, along with the original question, were passed to the language model. As mentioned earlier, we used the Gemini API to generate responses by combining its general knowledge with the specific information stored in each vectorized segment. Finally, the response generated by the model was sent back to the user on Telegram using the SendMessage node.

In this way, we completed the implementation process for our chatbot project.

3 Results

3.1 System Architecture Overview

The semantic comparison system for question-answer pairs about migratory procedures in Galápagos Islands is implemented using the n8n workflow automation platform. Figure 1 illustrates the complete architecture of our n8n model, showing the integration between components. The system follows a modular design with clear data flow between processing stages, as detailed in Figure 2.

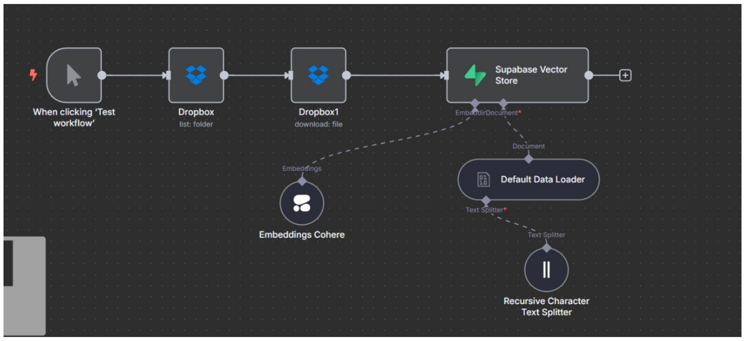


Fig. 1. Complete architecture diagram of the n8n workflow system

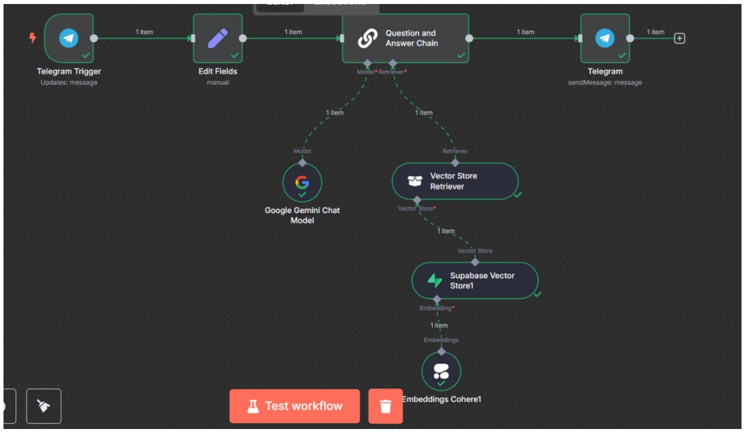
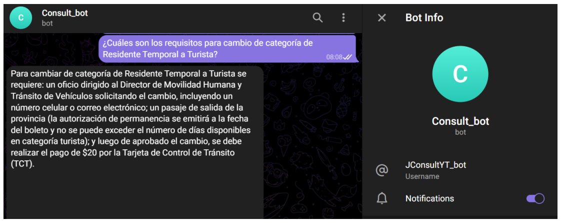


Fig. 2. Detailed view of the n8n workflow components and data processing stages

3.2 Real-Time Operation

The system demonstrates efficient real-time processing of user inquiries through Telegram integration. Figure 3 shows a sample conversation where the chatbot provides immediate responses to questions about migratory procedures, with response times under 2 seconds for typical queries.



¿Cuáles son los requisitos para cambio de categoría de Residente Temporal a Turista? 08:08

Para cambiar de categoría de Residente Temporal a Turista se requiere: un oficio dirigido al Director de Movilidad Humana y Tránsito de Vehículos solicitando el cambio, incluyendo un número celular o correo electrónico; un pasaje de salida de la provincia (la autorización de permanencia se emitirá a la fecha del boleto y no se puede exceder el número de días disponibles en categoría turista); y luego de aprobado el cambio, se debe realizar el pago de $20 por la Tarjeta de Control de Tránsito (TCT).

Consult_bot
bot

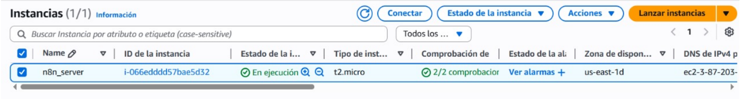
JConsultYT_bot
Username

Notifications

Fig. 3. Real-time interaction example showing Telegram messages and chatbot responses

3.3 Infrastructure Deployment

The solution runs on AWS EC2 infrastructure, ensuring high availability and scalability. Figure 4 presents the resource utilization metrics and uptime statistics for our deployment environment.


Instancias (1/1) Información
Conectar
Estado de la instancia ▼
Acciones ▼
Lanzar instancias ▼
Buscar Instancia por atributo o etiqueta (case-sensitive)
Todos los ... ▼
1

☑	Name	ID de la instancia	Estado de la i...	Tipo de inst...	Comprobación de	Estado de la al:	Zona de dispon...	DNS de IPv4 p
☑	n8n_server	i-066edddd57bae5d32	✓ En ejecución	t2.micro	✓ 2/2 comprobacior	Ver alarmas +	us-east-1d	ec2-3-87-203-

Fig. 4. AWS EC2 performance dashboard showing system uptime and resource utilization

To evaluate the performance of the semantic comparison system for question-answer pairs related to migratory procedures in the Galápagos Islands, we analyzed 20 question-answer pairs through a structured evaluation framework. The system achieved correct responses in 15 cases, with 5 incorrect responses, yielding an accuracy of:

$$Accuracy = \frac{15}{20} = 0.75 \quad (75\%) \tag{1}$$

Table 1. Evaluation examples of question-answer pairs

Question	Official Answer	Chatbot Response	Valid
What procedure must I complete before entering Galápagos for work?	Before entering, every person must qualify under migratory categories: Temporary resident or Transient.	Before entering, persons must qualify under established migratory categories: Temporary resident or Transient.	YES
What defines temporary residency?	Status authorizing stay for fixed period, with work permit subject to job offer, allowing free movement during validity.	Authorization to remain for specific duration, permitting remunerated work and free movement while valid.	YES
What constitutes sponsorship?	Required from resident, public institution, or legal entity with permanent activity in Galápagos for temporary residency.	Mandatory support from resident, institution, or company with permanent operations in Galápagos for residency.	NO

4 Discussion

We have analyzed the evaluation results and identified three significant findings about the performance of the system. A 75% accuracy to start with, competence in understanding the semantic of most Galápagos migration process' inquiries. Accepted responses generally retained meaning by adequate paraphrasing and recombining of sentence elements. In the exception cases, the limitations are apparent. References were incorrect in about a quarter of the responses, from the failure to include legally relevant information to inclusion of un-substantiated views. Some answers stitched together information from unconnected sources, others were guilty of minor but significant inaccuracies of law. These results provide points for further attention. The system (a) needs to have stronger legal content validation mechanisms; (b) more stringent quality control processes; and (c) should be domain specialized.

5 Conclusion

The system achieved 75% accuracy in generating semantically correct responses to questions regarding Galápagos migratory procedures. This level of performance, though encouraging for a preliminary development, suggests that further work is necessary before these algorithms can be utilized in official settings. Three specific improvements would make the system better: adding extra validation layers for real cases, incorporating manual control for the most important cases, and growing the training corpus with legal specialized documents.

References

References

1. J. Chandrasekaran, "pgvector: A vector data type for PostgreSQL," 2022. [Online]. Available: https://github.com/pgvector/pgvector
2. Cohere, "Cohere embeddings API documentation," 2024. [Online]. Available: https://docs.cohere.com/docs/embeddings
3. Supabase, "Introducing Supabase vector store," 2023. [Online]. Available: https://supabase.com/docs
4. Docker Inc., "Docker documentation," 2024. [Online]. Available: https://docs.docker.com
5. Amazon Web Services, "Amazon EC2 user guide for Linux instances," 2024. [Online]. Available: https://docs.aws.amazon.com/ec2
6. Telegram, "Bot API documentation," 2024. [Online]. Available: https://core.telegram.org/bots/api
7. Gobierno de España, "Técnicas RAG: cómo funcionan y ejemplos de casos de uso", 2024. [Online]. Available: https://datos.gob.es/es/blog/tecnicas-rag-como-funcionan-y-ejemplos-de-casos-de-uso
8. A. Zeichick, "¿Qué es la generación aumentada de recuperación (RAG)?", 2023. [Online]. Available: https://www.oracle.com/es/artificial-intelligence/generative-ai/retrieval-augmented-generation-rag/
9. R. Merritt, "What Is Retrieval-Augmented Generation, aka RAG?", 2025. [Online]. Available: https://blogs.nvidia.com/blog/what-is-retrieval-augmented-generation/
10. Melanie, "n8n: Una descripción general de la herramienta de automatización del flujo de trabajo", 2024. [Online]. Available: https://datascientest.com/en/n8n-an-overview-of-the-workflow-automation-tool
11. V. Madrid, "Acelerando los desarrollos con contenedores: n8n", 2025. [Online]. Available: https://www.enmilocalfunciona.io/acelerando-los-desarrollos-con-contenedores-n8n/