

Documentação do Projeto: Bot de Automação (Telegram + n8n)

Application ID: 86acf5hfk

Nome: Pedro Henrique da Silva Mendonça

1. Visão Geral do Projeto

1.1. Objetivo

Este projeto implementa um bot multifuncional no Telegram, orquestrado inteiramente pelo n8n. O bot é capaz de lidar com múltiplos casos de uso, desde a geração de documentos e manipulação de arquivos no Google Drive até a integração com APIs de terceiros para busca de notícias e transcrição de áudio por Inteligência Artificial.

A solução é construída com foco em ser robusta, escalável e de fácil manutenção, utilizando um sistema de gerenciamento de estado para permitir conversas complexas e de múltiplos passos com o usuário.

1.2. Handle do Bot no Telegram

O bot pode ser encontrado e testado no Telegram através do username:

[@TesteLowCoderBot](#)

1.3. Tecnologias Utilizadas

- **Orquestração:** n8n (Cloud)
- **Interface do Usuário:** Telegram Bot
- **Gerenciamento de Estado (Memória):** Google Sheets
- **Manipulação de Documentos:** Google Drive & Google Docs
- **Solução de Imagem (Bridge):** Google Apps Script
- **Transcrição de Áudio (IA):** Gladia.io (API Gratuita)
- **Busca de Notícias:** Feed RSS do G1 (via HTTP Request e XML)

2. Arquitetura da Solução

O "cérebro" do bot é um único workflow mestre no n8n. A arquitetura não se baseia em gatilhos (triggers) separados para cada comando, mas sim em um **Roteador Mestre** que gerencia o estado da conversa.

1. **Gatilho Único:** O **Telegram Trigger** recebe *todas* as mensagens (comandos, textos, imagens, áudios).
2. **Busca de Estado:** Imediatamente, o workflow consulta a planilha **EstadoBots** no Google Sheets (usando o **ChatID** como chave) para descobrir o que o usuário estava fazendo por último (o **EstadoAtual**).
3. **Normalização de Dados:** Um nó **Edit Fields (Set)** prepara todas as variáveis necessárias (ChatID, Mensagem, file_id, e o **EstadoAtual** que foi lido do Sheets).

4. **Roteador Mestre (Switch):** O fluxo é dividido com base na *mensagem* do usuário.
 - **Caminho 1: É um Comando (começa com "/")**
 - Um segundo **Switch** identifica *qual* comando foi (`/criardocs`, `/inserirtextimg`, etc.).
 - O bot, então, envia uma mensagem de resposta e **atualiza o estado** do usuário no Google Sheets (ex: `Set Estado = "aguardando_texto_pdf"`).
 - **Caminho 2: É uma Resposta (Texto, Imagem ou Áudio)**
 - O fluxo é direcionado para um **Switch** de **Contexto** que lê a variável `EstadoAtual`.
 - Se `EstadoAtual` for `"aguardando_texto_pdf"`, o bot executa o Caso 1.
 - Se `EstadoAtual` for `"aguardando_envio_da_img"`, o bot executa a segunda parte do Caso 3.
5. **Limpeza de Estado:** Após qualquer fluxo ser concluído, um nó final do Google Sheets é chamado para limpar o `Estado` e `DadosTemporarios` do usuário, deixando-o pronto para um novo comando.

3. Casos de Uso e Exemplos

Caso 1: Criar Documento (PDF Simples)

- **Comando:** `/criardocs`
- **Descrição:** O usuário solicita a criação de um documento. O bot define o estado como `aguardando_texto_pdf`, espera o texto do usuário, insere esse texto em um modelo de "papel timbrado" no Google Docs, gera um link de PDF e o envia de volta.
- **Exemplo de Uso:**
 1. **Usuário:** `/criardocs`
 2. **Bot:** "Mande o texto que você quer transformar em pdf."
 3. **Usuário:** "Este é um relatório de teste para a automação."
 4. **Bot:** [Envia o arquivo PDF e o link público do Google Drive]

Caso 2: Inserir Texto e Imagem em Marcadores

- **Comando:** `/inserirtextimg`
- **Descrição:** Este é o fluxo mais complexo, exigindo múltiplos passos e gerenciamento de estado. O bot primeiro solicita o texto e o armazena na coluna `ESTADO` do Google Sheets. Em seguida, solicita a imagem.
- **Desafio Técnico:** O nó padrão do Google Docs no n8n não permite substituir texto por imagem. Para contornar isso, foi criado um **Google Apps Script** (implantado como Web App). O fluxo n8n atualiza o texto (`<<texto>>`) e, em seguida, faz um **HTTP Request** para esse Apps Script, passando o `docId` e a `imageUrl`, que por sua vez insere a imagem no marcador `<>`.
- **Exemplo de Uso:**
 1. **Usuário:** `/inserirtextimg`
 2. **Bot:** "O escopo do arquivo está definido assim... Envie seu texto!"
 3. **Usuário:** "Relatório Fotográfico de Campo"
 4. **Bot:** "Texto recebido. Agora envie a imagem para `<>`."
 5. **Usuário:** [Envia uma foto]

6. **Bot:** [Envia o link do documento e PDF com o texto e a imagem inseridos nos locais corretos]

Caso 3: Pegar Notícias do G1

- **Comando:** `/buscarnoticia`
- **Descrição:** Um fluxo direto e sem estado. O bot faz um **HTTP Request** para o feed RSS oficial do G1 (<https://g1.globo.com/rss/g1/>). A resposta (XML) é convertida para JSON com um nó **XML**, e um nó **Code** formata as 5 primeiras notícias (título e link) em uma única mensagem de texto.
- **Exemplo de Uso:**
 1. **Usuário:** `/buscarnoticia`
 2. Bot: "Aqui estão as 5 principais notícias do G1:
 - Título da Notícia 1 - link...
 - Título da Notícia 2 - link...
 - ..."

Caso 4: Transcritor de Áudio (IA)

- **Comando:** `/transcreveraudio`
- **Descrição:** O bot define o estado como `aguardando_envio_de_audio`. Quando o usuário envia um áudio ou mensagem de voz, o bot baixa o arquivo (**Telegram: Get File**), envia o binário para a API da **Gladia.io** via **HTTP Request** (usando **Form-Data** e **n8n Binary File**). A resposta JSON da IA (uma lista de segmentos de transcrição) é processada e juntada em um único texto, que é enviado de volta ao usuário.
- **Exemplo de Uso:**
 1. **Usuário:** `/transcreveraudio`
 2. **Bot:** "Ok, me envie o áudio que você deseja transcrever."
 3. **Usuário:** [Envia uma mensagem de voz de 10 segundos]
 4. Bot: "Transcrição:
Bom dia, como está? Isso aqui é um teste."