



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

PROJETO E DESENVOLVIMENTO DE SOFTWARE
TURMA 1

TUTOR.IA - ASSISTENTE VIRTUAL

PROFESSOR: THALES LEVI AZEVEDO VALENTE

ALUNOS:

EMANUEL LOPES SILVA - 2021017818

LETÍCIA DELFINO DE ARAUJO-2021061763

THALES AYMAR FORTES DE SOUZA - 2021018145

GABRIEL FELIPE CARVALHO SILVA - 2023098664

VERÔNICA LIMA COSTA - 2022029493

São Luís – MA

22/01/2025

SUMÁRIO

1.0 - INTRODUÇÃO.....	3
1.1 - Visão Geral do Documento.....	3
1.2 - Escopo do Projeto.....	3
1.3 - Objetivo do Projeto.....	4
2.0 - DESCRIÇÃO GERAL DO SISTEMA.....	5
2.1 - Abrangência do Sistema.....	5
2.2 - Sistemas Similares.....	5
2.3 - Projeto de Desenvolvimento.....	6
3.0 - REQUISITOS.....	6
3.1 - Requisitos Funcionais.....	6
3.2 - Requisitos Não-Funcionais.....	8
3.3 - Prioridade dos Requisitos.....	9
4.0 - DIAGRAMAS DO SISTEMA.....	11
4.1 - Diagrama de Caso de Uso.....	11
4.1.1 - Diagrama de Recepção e Resposta no Whatsapp.....	11
4.1.2 - Diagrama de Recepção e Resposta no Telegram.....	12
4.1.3 - Diagrama de Notificação de e-mail.....	13
4.1.4 - Diagrama de Criação de Evento no Calendar.....	14
4.1.5 - Diagrama de Lembrete e Encerramento de Evento no Calendar.....	15
4.2 - Diagrama de Sequência.....	16
4.2.1 - Diagrama de Sequência de Mensagem Recebida no WhatsApp.....	16
4.2.2 - Diagrama de Sequência de Mensagem Recebida no Telegram.....	17
4.2.3 - Diagrama de Sequência de Classificação e Resumo de E-mails.....	18
4.2.4 - Diagrama de Sequência de Criação de Eventos.....	19
4.2.5 - Diagrama de Sequência de Encerramento de Eventos.....	20
4.3 - Diagrama de Atividade.....	21
4.3.1 - Diagrama de Atividade de Mensagem do Whatsapp.....	21
4.3.2 - Diagrama de Atividade de Mensagem do Telegram.....	22
4.3.3 - Diagrama de Atividade de Classificação e Resumo de E-mails.....	24
4.3.4 - Diagrama de Atividade de Notificação de Conclusão de Atividade.....	25
4.3.5 - Diagrama de Atividade de Gerenciamento de Eventos no Calendar.....	27
4.4 - Diagrama de Estado.....	28
4.4.1 - Diagrama de Estado de Evento.....	28
4.4.2 - Diagrama de Estado de Mensagem.....	30
5.0 - TECNOLOGIAS UTILIZADAS.....	31
5.1 - N8N.....	31
5.2 - WhatsApp Business API.....	32
5.3 - Telegram API.....	33
5.4 - Google Calendar API.....	33
5.5 - Gmail.....	33
6.0 - REFERÊNCIAS.....	34

1.0 - INTRODUÇÃO

1.1 - Visão Geral do Documento

Este documento apresenta uma visão abrangente do TutorIA, um bot desenvolvido para automação e gerenciamento de informações voltadas para alunos. Ele utiliza a plataforma n8n para integrar diferentes serviços, como WhatsApp, Telegram, Gmail e Google Calendar, com o objetivo de facilitar a comunicação e a organização de tarefas acadêmicas. Este documento descreve as funcionalidades do sistema, seu escopo, e os objetivos principais que orientam o projeto, fornecendo uma base sólida para entender como o TutorIA contribui para a eficiência e produtividade dos alunos.

1.2 - Escopo do Projeto

- **Problema**

No ambiente acadêmico, é comum que alunos enfrentem dificuldades em acompanhar suas atividades e prazos, resultando em um desempenho prejudicado. Esses problemas geralmente decorrem da falta de organização, dificuldade em gerenciar múltiplas tarefas e a ausência de sistemas integrados para notificações e lembretes.

Tabela 1 - Problemas identificados no contexto acadêmico.

Problema Identificado	Descrição do Problema
Esquecimento de prazos importantes	Muitos alunos não conseguem acompanhar os prazos de entrega de atividades ou eventos acadêmicos importantes devido à sobrecarga de tarefas e falta de organização
Falta de centralização das informações	Informações relevantes, como datas de provas, trabalhos e compromissos, estão dispersas em diferentes canais, dificultando o acesso rápido e eficiente.
Dificuldade em organizar múltiplas tarefas	A necessidade de gerenciar simultaneamente várias disciplinas e atividades pode gerar confusão e perda de prazos.

Fonte: Autoria própria (2025).

- **Justificativa**

Diante desses desafios, o desenvolvimento do TutorIA apresenta-se como uma solução eficiente para facilitar a vida acadêmica dos alunos. Esse sistema automatizado e integrado auxilia os estudantes a gerenciar melhor suas atividades, oferecendo lembretes,

notificações e suporte organizacional. Ao centralizar as informações acadêmicas e promover notificações automáticas, o TutorIA reduz os riscos de esquecimento e melhora o desempenho acadêmico.

Tabela 2 - Justificativas por problemas identificados.

Justificativa do Problema	Descrição da Justificativa
Melhoria no acompanhamento de prazos	O TutorIA pode enviar lembretes automáticos e notificações personalizadas, ajudando os alunos a não esquecerem prazos importantes.
Centralização das informações acadêmicas	Com o TutorIA, todas as informações relevantes, como prazos e compromissos, podem ser acessadas em um único local, proporcionando maior organização.
Auxílio na organização das tarefas	O TutorIA ajuda os alunos a priorizarem e gerenciarem suas tarefas de maneira eficiente, reduzindo a sobrecarga e aumentando a produtividade.

Fonte: Autoria própria (2025).

1.3 - Objetivo do Projeto

O objetivo geral do TutorIA é ser uma solução digital inovadora e acessível que auxilie os alunos no gerenciamento de suas atividades acadêmicas, promovendo maior organização, eficiência e produtividade. Ele busca resolver os principais desafios enfrentados pelos estudantes, como o esquecimento de prazos, a dificuldade em acompanhar várias tarefas simultâneas e a falta de integração de informações

Entre os objetivos específicos estão:

- **Auxiliar na organização dos alunos:** Fornecer lembretes e notificações automáticas sobre prazos, tarefas e eventos importantes.
- **Simplificar a comunicação:** Permitir interações rápidas e diretas via WhatsApp e Telegram, com respostas personalizadas.
- **Reduzir a carga manual:** Automatizar tarefas repetitivas, como leitura de e-mails, classificação de mensagens e atualização de calendários.
- **Oferecer suporte inteligente:** Utilizar inteligência artificial para processar dados, entender o contexto das mensagens e fornecer respostas relevantes.
- **Promover eficiência e produtividade:** Agilizar a rotina acadêmica dos alunos, permitindo que eles se concentrem nas suas atividades principais.

O TutorIA busca ser um assistente confiável e indispensável para estudantes, auxiliando-os a manterem-se organizados e preparados para suas atividades acadêmicas.

2.0 - DESCRIÇÃO GERAL DO SISTEMA

2.1 - Abrangência do Sistema

O TutorIA é um sistema automatizado que utiliza a plataforma n8n para auxiliar alunos em atividades acadêmicas, promovendo organização, eficiência e engajamento. Ele oferece suporte por meio de múltiplos canais de comunicação, como WhatsApp, Telegram, Gmail e Google Calendar, integrando funcionalidades baseadas em inteligência artificial para atender às necessidades específicas de estudantes.

O sistema abrange:

- **Recepção de mensagens e notificações:** Capacidade de receber e responder mensagens dos alunos via WhatsApp e Telegram.
- **Análise e classificação de e-mails:** Detecta e-mails importantes, classifica suas informações e gera resumos relevantes.
- **Gestão de atividades acadêmicas:** Cria e gerencia eventos no Google Calendar, incluindo lembretes e notificações de conclusão.
- **Personalização com IA:** Utiliza modelos avançados, como o OpenAI Chat Model, para gerar resumos, esboços de trabalhos e planejamentos de estudo personalizados.

2.2 - Sistemas Similares

Embora existam sistemas de gestão acadêmica e comunicação, o TutorIA se destaca por sua abordagem centrada no aluno e sua integração com tecnologias avançadas. Alguns sistemas semelhantes incluem:

- **Google Classroom:**
 - Oferece gerenciamento de aulas e envio de tarefas.
 - Foco limitado em lembretes personalizados e notificações integradas.
- **Slack e Discord (com bots):**
 - Permitem comunicação eficiente em grupos acadêmicos.
 - Carecem de recursos específicos para acompanhamento de prazos e organização acadêmica.
- **Calendários Integrados (Google/Outlook):**
 - Gerenciam eventos e lembretes básicos.
 - Não possuem funcionalidades inteligentes para criação automática de eventos baseados em e-mails ou mensagens.

O TutorIA combina essas funcionalidades e vai além, oferecendo integração direta entre sistemas e suporte inteligente com base em IA. Além disso, ele permite a redução significativa do esforço manual dos alunos ao automatizar tarefas, permite continuar conversas, entender histórico de mensagens e responder com base em interações anteriores e possui disponibilidade em tempo real, se destacando dentre seus concorrentes.

2.3 - Projeto de Desenvolvimento

O projeto de desenvolvimento do TutorIA seguiu um processo estruturado para assegurar que todas as necessidades acadêmicas dos alunos fossem atendidas de maneira eficaz. O objetivo principal foi criar um sistema de alta qualidade, integrado, personalizado e voltado exclusivamente para o suporte acadêmico. As etapas do desenvolvimento incluíram:

1. Levantamento de Necessidades e Requisitos
2. Diagramação
3. Desenvolvimento do Sistema
4. Testes e Validação do Sistema

3.0 - REQUISITOS

3.1 - Requisitos Funcionais

Basicamente, os requisitos funcionais descrevem as funções específicas que o software deve realizar para atender às necessidades dos usuários e alcançar os objetivos essenciais do projeto, segundo o livro UML 2- Uma abordagem prática, feito por Gilleanes Guedes, em 2018, que oferece uma ideia geral de como o projeto de software deve ser realizado. Os requisitos funcionais deste projeto podem ser vistos na Tabela 3 a seguir.

Tabela 3 - Requisitos Funcionais

Identificação	Requisitos	Descrição do Requisito
RF 1	Processamento de Mensagens em Tempo Real	O sistema deve capturar mensagens recebidas via WhatsApp e Telegram usando gatilhos específicos, e logo após isso serem processadas para determinar seu conteúdo e direcionadas a respostas automáticas ou fluxos subsequentes.
RF 2	Envio de Mensagens Automatizadas	Permitir o envio de respostas personalizadas via WhatsApp e Telegram, dependendo do contexto presente.
RF 3	Classificação de e-mails	Analisar e classificar textos recebidos, como a verificação de “Atividades” usando um nó de classificação, garantindo que as mensagens sejam tratadas

		conforme o contexto.
RF 4	Uso de IA para Respostas Contextuais	Utilizar modelos de IA via OpenAI para gerar respostas baseadas em texto, processar solicitações dos usuários e criar resumos detalhados.
RF 5	Execução Condicional de Fluxos	Implementar verificações para determinar se uma mensagem ou dado recebido atende a condições específicas antes de continuar o processamento
RF 6	Notificações em Múltiplos Canais	Enviar notificações simultaneamente para diferentes plataformas, como Telegram e WhatsApp, com base em eventos ou gatilhos.
RF 7	Geração de Resumos	Gerar resumos automáticos de e-mails recebidos, destacando prazos e tarefas importantes, e enviá-los para os usuários via WhatsApp e Telegram
RF 8	Geração de Lembretes	Configurar lembretes automáticos baseados nos eventos criados no Google Calendar, com envio programado para 1 dia antes do prazo final.
RF 9	Armazenar Memória de Contexto	Armazenar e reutilizar informações anteriores de interações para manter o contexto em diálogos contínuos.

Fonte: Autoria própria (2025)

3.2 - Requisitos Não-Funcionais

Os requisitos não-funcionais do sistema basicamente são especificações que definem os critérios de qualidade e restrições do sistema que não estão diretamente relacionados às funcionalidades específicas que o sistema deve executar. Tais requisitos indicam como o sistema deve se comportar, operar e interagir dentro de determinados

contextos (Guedes, 2018). Dito isso, os requisitos não funcionais do sistema podem ser vistos abaixo, na Tabela 4.

Tabela 4 - Requisitos Não-Funcionais

Identificação	Requisitos	Descrição do Requisito
RF 1	Integração com o Gmail	Deve-se monitorar a caixa de entrada em intervalos regulares com baixa latência para capturar e-mails rapidamente, configurado para verificar a cada minuto.
RF 2	Integração com o Whatsapp	Possibilita o envio e recebimento de mensagens automatizadas, respeitando as políticas da API, usando modelos para notificações personalizadas e mantendo um histórico de conversas para maior contextualização
RF 3	Integração com o Telegram	Permite a interação bidirecional entre usuários e o sistema por meio de bots, permitindo notificações em tempo real, respostas automatizadas e suporte a grupos
RF 4	Integração com o Calendar	Organiza tarefas e prazos criando eventos diretamente no calendário, monitorando suas atualizações e sincronizando lembretes, garantindo que os usuários estejam sempre informados sobre suas atividades e compromissos.
RF 5	Desempenho	O sistema deve processar mensagens e eventos com baixa latência para garantir respostas em tempo real.
RF 6	Segurança	Proteger os dados de mensagens, eventos e credenciais das APIs integradas usando criptografia e práticas

		seguras de autenticação
RF 7	Escalabilidade	Suportar um grande volume de mensagens simultâneas e múltiplas integrações sem degradação de desempenho.
RF 8	Manutenibilidade	As automações devem ser fáceis de ajustar ou expandir, com documentação clara dos fluxos e nós utilizados.
RF 9	Confiabilidade	Garantir que as mensagens e eventos sejam capturados e processados com precisão, mesmo em cenários de alta carga.

Fonte: Autoria própria (2025)

3.3 - Prioridade dos Requisitos

Para o desenvolvimento contemplando esses requisitos foram adotadas as seguintes denominações para estabelecer a prioridade dos requisitos: Crítico, Relevante e Opcional, que permite que a equipe apresente uma melhor gestão do projeto, priorizando a entrega das funcionalidades críticas e assegurando que o sistema atenda aos seus objetivos principais desde o início. Ademais, durante o mesmo período de tempo, essa prática permite a manutenção da flexibilidade do sistema, permitindo a incorporação de melhorias e adição de recursos adicionais em versões futuras, dependendo das necessidades dos usuários, tendo em foco modificações de práticas vigentes ou adições de novas funcionalidades, como o suporte para outros aplicativos de mensagem.

Aqui estão as denominações em mais detalhe, logo abaixo:

- **Crítico:** Requisitos que são indispensáveis para o funcionamento básico do sistema, onde sua ausência inviabiliza qualquer operação.
- **Relevante:** Requisitos que, embora não essenciais para o funcionamento básico, afetam significativamente a usabilidade, eficiência ou valor agregado do sistema.
- **Opcional:** Requisitos que agregam funcionalidades extras ou melhorias incrementais, mas que não são obrigatórios para o sistema cumprir sua finalidade inicial.

Com as definições bem detalhadas, os requisitos funcionais e não funcionais podem ser separados dentre as categorias acima da seguinte forma:

- Requisitos Funcionais
 - Crítico
 - Processamento de Mensagens em Tempo Real

- Geração de Lembretes.
 - Geração de Resumos
 - Envio de Mensagens Automatizadas
 - Uso de IA para Respostas Contextuais
- Relevante
 - Armazenar Memória de Contexto
 - Classificação de e-mails
 - Execução Condicional de Fluxos
- Opcional
 - Notificações em Múltiplos Canais
- Requisitos Não-Funcionais
 - Crítico
 - Integração com o Gmail
 - Integração com o Whatsapp
 - Integração com o Telegram
 - Integração com o Calendar
 - Desempenho
 - Relevante
 - Escalabilidade
 - Segurança
 - Confiabilidade
 - Opcional
 - Manutenibilidade

4.0 - DIAGRAMAS DO SISTEMA

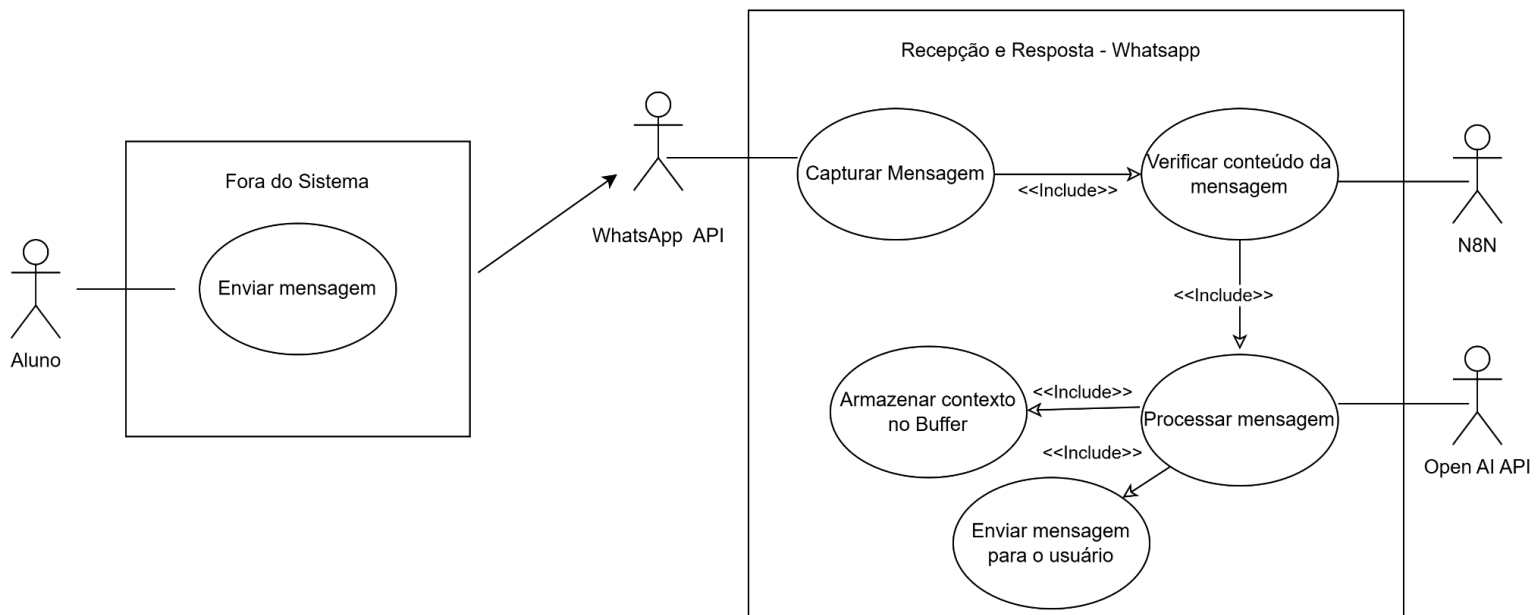
4.1 - Diagrama de Caso de Uso

Os diagramas de caso de uso são representações gráficas que mostram as interações entre os atores e o sistema. Eles descrevem o que o sistema é capaz de fazer do ponto de vista funcional, destacando suas principais funcionalidades. Esse tipo de diagrama é útil para capturar os requisitos do sistema e fornecer uma visão geral das funcionalidades esperadas.

4.1.1 - Diagrama de Recepção e Resposta no Whatsapp

O diagrama de caso de uso **Envia Mensagem pelo WhatsApp** (Figura 1) demonstra as interações entre o sistema e os atores no envio de mensagens por meio do canal WhatsApp.

Figura 1 - Caso de Uso Envia Mensagem pelo WhatsApp



Fonte: Autoria própria (2025)

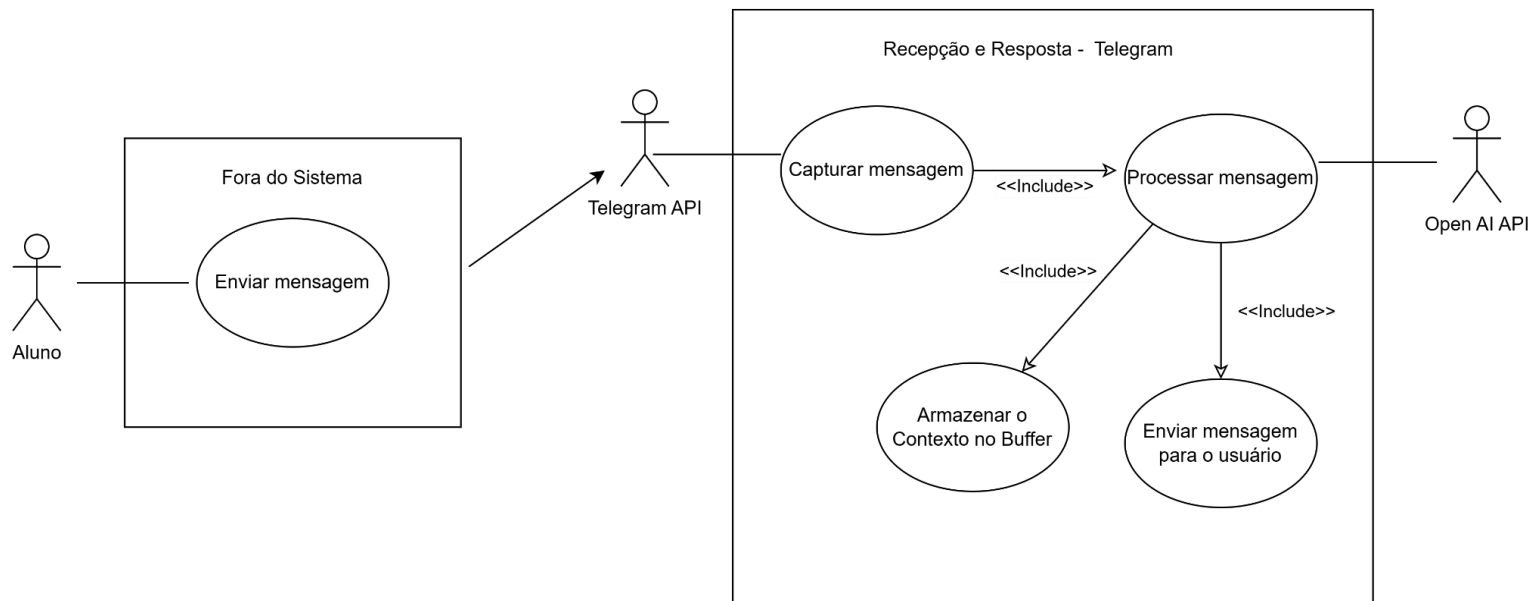
O fluxo começa com o envio de uma mensagem por um usuário, que é capturada pelo sistema. Essa captura inclui o registro do conteúdo, remetente e outros metadados importantes, como o horário de envio. Em seguida, o sistema realiza uma verificação detalhada do conteúdo, identificando inconsistências, como mensagens em formatos inválidos ou contendo linguagens inadequadas.

Após a verificação, o sistema armazena a mensagem no buffer para manter o contexto das interações, permitindo que futuras mensagens sejam processadas com base no histórico do usuário. A mensagem é então analisada por algoritmos de inteligência artificial para compreender sua intenção e gerar uma resposta adequada. Por fim, o sistema realiza o envio da resposta ao destinatário por meio do WhatsApp, garantindo que a comunicação seja eficiente, segura e personalizada para atender às necessidades do usuário.

4.1.2 - Diagrama de Recepção e Resposta no Telegram

O diagrama de caso de uso **Envia Mensagem pelo Telegram** (Figura 2) descreve as interações entre o sistema e os atores envolvidos no envio de mensagens por meio do canal Telegram.

Figura 2 - Caso de Uso Envia Mensagem pelo Telegram



Fonte: Autoria própria (2025)

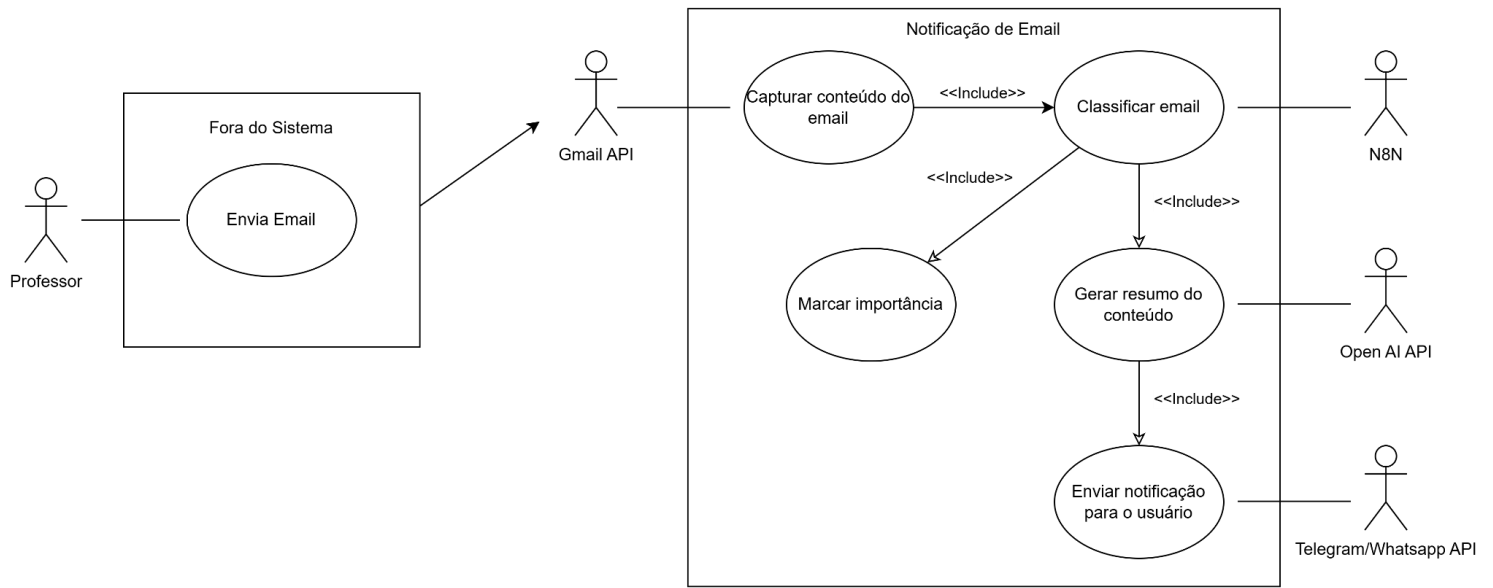
Esse processo inicia com o recebimento de uma mensagem enviada por um usuário, seguida da captura pelo sistema, que armazena o conteúdo e os metadados associados, como o remetente e o horário. Após a captura, o sistema realiza uma análise para verificar a integridade e a relevância do conteúdo, garantindo que apenas mensagens válidas sejam processadas.

Posteriormente, o sistema utiliza inteligência artificial para analisar o contexto da mensagem e armazená-la temporariamente no buffer, possibilitando um histórico de interações que pode ser usado em respostas futuras. Após o processamento, o sistema gera uma resposta apropriada e a encaminha para o destinatário por meio do Telegram. Essa funcionalidade assegura uma comunicação rápida, eficiente e contextualizada, promovendo uma experiência de usuário fluida e personalizada.

4.1.3 - Diagrama de Notificação de e-mail

O diagrama de caso de uso **Notificação do e-mail** (Figura 3) demonstra as interações do sistema para o processamento e envio de e-mails.

Figura 3 - Caso de Uso de Notificação do e-mail



Fonte: Autoria própria (2025)

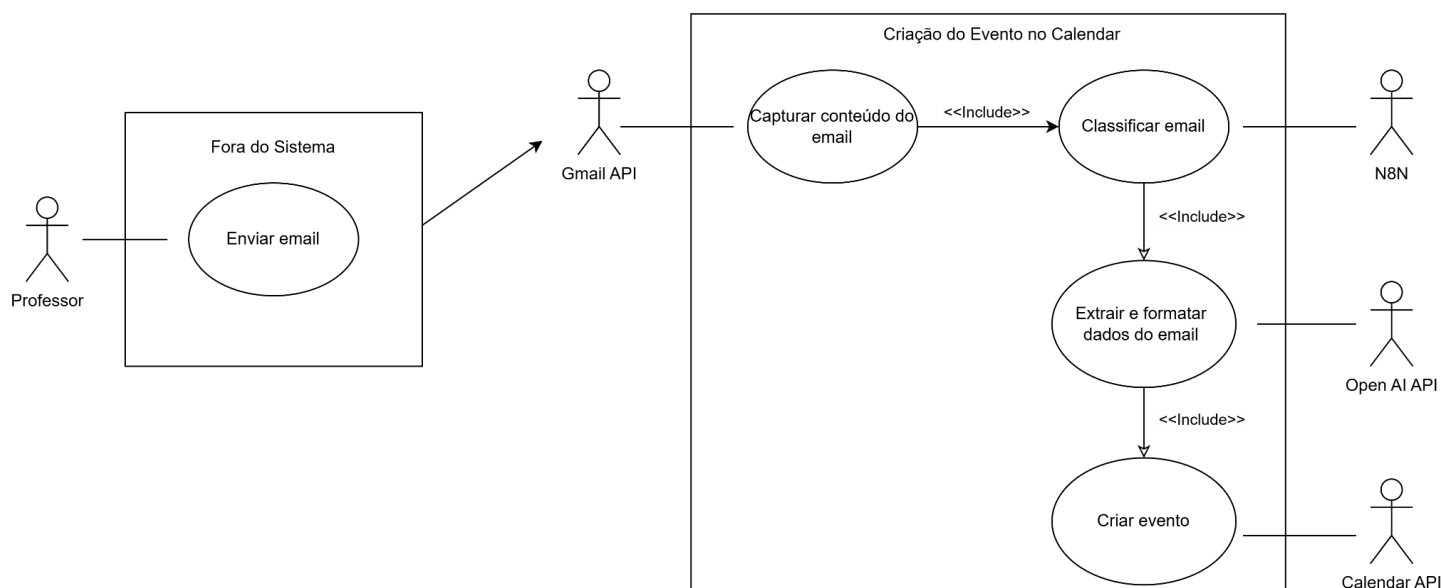
O fluxo inclui a captura do conteúdo do e-mail, a classificação quanto à relevância, a marcação de e-mails importantes, e a geração de resumos para facilitar a leitura. O sistema organiza e prioriza informações para garantir que os e-mails mais relevantes sejam destacados.

Notificações automáticas são enviadas aos destinatários para informá-los sobre o conteúdo do e-mail. Esse fluxo otimiza a comunicação e o gerenciamento de informações importantes, promovendo agilidade no tratamento de dados e reduzindo o tempo necessário para acessar informações críticas.

4.1.4 - Diagrama de Criação de Evento no Calendar

O diagrama de caso de uso **Criação de Eventos no Calendar** (Figura 4) descreve como o sistema processa e-mails relevantes para extrair e organizar informações necessárias para a criação de eventos no calendário.

Figura 4 - Caso de Uso de Criação de Evento



Fonte: Autoria própria (2025)

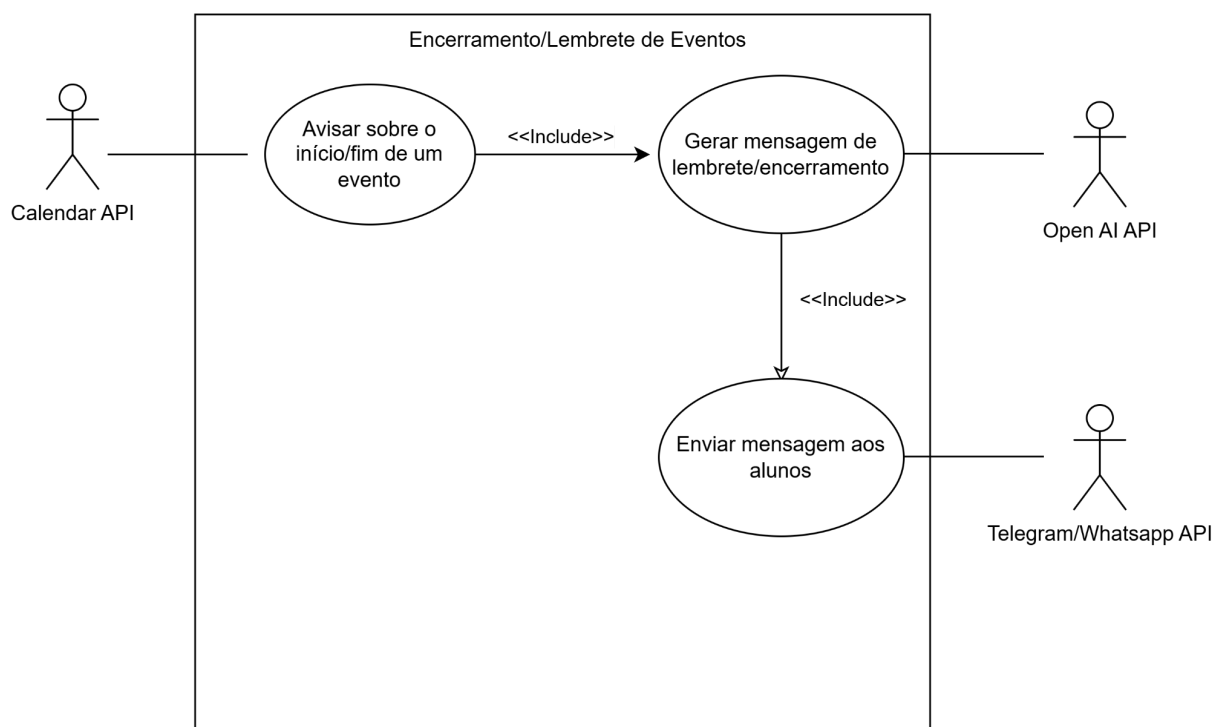
O fluxo começa com a funcionalidade "Pega o conteúdo do e-mail", onde o sistema acessa os e-mails recebidos para identificar dados como título, descrição, e datas relacionadas. Essa etapa inicial garante que todas as informações contidas no e-mail sejam consideradas para o processamento posterior.

Após obter o conteúdo, o sistema realiza a classificação do e-mail. Nesse momento, os e-mails são analisados e categorizados como relevantes ou irrelevantes, com base em regras pré-definidas. Apenas os e-mails relevantes avançam para a próxima etapa, Extração e Formatação dos Dados do Evento. Aqui, o sistema organiza e padroniza os dados extraídos, preparando-os para a criação do evento. Finalmente, na etapa Criação do Evento, as informações processadas são utilizadas para gerar um evento no calendário, garantindo que ele esteja completo e corretamente configurado para o usuário.

4.1.5 - Diagrama de Lembrete e Encerramento de Evento no Calendar

O diagrama de caso de uso **Lembrete de Eventos/Finalização de Eventos** (Figura 5) descreve as interações do sistema para avisar os alunos sobre o início ou término de eventos programados no calendário.

Figura 5 - Caso de Uso de Lembrete de Eventos/Finalização de Eventos



Fonte: Autoria própria (2025)

O fluxo começa com a funcionalidade "Avisa sobre o Início/Fim de um evento", onde o sistema detecta automaticamente que um evento está prestes a começar ou foi concluído. Essa etapa inicial garante que os alunos estejam cientes dos compromissos importantes e não percam informações relevantes.

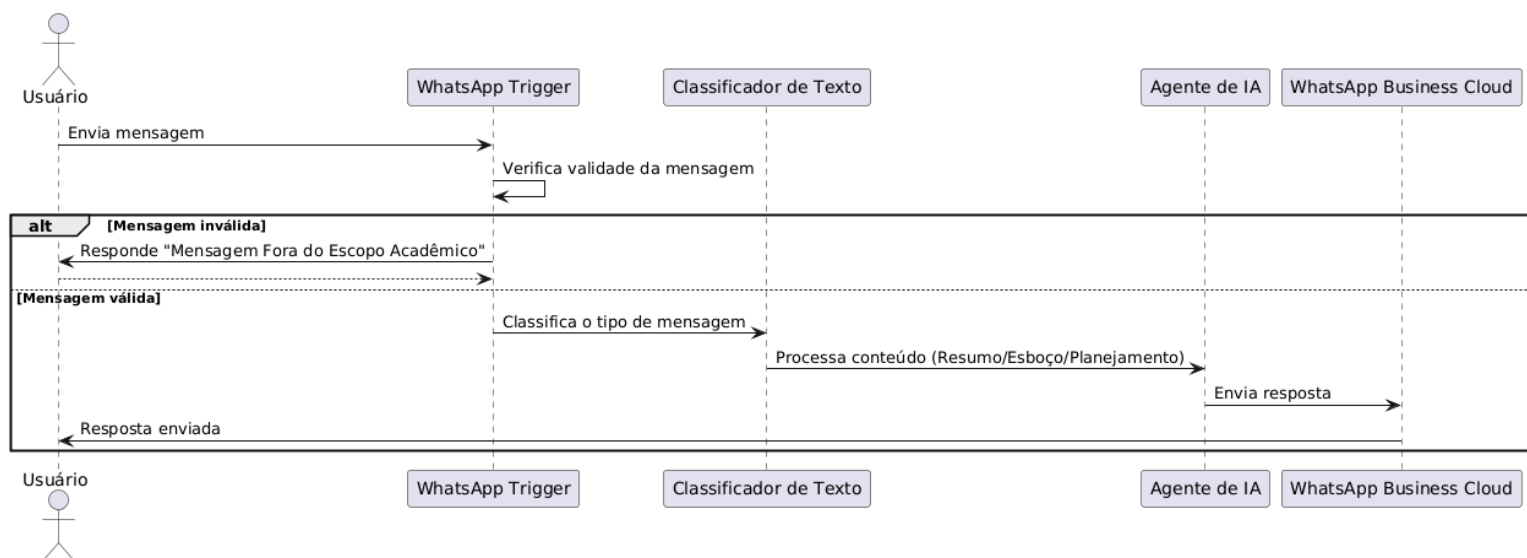
Após detectar o status do evento, o sistema aciona a funcionalidade Gera uma mensagem de Lembrete/Encerramento. Essa mensagem é personalizada com detalhes do evento, como título, horário de início ou término, e informações adicionais sobre possíveis tarefas pendentes. Por fim, a funcionalidade **Envia a mensagem aos alunos** é responsável por distribuir essas notificações aos participantes pelos canais configurados, como WhatsApp ou Telegram. Esse caso de uso garante que os alunos mantenham-se organizados e informados sobre suas atividades acadêmicas, promovendo maior eficiência no gerenciamento de tempo e tarefas.

4.2 - Diagrama de Sequência

Os diagramas de sequência, por outro lado, mostram como os objetos ou componentes do sistema interagem entre si em um fluxo temporal. Eles descrevem a ordem em que as mensagens são enviadas e recebidas para realizar uma tarefa específica. Esses diagramas são amplamente usados para detalhar o comportamento dinâmico do sistema, oferecendo uma visão clara de como as interações ocorrem ao longo do tempo.

4.2.1 - Diagrama de Sequência de Mensagem Recebida no WhatsApp

Figura 6 - Diagrama de Sequência de Mensagem Recebida no Whatsapp



Fonte: Autoria própria (2025)

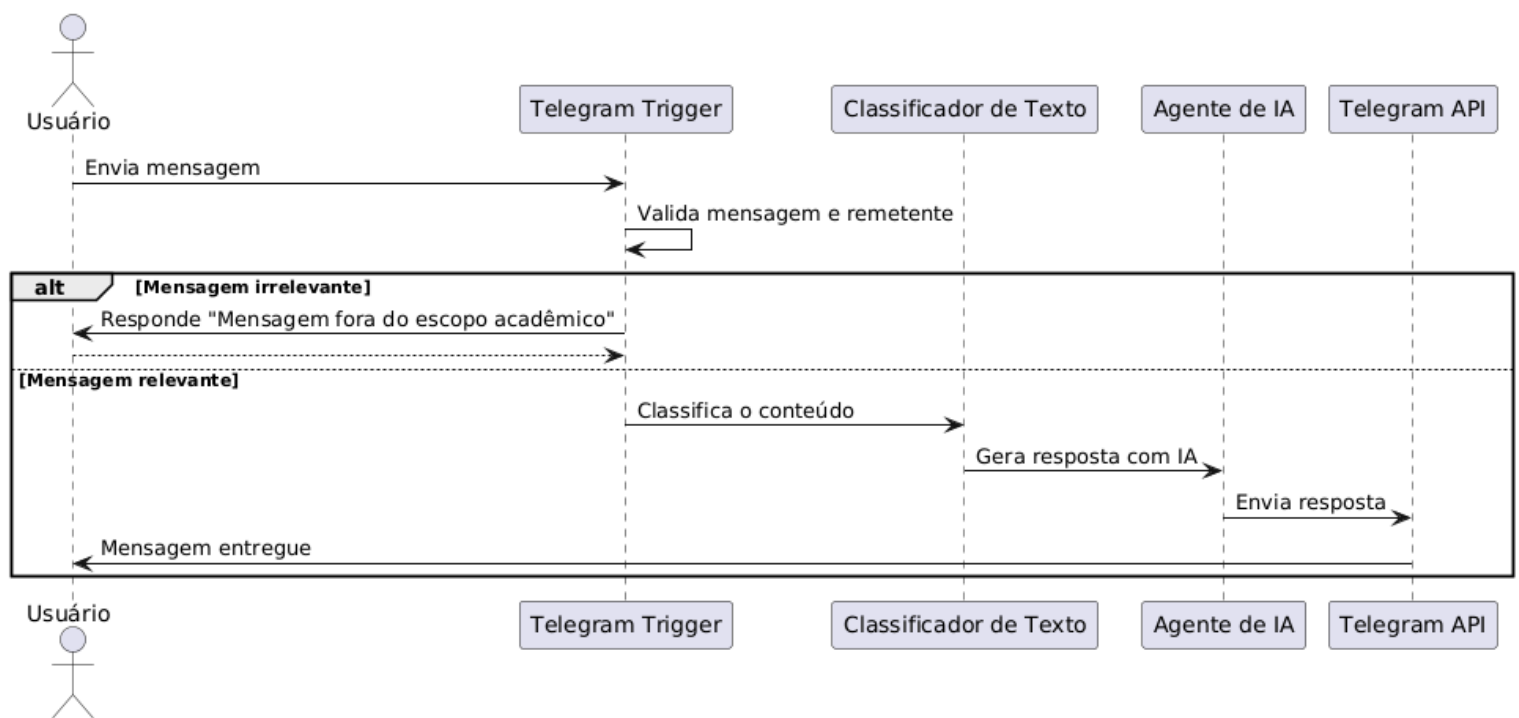
No diagrama apresentado acima, o processo inicia com o usuário enviando uma mensagem ao sistema por meio do WhatsApp Trigger. Em seguida, o sistema verifica a validade da mensagem utilizando o Classificador de Texto, que analisa se o conteúdo está dentro do escopo acadêmico definido. Caso a mensagem seja inválida, o sistema responde imediatamente ao usuário com a mensagem "Mensagem Fora do Escopo Acadêmico". Já se a mensagem for considerada válida, ela é classificada de acordo com seu tipo, como Resumo, Esboço ou Planejamento.

O diagrama fornece uma visão clara e detalhada do fluxo de mensagens no sistema, desde a recepção até a resposta ao usuário. Após a classificação, o conteúdo válido é enviado ao Agente de IA, que processa as informações e gera a resposta adequada, como um resumo ou plano detalhado. Finalmente, a resposta gerada é enviada ao usuário por meio do

WhatsApp Business Cloud, completando o ciclo de comunicação. O diagrama utiliza os termos "Usuário" para representar quem envia a mensagem, "WhatsApp Trigger" e "Agente de IA" para definir os agentes internos do sistema, e "WhatsApp Business Cloud" como o canal de entrega da resposta. Setas conectando esses elementos ilustram claramente o fluxo de informações no sistema.

4.2.2 - Diagrama de Sequência de Mensagem Recebida no Telegram

Figura 7 - Diagrama de Sequência de Mensagem Recebida no Telegram



Fonte: Autoria própria (2025)

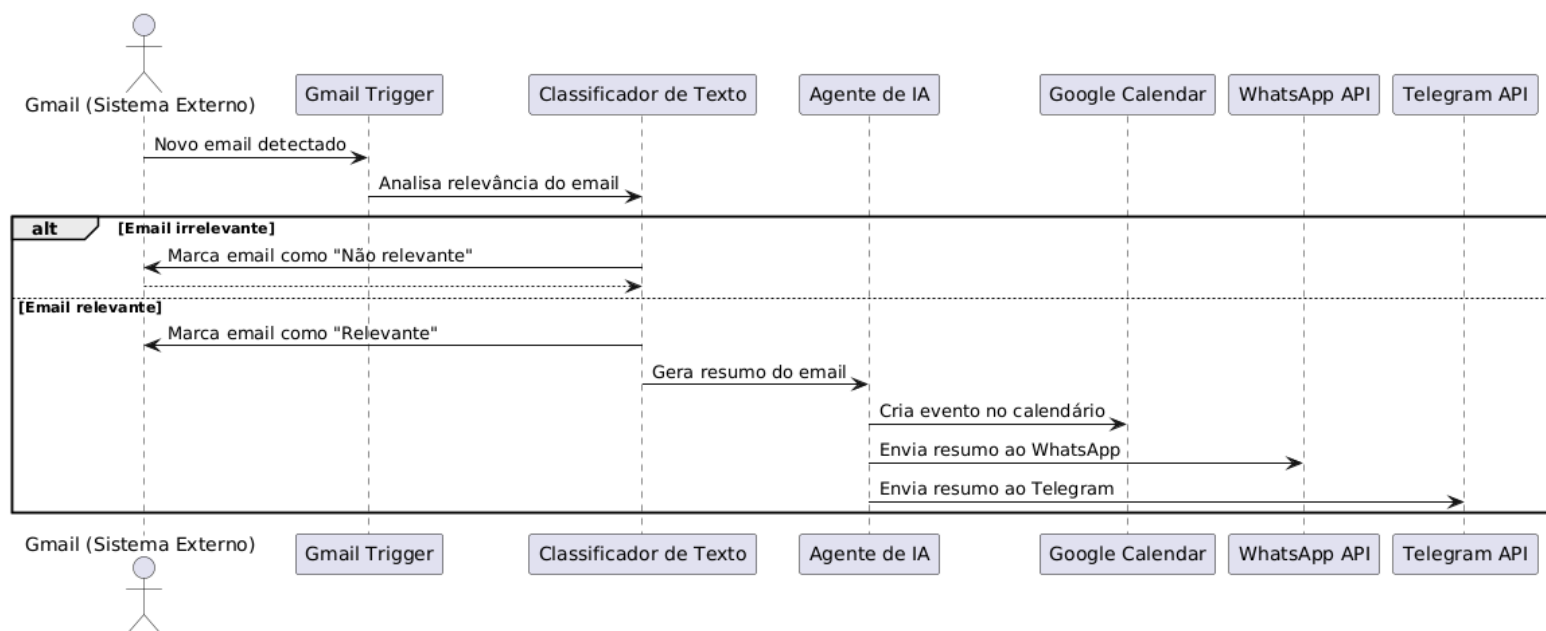
No diagrama presente na Figura 7, o processo inicia com o sistema Gmail Trigger detectando um novo e-mail recebido. Esse e-mail é imediatamente encaminhado ao Classificador de Texto, que analisa sua relevância com base em critérios previamente definidos. Caso o e-mail seja considerado irrelevante, ele é marcado como "Não relevante" e o fluxo é encerrado. No entanto, se o e-mail for classificado como relevante, o fluxo continua com o envio das informações ao Agente de IA.

O Agente de IA processa o e-mail relevante e gera um resumo com as informações principais. Esse resumo é então utilizado para criar ou atualizar um evento no Google Calendar. Além disso, o resumo é enviado simultaneamente para os usuários pelos canais de comunicação WhatsApp API e Telegram API, garantindo que os participantes sejam

notificados de forma eficaz. O diagrama utiliza o termo "Gmail Trigger" para indicar o ponto de entrada do fluxo, "Classificador de Texto" e "Agente de IA" como responsáveis pelo processamento, e "WhatsApp API" e "Telegram API" como canais de notificação. As setas conectando esses elementos ilustram claramente o fluxo de informações e ações no sistema, fornecendo uma visão detalhada e sequencial do processo.

4.2.3 - Diagrama de Sequência de Classificação e Resumo de E-mails

Figura 8 - Diagrama de Sequência de Classificação e Resumo de E-mails



Fonte: Autoria própria (2025)

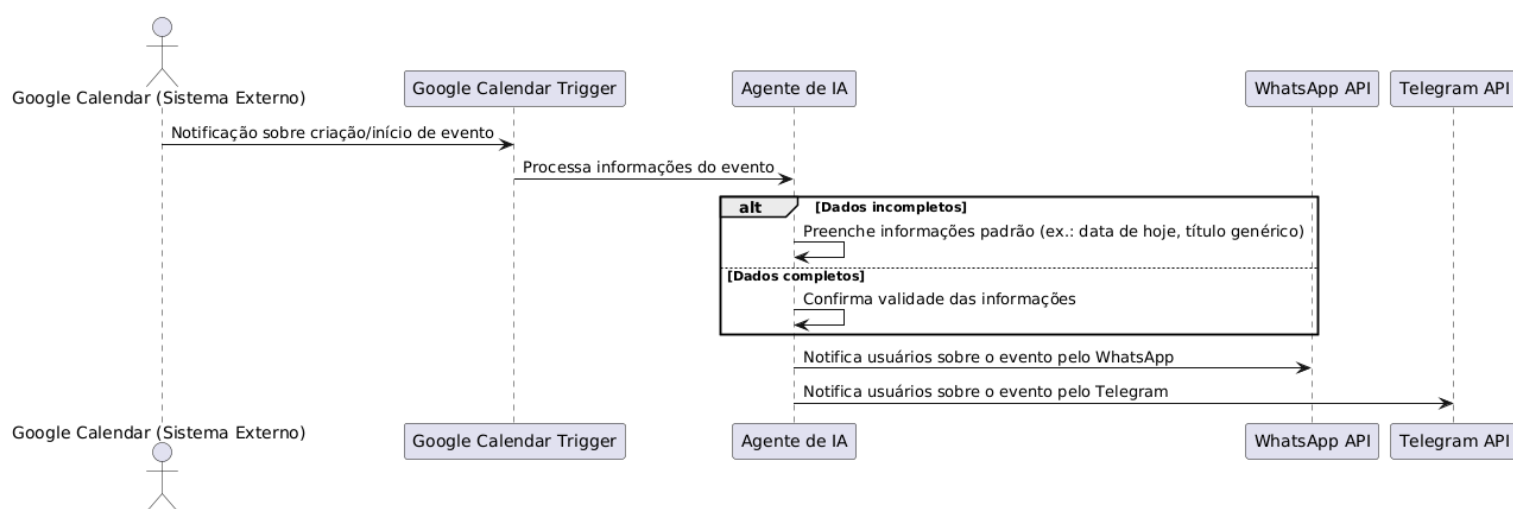
O diagrama de sequência da figura 8 descreve o processo de manipulação de e-mails detectados pelo sistema a partir do Gmail Trigger. Quando o Gmail detecta um novo e-mail, este é encaminhado ao Gmail Trigger, que aciona o Classificador de Texto para determinar a relevância do e-mail. Caso o e-mail seja considerado irrelevante, ele é marcado como "Não relevante" e o fluxo é interrompido. Por outro lado, se o e-mail for classificado como relevante, o sistema continua o processamento.

No caso de um e-mail relevante, o Classificador de Texto o marca como "Relevante" e o encaminha ao Agente de IA. Este agente é responsável por processar o conteúdo e gerar um resumo das informações mais importantes do e-mail. Em seguida, o resumo gerado é utilizado para criar ou atualizar eventos no Google Calendar. Além disso, o mesmo resumo é distribuído por dois canais de comunicação: WhatsApp API e Telegram API, garantindo que os usuários sejam informados de maneira rápida e eficiente.

O diagrama apresenta uma estrutura que permite visualizar claramente os diferentes caminhos que o e-mail pode seguir, dependendo da sua classificação inicial. Ele demonstra como os elementos interagem entre si e como os processos de filtragem, processamento e notificação são realizados. A separação dos fluxos por relevância do e-mail permite uma gestão eficiente e orientada a prioridades, enquanto o uso de múltiplos canais de comunicação melhora a acessibilidade e a distribuição das informações para os usuários.

4.2.4 - Diagrama de Sequência de Criação de Eventos

Figura 9 - Diagrama de Sequência de Criação de Eventos



Fonte: Autoria própria (2025)

Este diagrama de sequência acima retrata o fluxo de notificações relacionadas à criação ou início de eventos no Google Calendar. O processo inicia com o Google Calendar enviando uma notificação ao Google Calendar Trigger, indicando que um evento foi criado ou está prestes a começar. A partir dessa notificação, o Google Calendar Trigger encaminha as informações do evento ao Agente de IA para processamento.

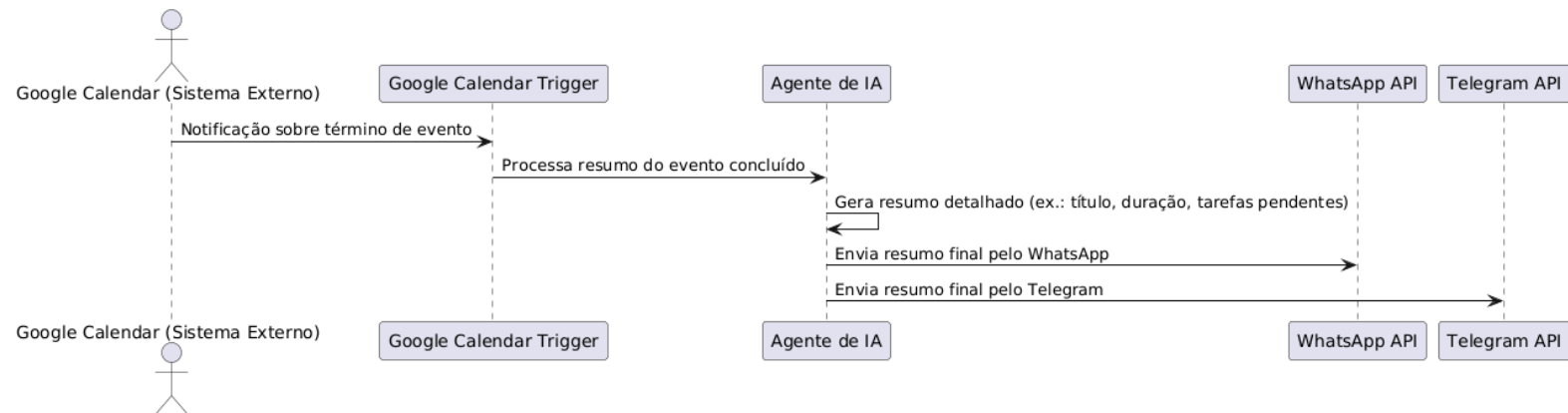
Neste momento, ocorre uma bifurcação no fluxo. Se os dados recebidos estiverem incompletos, o Agente de IA preenche automaticamente as informações ausentes com valores padrão, como a data atual ou um título genérico. Caso os dados estejam completos, o agente valida as informações e segue para as etapas finais. Essa divisão de fluxo, representada pelo bloco “alt”, garante que todos os eventos sejam processados corretamente, mesmo que algumas informações estejam ausentes inicialmente.

Com as informações validadas ou ajustadas, o Agente de IA realiza duas ações principais de notificação. Primeiro, ele utiliza a WhatsApp API para informar os usuários sobre o evento por meio desse canal. Em seguida, uma mensagem semelhante é enviada através da Telegram API, garantindo que os participantes recebam as atualizações em diferentes plataformas de comunicação. Esse fluxo demonstra a robustez do sistema ao lidar

com cenários de dados incompletos, além de destacar a abordagem multicanal para engajar os usuários.

4.2.5- Diagrama de Sequência de Encerramento de Eventos

Figura 10 - Diagrama de Sequência de Finalização de Eventos



Fonte: Autoria própria (2025)

Este diagrama de sequência representa o fluxo de notificações após o término de um evento no Google Calendar. O ciclo inicia quando o Google Calendar envia uma notificação ao Google Calendar Trigger, informando que um evento foi concluído. Essa notificação serve como ponto de partida para o processamento do resumo do evento.

O Google Calendar Trigger encaminha os dados ao Agente de IA, que processa essas informações para gerar um resumo detalhado do evento. Este resumo inclui elementos como o título do evento, a duração e possíveis tarefas que ainda estejam pendentes. O foco aqui é consolidar todas as informações relevantes do evento para que os participantes tenham uma visão clara de sua conclusão.

Após gerar o resumo, o Agente de IA utiliza dois canais de comunicação para distribuir o conteúdo: a WhatsApp API e a Telegram API. Por meio do WhatsApp, os participantes recebem o resumo diretamente no aplicativo, garantindo acessibilidade e praticidade. Em seguida, o mesmo resumo é enviado via Telegram, assegurando que todos os usuários sejam alcançados independentemente do canal que utilizem. Este diagrama ilustra a eficiência do sistema em gerenciar notificações pós-evento de maneira automatizada e acessível.

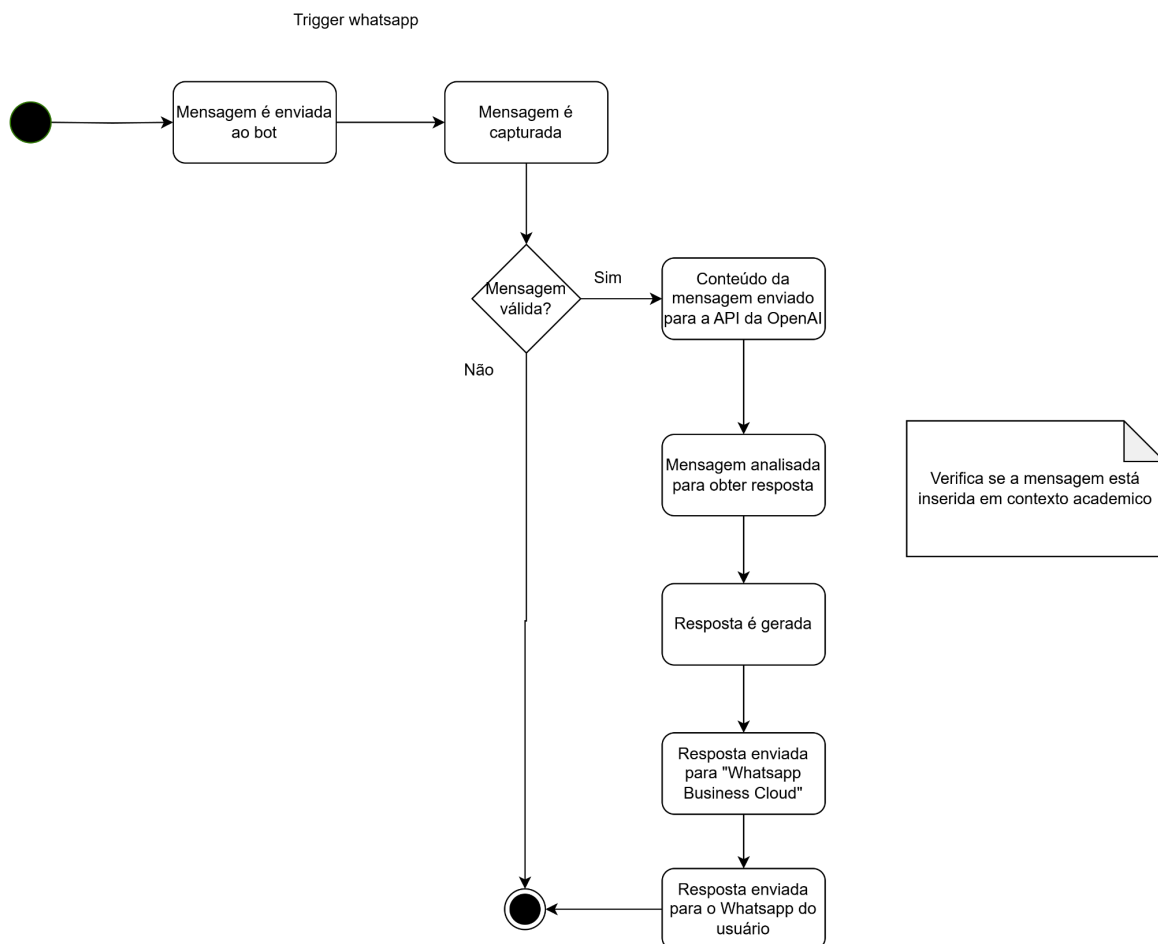
4.3 - Diagrama de Atividade

Os diagramas de atividade têm o objetivo de descrever o fluxo de trabalho ou os processos dentro do sistema. Eles representam as atividades realizadas, os pontos de decisão e os caminhos alternativos. Por serem focados em fluxos de trabalho, eles são amplamente utilizados para mapear processos de negócios e identificar possíveis gargalos ou melhorias.

4.3.1 - Diagrama de Atividade de Mensagem do Whatsapp

O diagrama de atividade apresentado na Figura 11 ilustra o fluxo de ações realizadas quando um usuário interage com o bot para enviar uma mensagem e receber uma resposta.

Figura 11 - Diagrama de Atividade de Mensagem do Whatsapp



Fonte: Autoria própria (2025)

Os participantes principais no fluxo são o Usuário, o Bot e a API da OpenAI. Esse fluxo é projetado para garantir a validação da mensagem e a geração de uma resposta relevante ao usuário.

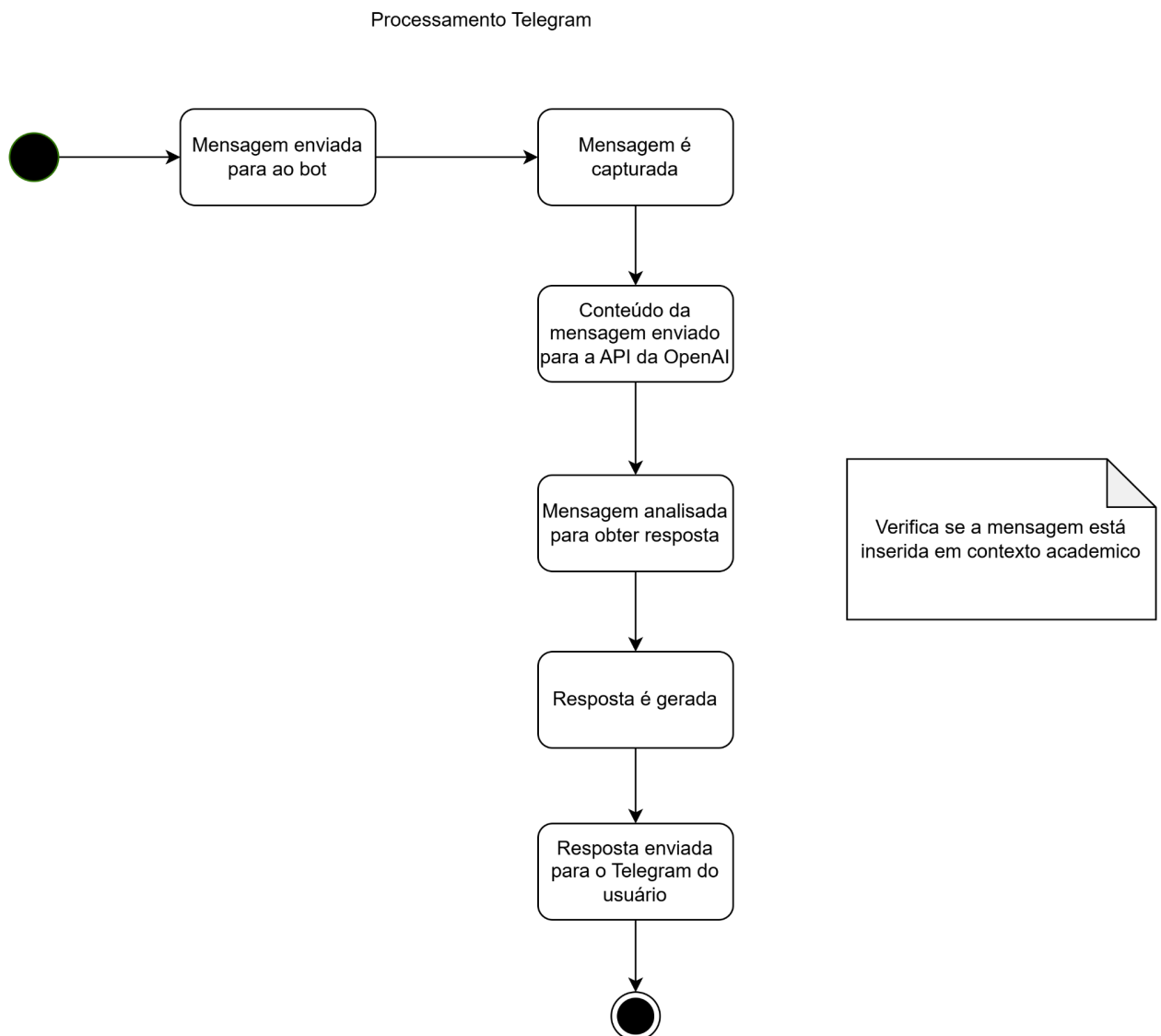
O processo começa quando o usuário envia uma mensagem ao bot, que captura o conteúdo enviado. Em seguida, o sistema realiza uma verificação para validar se a mensagem

está no formato esperado e se é adequada para processamento. Caso a mensagem seja considerada válida, ela passa por uma análise adicional para confirmar se está dentro de um contexto acadêmico. Após essas validações, a mensagem é enviada para a API da OpenAI, que processa o conteúdo e retorna uma resposta baseada no objetivo do usuário. O sistema então envia essa resposta para o WhatsApp Business Cloud, que entrega o conteúdo diretamente ao usuário no WhatsApp, completando o ciclo de interação.

4.3.2 - Diagrama de Atividade de Mensagem do Telegram

O diagrama de atividade da Figura 12 consegue detalhar o fluxo de processamento de mensagens enviadas ao bot por meio do Telegram, destacando todas as etapas que garantem a captura, validação, análise e resposta ao usuário.

Figura 12 - Diagrama de Atividade de Mensagem do Telegram



Fonte: Autoria própria (2025)

O processo começa quando o usuário envia uma mensagem ao bot pelo Telegram, servindo como o ponto inicial de interação no sistema. Essa mensagem é imediatamente capturada pelo bot, que a encaminha para as etapas subsequentes de validação e análise.

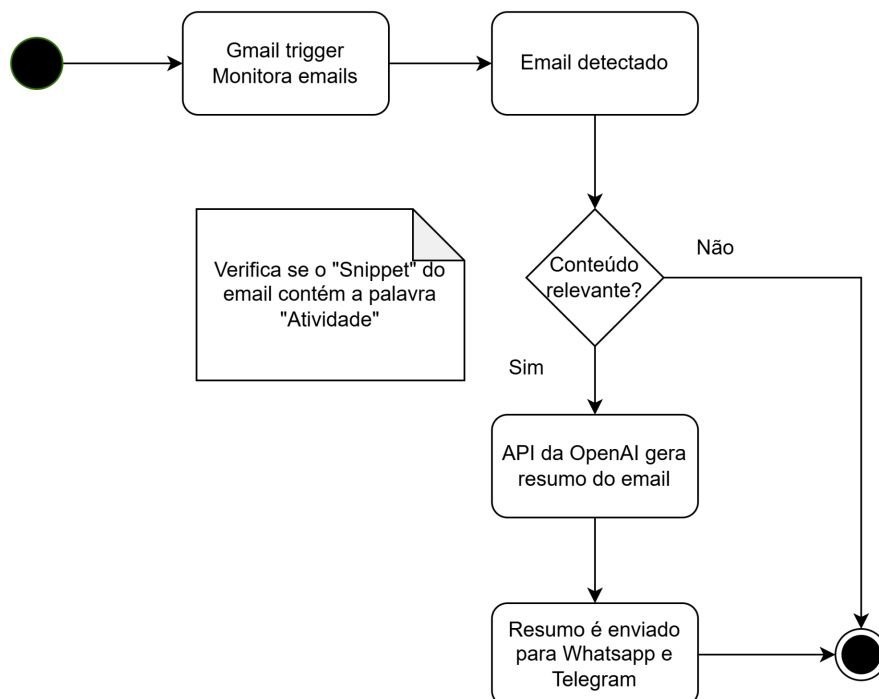
Após a captura da mensagem, o conteúdo é enviado para a API da OpenAI, onde ocorre uma análise preliminar. Essa integração é fundamental para interpretar o conteúdo da mensagem e determinar as informações necessárias para gerar uma resposta relevante. Durante esse fluxo, o sistema realiza uma verificação importante para garantir que a mensagem esteja inserida dentro de um contexto acadêmico. Essa verificação é essencial para alinhar a funcionalidade do bot com seu propósito educacional, filtrando mensagens irrelevantes e garantindo que apenas conteúdos válidos sejam processados.

Com a validação concluída e o contexto acadêmico confirmado, o sistema procede para a etapa de análise detalhada da mensagem. A API da OpenAI processa o conteúdo, retornando uma resposta apropriada com base nos parâmetros definidos e no objetivo da interação. Essa resposta é gerada de forma personalizada, considerando o conteúdo da mensagem enviada pelo usuário e garantindo que a solução apresentada seja adequada ao problema ou consulta realizada.

Finalmente, a resposta gerada é enviada diretamente ao usuário por meio do Telegram, concluindo o ciclo de interação. Essa etapa de entrega garante que o usuário receba um retorno claro e relevante no mesmo canal utilizado para iniciar a comunicação. Esse fluxo destaca a eficiência e a organização do sistema, demonstrando como ele integra validação, análise de contexto e geração de resposta para fornecer uma experiência alinhada às necessidades acadêmicas do usuário.

4.3.3 - Diagrama de Atividade de Classificação e Resumo de E-mails

Figura 13 - Diagrama de Atividade Resumo e E-mails



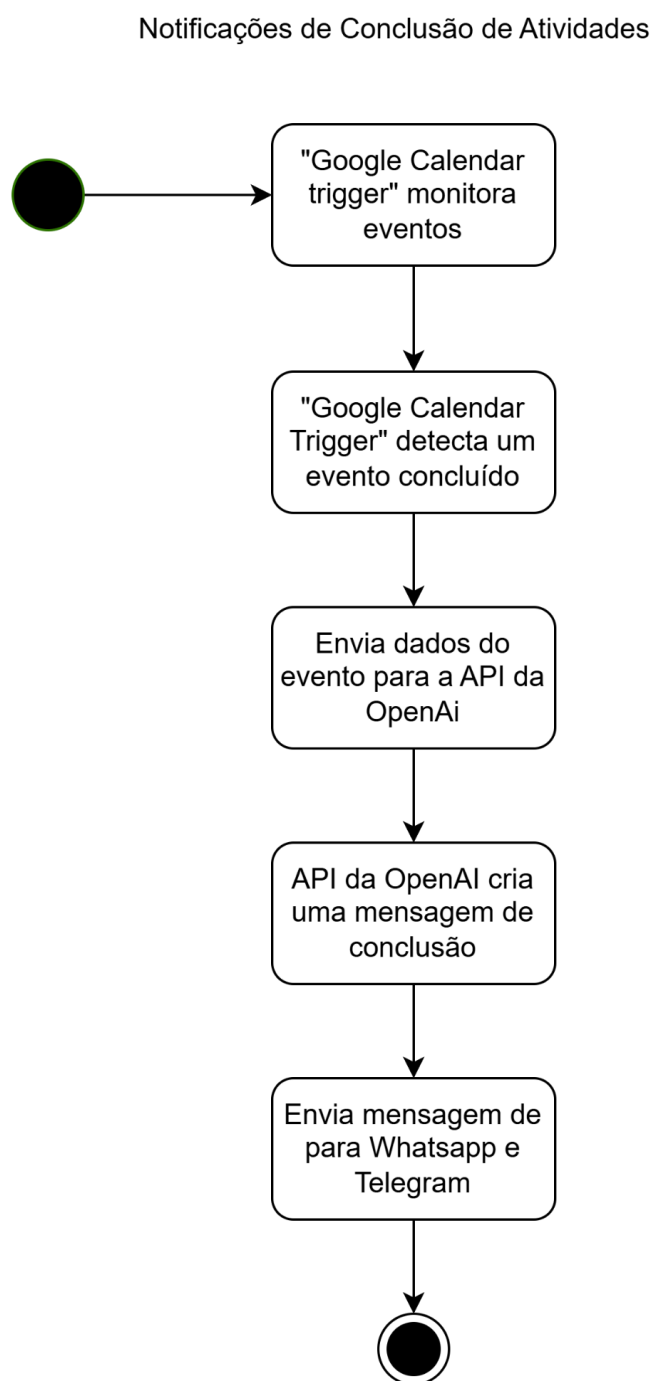
Fonte: Autoria própria (2025)

O diagrama de atividade acima ilustra as etapas envolvidas no processo de monitorar, classificar, e filtrar e-mails importantes recebidos pelo usuário, incluindo também a geração de resumo desses e-mails, e o envio de um resumo para os canais de comunicação do usuário, como WhatsApp e Telegram. Ele destaca como o sistema automatiza a triagem de e-mails e a geração de informações úteis.

Inicialmente, o trigger do Gmail fica monitorando a chegada de e-mails. Quando um novo e-mail é detectado, o trigger verifica o snippet (um pequeno trecho do conteúdo do e-mail) para decidir se o conteúdo dele é relevante academicamente, buscando a palavra-chave "Atividade". Essa verificação funciona como um filtro para determinar se o e-mail possui informações importantes que devem ser processadas, filtrando os e-mails que de fato são direcionados do professor para o aluno. Caso a palavra-chave não seja detectada, o conteúdo do e-mail não é capturado e o fluxo se encerra. Mas caso ela seja detectada, a API da OpenAI gera um resumo geral dos principais pontos do conteúdo do e-mail e encaminha para o Whatsapp e Telegram do usuário.

4.3.4 - Diagrama de Atividade de Notificação de Conclusão de Atividade

Figura 14 - Diagrama de Atividade Conclusão de Atividade



Fonte: Autoria própria (2025)

Fazendo uma análise mais detalhada do fluxograma representado, podemos dizer que ele representa o fluxo de trabalho relacionado ao monitoramento e notificação de conclusão de eventos utilizando o Google Calendar e ferramentas de comunicação como WhatsApp e

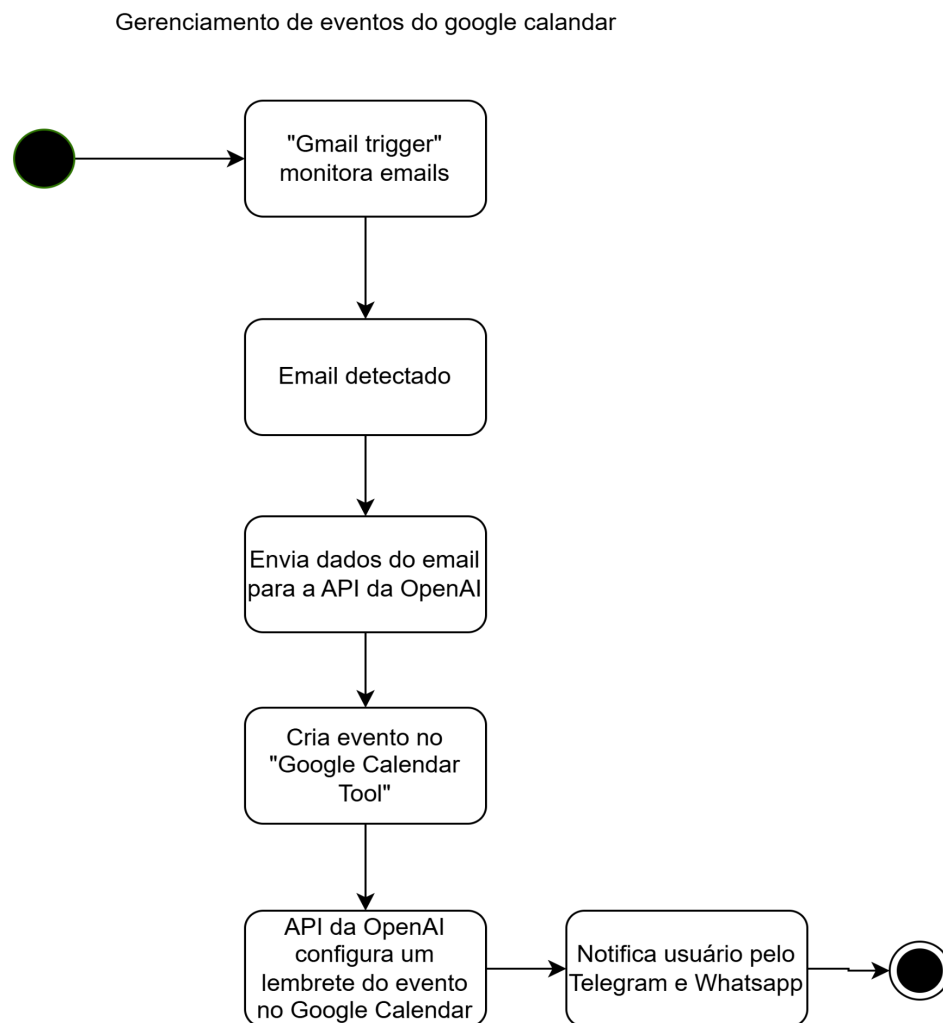
Telegram. Ele ilustra como o sistema automatiza o processo desde a detecção de um evento concluído até o envio de notificações para os participantes.

O processo inicia com o Google Calendar Trigger, que está constantemente monitorando os eventos registrados no calendário. Quando o Trigger detecta que um evento foi concluído, ele aciona o próximo passo no fluxo: o envio dos dados do evento para a API da OpenAI. Essa API desempenha um papel central na geração de uma mensagem de conclusão. Os dados do evento são analisados, e uma mensagem personalizada é criada para resumir os principais detalhes do evento finalizado.

Após a geração da mensagem pela API da OpenAI, o sistema a encaminha para os canais de comunicação escolhidos, neste caso, WhatsApp e Telegram. Essa etapa final garante que os participantes sejam informados sobre a conclusão do evento de forma rápida e eficaz, independentemente do canal de comunicação que utilizem.

4.3.5 - Diagrama de Atividade de Gerenciamento de Eventos no Calendar

Figura 15 - Diagrama de Atividade de Gerenciamento de Eventos no Google Calendar



Fonte: Autoria própria (2025)

O diagrama de atividade acima descreve o processo automatizado para a gestão de e-mails, criação de compromissos no Google Calendar e envio de alertas ao usuário por meio do Telegram e WhatsApp. Ele detalha cada etapa do fluxo, desde a identificação de um e-mail até a comunicação das informações ao usuário.

O ciclo tem início com o Gmail Trigger, que monitora a chegada de novos e-mails na caixa de entrada. Quando um e-mail relevante é identificado, o conteúdo é extraído e enviado para a API da OpenAI, onde os dados são processados para definir os detalhes do evento, como nome, horário e participantes.

Após o processamento, o sistema utiliza a Google Calendar Tool para registrar o evento no calendário, garantindo que as informações estejam organizadas e acessíveis. Além disso, um lembrete é configurado automaticamente no Google Calendar, assegurando que o usuário seja alertado a tempo. Por último, o sistema utiliza os canais Telegram e WhatsApp para enviar notificações personalizadas, informando o usuário sobre o novo evento.

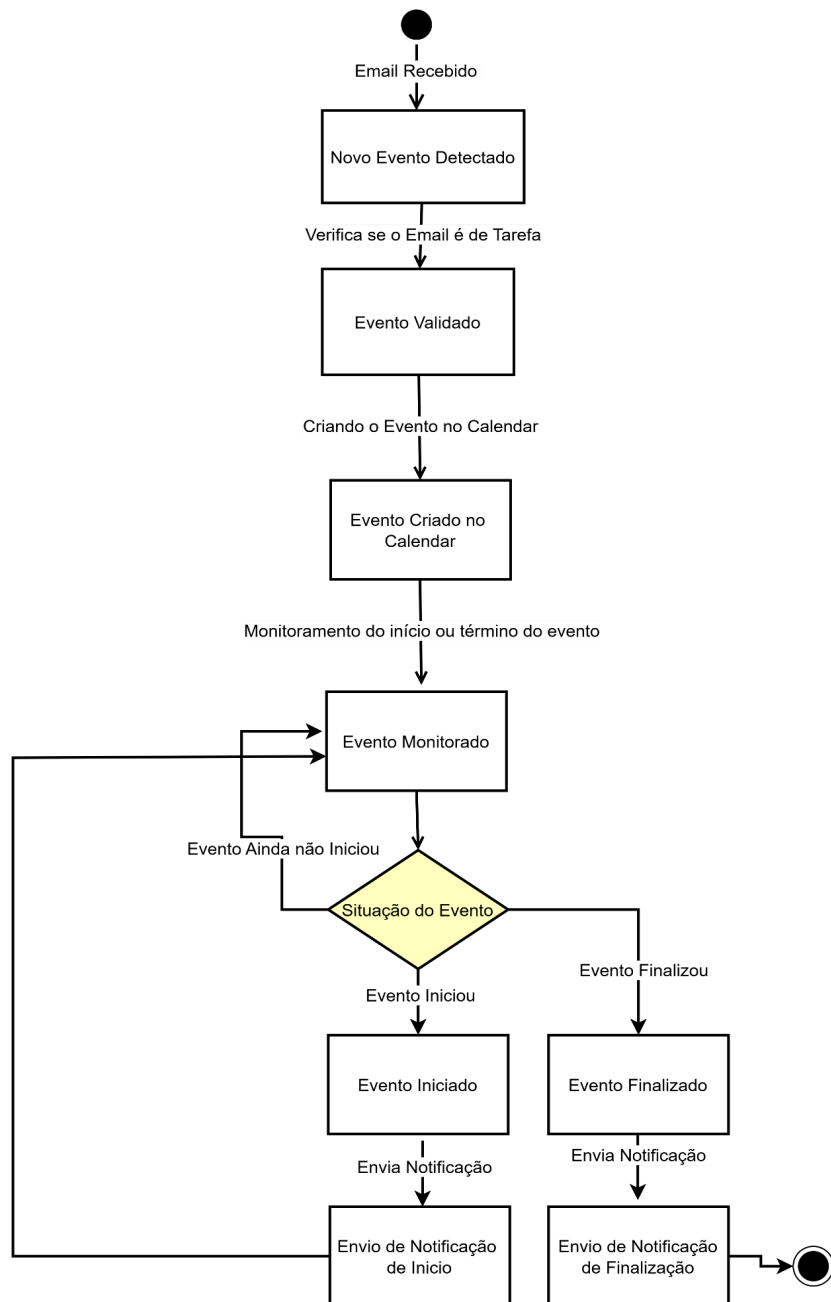
4.4 - Diagrama de Estado

O diagrama de estado é uma ferramenta utilizada para modelar o comportamento de um objeto dentro de um sistema, destacando os diferentes estados pelos quais ele pode passar ao longo de sua vida útil. Ele é composto por estados que representam condições ou situações específicas do objeto e transições, que mostram como e por que o objeto muda de estado, geralmente acionadas por eventos ou ações. Cada estado pode ter comportamentos ou atividades associados que são realizados enquanto o objeto está naquele estado. Por exemplo, em um sistema de pedidos, um pedido pode começar no estado "Criado", transitar para "Aguardando Pagamento" quando um cliente finaliza a compra, e avançar para "Pago" após a confirmação do pagamento, terminando em "Enviado" quando o pedido é despachado.

Esse tipo de diagrama é particularmente útil em sistemas onde o ciclo de vida dos objetos é complexo e dependente de eventos ou condições externas. Ele permite que os desenvolvedores visualizem claramente como as regras de negócio afetam o comportamento do sistema, além de facilitar a identificação de possíveis problemas, como transições não tratadas ou estados irrelevantes. Por exemplo, em um sistema bancário, um cartão de crédito pode alternar entre estados como "Ativo", "Bloqueado" e "Cancelado", dependendo de eventos como tentativas de fraude ou solicitações do cliente.

4.4.1- Diagrama de Estado de Evento

Figura 16 - Diagrama de Estado de Evento



Fonte: Autoria própria (2025)

O diagrama de estado apresentado detalha o ciclo de vida de um evento no sistema, desde sua detecção inicial até a conclusão. O processo começa com a chegada de um e-mail recebido, que aciona o sistema para verificar se o conteúdo corresponde a uma tarefa ou evento. Caso o e-mail seja identificado como relevante, ele segue para o estado de Evento Validado, onde suas informações são analisadas e preparadas para a próxima etapa.

Uma vez validado, o evento passa pelo processo de criação no calendário, resultando no estado de Evento Criado no Calendar. Nesse ponto, o evento é monitorado continuamente pelo sistema para determinar seu status em tempo real, entrando no estado de Evento Monitorado. O monitoramento verifica se o evento já começou ou foi concluído, movendo-o para um dos estados finais com base nessa condição.

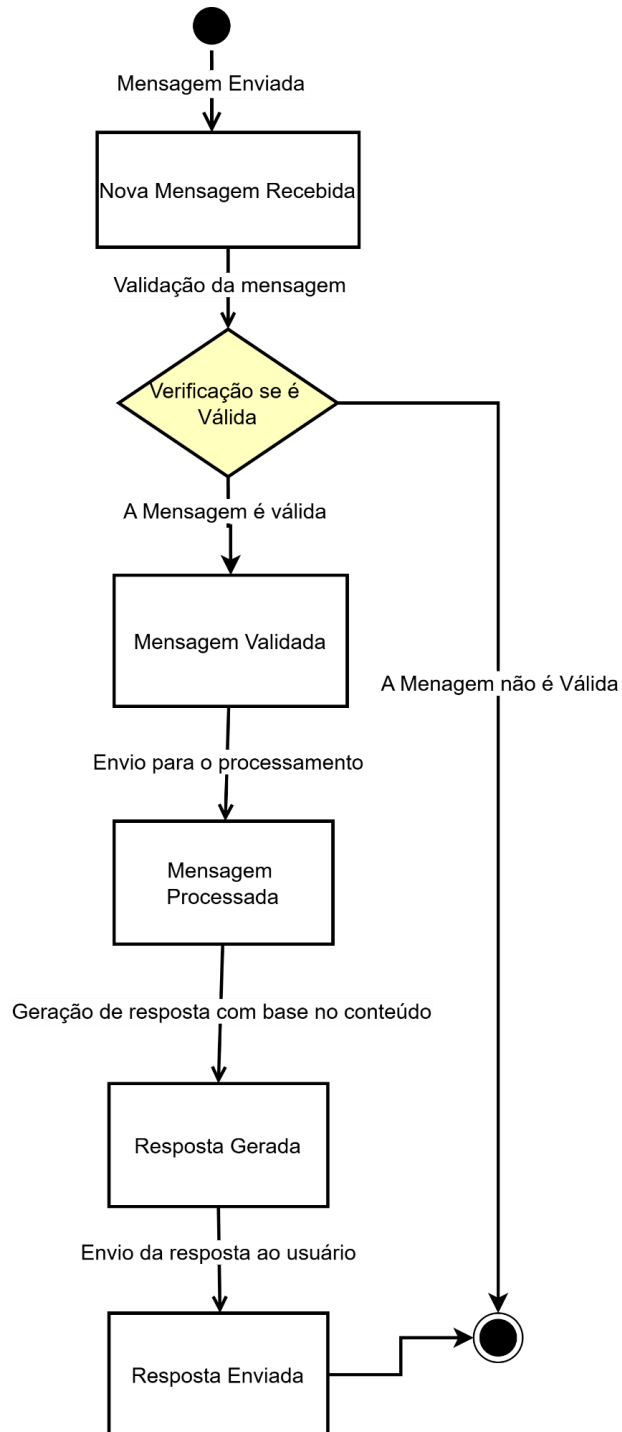
Se o evento ainda não iniciou, ele permanece monitorado até que a condição mude. Quando o evento inicia, o sistema entra no estado de Evento Iniciado e imediatamente dispara uma notificação de início aos participantes, informando sobre o início do compromisso. Por outro lado, se o sistema detecta que o evento foi finalizado, ele muda para o estado de Evento Finalizado e envia uma notificação de conclusão com um resumo do que aconteceu.

4.4.2- Diagrama de Estado de Mensagem

O diagrama abaixo, na figura 17, ilustra o fluxo completo do processamento de uma mensagem no sistema, desde o envio até a entrega da resposta ao usuário. O processo começa com o usuário enviando uma mensagem, que é reconhecida pelo sistema como Nova Mensagem Recebida. Em seguida, a mensagem passa por uma etapa de Validação, onde é verificada sua conformidade com os critérios estabelecidos pelo sistema. Durante essa etapa, uma decisão é tomada: se a mensagem for inválida, o fluxo termina com uma possível resposta genérica ou descarte; se for válida, ela é marcada como Mensagem Validada, e o processo continua.

Com a mensagem validada, ela é enviada para o módulo de Processamento, onde seu conteúdo é analisado por ferramentas ou algoritmos, como IA, para entender sua intenção e gerar a resposta apropriada. Após o processamento, o sistema avança para a etapa de Geração de Resposta, criando um conteúdo específico com base no que foi solicitado. Por fim, a resposta é enviada ao usuário por meio do canal adequado, concluindo o ciclo no estado de Resposta Enviada.

Figura 17 - Diagrama de Estado de Mensagem



Fonte: Autoria própria (2025)

5.0 - TECNOLOGIAS UTILIZADAS

5.1 - N8N

A ferramenta chamada de N8N é apropriadamente a tecnologia mais importante presente neste projeto. Isso se dá pois o N8N é uma plataforma de automação de fluxos de trabalho projetada para integrar diferentes serviços, sistemas e aplicativos de forma eficiente e flexível. Ele possibilita a criação de processos automatizados por meio de uma interface gráfica, eliminando a necessidade de conhecimento avançado em programação.

Com uma abordagem de código aberto o n8n oferece liberdade para personalizações e implementação em ambientes autogerenciados. Por causa dessa característica, ele foi selecionado para ser aplicado neste projeto, dado que a equipe possui mais controle dos dados inseridos no sistema. Além disso, o n8n pode ser auto-hospedado em servidores locais ou na nuvem, garantindo flexibilidade e segurança para atender a exigências regulatórias e corporativas.

Ademais, a interface do N8N adota o conceito de pegar o agente e soltar, permitindo que fluxos de trabalho sejam criados de maneira visual e intuitiva. Cada fluxo é composto por nós (nodes), que representam ações, gatilhos ou processos intermediários. Isso torna a ferramenta acessível tanto para desenvolvedores quanto para usuários com pouco conhecimento técnico.

Outrossim, outro aspecto importante do N8N é que a ferramenta é altamente extensível, suportando integração com APIs personalizadas, além de centenas de serviços e aplicativos populares. Para casos de uso mais específicos, os usuários podem adicionar lógica personalizada por meio de scripts em JavaScript, o que expande ainda mais as possibilidades da ferramenta.

5.2 - WhatsApp Business API

O WhatsApp é um aplicativo de mensagens instantâneas amplamente utilizado em todo o mundo para comunicação pessoal e empresarial. Desenvolvido pela empresa WhatsApp Inc., foi lançado inicialmente em 2009 e posteriormente adquirido pelo Facebook, agora Meta Platforms, em 2014. Desde então, o WhatsApp se consolidou como uma das plataformas mais populares para troca de mensagens, suportando texto, voz, vídeo e diversas outras formas de comunicação digital.

O WhatsApp pode ser integrado a sistemas empresariais, como CRMs e plataformas de automação, por meio do WhatsApp Business API. Essa integração permite automação de respostas, envio de notificações e gestão centralizada de interações.

A API permite que empresas enviem mensagens em massa de maneira personalizada. Isso inclui notificações transacionais como confirmações de pedido, atualizações de envio ou lembretes de pagamento e mensagens de engajamento, como promoções e campanhas de marketing.

A integração da API do WhatsApp com o n8n permite que alguns processos de comunicação sejam automatizados, otimizando interações com clientes de maneira escalável e personalizada. O n8n, como uma plataforma de automação, pode usar a API do

WhatsApp para conectar o WhatsApp Business a outros serviços, como ferramentas de e-mail. Essa integração combina as capacidades da API do WhatsApp, como envio de mensagens e notificações, com os recursos do n8n, como automação condicional, manipulação de dados e fluxo de trabalho visual.

A interação é viabilizada por meio de nós específicos no n8n que trabalham diretamente com a API do WhatsApp. Esses nós permitem configurar gatilhos, enviar mensagens e processar dados recebidos.

5.3 - Telegram API

O Telegram é uma plataforma de mensagens instantâneas baseada na nuvem, que foi desenvolvido com o objetivo de oferecer uma alternativa rápida, segura e confiável para comunicação digital, diferenciando-se de outros aplicativos de mensagens por sua forte ênfase em privacidade e funcionalidade avançada. O aplicativo utiliza uma infraestrutura robusta de servidores distribuídos globalmente, permitindo a sincronização automática de mensagens entre dispositivos, garantindo acesso contínuo aos dados de qualquer lugar.

Sua API aberta é um diferencial significativo, possibilitando o desenvolvimento de bots personalizados para automatização de tarefas, integração com sistemas externos e execução de comandos avançados, sendo esse um dos principais motivos para a escolha desta ferramenta, dado a facilidade da implementação de sua API.

A Bot API do Telegram permite criar bots que podem enviar mensagens, responder a comandos, receber dados dos usuários e interagir de várias formas com os participantes. Um bot é essencialmente uma conta do Telegram que não é controlada por um ser humano, mas por um sistema automatizado. Tal API foi utilizada para a criação do bot Tutor.IA, que está presente na implementação deste sistema.

5.4 - Google Calendar API

O Google Calendar é uma plataforma de calendário online que permite aos usuários criar, gerenciar e compartilhar eventos de forma simples e eficiente. Com ele, é possível agendar compromissos, adicionar lembretes, configurar notificações e sincronizar com outros dispositivos. O serviço é amplamente utilizado por indivíduos e empresas para organização de agendas, reuniões, eventos e compromissos diários, além de permitir o compartilhamento de calendários com colegas e familiares, facilitando a coordenação de agendas e o planejamento de atividades. A interface intuitiva também possibilita a integração com outros aplicativos do Google, como o Gmail, o que torna a gestão de tarefas ainda mais prática.

Ademais, a utilização de sua API oferece aos desenvolvedores uma maneira de integrar as funcionalidades do calendário em suas aplicações, automatizando processos e melhorando a experiência do usuário. Com a API, é possível criar, editar, excluir e listar eventos, bem como gerenciar convites e a visibilidade de eventos..

Ao integrar o n8n com o Google Calendar, é possível automatizar diversas tarefas relacionadas à gestão de eventos e compromissos, eliminando a necessidade de intervenção manual e otimizando a produtividade dos usuários. Uma das funcionalidades

mais comuns de automação do n8n com o Google Calendar é a criação de eventos automaticamente com base em outros dados ou ações. Esta é a funcionalidade principal que será utilizada na implementação deste sistema, ao criar eventos utilizando os dados presentes nos e-mails de atividade que são recebidos pelos alunos.

5.5 - Gmail

O Gmail é um serviço de e-mail gratuito oferecido pelo Google, amplamente utilizado tanto por indivíduos quanto por empresas. Ele permite o envio, recebimento e organização de e-mails, oferecendo funcionalidades como filtros, etiquetas, arquivamento e pesquisa avançada. Além disso, o Gmail se integra com outros produtos do Google, como Google Drive, Google Calendar e Google Contacts, tornando-se uma plataforma central para comunicação e colaboração. Sua interface é intuitiva, e oferece um poderoso sistema de spam, além de recursos como resposta automática, organização de mensagens em pastas e integração com outras ferramentas de produtividade.

A API do Gmail é uma interface poderosa e flexível fornecida pelo Google que permite que desenvolvedores integrem funcionalidades de e-mail do Gmail em seus próprios aplicativos ou sistemas. Ela permite o acesso a diversas funcionalidades do Gmail, como enviar, ler e organizar e-mails, gerenciar rótulos, e até mesmo configurar filtros ou configurar respostas automáticas. A API é projetada para interagir diretamente com as contas do Gmail de usuários autenticados e pode ser usada para automatizar, personalizar e integrar a comunicação por e-mail em processos empresariais ou em fluxos de trabalho pessoais.

A integração entre a API do Gmail e o n8n oferece uma maneira poderosa e eficiente de automatizar tarefas de e-mail e criar fluxos de trabalho personalizados sem a necessidade de programação. Para o caso deste sistema em específico, a função adotada foi a análise dos e-mails recebidos pela conta vinculada, permitindo que o sistema verifique se o e-mail que chegou é um e-mail válido de tarefas ou não.

6.0 - REFERÊNCIAS

GUEDES, Gilleanes T. A. UML2 Uma Abordagem Prática, 3ª Edição. Novatec Editora. São Paulo, Brasil, 2018.

N8N, "N8N Documentation," [Online]. Disponível em: <https://docs.n8n.io/>. Acessado em 26 de jan, 2025.