

BOT TELEGRAM COM N8N E IA

Automação de Processos com Inteligência Artificial

Projeto: Desafio Técnico B2BIT

Candidato: Matheus Bezerra Dantas Saraiva

Data: 25 de Outubro de 2025

Bot: @MeuBot8n8_bot

SUMÁRIO

1. Introdução
2. Visão Geral do Sistema
3. Casos de Uso Implementados
 - 3.1. Caso 1: Criar Documento Simples
 - 3.2. Caso 2: Documento com Marcadores e Imagem
 - 3.3. Caso 3: Buscar Notícias do G1
 - 3.4. Caso 4: Transcrição de Áudio
4. Assistente Virtual com IA
5. Hospedagem e Infraestrutura
6. Desafios

1. INTRODUÇÃO

Este projeto consiste no desenvolvimento de um **bot automatizado no Telegram** que utiliza o **N8N** como plataforma de orquestração de workflows e **Inteligência Artificial** para oferecer uma experiência interativa aos usuários.

Objetivo do Projeto

Criar uma solução completa de automação que permita aos usuários:

- Gerar documentos do Google Docs de forma automatizada
- Inserir imagens e textos em templates personalizados
- Consultar notícias em tempo real
- Transcrever áudios utilizando IA
- Interagir com um assistente virtual inteligente

Tecnologias Utilizadas

- **N8N**: Plataforma de automação low-code/no-code
- **Telegram Bot API**: Interface de comunicação com usuários
- **Google Workspace**: Drive e Docs para geração de documentos
- **OpenAI GPT-4o-mini**: Modelo de linguagem para assistente virtual
- **Gladia API**: Serviço de transcrição de áudio
- **Cloudinary**: Armazenamento temporário de arquivos
- **HTTP Requests**: Scraping de notícias do portal G1

2. VISÃO GERAL DO SISTEMA

O sistema foi projetado seguindo uma arquitetura orientada a eventos, onde cada mensagem recebida no Telegram aciona o workflow do N8N, que processa a solicitação e retorna a resposta apropriada ao usuário. O workflow final é mostrado na Figura 1.

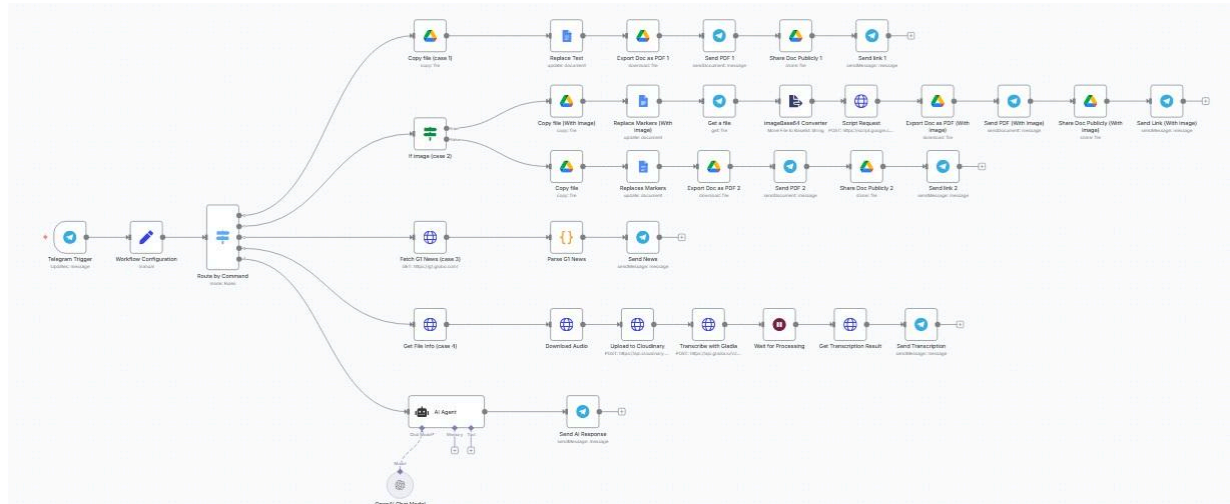


Figura 1: Workflow completo

Fluxo de Funcionamento

1. **Usuário** envia uma mensagem ou comando no Telegram
2. **Telegram Bot API** captura a mensagem via webhook
3. **N8N Workflow** recebe o trigger e processa a requisição
4. **Route by Command** (Switch) identifica o tipo de solicitação
5. **Processamento** específico para cada caso de uso
6. **Integrações externas** (Google, APIs, IA) são acionadas
7. **Resposta** é enviada de volta ao usuário no Telegram

Arquitetura de Integração

O N8N atua como orquestrador central, conectando diversas tecnologias:

Comunicação:

- Telegram Bot API para recebimento e envio de mensagens
- Webhooks para comunicação em tempo real

Armazenamento:

- Google Drive para templates e documentos gerados
- Google Docs para manipulação de conteúdo
- Cloudinary para armazenamento temporário de áudios

Inteligência Artificial:

- OpenAI GPT-4o-mini para assistente conversacional
- Gladia API para transcrição de áudio com 98%+ de precisão

Dados Externos:

- HTTP Requests para scraping do portal G1
- Parsing customizado de HTML para extração de notícias

Na Figura 2 temos o esboço do fluxograma simplificado gerado a partir das etapas de disparo do gatilho, processamento e entrega dos respostas.

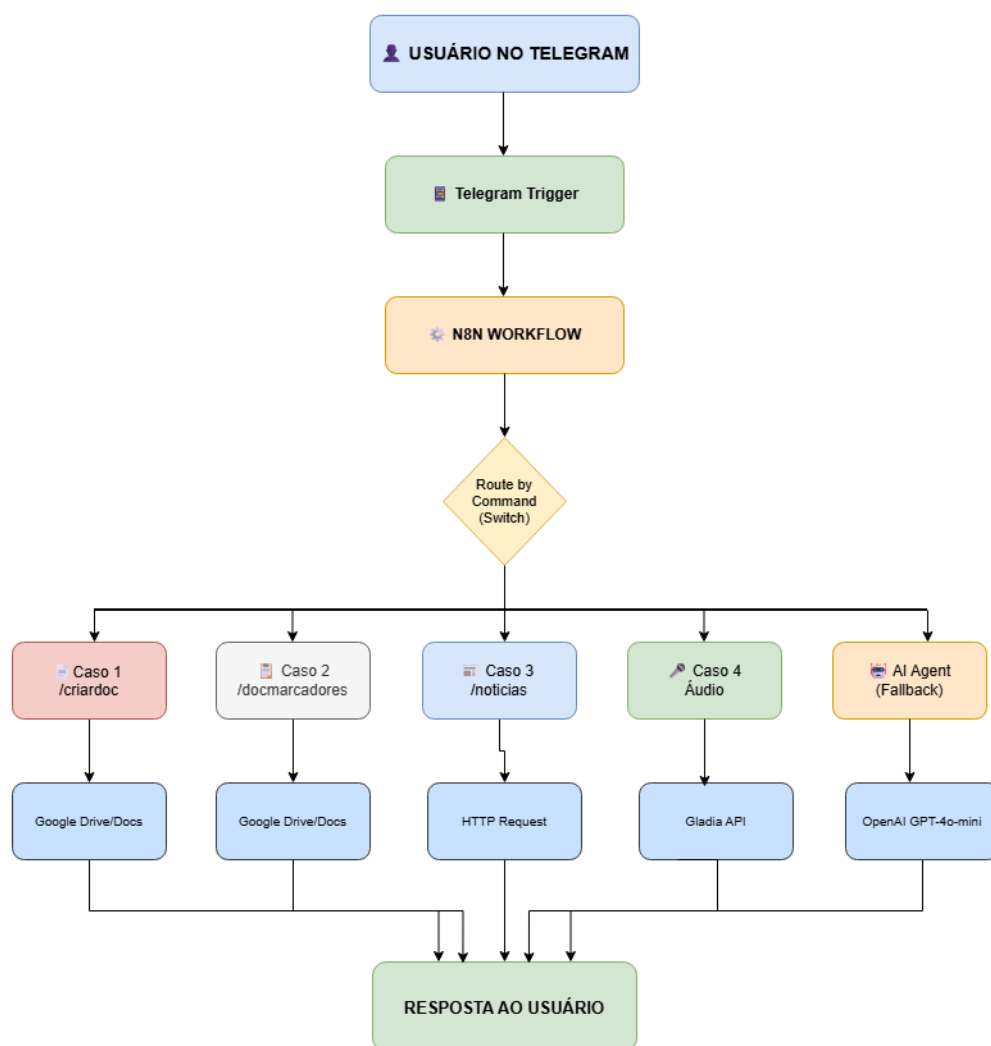


Figura 2: Fluxograma simplificado

3. CASOS DE USO IMPLEMENTADOS

3.1. CASO 1: Criar Documento Simples

Descrição

Permite ao usuário criar um documento do Google Docs contendo texto personalizado. O sistema gera automaticamente o documento, exporta em PDF e compartilha um link público. Para esse caso de uso foram usados o Google Drive API, Google Docs API e Telegram Bot API.

Comando

```
/criardoc [seu texto aqui]
```

Fluxo técnico

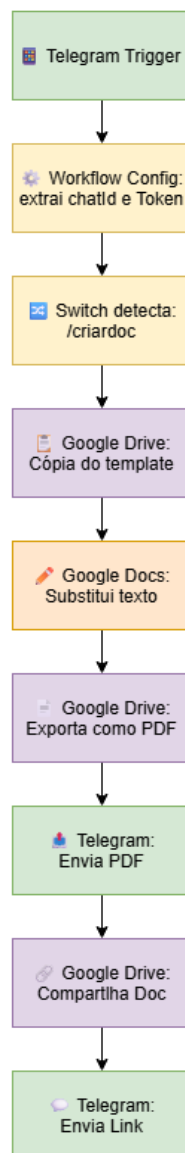


Figura 3: Fluxograma do caso de uso 1

Etapas

1. **Telegram Trigger:** Captura a mensagem do usuário
2. **Workflow Configuration:** Extrai chatId e token do bot
3. **Switch (Route by Command):** Detecta o comando `/criardoc`
4. **Copy Template (Google Drive):** Duplica o documento base
5. **Replace Text (Google Docs):** Substitui os marcadores `[texto]` e `[Data]` pelo conteúdo fornecido
6. **Export as PDF (Google Drive):** Converte o documento para formato PDF
7. **Send PDF (Telegram):** Envia o arquivo PDF ao usuário
8. **Share Publicly (Google Drive):** Torna o documento acessível publicamente
9. **Send Link (Telegram):** Envia o link editável do Google Docs

Resultados

O usuário recebe:

- Arquivo PDF do documento
- Link público para visualização/edição no Google Docs
- Confirmação com data/hora de criação

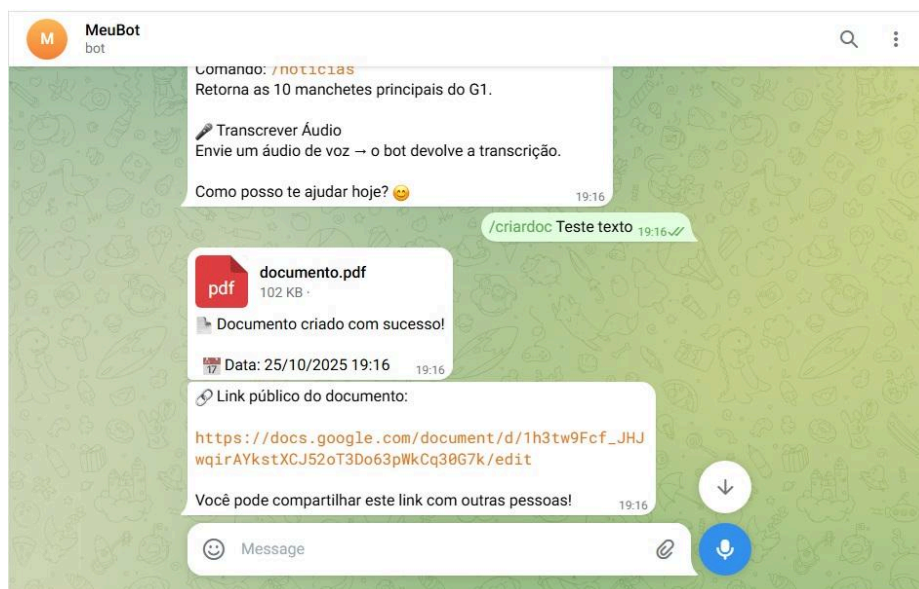


Figura 4: Teste do caso de uso 1

3.2. CASO 2: Documento com Marcadores e Imagem

Descrição

Funcionalidade avançada que permite criar documentos com marcadores personalizados e inserção de imagens. Ideal para relatórios, currículos e documentos formatados.

Para esse caso de uso foi interessante dividir o fluxo em 2 etapas: com imagem anexada e sem imagem anexada, para isso foi necessário adicionar um nó IF. Para o caso sem imagem o fluxograma é bem semelhante ao caso de uso 1. Para o caso com imagem foram necessários adicionar alguns nós a mais para fazer o download da imagem no telegram, fazer a conversão do binário para imageBase64 (visto que o n8n apenas armazena uma referência nesse valor) e, por fim, fazer uma requisição HTTP para um script externo que faz a inserção da imagem no .doc.

Fluxograma:

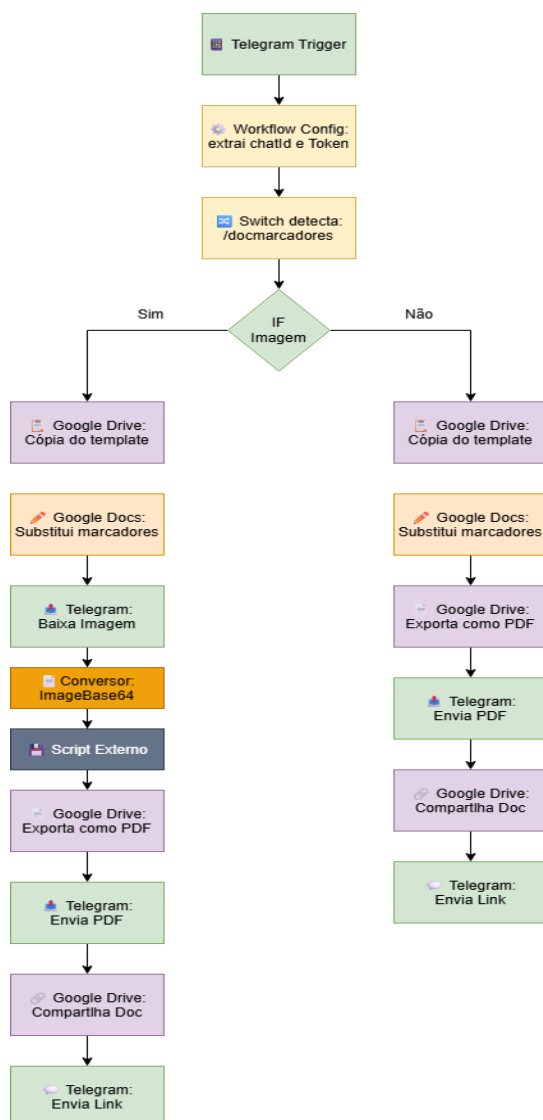


Figura 5: Fluxograma do caso de uso 2

Comando

/docmarcadores (imagem anexada ou não)

Título: [seu título]

Texto: [seu conteúdo]

Fluxo COM Imagem:

1. **Telegram Trigger:** captura mensagem com foto
2. **IF:** detecta presença de imagem
3. **Google Drive:** Cópia do template
4. **Google Docs:** Substitui marcadores <<título>>, <<texto>>, <<data>>
5. **Get File (Telegram):** Baixa a imagem enviada
6. **ImageBase64 Converter:** Converte para Base64
7. **Script Externo (Google Apps Script):** Insere imagem no marcador <<imagem>>
8. Export as PDF e envio ao usuário

Fluxo SEM Imagem:

- 1 - 4. Mesmos passos iniciais
5. Pula etapas de imagem
6. Export as PDF e envio ao usuário

Resultado

- Documento formatado com marcadores substituídos
- Imagem inserida (se fornecida)
- PDF gerado
- Link público compartilhável

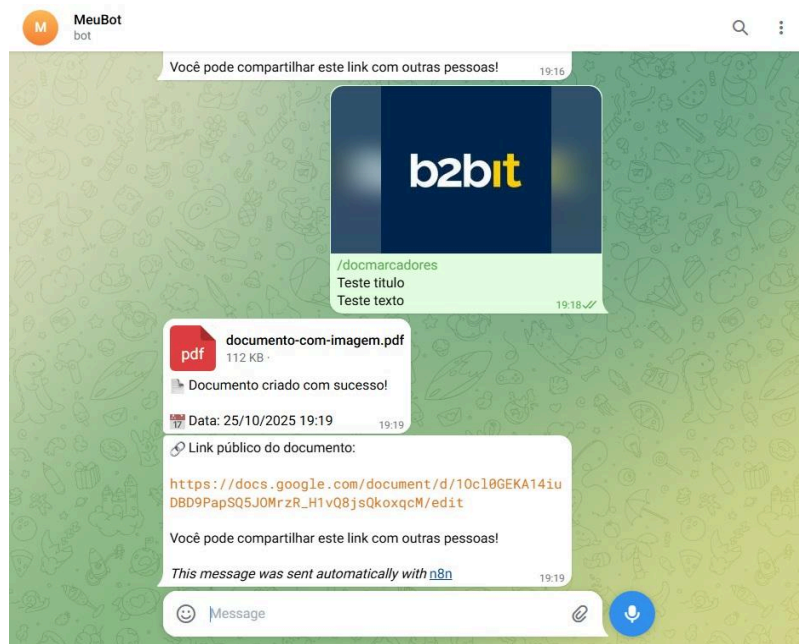


Figura 6: Teste do caso de uso 2


DESAFIO BOT TELEGRAM B2BIT

Data: 25/10/2025 19:19

TÍTULO: Teste titulo

DESCRIÇÃO: Teste texto



Figura 7: Documento obtido pelo teste da Figura 6

3.3. CASO 3: Buscar Notícias do G1

Descrição

Realiza web scraping do portal G1 e retorna as 10 principais manchetes do momento com links diretos para as notícias.

Fluxograma:

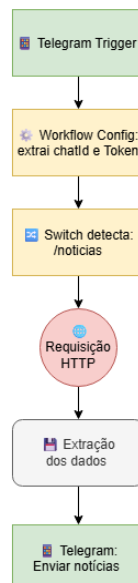


Figura 8: Fluxograma do caso de uso 3

Comando

/noticias

Fluxo Técnico

1. **Telegram Trigger:** Recebe comando `/noticias`
2. **HTTP Request:** Faz requisição GET para `https://g1.globo.com/`
3. **Parse HTML (Code Node):** Analisa o HTML retornado
 - Utiliza RegEx para encontrar links do G1
 - Extrai título e URL de cada notícia
 - Filtra resultados (mínimo 10 caracteres, sem JavaScript)
 - Limita a 10 notícias
4. **Format Message:** Estrutura a resposta em Markdown
5. **Send News (Telegram):** Envia lista formatada ao usuário

Resultado

- O usuário recebe uma lista formatada

Tecnologias Envolvidas

1. HTTP Request (Node.js)
2. RegEx para parsing de HTML
3. Formatação Markdown

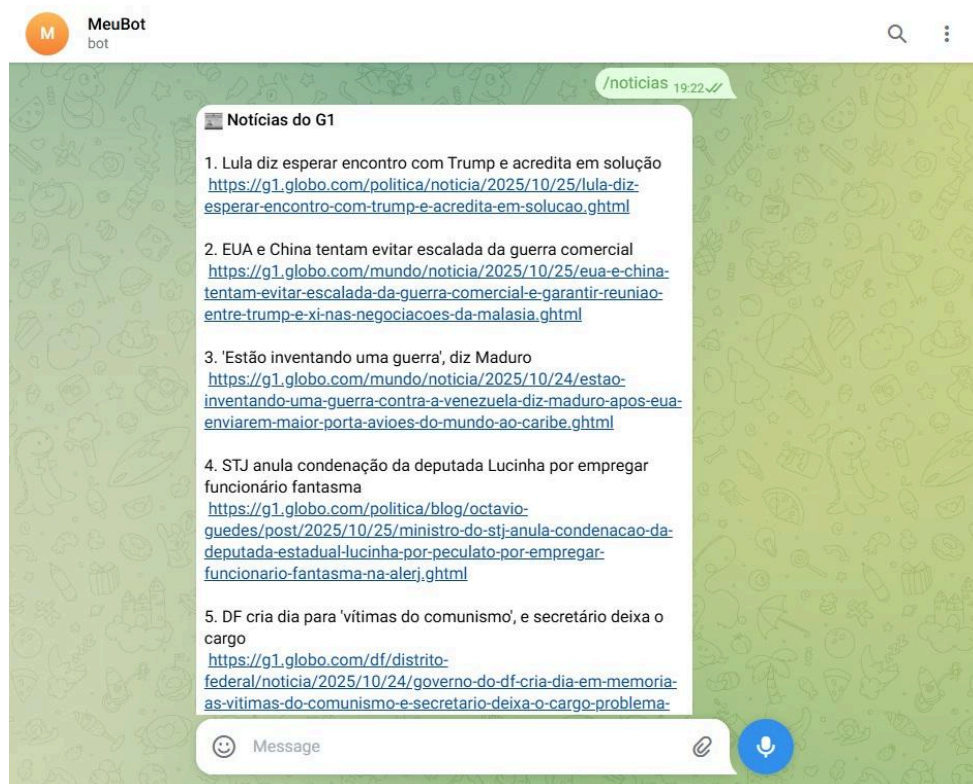


Figura 9: Teste do caso de uso 3

3.4. CASO 4: Transcrição de Áudio

Descrição

Transcreve automaticamente áudios de voz enviados pelo usuário utilizando Inteligência Artificial. Funciona com áudios em português de 5 a 30 segundos. Um detalhe importante é que a princípio apenas áudios gravados via celular estão funcionando.

Fluxograma:



Figura 10: Fluxograma do caso de uso 4

Como Usar

1. Grave um áudio de voz no Telegram
2. Envie para o bot
3. Aguarde a transcrição (10-15 segundos)

Fluxo Técnico

Telegram Trigger: Detecta mensagem de voz

Get File Info: Obtém metadados do áudio via Telegram API

Download Audio: Baixa o arquivo OGG do Telegram

Upload to Cloudinary: Armazena temporariamente o áudio

Transcribe with Gladia: Envia para API de transcrição

- Configuração: idioma português (pt)
- Modelo: general-purpose

Wait 10 seconds: Aguarda processamento assíncrono

Get Transcription Result: Busca resultado via result_url

Send Transcription: Retorna texto transcrito ao usuário

Resultado

- Transcrição do áudio enviado ao usuário

Tecnologias Envolvidas

- Telegram File API
- Cloudinary Storage
- Gladia AI (transcrição)
- Processamento assíncrono

Foi realizado um teste conforme mostra a Figura 11. Neste teste foi enviado um áudio via celular com a expressão: Alô, testando...

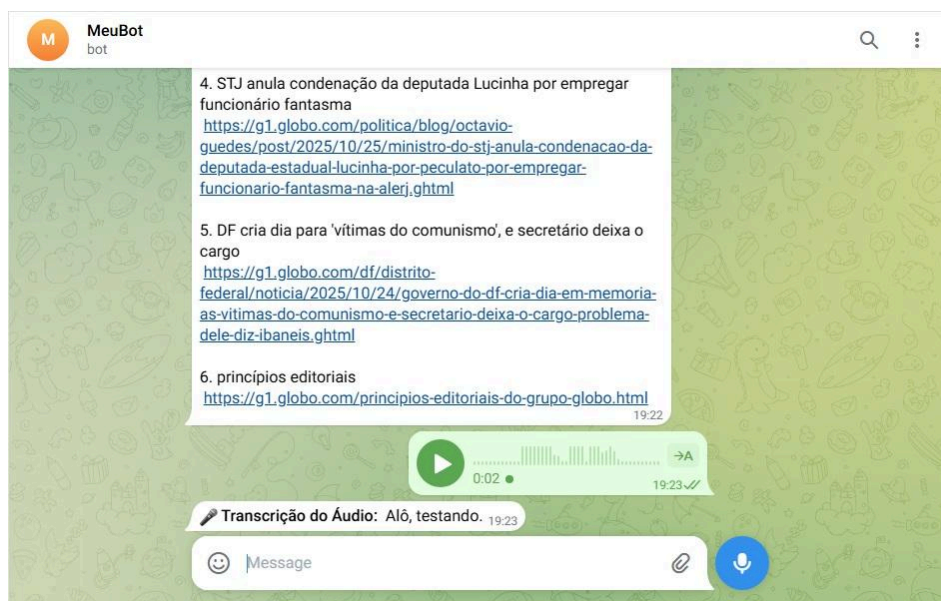


Figura 11: Teste do caso de uso 4

4. ASSISTENTE VIRTUAL COM IA

Descrição

O bot conta com um Agente de Inteligência Artificial (AI Agent) que atua como assistente virtual, respondendo perguntas sobre funcionalidades, orientando usuários e fornecendo suporte conversacional.

Fluxograma:

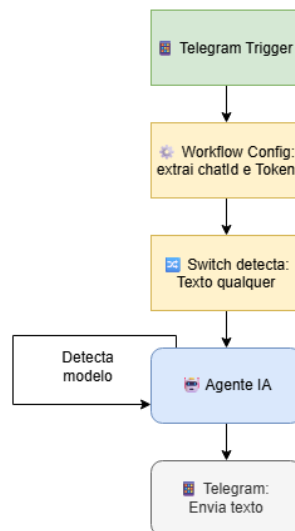


Figura 12: Fluxograma do Assistente de IA

Tecnologia

- **Modelo:** OpenAI GPT-4o-mini
- **Função:** Assistente conversacional
- **Ativação:** Qualquer mensagem que não seja comando específico

Capacidades do AI Agent

1. Explicar Funcionalidades

- Responde dúvidas sobre comandos
- Fornece exemplos práticos de uso
- Orienta novos usuários

2. Informações sobre a B2BIT

- Descreve serviços da empresa
- Explica áreas de atuação

3. Suporte Contextual

- Entende perguntas em linguagem natural
- Fornece respostas personalizadas
- Mantém tom amigável e profissional

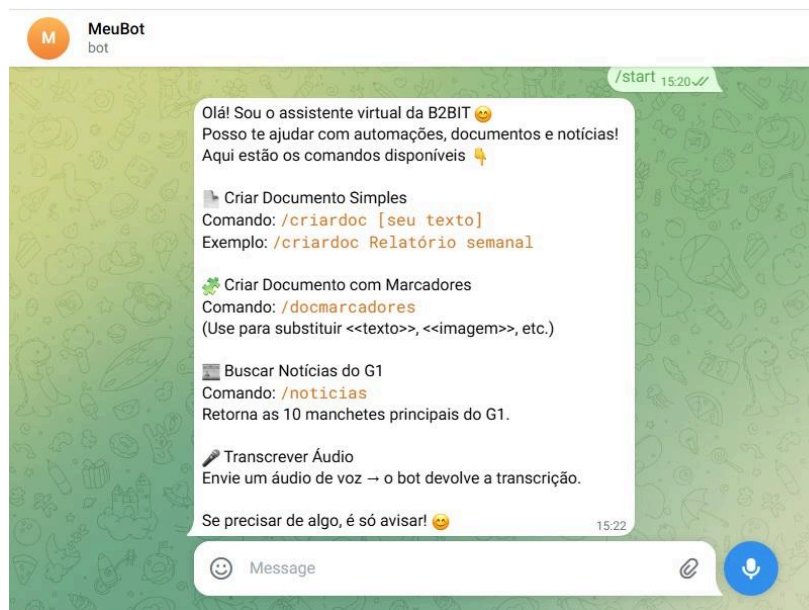


Figura 13: Teste assistente de IA

5. HOSPEDAGEM E INFRAESTRUTURA

Estado Atual

Houveram algumas tentativas de hospedagem nas plataformas Render e Railway.app, ambas gratuitas, porém sem sucesso devido problemas de importação das credenciais. Com isso, a melhor alternativa foi optar por priorizar a qualidade da implementação das funcionalidades e a robustez das integrações ao invés de focar em hospedagem própria. Por fim, o bot ficou hospedado apenas na própria n8n.

6. DESAFIOS

Desafio 1: Inserção de Imagens no Google Docs

Problema: A API do Google Docs não suporta inserção direta de imagens via Node.js.

Solução: Implementação de um **Google Apps Script** externo que:

1. Recebe imagem em Base64
2. Converte para Blob
3. Insere no documento via Apps Script API
4. Retorna confirmação

Desafio 2: Transcrição de Áudio em Tempo Real

Problema: Telegram envia áudios em formato OGG, que precisa de conversão e upload para API externa.

Solução:

1. Cloudinary para storage temporário
2. Gladia API com processamento assíncrono
3. Wait node de 10s para aguardar processamento
4. Polling do resultado via result_url

Desafio 3: AI Agent com Context

Problema: Manter contexto da conversa sem banco de dados.

Solução: System prompt detalhado com:

- Instruções claras sobre funcionalidades
- Exemplos de interação
- Diretrizes de tom e estilo
- Limitações explícitas

Informações adicionais

Link do Bot: https://t.me/MeuBot8n8_bot

Demais arquivos serão enviados em anexo