IFgenia: Uma IA Generativa com Abordagem RAG utilizando *n8n* para Respostas Humanizadas sobre os Editais do Programa Autonomia e Renda

Adryan N. Reis[1], **Rafaela C. S de Oliveira**[2], **Ytallo G. S. S. de Lima**[2], **Yulle G. S. de Lima**[2]

[1]Instituto Federal de Pernambuco (IFPE) - Campus Igarassu
53659-899 – Igarassu – PE – Brasil

[2]Instituto Federal de Pernambuco (IFPE) - Campus EAD

{anr,rcso2,ygs}@discente.ifpe.edu.br, yulle.gustavo@ead.ifpe.edu.br

Abstract. *The bureaucracy of public program notices represents a significant barrier for populations in situations of socioeconomic vulnerability. This paper presents IFgenia, a virtual assistant based on Retrieval-Augmented Generation (RAG) integrated with WhatsApp, developed to answer questions about the notices of the Petrobras Autonomia e Renda Program. The solution uses the low-code platform n8n to orchestrate the flow between the Evolution API, the PostgreSQL vector database with pgvector, and the Google Gemini language model. Preliminary results indicate that IFgenia has the potential to reduce the number of divergent responses compared to direct LLM queries, with the system currently in the homologation phase for integration into ServidorOn, an institutional platform of IFPE.*

Resumo. *A burocracia dos editais de programas sociais representa uma barreira significativa para populações em situação de vulnerabilidade socioeconômica. Este trabalho apresenta o IFgenia, um assistente virtual baseado em Geração Aumentada por Recuperação (RAG) integrado ao WhatsApp, desenvolvido para responder dúvidas sobre os editais do Programa Autonomia e Renda Petrobras. A solução utiliza a plataforma low-code n8n para orquestrar o fluxo entre a Evolution API, o banco de dados vetorial PostgreSQL com pgvector e o modelo de linguagem Google Gemini. Os resultados preliminares indicam que o IFgenia tem o potencial de reduzir a quantidade de respostas divergentes em relação a consultas diretas ao LLM, com o sistema atualmente em fase de homologação para integração ao ServidorOn, uma plataforma institucional do IFPE.*

1. Introdução

O Programa Autonomia e Renda Petrobras é uma iniciativa que visa ampliar oportunidades de trabalho para moradores das áreas de abrangência das operações da Petrobras, realizado em parceria com os Institutos Federais de Educação, Ciência e Tecnologia (IFs) e o SENAI. O público prioritário são pessoas em condições de vulnerabilidade socioeconômica, sem vínculo formal de emprego, com atenção especial a grupos minorizados, como mulheres, pessoas negras, indígenas, quilombolas e pessoas com deficiência [Programa Autonomia e Renda 2026].

A burocracia dos editais representa uma barreira de linguagem técnica, prazos e documentações exigidas frequentemente resultam em abstenções evitáveis. Dados oficiais indicam que o telefone celular é o dispositivo exclusivo de acesso à rede para a maior parte das classes D e E no Brasil, tornando os aplicativos de mensageria instantânea as interfaces mais democráticas para a comunicação pública [Comitê Gestor da Internet no Brasil 2025]. Nesse contexto, este trabalho apresenta o IFgenia, um assistente virtual baseado na arquitetura RAG integrado ao WhatsApp, que orquestra o fluxo entre a Evolution API, o banco de dados vetorial PostgreSQL com pgvector e o modelo de linguagem Google Gemini por meio da plataforma low-code n8n. O sistema oferece suporte contínuo (24h/7d), acessível e humanizado, sem exigir a instalação de novos softwares ou aprendizado de interfaces complexas.

2. Fundamentação Teórica

2.1. Retrieval-Augumented Generation (RAG)

Embora os LLMs apresentem capacidades notáveis, enfrentam desafios como conhecimento desatualizado e geração de informações incorretas (alucinações). A arquitetura RAG integra técnicas de recuperação de informações de bases externas ao processo de geração de texto [Gao et al. 2023], operando em três etapas: (1) indexação de documentos; (2) recuperação de trechos relevantes com base na consulta do usuário; e (3) geração da resposta final com o contexto recuperado. Ao utilizar os editais do Programa Autonomia e Renda como base verificável, o RAG aumenta a precisão e a rastreabilidade das informações fornecidas ao candidato [Swacha and Gracel 2025].

2.2. Automação *low-code* com n8n

O n8n é uma plataforma de automação de workflows de código aberto com interface visual baseada em nós, onde cada nó representa uma ação específica, como: uma requisição HTTP, execução de código JavaScript ou interação com APIs de modelos de linguagem [Barros 2026]. Por ser *self-hosted*, oferece controle total sobre dados e infraestrutura, requisitos relevantes em contextos institucionais. Sua abordagem *low-code* permite que fluxos complexos sejam criados com menor esforço de desenvolvimento, integrando nativamente serviços como WhatsApp e modelos como Gemini, Anthropic e OpenAI [n8n 2026].

2.3. Embeddings e Banco de Dados Vetoriais

Embeddings são representações numéricas de textos na forma de vetores de alta dimensão gerados por modelos pré-treinados, de modo que textos semanticamente próximos ocupam regiões vizinhas no espaço vetorial [Mikolov et al. 2013]. Isso permite que uma consulta como "Quando é o prazo para se inscrever?"recupere trechos que mencionam "data limite de submissão de candidaturas", superando as limitações das buscas por palavras-chave [Manning et al. 2008]. Para armazenar e recuperar esses vetores eficientemente, utilizou-se a extensão pgvector para PostgreSQL, uma solução de código aberto que adiciona suporte nativo à busca vetorial dentro do próprio banco relacional já utilizado pela aplicação, eliminando a necessidade de infraestrutura adicional [pgvector 2024].

3. Arquitetura e Implementação do IFgenia

A infraestrutura do IFgenia foi provisionada por meio de *containers* Docker organizados em rede interna, garantindo isolamento entre serviços, reprodutibilidade do ambiente e facilidade de implantação. O ambiente é composto por cinco serviços principais: *postgres*, *evolution-api*, n8n, *redis* e pgadmin. O banco de dados estende o PostgreSQL 15 com suporte nativo ao *pgvector* para armazenamento e recuperação dos embeddings dos editais; o Redis é utilizado pela Evolution API para gerenciamento de filas e estado das conexões WhatsApp.

A arquitetura foi implementada com três workflows distintos no n8n, descritos a seguir.

3.1. Coleta e Indexação dos dados

O *workflow* de carregamento, ilustrado na Figura 1, é responsável por popular a base de conhecimento vetorial com o conteúdo dos editais do Programa Autonomia e Renda.

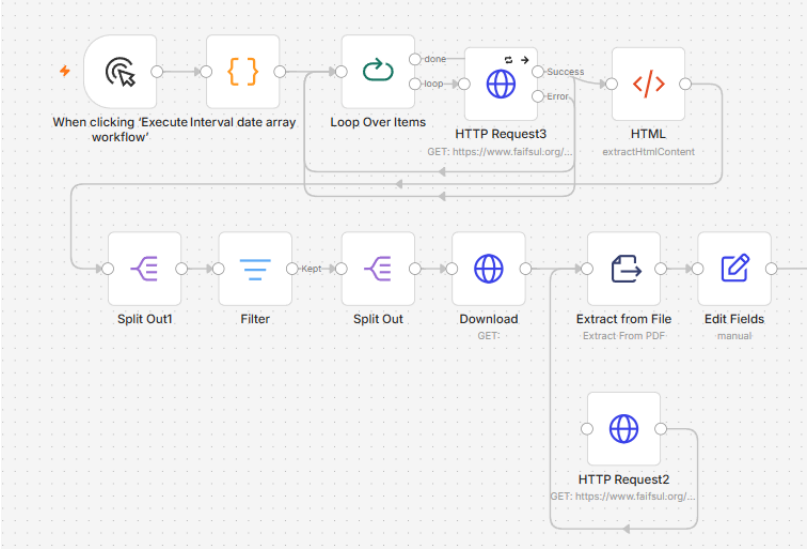


Figura 1. Workflow de download dos editais no site da FAIFSUL.

Durante essa etapa, o nó Loop Over Items itera sobre um *array* de intervalos de datas, realizando requisições HTTP (GET) ao site do FAIFSUL. O conteúdo HTML retornado é processado e os links dos editais disponíveis são extraídos. Em seguida, apenas os itens relevantes são isolados e filtrados, e então separados para serem processados individualmente. Depois, é realizado o *download* de cada PDF, logo em seguida, o conteúdo é extraído no formato JSON[1] (JavaScript Object-Notation) pelo nó *Extract From PDF* para facilitar a manipulação dos dados ao longo do fluxo. O nó Edit Fields permite a anotação manual de metadados antes do carregamento. Como alternativa ao *loop*, o *workflow* disponibiliza um caminho auxiliar para o carregamento avulso de um único documento, através do nó *HTTP Request2*.

Durante a etapa de carregamento, ilustrada na Figura 2, cada edital extraído é segmentado em trechos pelo *Recursive Character Text Splitter*, estratégia que preserva a

[1] https://www.rfc-editor.org/rfc/rfc8259

coerência semântica local ao manter trechos próximos do texto original. A cada trecho, são adicionados os metadados anotados na etapa anterior, permitindo a rastreabilidade da fonte durante a recuperação. Os trechos são então vetorizados pelo modelo *Embeddings Google Gemini* e persistidos no *Postgres PGVector Store*.

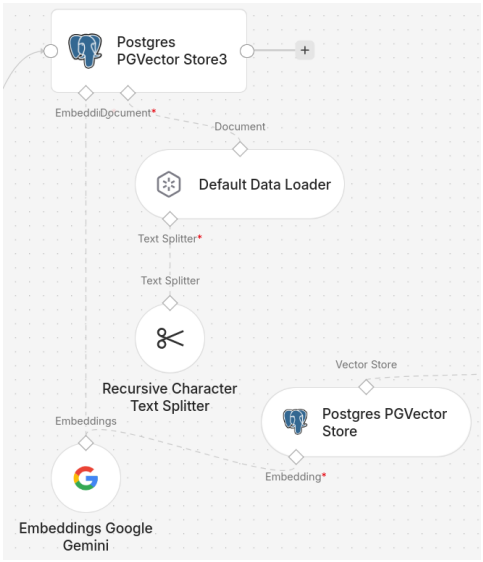


Figura 2. Workflow de carregamento dos editais

3.2. Agente de IA e geração das respostas

O *workflow* do agente, ilustrado na Figura 3, é acionado a cada mensagem recebida via WhatsApp. Um nó *Webhook* recebe a requisição e a encaminha ao nó central *AI Agent*, que coordena três subcomponentes conectados.

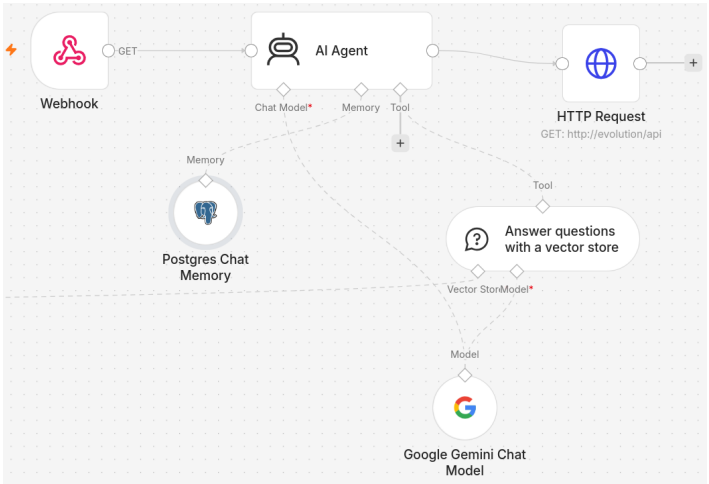


Figura 3. *Workflow* do Agente de IA

O *Google Gemini Chat Model*, vinculado ao agente, é responsável pela geração das respostas em linguagem natural. O *Postgres Chat Memory* persiste o histórico de cada conversa no PostgreSQL, conferindo continuidade ao diálogo entre sessões distintas. A ferramenta *Answer Questions with a Vector Store* consulta o *Postgres PGVector*

Store para recuperar os trechos dos editais semanticamente mais próximos à pergunta do usuário, utilizando o mesmo modelo de embeddings empregado na indexação, garantindo consistência das informações.

A resposta gerada pelo agente é encaminhada ao nó *HTTP Request*, que a repassa à Evolution API para entrega ao candidato via WhatsApp, completando o ciclo de interação. O comportamento do agente é orientado por um *system prompt* institucional, que instrui o modelo a responder de forma acessível, humanizada e restrita ao conteúdo dos editais do Programa Autonomia e Renda. Essa restrição de escopo é importante para mitigar alucinações e garantir que o assistente forneça apenas informações verificáveis acerca do Programa Autonomia e Renda.

4. Resultados e Discussões

A solução foi integrada ao ServidorOn, plataforma institucional de gerenciamento do WhatsApp desenvolvida e validada no âmbito do Instituto Federal de Pernambuco (IFPE), respaldada por iniciativas de extensão tecnológica homologadas pela Pró-Reitoria de Extensão [Lima 2025a, Lima 2025b]. A execução é financiada no escopo do Programa Autonomia e Renda Petrobras, com governança administrativa gerenciada pela Fundação Ennio de Jesus Pinheiro Amaral (FAIFSul), por meio de termo de cooperação técnica vinculado ao Edital REI/IFPE nº 31/2025-GR [Fundação Ennio de Jesus Pinheiro Amaral 2025].

Atualmente, a arquitetura encontra-se em fase de homologação, com ciclos de testes conduzidos por membros e técnicos do Programa Autonomia e Renda. O objetivo é mapear e mitigar lacunas no processo de injeção de contexto do LLM, dado que respostas divergentes das regras oficiais dos editais representam riscos críticos para as inscrições dos candidatos. A validação é conduzida por meio de uma matriz que avalia: (i) cenários de exceção e ambiguidade, verificando a fidelidade estrita ao texto original dos editais mediante prompts complexos ou contraditórios; (ii) identificação de lacunas de contexto, avaliando se a segmentação e recuperação dos dados geraram respostas acessíveis e condizentes com os editais; e (iii) consistência e humanização, analisando se as respostas são assertivas e acessíveis, de modo a desafogar as demandas repetitivas dos trabalhadores do programa.

Os resultados preliminares indicam que a abordagem RAG tem o potencial de reduzir as taxas de divergência em comparação com consultas diretas ao LLM. Os episódios pontuais de inconsistência identificados pela auditoria humana estão sendo catalogados para o refinamento contínuo dos *system prompts*, garantindo que o futuro lançamento do módulo de respostas por IA generativa no ServidorOn forneça aos participantes um suporte contínuo, rápido e confiável a qualquer hora do dia.

5. Conclusão e Trabalhos futuros

Este trabalho apresentou o IFgenia, um assistente virtual baseado na arquitetura RAG integrado ao WhatsApp, desenvolvido para democratizar o acesso às informações dos editais do Programa Autonomia e Renda. Os resultados preliminares indicam que a abordagem RAG tem o potencial de reduzir significativamente as divergências em relação a consultas diretas ao LLM, tornando o assistente uma alternativa viável ao atendimento manual para um público em situação de vulnerabilidade socioeconômica. A integração ao ServidorOn,

plataforma já consolidada no IFPE, amplia o alcance institucional da solução e favorece sua adoção em larga escala.

Como limitação e trabalhos futuros, com a conclusão da fase de homologação, pretende-se: (1) implantar o IFgenia em produção e conduzir uma validação quantitativa sistemática das respostas; (2) aprofundar os protocolos de privacidade e LGPD para o atendimento via WhatsApp; e (3) expandir a base de conhecimento para outros programas sociais do IFPE, avaliando a escalabilidade da arquitetura.

Referências

Barros, T. (2026). Página inicial | n8n brasil. Acesso em: 16 fev. 2026.

Comitê Gestor da Internet no Brasil (2025). *Pesquisa sobre o Uso das Tecnologias de Informação e Comunicação nos Domicílios Brasileiros: TIC Domicílios 2024*. Comitê Gestor da Internet no Brasil, São Paulo. Livro eletrônico.

Fundação Ennio de Jesus Pinheiro Amaral (2025). Carta de anuência para execução de projeto - edital REI/IFPE nº 31/2025-GR - proposta: *Servidor On: Otimização e Expansão de uma API de Comunicação para Gestão de Matrículas em Programas Sociais*. Documento técnico de cooperação, FAIFSul / Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco (IFPE), Pelotas. Parceria com o Programa Autonomia e Renda Petrobras.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, H., Wang, H., et al. (2023). Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1):32.

Lima, Y. G. S. (2025a). Servidoron: Otimização e expansão de uma API de comunicação para gestão de matrículas em programas sociais. Projeto de extensão tecnológica - edital REI/IFPE nº 31/2025-GR, Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco (IFPE), Recife. Homologado e Aprovado com Bolsa. Parceria com a Fundação FAIFSul e Programa Autonomia e Renda Petrobras.

Lima, Y. G. S. (2025b). Servidoron: Propagação e treinamento do *software* de comunicação descentralizado para instituições federais de ensino. Programa de extensão - edital PIBEX 2025-2026, Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco (IFPE), Recife. Homologado e Aprovado com Bolsa.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to information retrieval*. Cambridge university press.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

n8n (2026). n8n integrations: Communication categories. Acesso em: 17 jun. 2026.

pgvector (2024). Open-source vector similarity search for Postgres. `https://github.com/pgvector/pgvector`. Acesso em: 17 jun. 2026.

Programa Autonomia e Renda (2026). Sobre o programa. Acesso em: 17 jun. 2026.

Swacha, J. and Gracel, M. (2025). Retrieval-augmented generation (rag) chatbots for education: A survey of applications. *Applied Sciences*, 15(8):4234.