



UNIVERSIDADE FEDERAL DO MARANHÃO
CAMPUS DE SÃO LUÍS - CIDADE UNIVERSITÁRIA
ENGENHARIA DA COMPUTAÇÃO

PROJETO E DESENVOLVIMENTO DE SOFTWARE

TURMA 1

TUTOR.IA - ASSISTENTE VIRTUAL
MANUAL DE INSTALAÇÃO

PROFESSOR:THALES LEVI AZEVEDO VALENTE

ALUNOS:

EMANUEL LOPES SILVA - 2021017818

GABRIEL FELIPE CARVALHO SILVA - 2023098664

LETÍCIA DELFINO DE ARAUJO-2021061763

THALES AYMAR FORTES DE SOUZA - 2021018145

VERÔNICA LIMA COSTA - 2022029493

São Luís – MA

22/01/2025

Manual do Instalação

TUTOR.IA

Seu assistente acadêmico inteligente



Desenvolvido por : ADA.NET

Versão 1.0 – Fevereiro de 2025

Índice

1.0 - Introdução.....	4
2.0 - Instalação.....	5
3.0 - Execução.....	7
4.0 - Nós e Funcionalidades.....	8
4.1 - Webhooks.....	8
4.2 - Bots.....	9
4.3 - Classificadores de Texto.....	10
4.4 - Nós de Serviço do Google.....	11
4.5 - Nós de Envio de Mensagem.....	13
5.0 - Conclusão.....	15

1.0 - Introdução

Este documento tem por finalidade orientar os usuários na instalação e configuração do Tutor.IA, garantindo que o sistema funcione corretamente e possa desempenhar todas as suas funcionalidades com eficiência. Aqui serão apresentadas instruções detalhadas sobre os requisitos necessários, os passos de instalação e os procedimentos para ativação e testes, permitindo que você configure o Tutor.IA sem dificuldades.

O Tutor.IA foi desenvolvido pela ADA.NET para auxiliar estudantes no gerenciamento de suas atividades acadêmicas, eliminando o risco de esquecer prazos de entrega, provas e compromissos importantes. Ele automatiza o envio de lembretes personalizados via WhatsApp, Telegram e E-mail (Gmail) e realiza a integração com o Google Calendar, garantindo que todas as atividades estejam organizadas e acessíveis em um só lugar.

Além disso, o sistema conta com um chatbot acadêmico capaz de fornecer suporte instantâneo para diversas necessidades do estudante, como:

- Resumos automáticos de conteúdos e textos acadêmicos;
- Esboços estruturados para organização de trabalhos;
- Planejamento de estudos personalizado, com base nas necessidades do usuário;
- Consulta de datas de atividades cadastradas no Google Calendar.

O Tutor.IA funciona sem interface gráfica, operando por meio de automação no n8n e integração com diversas plataformas, tornando a experiência do usuário prática e intuitiva. Com ele, estudantes podem otimizar sua rotina acadêmica, evitar esquecimentos e melhorar significativamente seu desempenho acadêmico.

Nos capítulos a seguir, você encontrará todas as orientações necessárias para instalar, configurar e ativar o Tutor.IA, garantindo que ele esteja pronto para automatizar suas atividades acadêmicas de forma eficaz.

2.0 - Instalação

Para instalar e configurar o Tutor.IA, é necessário preparar o ambiente do n8n, uma plataforma de automação de fluxos de trabalho que permitirá a execução dos processos do sistema. Antes de iniciar a instalação, é fundamental garantir que o sistema atenda aos requisitos necessários. O n8n pode ser instalado em Windows 10, macOS ou Linux, sendo essencial ter o Node.js na versão 20.16.0 ou superior, juntamente com o NPM (Node Package Manager). Além disso, recomenda-se o uso do Docker para um ambiente mais seguro e organizado, especialmente em produção.

A instalação do n8n pode ser feita de diferentes formas, dependendo da preferência e do ambiente de execução. Para aqueles que desejam instalá-lo manualmente via NPM, basta primeiro instalar o Node.js a partir do site oficial, e então executar o comando, como segue:

Algoritmo 1 : Código de Instalação do n8n

1. `npm install -g n8n`
-

Após executar esse comando no terminal, o que instalará o n8n globalmente no sistema. Após a instalação, o serviço pode ser iniciado digitando n8n no terminal, e ele ficará acessível via host local, na porta 5678, ou de forma mais simples, <http://localhost:5678>.

Entretanto, **de forma optativa**, para criar um ambiente mais estável e seguro, recomenda-se a instalação via Docker. Para isso, é necessário instalar o Docker e, em seguida, executar o comando (Algoritmo 2):

Algoritmo 2 : Código de Instalação do n8n com o Docker

1. `docker volume create n8n_data`
 - 2.
 3. `docker run -it --rm --name n8n -p 5678:5678 -v n8n_data:/home/node/.n8n docker.n8n.io/n8nio/n8n`
-

Esse comando iniciará o n8n. Outra opção, mais robusta, é configurar o Docker Compose, permitindo que o n8n continue rodando em segundo plano sem a necessidade de reiniciar manualmente. Para isso, deve-se criar um arquivo chamado “docker-compose.yml” com as configurações adequadas e rodar o comando “docker-compose up -d”, o que fará com que o n8n inicie automaticamente sempre que o sistema for ligado.

Após a instalação do n8n, é necessário importar o fluxo do Tutor.IA. Para isso, deve-se acessar a interface do n8n através do navegador, no endereço <http://localhost:5678>, e

fazer login, caso tenha sido configurada uma autenticação. No menu principal, a opção Import Workflow permitirá o carregamento do arquivo JSON do Tutor.IA, que contém todas as configurações do fluxo de automação. Após a importação, será necessário configurar as credenciais dos serviços utilizados, incluindo WhatsApp API, Telegram, Gmail e Google Calendar, garantindo que o Tutor.IA consiga interagir corretamente com essas plataformas.

Para testar se o Tutor.IA está funcionando corretamente, o usuário pode enviar uma mensagem de teste no WhatsApp ou Telegram e verificar se o sistema responde conforme esperado. Além disso, deve-se conferir se as atividades acadêmicas estão sendo registradas corretamente no Google Calendar e se os e-mails são enviados pelo Gmail. Caso todos os testes sejam bem-sucedidos, o Tutor.IA estará totalmente configurado e pronto para auxiliar os estudantes no gerenciamento de suas atividades acadêmicas, garantindo organização, pontualidade e eficiência nos estudos.

Além disso, opcionalmente, existe a ferramenta Ngrok, que permite a exposição de servidores locais para a internet, criando túneis seguros que facilitam a comunicação entre aplicações locais e serviços externos. No contexto do Tutor.IA, o Ngrok pode ser utilizado para expor a instância do n8n rodando localmente, permitindo que webhooks do WhatsApp, Telegram e Gmail funcionem corretamente, mesmo sem um servidor público. Isso é especialmente útil para testes ou implementações temporárias antes da implantação em um servidor dedicado. O ngrok pode ser baixado e instalado diretamente de seu site oficial e cria-se o túnel para o n8n com o seguinte comando, :

Algoritmo 3 : Código de Criação de Túnel com o Ngrok

1. `ngrok http 5678`

Antes de começar a usar o Ngrok, é necessário obter o token de autenticação. Para obter o token de autenticação do serviço, o primeiro passo é criar uma conta na plataforma. Deve-se acessar o site oficial do Ngrok e clicar na opção Sign Up para registrar-se, podendo utilizar um e-mail ou uma conta do Google/GitHub para facilitar o login. Após acessar o painel, o usuário deve adentrar a seção Setup & Installation, onde será encontrado o authtoken pessoal. Esse token é essencial para que o Ngrok possa autenticar a conexão e permitir a criação de túneis seguros para expor serviços locais.

Depois de copiar o authtoken, deve-se realizar o comando do Algoritmo 4, esse comando salvará a autenticação no sistema, tornando o uso do Ngrok mais seguro.

Algoritmo 4 : Código de Autenticação do Ngrok

1. `ngrok config add-authtoken AUTH_TOKEN`

3.0 - Execução

Após a instalação e configuração do n8n, a execução do Tutor.IA deve ser feita ativando o workflow correspondente dentro da plataforma. Para isso, o primeiro passo é acessar o n8n. Se o sistema estiver rodando localmente, basta abrir o navegador e acessar <http://localhost:5678>. Caso o n8n esteja sendo executado em um servidor remoto, será necessário utilizar o endereço IP ou domínio configurado. Se a autenticação estiver habilitada, o login com usuário e senha será necessário antes de prosseguir.

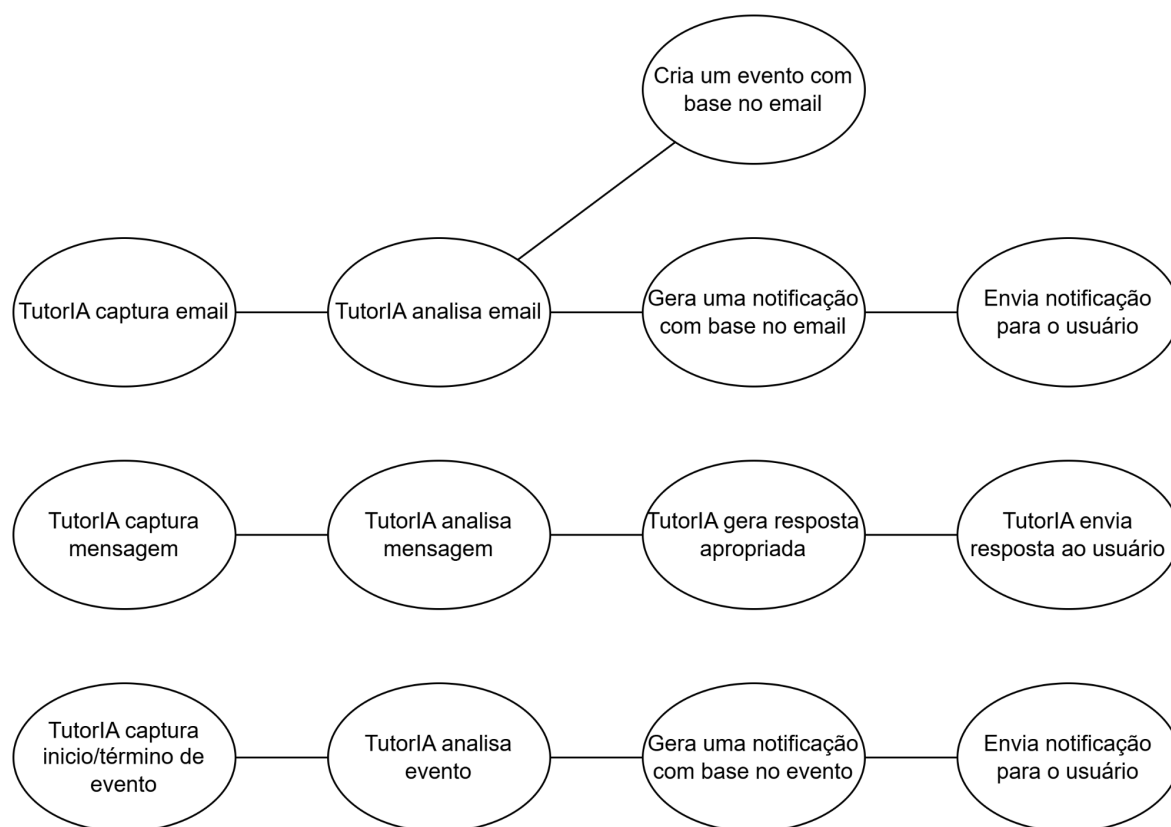
Dentro do painel do n8n, o próximo passo é localizar o workflow do Tutor.IA. Para isso, o usuário deve acessar o menu principal e navegar até a seção de Workflows. Uma vez na tela de workflows, deve-se procurar pelo fluxo correspondente ao Tutor.IA e abri-lo para visualizar os detalhes.

Após abrir o workflow, a ativação pode ser feita clicando no botão cinza que está no canto superior direito, que está escrito “Inativo”. Após clicar, o botão deve ficar da cor verde, e estar escrito Ativo. Esse processo coloca o fluxo em execução contínua, garantindo que o Tutor.IA esteja sempre pronto para processar mensagens, gerar lembretes e responder aos usuários automaticamente. Para testar se a ativação foi bem-sucedida, pode-se enviar uma mensagem via WhatsApp ou Telegram, e o sistema deve responder conforme configurado. Caso os lembretes estejam sendo gerados corretamente e os eventos adicionados ao Google Calendar, significa que o Tutor.IA está funcionando conforme esperado e está pronto para auxiliar os alunos na organização acadêmica.

4 - Nós e Funcionalidades

No contexto do n8n, a construção de fluxos automatizados depende de elementos fundamentais que possibilitam a integração entre sistemas e a execução de tarefas de forma inteligente. Neste tópico, exploraremos a estrutura e as funcionalidades que tornam essa automação possível, destacando sua importância e impacto na otimização de processos. A figura 1 mostra 3 fluxogramas que representam as 3 funcionalidades principais do Tutor.Ia, onde o fluxograma de cima simboliza o envio de notificações de emails importantes, o do meio descreve a interação com o chatbot, e o de baixo mostra o envio de notificações de eventos iniciados ou concluídos.

Figura 1 - Fluxogramas das funções do TutorIA



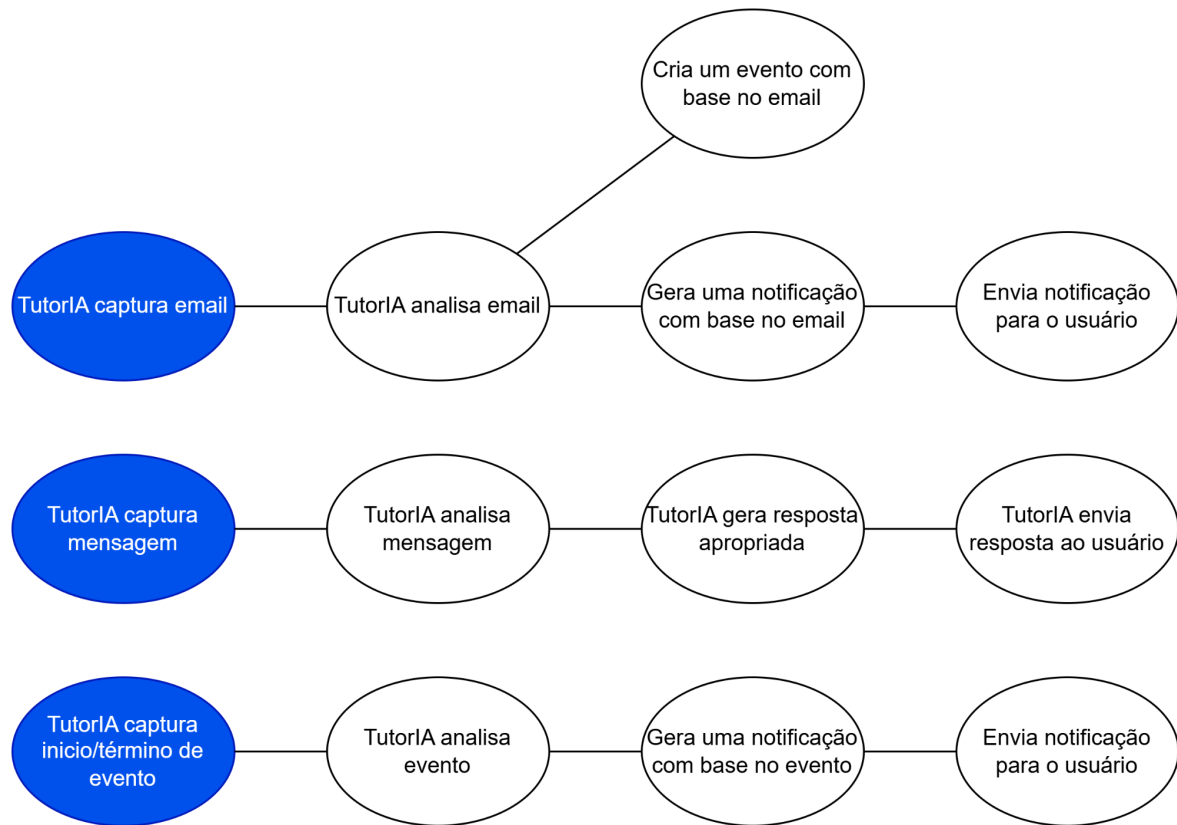
Autoria própria (2025)

4.1 - Webhooks

Os webhooks são um dos componentes fundamentais para a comunicação entre diferentes sistemas na web. No Tutor.IA, eles são utilizados para garantir que eventos importantes, como mensagens enviadas por alunos ou atualizações no calendário acadêmico,

sejam processados de forma imediata e automatizada. A Figura 2 mostra em destaque em quais fluxos esses nós são utilizados.

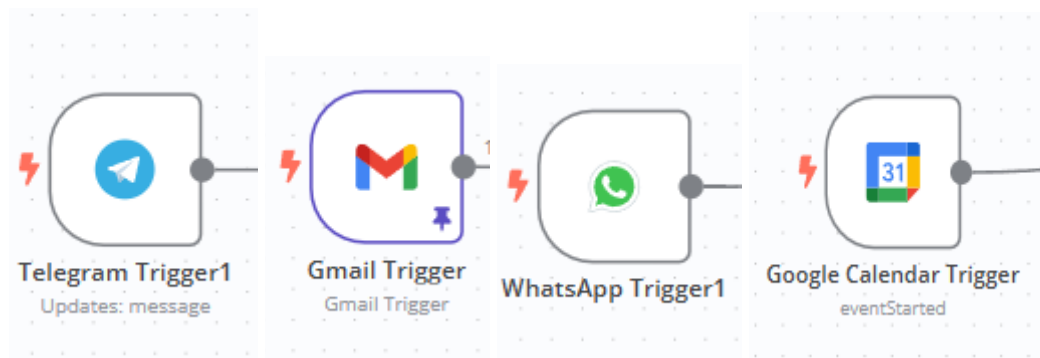
Figura 2 - Fluxos onde Webhooks são usados



Autoria Própria (2025)

Um webhook é um mecanismo que permite que sistemas enviem notificações automaticamente para outros sistemas sempre que um evento específico ocorre. Diferente de APIs tradicionais, onde um sistema precisa constantemente requisitar informações de outro, os webhooks funcionam de forma reativa, ou seja, eles são ativados automaticamente quando uma ação acontece. Os webhooks usados no Sistema podem ser vistos na Figura 3 abaixo:

Figura 3 - Webhooks do Tutor.IA



Autoria Própria (2025)

No Tutor.IA, os webhooks são utilizados para capturar e processar eventos em três áreas principais. O primeiro tipo de webhook é o de mensagens enviadas via WhatsApp e Telegram. Sempre que um aluno envia uma mensagem, um webhook aciona automaticamente o n8n, que processa a solicitação, classifica a mensagem e gera uma resposta através da IA. Dessa forma, o aluno recebe um atendimento ágil e personalizado sem a necessidade de interação humana.

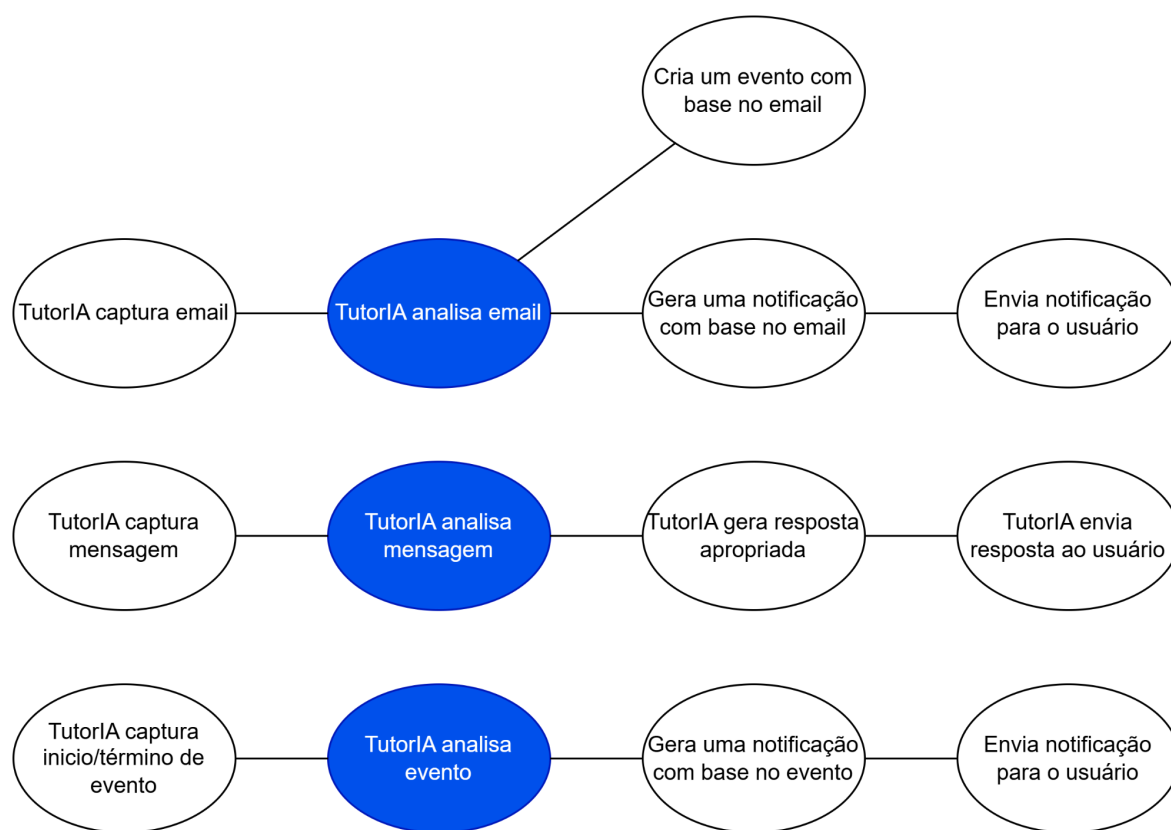
O segundo tipo de webhook está integrado ao Google Calendar. Toda vez que um evento acadêmico é criado, modificado ou está prestes a ocorrer, um webhook do Google Calendar dispara um alerta para o Tutor.IA, garantindo que os alunos sejam notificados sobre prazos e compromissos importantes. Se um evento estiver próximo de sua data limite, o sistema gera automaticamente um lembrete e o envia via WhatsApp ou Telegram, assegurando que o aluno não esqueça suas atividades.

Já o terceiro tipo de webhook, está relacionado ao monitoramento de e-mails acadêmicos. O sistema fica atento a e-mails recebidos por professores e instituições de ensino. Quando um novo e-mail contendo detalhes sobre uma atividade acadêmica chega, um webhook do Gmail captura a mensagem, a envia para análise da IA, que extrai informações importantes como título, prazo de entrega e descrição da atividade. Essas informações são então convertidas em um lembrete que pode ser enviado ao aluno por WhatsApp, Telegram ou adicionado ao Google Calendar.

4.3 - Classificadores de Texto

Os classificadores de texto desempenham um papel essencial no funcionamento do Tutor.IA, permitindo que o sistema analise, compreenda e organize automaticamente as mensagens recebidas dos alunos e os e-mails acadêmicos. Eles utilizam inteligência artificial e processamento de linguagem natural (NLP) para interpretar a intenção das mensagens e categorizá-las corretamente. Dessa forma, o Tutor.IA pode fornecer respostas precisas e relevantes, tornando a interação com o aluno mais fluida e eficiente. A Figura 4 evidencia em que parte das funcionalidades essas ferramentas atuam.

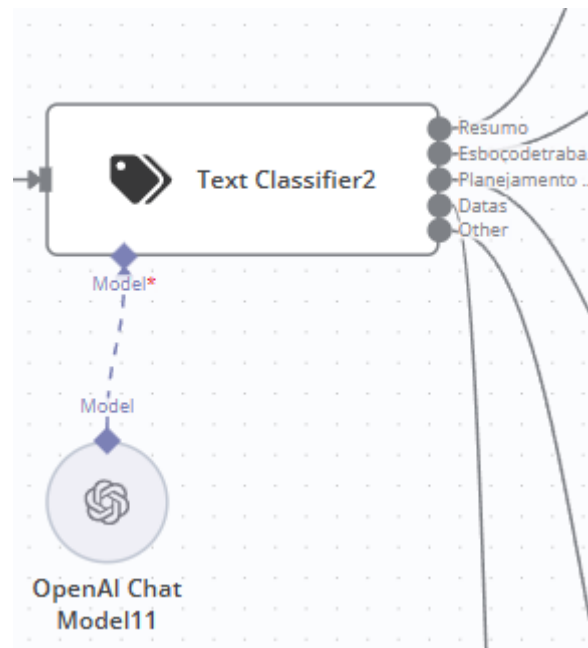
Figura 4 - Fluxos onde classificadores de texto são usados



Autoria própria (2025)

Quando um aluno envia uma mensagem pelo WhatsApp ou Telegram, o classificador de texto analisa seu conteúdo, identificando palavras-chave e determinando a intenção do usuário. Com base nessa análise, o classificador direciona a solicitação para a funcionalidade correta dentro do Tutor.IA. Algumas das categorias reconhecidas incluem consulta de prazos, solicitação de resumos, esboço de trabalhos e planejamento de estudos. Além disso, graças à memória contextual, o classificador consegue manter um histórico das interações recentes, permitindo que o bot continue uma conversa sem que o aluno precise repetir informações já fornecidas. Um classificador pode ser visto na Figura 5 abaixo:

Figura 5 - Classificador de Texto



Autoria Própria (2025)

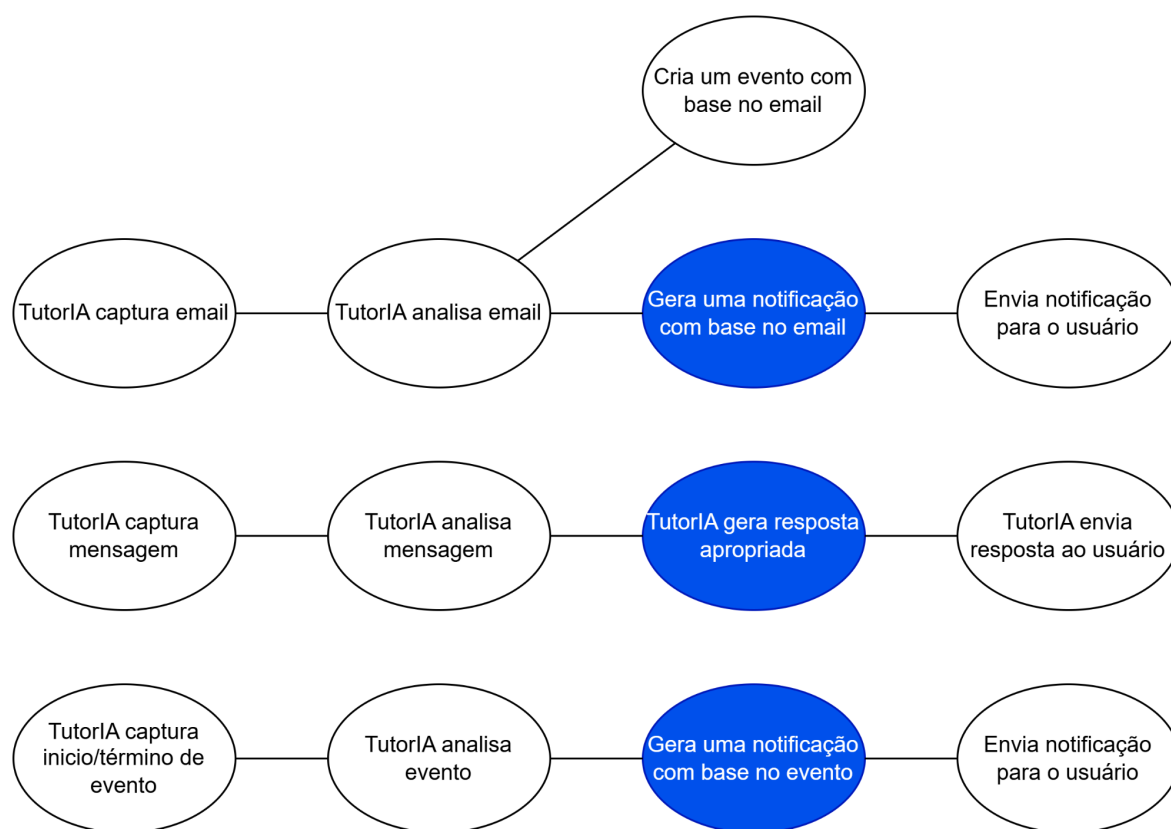
Outro uso fundamental dos classificadores de texto no Tutor.IA está na análise automática de e-mails acadêmicos. O sistema monitora a caixa de entrada do Gmail do aluno e, sempre que um novo e-mail de um professor ou instituição acadêmica é recebido, o classificador analisa seu conteúdo para identificar se a mensagem contém informações relevantes sobre atividades, provas ou comunicados importantes. Caso o e-mail seja classificado como um aviso acadêmico, a inteligência artificial extrai os dados essenciais, como título da atividade, data de entrega e descrição, e envia uma notificação automatizada ao aluno via WhatsApp, Telegram ou Google Calendar.

O uso de classificadores de texto no Tutor.IA traz diversos benefícios. Primeiramente, ele permite que o sistema forneça respostas rápidas e precisas, pois entende a intenção do usuário e retorna informações adequadas de maneira automática. Além disso, a automação reduz a necessidade de intervenção humana, tornando o Tutor.IA altamente eficiente na resolução de dúvidas acadêmicas sem precisar de suporte manual. Outro ponto importante é a organização estruturada das informações, pois o sistema processa e classifica mensagens e e-mails, garantindo que os alunos nunca percam um prazo importante. Por fim, a memória contextual aprimora a interação com o chatbot, tornando as conversas mais naturais e personalizadas.

4.2 - Bots

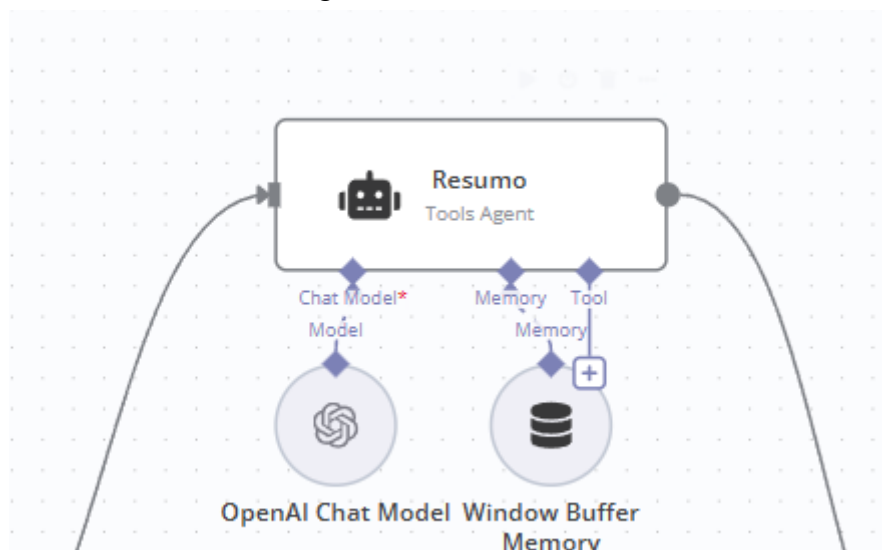
Os bots do Tutor.IA são assistentes virtuais que interagem diretamente com os alunos via WhatsApp e Telegram, ajudando na organização acadêmica ao responder dúvidas, enviar lembretes e fornecer resumos e planejamentos de estudo de forma automatizada. A Figura 6 evidencia onde os bots agem, enquanto a Figura 7 apresenta o Bot responsável pelo resumo dentro do Tutor.IA.

Figura 6 - Fluxos onde bots são usados



Autoria própria (2025)

Figura 7 - Bot de resumo



Autoria Própria (2025)

Focados exclusivamente em temas acadêmicos, os bots do Tutor.IA garantem que todas as interações sejam direcionadas ao apoio educacional, evitando conversas fora desse contexto. Sempre que um aluno envia uma mensagem, os bots processam a solicitação utilizando inteligência artificial integrada ao n8n, classificando a pergunta em categorias como consulta de prazos, solicitação de resumos, esboços de trabalhos e planejamento de estudos. Além de responder interações diretas, os bots também são responsáveis por notificar os alunos sobre atividades e eventos acadêmicos, garantindo que prazos importantes não sejam esquecidos.

Uma das funcionalidades mais importantes do Tutor.IA é a capacidade de processar e-mails acadêmicos e transformar essas informações em notificações automatizadas. Sempre que um professor envia um e-mail contendo detalhes sobre uma atividade, um bot monitora a caixa de entrada do Gmail e captura a mensagem. O conteúdo do e-mail é então analisado por um modelo de inteligência artificial, que extrai informações como título da atividade, data de entrega e descrição. Com esses dados, o bot gera um resumo da atividade e adiciona um lembrete no Google Calendar. Além disso, o aluno recebe uma notificação imediata via WhatsApp ou Telegram, garantindo que esteja sempre atualizado sobre seus compromissos acadêmicos.

Ademais, eles também auxiliam na geração automática de resumos e planejamento de estudos, permitindo que os alunos solicitem informações detalhadas sobre conteúdos acadêmicos. Caso um aluno precise de um resumo de determinado assunto, basta enviar um pedido ao bot, que utilizará a IA para gerar uma versão simplificada e bem estruturada. Da mesma forma, se o aluno precisar de um cronograma de estudos, o bot pode sugerir uma organização eficiente do tempo, distribuindo as tarefas de forma equilibrada.

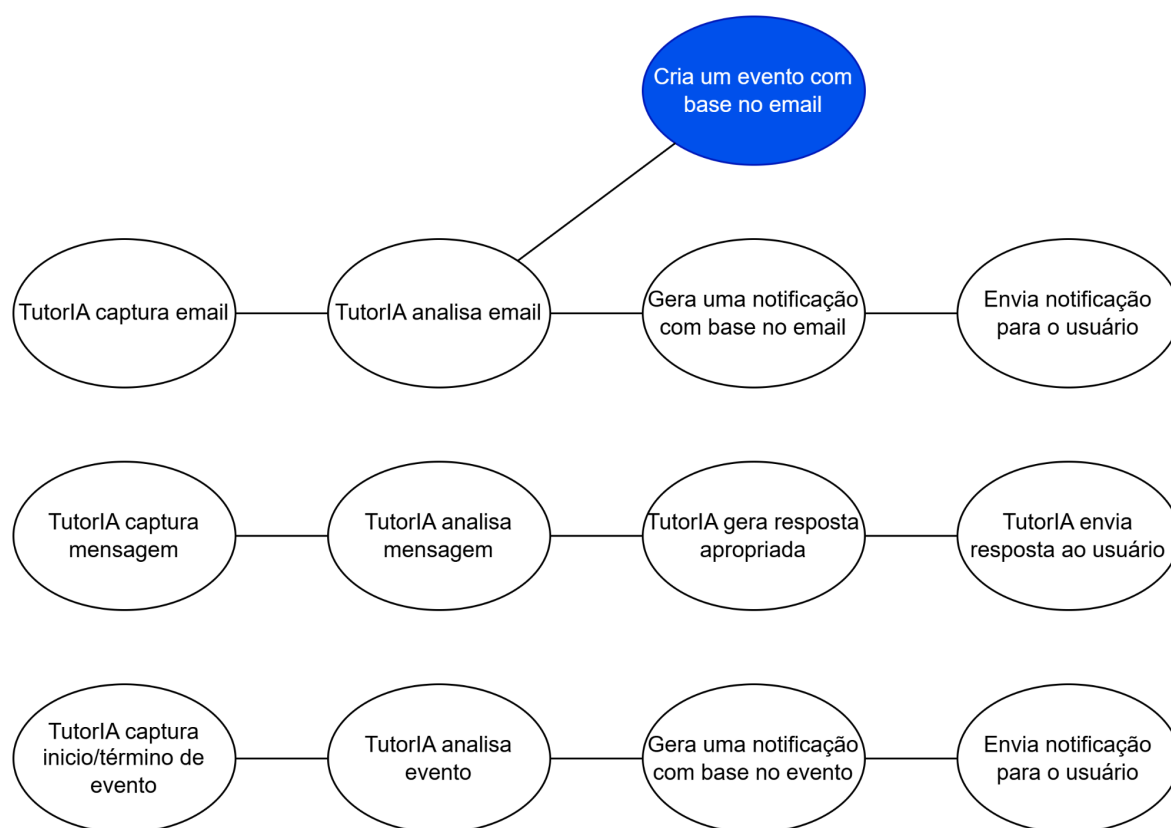
Além da interação direta com os alunos, os bots do Tutor.IA também desempenham um papel fundamental na notificação de atividades e eventos acadêmicos. Com a integração ao Google Calendar, os bots são responsáveis por lembrar os alunos sobre prazos de entrega de trabalhos e datas de provas. Assim, quando uma atividade cadastrada está próxima da data de vencimento, o bot automaticamente envia uma mensagem ao aluno via WhatsApp ou Telegram, garantindo que ele não esqueça de suas obrigações acadêmicas.

Outro aspecto dos bots do Tutor.IA é a capacidade de manter um histórico de conversas e interações anteriores. Utilizando memória de contexto, os bots conseguem lembrar informações de interações passadas para oferecer um diálogo mais fluido e eficiente. Isso significa que, se um aluno perguntar sobre uma atividade hoje e retomar a conversa amanhã, o bot ainda terá acesso ao que foi discutido anteriormente, sem a necessidade de repetir todo o contexto. Esse recurso torna a comunicação mais natural e evita que o aluno tenha que reformular suas perguntas constantemente.

4.4 - Nós de Serviço do Google

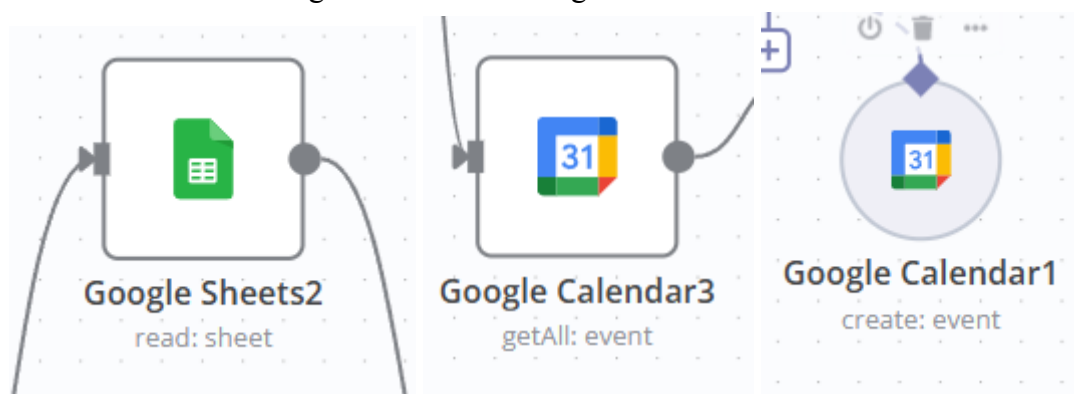
O Tutor.IA utiliza diversos serviços do Google para organizar e gerenciar as atividades acadêmicas dos alunos de maneira eficiente. Entre os principais nós de serviço do Google, destacam-se os relacionados ao Google Sheets e ao Google Calendar, que permitem armazenar e processar informações sobre os alunos e seus compromissos acadêmicos. Na figura 8 é possível ver em destaque o caso onde esses nós são utilizados. Eles garantem que o sistema seja capaz de capturar, registrar e notificar os usuários sobre eventos importantes sem a necessidade de ações manuais (Figura 9).

Figura 8 - Fluxos onde nós de serviço do Google são usados



Autoria própria (2025)

Figura 9 - Nós do Google Sheet e Calendar



Autoria Própria (2025)

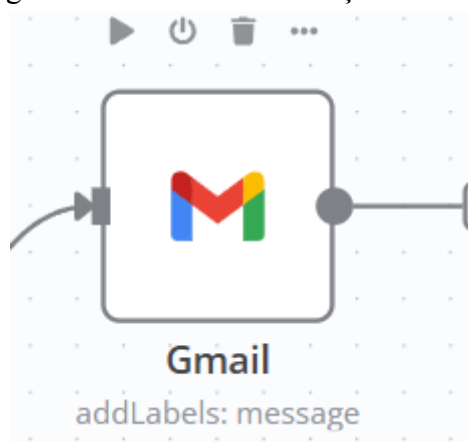
Primeiramente, o Google Sheets, representado na Figura 9, é utilizado para armazenar informações acadêmicas essenciais, como a lista de alunos cadastrados no sistema, suas atividades pendentes e o histórico de interações com o chatbot. Sempre que um novo aluno é adicionado ao Tutor.IA, seus dados são registrados em uma planilha no Google Sheets, permitindo que o sistema gere quais usuários devem receber lembretes personalizados. Além disso, essa planilha também pode ser usada para armazenar respostas do chatbot e

analisar padrões de comportamento dos alunos, possibilitando melhorias na interação e maior precisão nas notificações.

Ademais, outro serviço fundamental é o Google Calendar, também presente na Figura 9, que permite que o Tutor.IA crie e gerencie eventos acadêmicos automaticamente. Quando uma nova atividade é identificada, seja por meio da análise de e-mails acadêmicos ou de solicitações diretas dos alunos, o sistema utiliza o nó de criação de eventos do Google Calendar para registrar a atividade na agenda do aluno. Esse evento inclui informações como título da atividade, data de vencimento, descrição detalhada e lembretes automáticos, garantindo que o aluno não perca prazos importantes. Dessa forma, o Tutor.IA reduz o trabalho manual do estudante, tornando sua rotina acadêmica mais organizada.

Outrossim, além de criar novos eventos, o Tutor.IA também precisa acessar os eventos já cadastrados no Google Calendar para fornecer informações atualizadas aos alunos. O nó de obtenção de eventos existentes permite que o sistema consulte a agenda do aluno e forneça respostas personalizadas a perguntas como: "Quais são minhas atividades desta semana?". Sempre que uma solicitação desse tipo é feita pelo aluno via WhatsApp ou Telegram, o Tutor.IA consulta automaticamente o Google Calendar e retorna uma resposta clara e estruturada.

Figura 10 - Nó de Classificação de Email



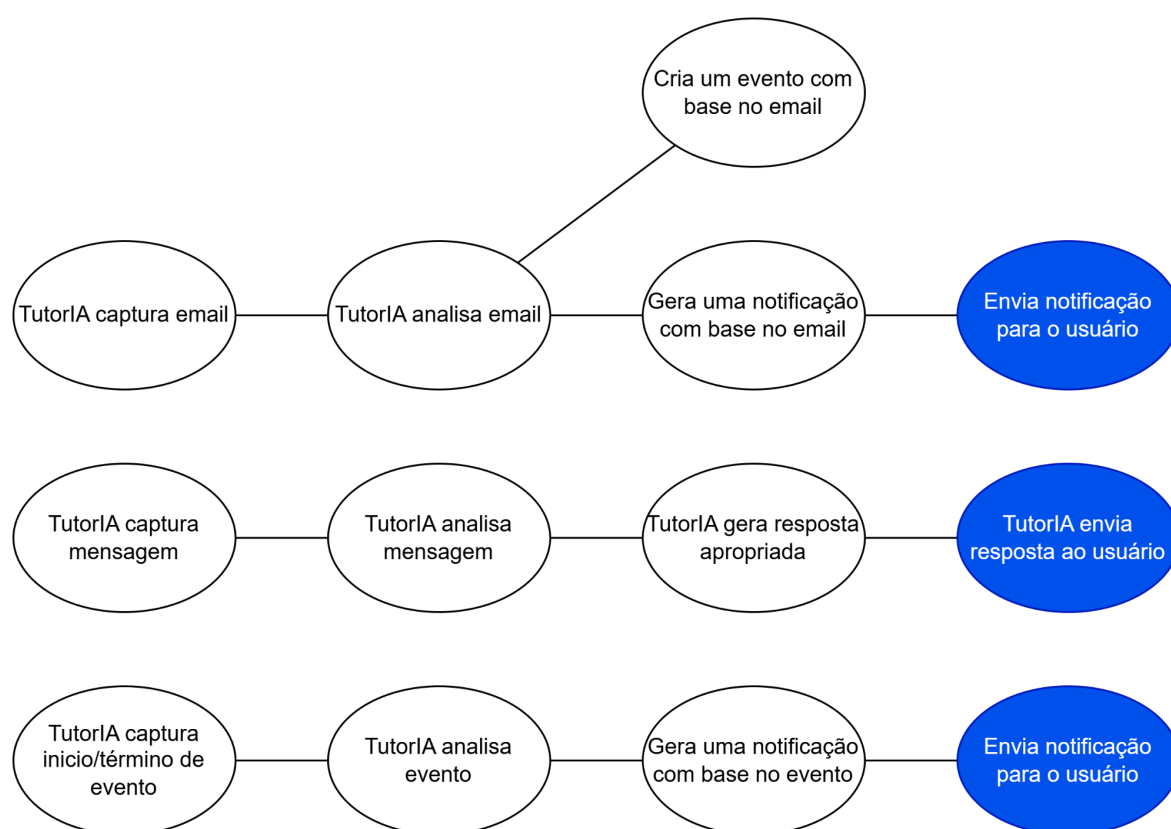
Autoria Própria (2025)

Além disso, para facilitar a organização, o Tutor.IA utiliza o Nó de Classificação de Emails para marcar os emails de atividade como importante, que pode ser visto na Figura 10. Isso auxilia na organização do aluno, garantindo que ele possa encontrá-la facilmente quando precisar.

4.5 - Nós de Envio de Mensagem

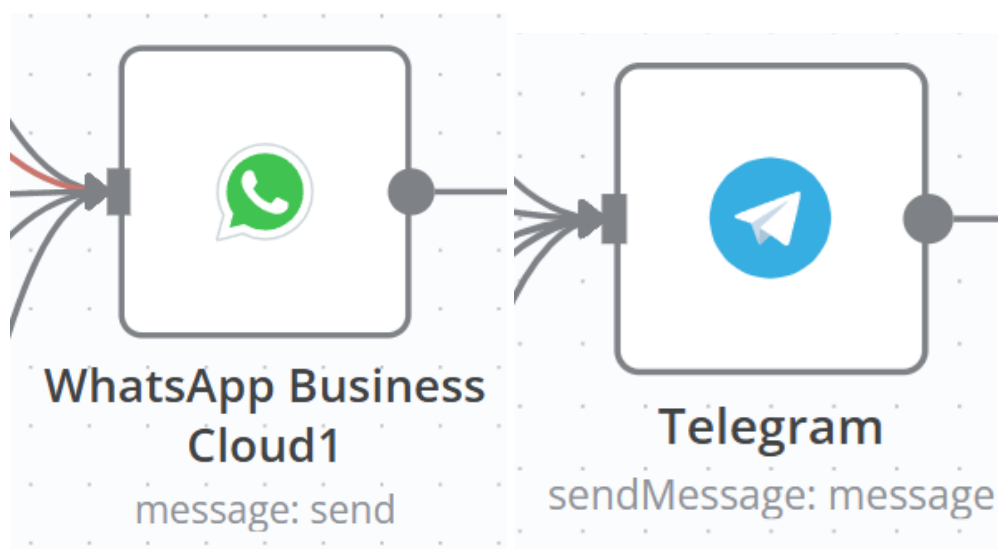
Os nós de envio de mensagens do Tutor.IA, são fundamentais para garantir a comunicação eficiente entre o sistema e os alunos, utilizando as APIs do WhatsApp Business Cloud e do Telegram Bot API para entregar respostas personalizadas, lembretes e notificações sobre eventos acadêmicos. Os fluxogramas da Figura 11 apontam quando esses nós são ativados.

Figura 11 - Fluxos onde nós de envio de mensagem são usados



Autoria própria (2025)

Figura 12 - Nós de Envio de Mensagem



Autoria Própria (2025)

Esses nós, que podem ser vistos na Figura 12 acima, trabalham diretamente com os nós de recepção de mensagens e os classificadores de texto, permitindo que os alunos recebam informações precisas no momento certo, sem necessidade de consultas manuais. A principal função desses nós é garantir que as mensagens sejam enviadas corretamente, utilizando IDs exclusivos que identificam cada usuário, seja no WhatsApp ou no Telegram.

De início, pode-se começar pelo WhatsApp. No WhatsApp, o nó de envio utiliza a API do WhatsApp Business Cloud, onde cada aluno possui um ID único vinculado ao seu número de telefone. Esse identificador é capturado pelo sistema sempre que o aluno interage com o chatbot. Isso garante que a mensagem seja entregue corretamente ao destinatário.

Outrossim, no caso do Telegram, o Tutor.IA utiliza a Telegram Bot API para enviar mensagens automatizadas com um funcionamento semelhante ao do WhatsApp. Cada usuário do Telegram possui um ID único, capturado quando o aluno interage pela primeira vez com o chatbot. O nó de envio do Telegram armazena esse ID e o utiliza sempre que precisa encaminhar uma resposta ou um lembrete.

Ambos os nós encaminham para o usuário as mensagens geradas pelos Bots, que foram descritos anteriormente neste documento. São eles que permitem que o usuário receba a resposta correta.

Além das respostas interativas, esses nós também são responsáveis por enviar notificações automáticas quando eventos importantes se aproximam. Por exemplo, se um prazo de atividade está se aproximando, o Tutor.IA acessa o Google Calendar, identifica a atividade e dispara um alerta automático para o aluno pelo WhatsApp ou Telegram, utilizando o ID previamente armazenado. Dessa forma, mesmo que o aluno não tenha

interagido recentemente com o chatbot, ele ainda receberá lembretes importantes sem precisar solicitar manualmente.

5 - Conclusão

O Tutor.IA foi desenvolvido para ser um assistente acadêmico automatizado, garantindo que os alunos se mantenham organizados e informados sobre suas atividades acadêmicas. Com a integração de tecnologias como Google Sheets, Google Calendar, Gmail, WhatsApp e Telegram, o sistema é capaz de capturar informações acadêmicas, processá-las e entregar notificações e respostas personalizadas automaticamente. Para garantir uma experiência fluida e eficiente, a instalação correta do Tutor.IA é essencial, permitindo que todas as funcionalidades operem de maneira integrada.

Ao longo deste manual, foram abordadas todas as etapas necessárias para a instalação e configuração do Tutor.IA, desde a configuração do ambiente utilizando Docker ou Node.js, passando pela importação do fluxo no n8n, configuração das credenciais das APIs externas e ativação dos nós responsáveis pelo recebimento e envio de mensagens, classificadores de texto e webhooks. Além disso, foi explicado como conectar o Tutor.IA aos serviços do Google e integrar o chatbot com WhatsApp e Telegram para garantir notificações instantâneas e suporte acadêmico aos alunos.

Com a correta configuração e instalação, o Tutor.IA estará pronto para operar de forma contínua e eficiente, automatizando lembretes acadêmicos, organização de prazos, resumos de conteúdos e planejamentos de estudo, tudo acessível diretamente via WhatsApp e Telegram. A automação elimina a necessidade de monitoramento manual, otimizando o tempo do usuário e garantindo uma experiência acadêmica organizada e produtiva.

Reconhecimentos e Direitos Autorais

@autor: Emanuel Lopes Silva, Letícia Delfino De Araujo, Thales Aymar Fortes De Souza , Gabriel Felipe Carvalho Silva, Verônica Lima Costa.

@contato: emmanuel.silva@discente.ufma.br

@data última versão: 10/02/2025

@versão: 1.0

@outros repositórios:

<https://github.com/EmanuelSilva69/Tutor-IA>, <https://github.com/lwticias-lelet>,
<https://github.com/ThalesAymar>, <https://github.com/gfcarvalhos>,
<https://github.com/Veronicacosta-ui><https://github.com/Veronicacosta-ui>

@Agradecimentos: Universidade Federal do Maranhão (UFMA), Professor Doutor Thales Levi Azevedo Valente, e colegas de curso.

Copyright/License

Este material é resultado de um trabalho acadêmico para a disciplina "Projeto e Desenvolvimento de Software", sob a orientação do professor Dr. THALES LEVI AZEVEDO VALENTE, semestre letivo 2024.2, curso Engenharia da Computação, na Universidade Federal do Maranhão (UFMA). Todo o material sob esta licença é software livre: pode ser usado para fins acadêmicos e comerciais sem nenhum custo. Não há papelada, nem royalties, nem restrições de "copyleft" do tipo GNU. Ele é licenciado sob os termos da Licença MIT, conforme descrito abaixo, e, portanto, é compatível com a GPL e também se qualifica como software de código aberto. É de domínio público. Os detalhes legais estão abaixo. O espírito desta licença é que você é livre para usar este material para qualquer finalidade, sem nenhum custo. O único requisito é que, se você usá-los, nos dê crédito.

Licenciado sob a Licença MIT. Permissão é concedida, gratuitamente, a qualquer pessoa que obtenha uma cópia deste software e dos arquivos de documentação associados (o "Software"), para lidar no Software sem restrição, incluindo sem limitação os direitos de usar, copiar, modificar, mesclar, publicar, distribuir, sublicenciar e/ou vender cópias do Software, e permitir pessoas a quem o Software é fornecido a fazê-lo, sujeito às seguintes condições:

Este aviso de direitos autorais e este aviso de permissão devem ser incluídos em todas as cópias ou partes substanciais do Software.

O SOFTWARE É FORNECIDO "COMO ESTÁ", SEM GARANTIA DE QUALQUER TIPO, EXPRESSA OU IMPLÍCITA, INCLUINDO MAS NÃO SE LIMITANDO ÀS GARANTIAS DE COMERCIALIZAÇÃO, ADEQUAÇÃO A UM DETERMINADO FIM E NÃO INFRINGÊNCIA. EM NENHUM CASO OS AUTORES OU DETENTORES DE DIREITOS AUTORAIS SERÃO RESPONSÁVEIS POR QUALQUER RECLAMAÇÃO, DANOS OU OUTRA RESPONSABILIDADE, SEJA EM AÇÃO DE CONTRATO, TORT OU OUTRA FORMA, DECORRENTE DE, FORA DE OU EM CONEXÃO COM O SOFTWARE OU O USO OU OUTRAS NEGOCIAÇÕES NO SOFTWARE.

Para mais informações sobre a Licença MIT: <https://opensource.org/licenses/MIT>