

IMERSÃO Agentes de IA

APOSTILA COMPLETA

AULA 3

Multiagentes: Criando um
Exército de Agentes de IA que
trabalham para você

PARTE

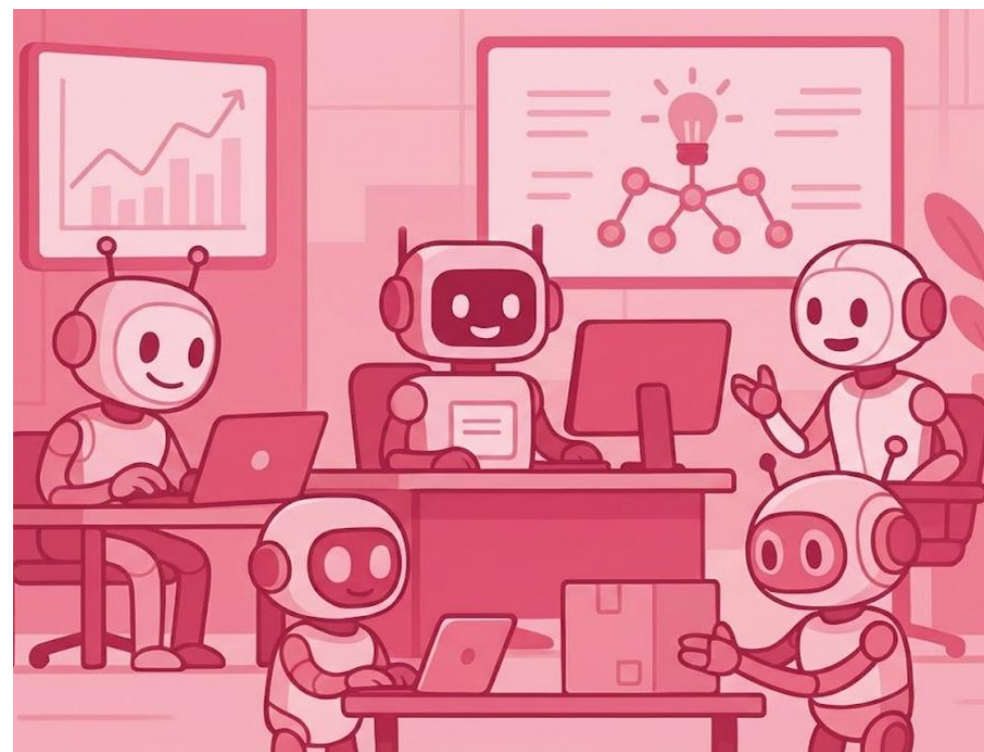
1

ENTENDENDO O CONCEITO DE MULTIAGENTES

Fala, **Impressionador!**

Nesta aula, vamos construir uma automação mais avançada no n8n utilizando o conceito de **multiagentes de IA**. Em vez de uma única inteligência artificial lidando com todas as solicitações, o fluxo será estruturado com agentes especializados, cada um responsável por um tipo específico de atendimento.

A automação conta com um agente principal, que recebe a mensagem do usuário, identifica a intenção e direciona a solicitação para o agente mais adequado. Esse agente não responde diretamente, apenas orquestra o fluxo e retorna ao usuário(a) a resposta gerada pelo agente especialista, tornando o processo mais organizado, escalável e próximo do funcionamento de uma equipe real.



PARTE 2

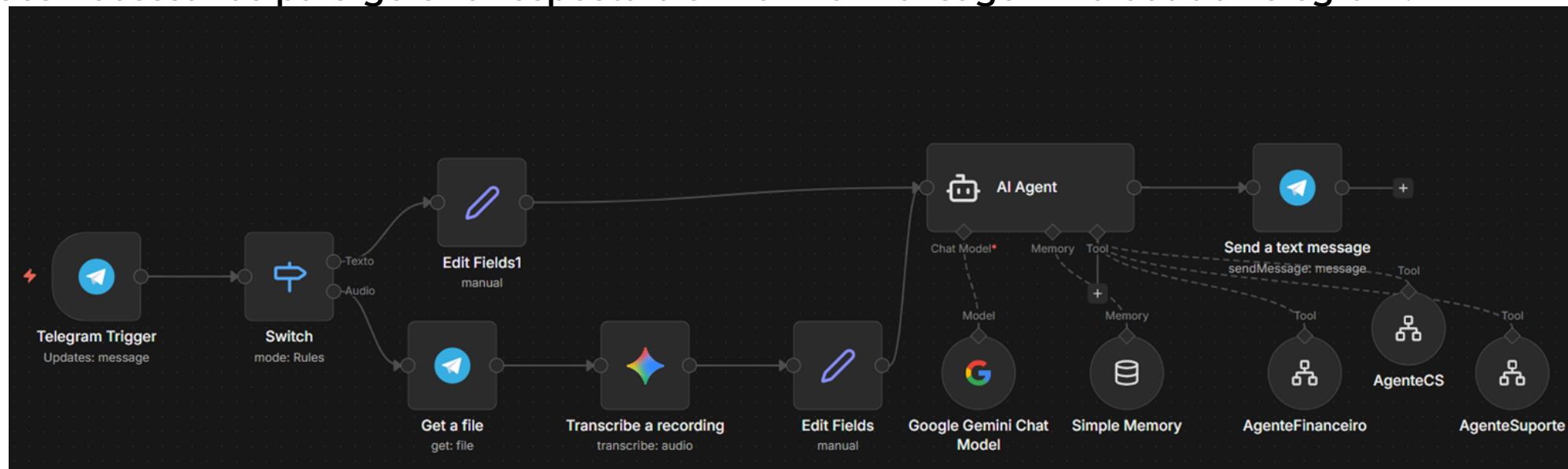
CONHECENDO A AUTOMAÇÃO

PARTE 2

Conhecendo a automação

Imagine que você trabalha em uma consultoria financeira e precisa automatizar o atendimento do setor de suporte que tira dúvidas técnicas sobre os investimentos, o sucesso do cliente que acompanha o cliente e o financeiro que trata questões de pagamento. Para isso, podemos usar uma automação com multiagentes de IA.

Vamos receber uma mensagem de um bot no Telegram, que pode receber mensagens de texto ou de áudio, enviar para um agente de IA que vai identificar a solicitação para acionar o agente específico com as informações necessárias para gerar a resposta e enviar via mensagem no bot do Telegram.



PARTE 3

COMEÇANDO PELO GATILHO

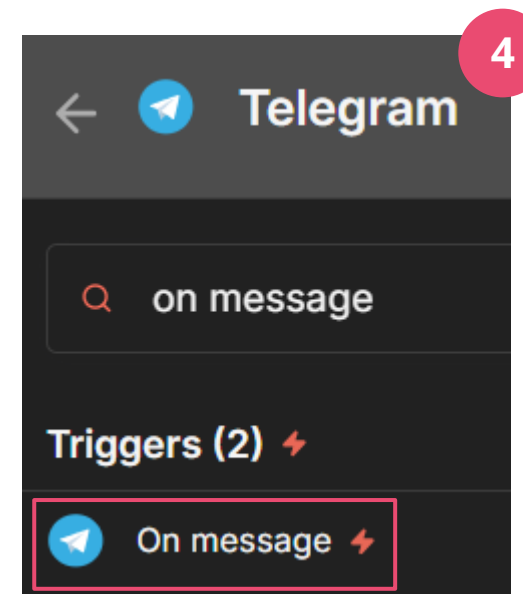
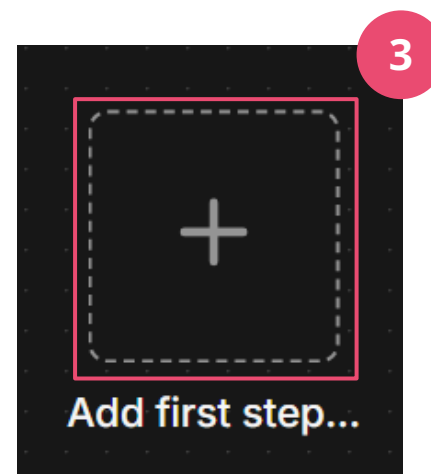
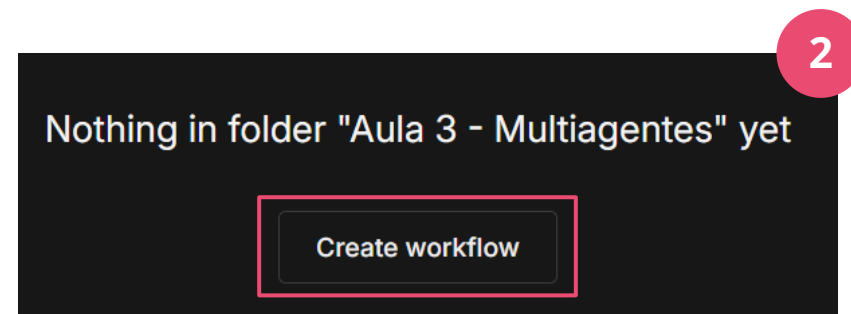
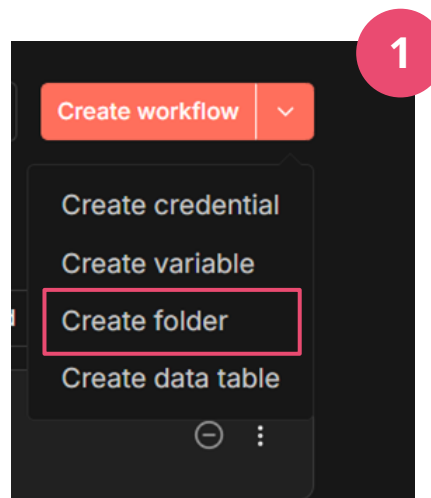
PARTE 3 Começando pelo gatilho

O primeiro passo para construir essa automação é criar uma pasta no n8n (Imagem 1) com a opção **Create folder** que vai agrupar o workflow principal e os workflows secundários.

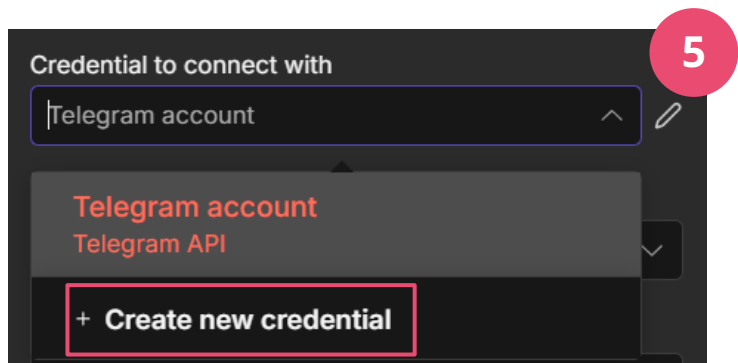
Para manter tudo organizado, podemos dar o nome para essa pasta de "Aula 3 - Multiagentes", por exemplo.

Dentro da pasta vamos criar o workflow principal com o botão **Create workflow** no centro da tela (Imagem 2). Como teremos mais de um fluxo, vamos renomear o fluxo como "Workflow Principal".

O gatilho da aplicação vai ser **on message** do **Telegram**. Para isso, clique no botão **Add first step...**, busque por **Telegram** e selecione a opção **on message**.

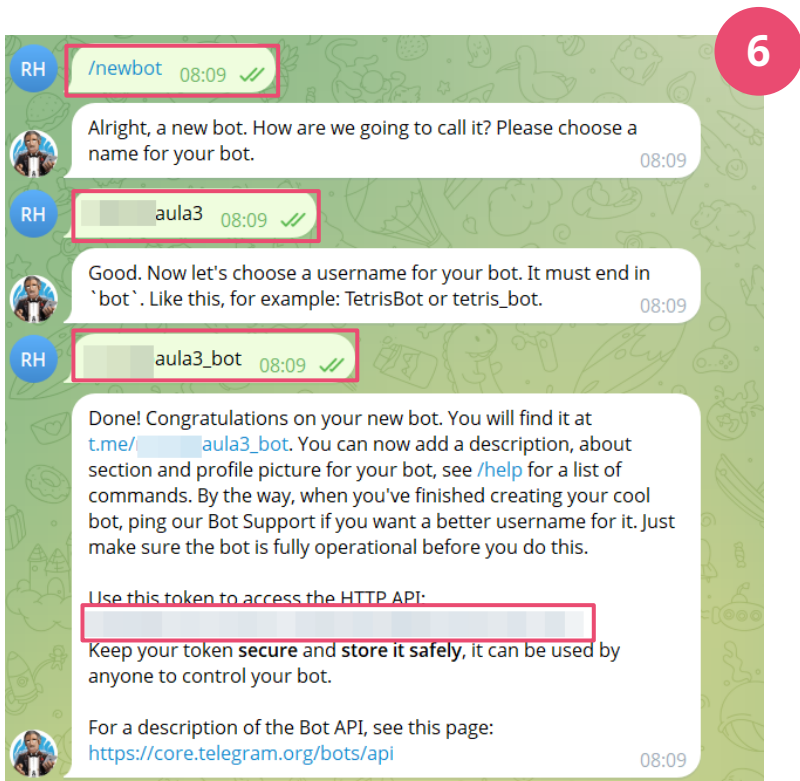


PARTE 3 Começando pelo gatilho



Em seguida, é necessário configurar os parâmetros desse gatilho.

No campo **Credential to Connect With**, clicamos em **Create new credential** (Imagem 5) e devemos preencher o campo **Access Token**.



Para isso, usaremos o **BotFather**, a ferramenta oficial para criação e gerenciamento de bots. Ao abrir a conversa, clique em **Start** e envie o comando **"/newbot"** para iniciar a criação do bot.

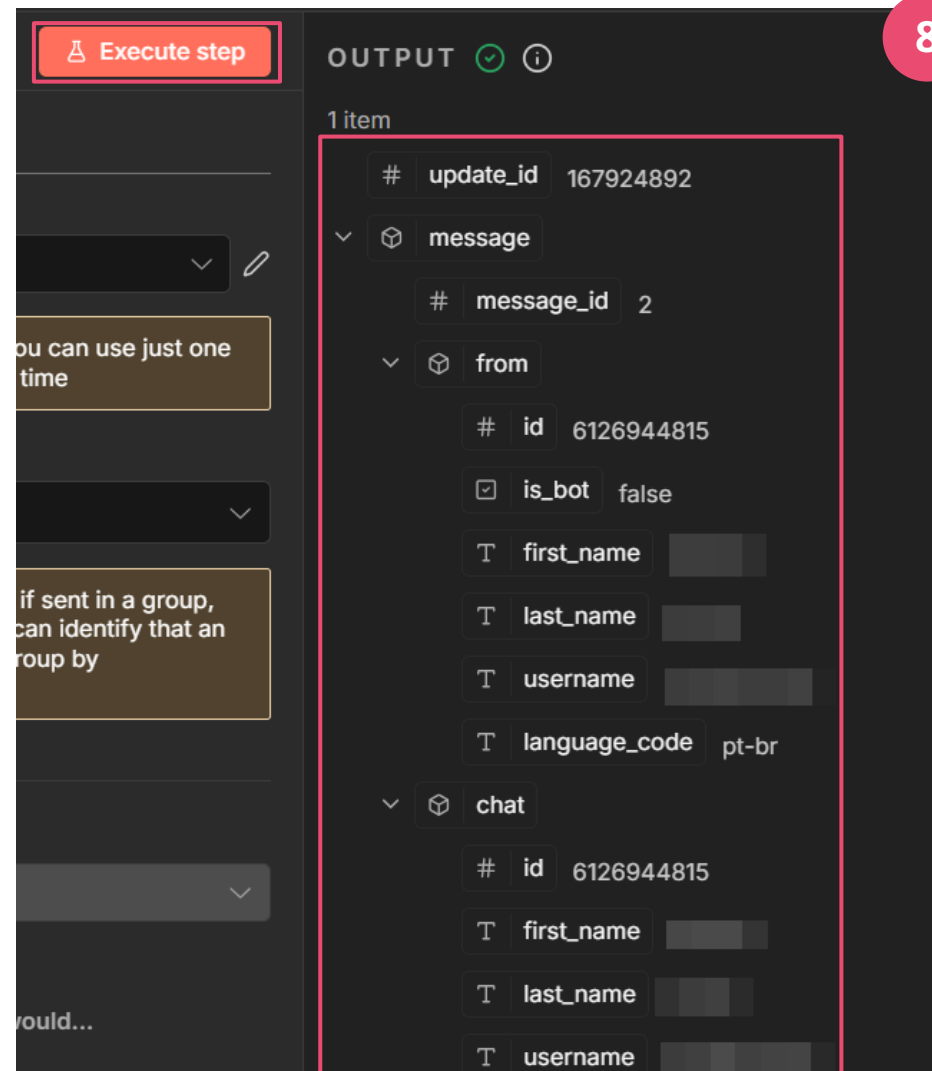
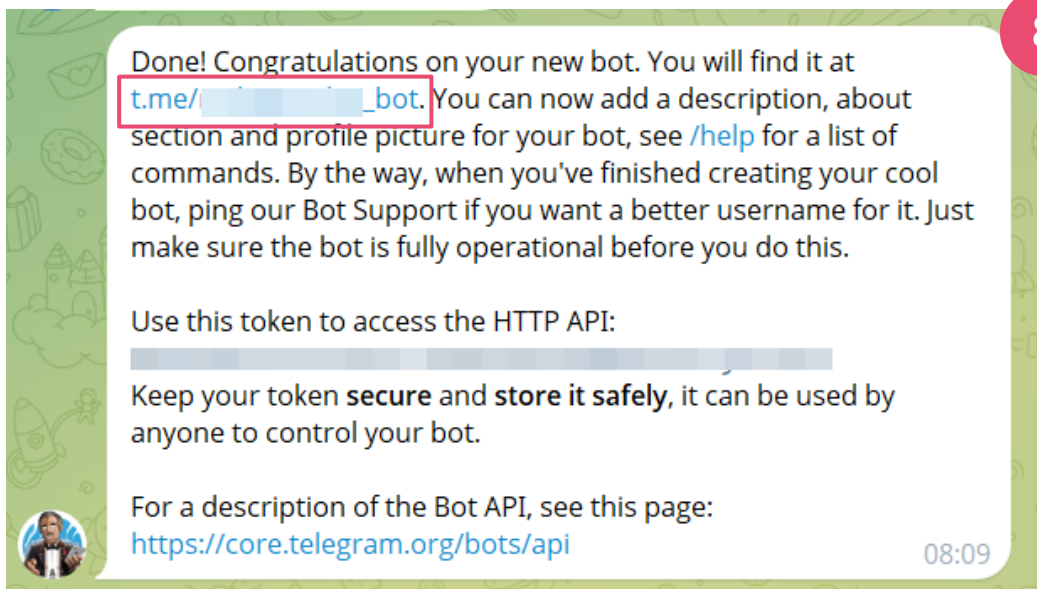
Em seguida, defina um nome para o bot (Ex.: "seunomeaula3"). Depois, escolha um adicione **"_bot"** ao final do nome do bot (Ex.: **seunomeaula3_bot**).

Ao concluir, o BotFather enviará o link de acesso ao bot e o token de acesso que precisamos para integrar a automação (Imagem 6).

PARTE 3 Começando pelo gatilho

Para testar este passo, basta clicar no botão **Execute step** (Imagem 7) e enviar uma mensagem no link que o BotFather disponibilizou para acessar a conversa com o bot que você criou (Imagem 8).

Caso tudo esteja configurado corretamente, as informações da mensagem deverão aparecer na área de **Output** na lateral direita dentro das configurações do nó (Imagem 7).



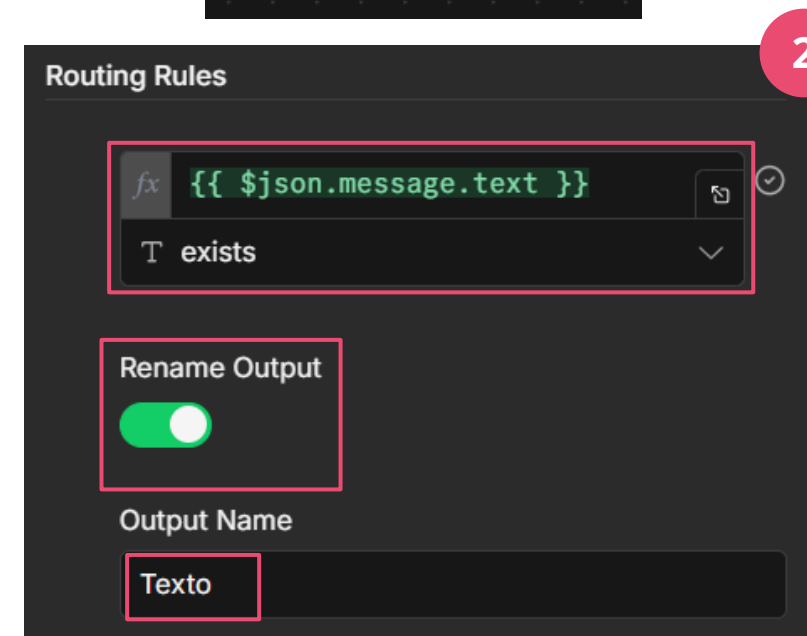
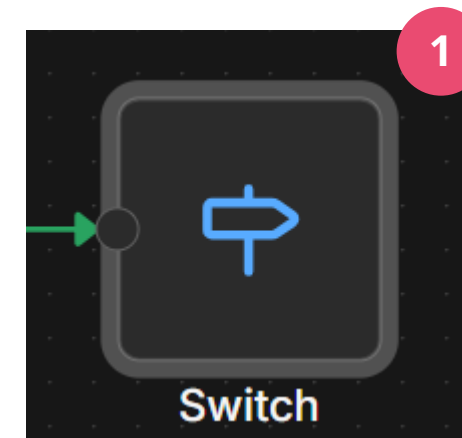
PARTE 4

IDENTIFICANDO O TIPO DE MENSAGEM

Após o gatilho, é necessário identificar o tipo de mensagem recebida. Para isso, utiliza-se um nó **Switch** (Imagem 1) que permite criar diferentes caminhos de execução com base na existência de determinados campos.

O primeiro caminho que vamos configurar é o caminho da mensagem de texto, portanto, no campo **value 1** arraste da área de **Input** o campo "text" do nó anterior e na condição preencha com "exists".

Em seguida, habilite a opção **Rename Output** e preencha o campo **Output Name** com "Texto" para rotular o caminho (Imagem 2).

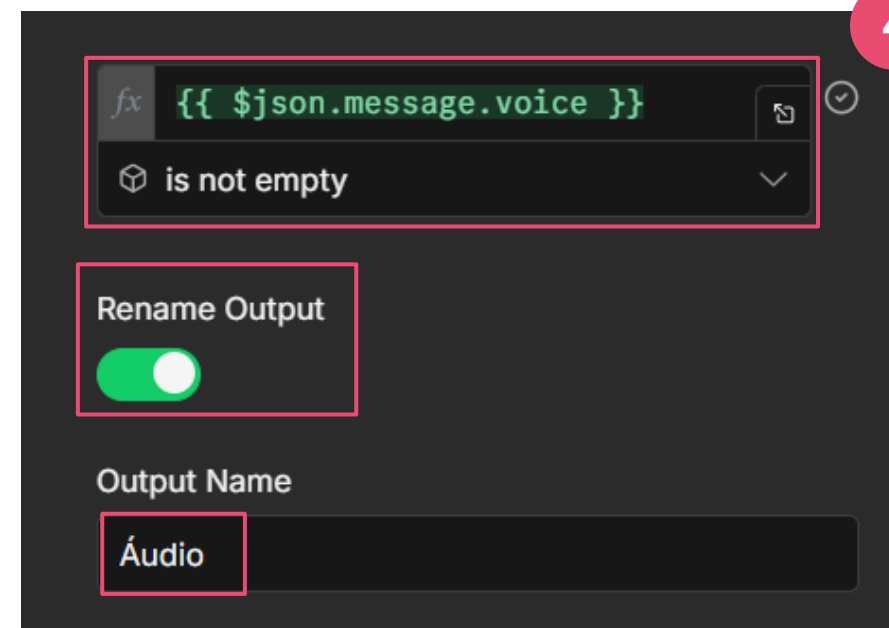
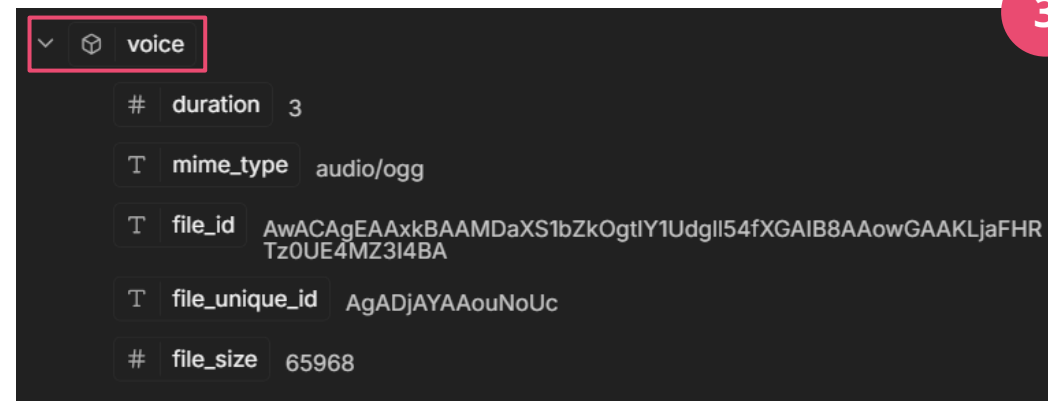


Após construir o caminho de texto, vamos clicar no botão **Add Routine Rule** para adicionar caminho da mensagem se for um áudio.

Para construir esse caminho, precisamos executar o gatilho novamente e mandar uma mensagem de voz no bot do Telegram (Imagem 3).

No **Output** as informações do áudio vão aparecer, e é ela que arrastaremos para o campo **value 1** e nas condições colocaremos a opção "is not empty".

Neste caminho, também habilitaremos a opção **Rename Output** e preencha o campo **Output Name** com "Áudio" para rotular o caminho (Imagem 4).



PARTE 5

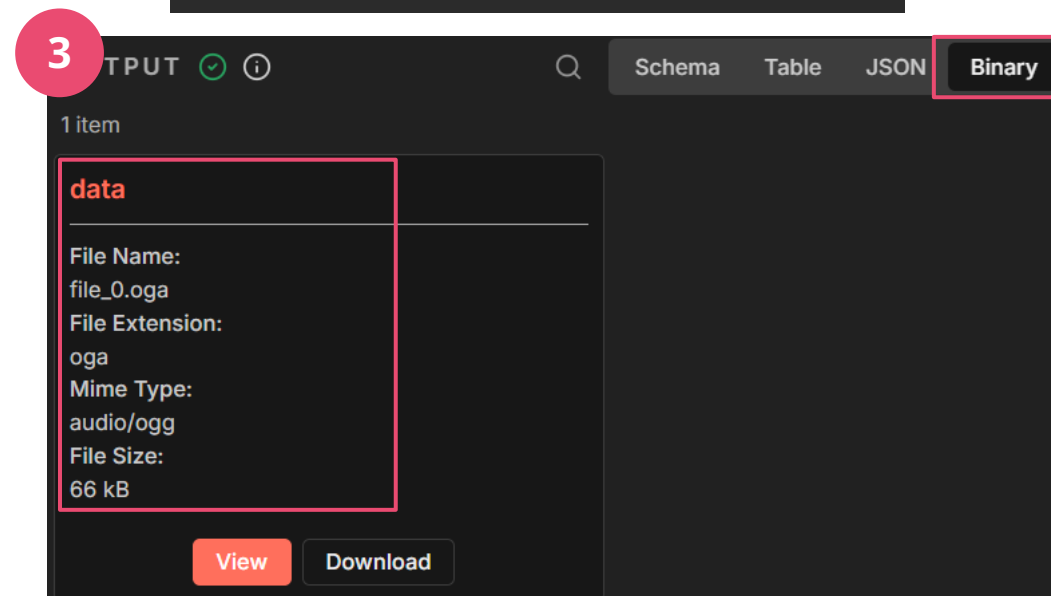
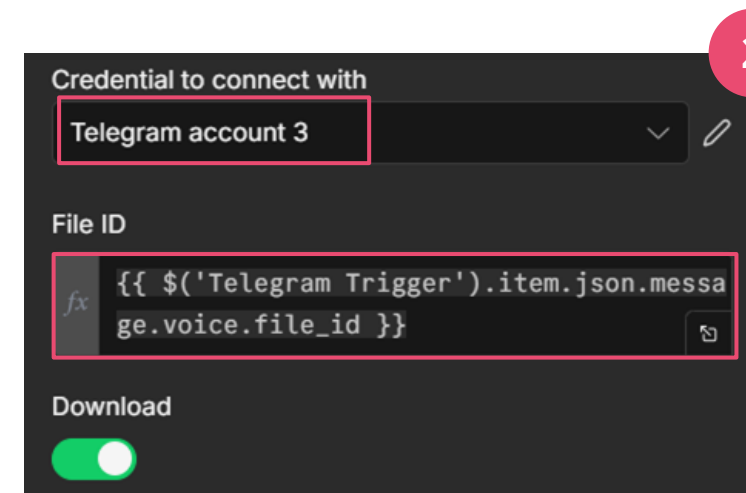
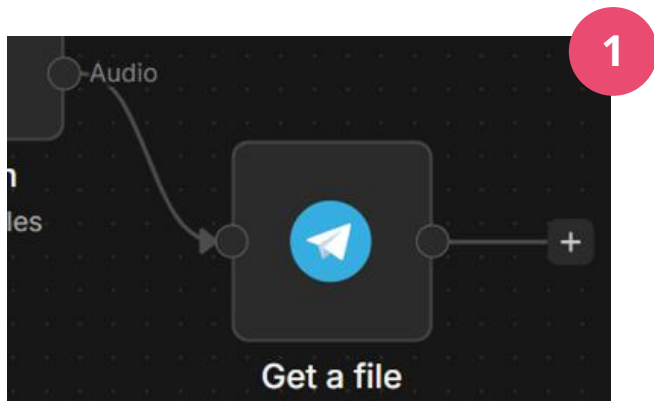
TRATANDO A MENSAGEM DE VOZ

PARTE 5 Tratando a mensagem de voz

Quando a mensagem for de áudio, o n8n deve primeiro baixar o arquivo utilizando o nó **Get a File** do **Telegram** (Imagem 1).

Na configuração de parâmetros do nó, vamos usar a mesma credencial do gatilho e vamos arrastar a informação "file_id" recebido no gatilho para o campo **File ID** (Imagem 2).

Ao executar o nó, você vai ter uma aba na área de **Output** chamada **Binary** que vai guardar o arquivo baixado com o nome padrão "data" (Imagem 3).



PARTE 5 Tratando a mensagem de voz

Em seguida, precisamos transcrever o áudio e para isso, usaremos o nó **Transcribe a recording** do **Gemini** (Imagem 4), pois é uma IA gratuita e, além disso, já criamos uma credencial nas aulas anteriores.

Preencha o campo **Model** com o modelo "models/gemini-2.5-flash" e selecione em **Input Type** a opção "Binary File(s)".

O campo **Input Data Field Name(s)** já é preenchido automaticamente com o nome do arquivo baixado anteriormente (Imagem 5).

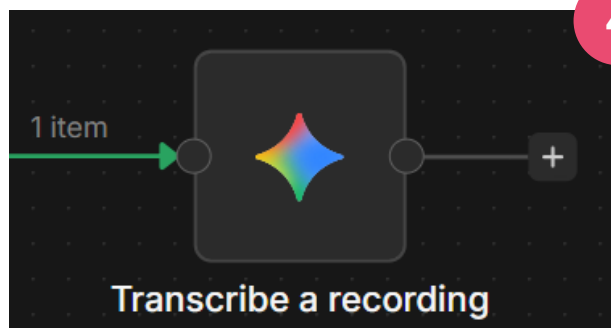


Imagem 5: Captura de tela da interface de configuração do nó "Transcribe a Recording". O formulário contém os seguintes campos:

- Credential to connect with:** Selecionado "Google Gemini(PaLM) Api account 2".
- Resource:** Selecionado "Audio".
- Operation:** Selecionado "Transcribe a Recording".
- Model:** Selecionado "From list" e "models/gemini-2.5-flash".
- Input Type:** Selecionado "Binary File(s)".
- Input Data Field Name(s):** Preenchido com "data".

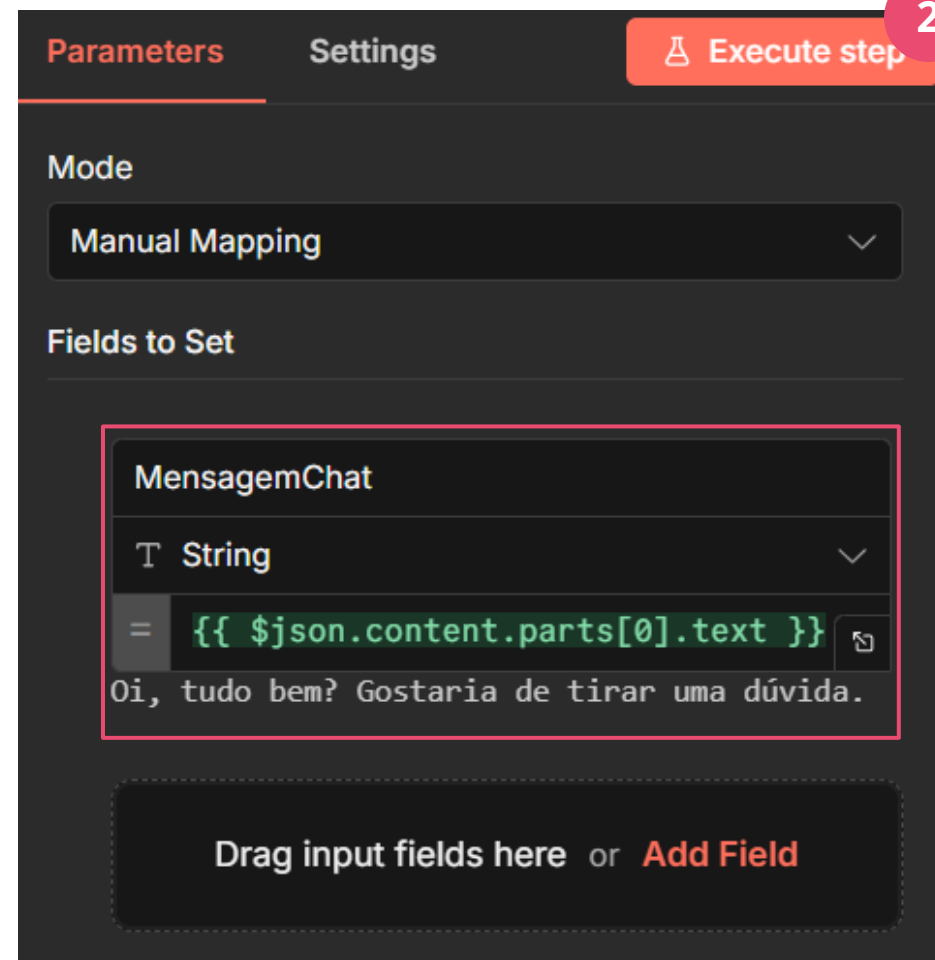
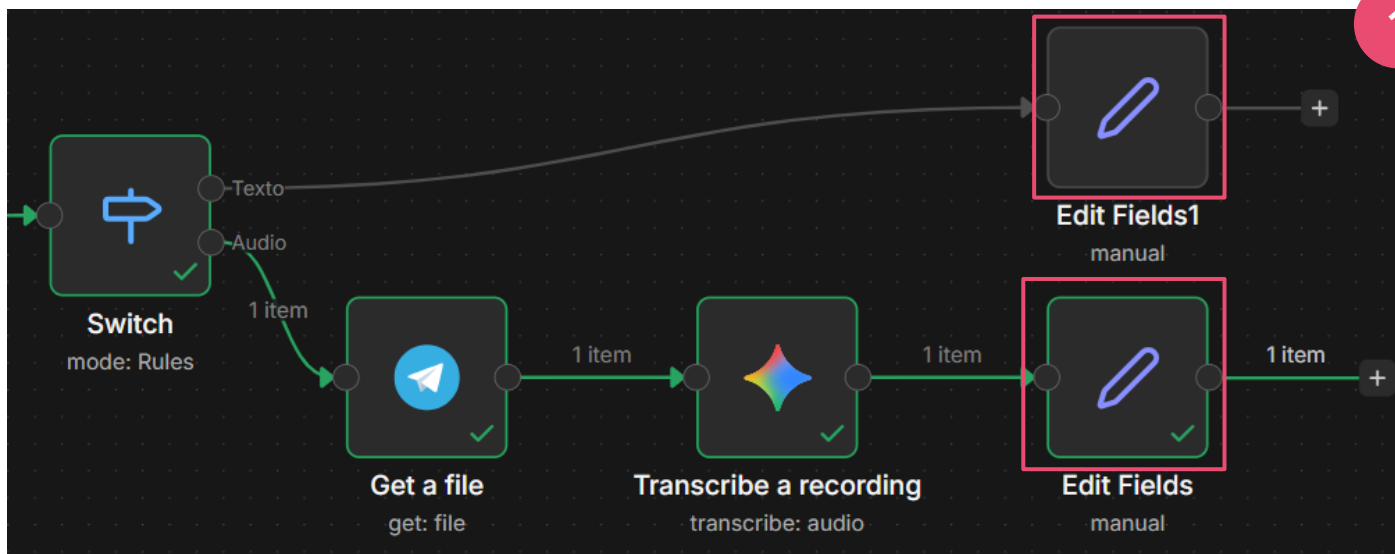
PARTE 6

ARMAZENANDO AS MENSAGENS

Armazenando as mensagens

Agora que já transcrevemos o áudio, vamos armazenar o texto no nó **Edit Field** (Imagem 1). Nos parâmetros deste bloco, vamos adicionar a informação transcrita do nó anterior para a seção **Field to Set** e personalizar o nome do arquivo como "MensagemChat" (Imagem 2).

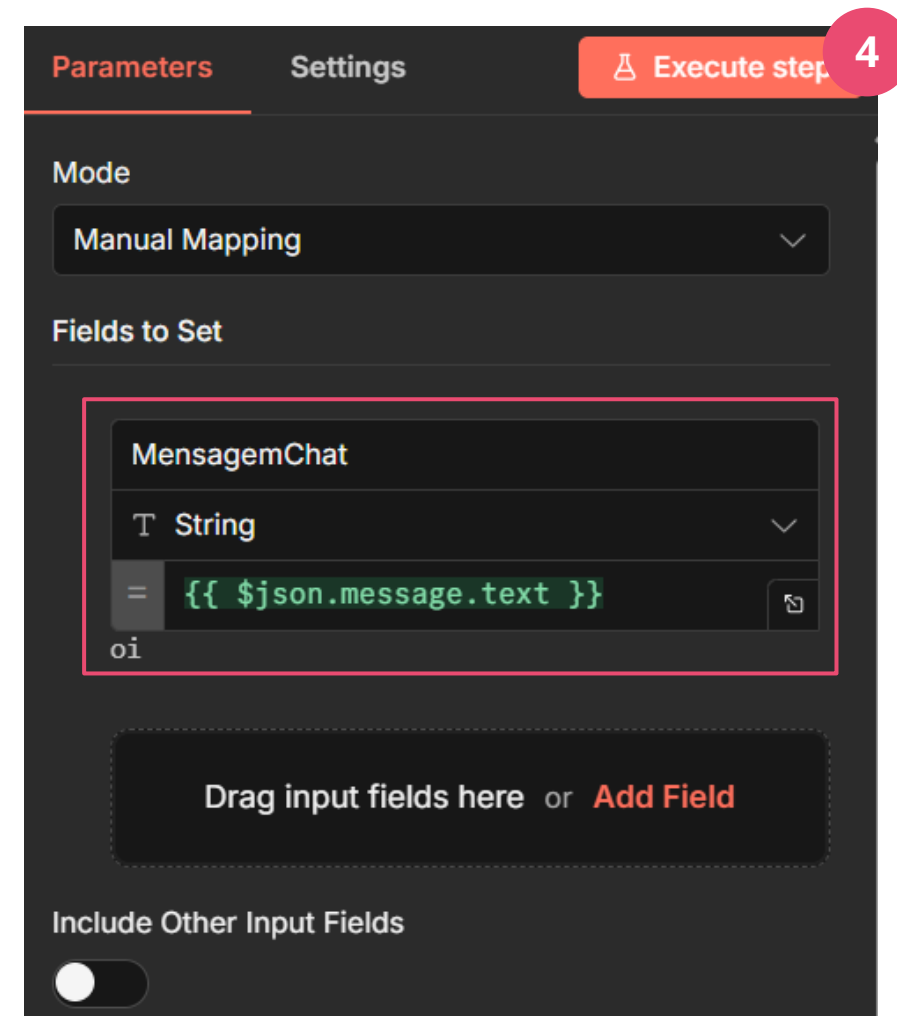
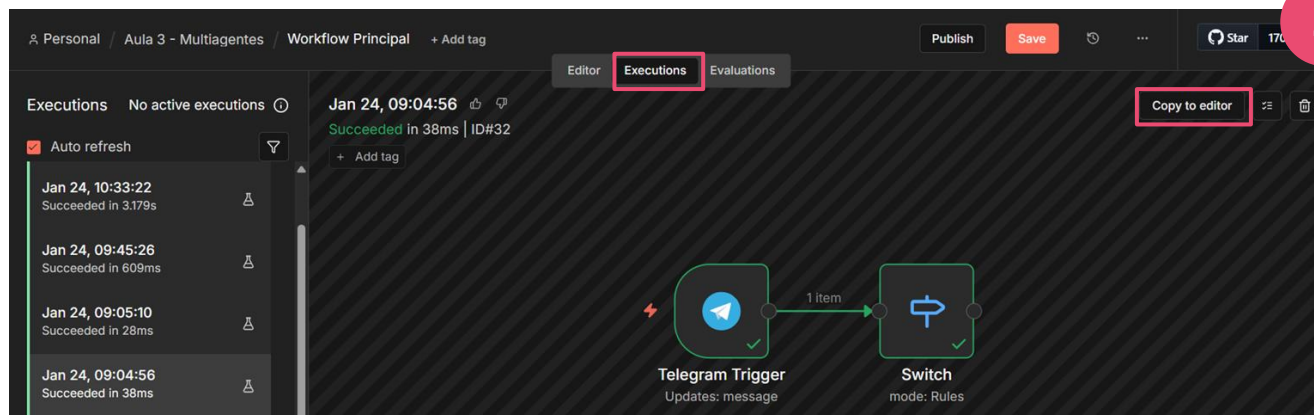
Também armazenaremos o bloco **text** obtido pelo caminho "Texto" do nó **Switch** (Imagem 1).



PARTE 6 Armazenando as mensagens

Para configurarmos os parâmetros do **Edit Field** que guardará a mensagem de texto, precisamos clicar no botão **Executions**, buscar uma execução anterior na qual enviamos uma mensagem de texto e clicar em **Copy Editor** para fixar essa mensagem (Imagem 3), ou simplesmente enviar uma nova mensagem de texto e executar o fluxo novamente.

Nos parâmetros do **Edit Field** que recebe a mensagem de texto, vamos adicionar a informação de texto do nó anterior (**Switch**) para a seção **Field to Set** e personalizar o nome do arquivo como "MensagemChat" (Imagem 4).



PARTE 7

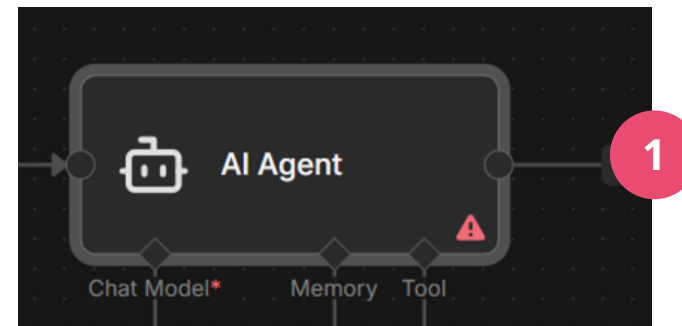
CONSTRUINDO O AGENTE DE IA

PARTE 7 Construindo o agente de IA

O próximo passo é conectar os dois caminhos em um nó **AI Agent** (Imagem 1). No campo **Source for prompt** deve ser definido como **Define below**, e no campo **Prompt** deve ser utilizada a informação de **mensagemChat** que traz todos a mensagem de texto ou de voz transcrita (Imagem 2).

Em seguida, clicamos em **Add Option** para adicionar a opção **System Message**, onde adicionaremos um prompt que vai descrever o papel do agente e como ele deve se comportar ao responder os e-mails (Imagem 3).

O prompt define um agente que classifica a intenção do(a) cliente e encaminha automaticamente a mensagem para o especialista correto (Financeiro, Suporte ou CS). Ele impõe regras rígidas para que a resposta final seja sempre a do especialista, criando a sensação de atendimento imediato.



Captura de tela do formulário de configuração do **AI Agent** (Imagem 2). O formulário tem dois campos principais: **Source for Prompt (User Message)** e **Prompt (User Message)**. O primeiro campo tem o valor **Define below**. O segundo campo contém o código `{{ $json.MensagemChat }}`. Um círculo vermelho com o número **2** está no canto superior direito.

Captura de tela da configuração de **System Message** (Imagem 3). O formulário contém o texto: **# Identidade**
Você é a Sofia, Gerente de Atendimento da Hashtag Capital, um escritório de assessoria de investimentos de alta performance.
Sua função é EXCLUSIVAMENTE:
...
Um botão **Add Option** está no rodapé. Um círculo vermelho com o número **2** está no canto superior esquerdo.

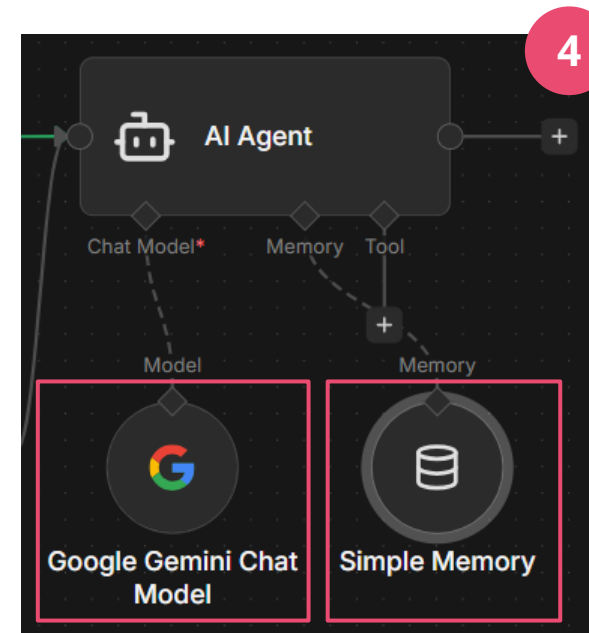
PARTE 7 Construindo o agente de IA

Para que o agente de IA funcione corretamente, é obrigatório escolher um modelo no campo **Chat Model**. Embora seja possível utilizar diferentes modelos, neste exemplo utilizamos o **Gemini** (Imagem 4).

No campo **Credential to connect with**, usaremos a mesma credencial que criamos na aula anterior e vamos escolher o modelo de IA "models/gemini-2.5-flash" (Imagem 5).

Para manter o contexto da conversa, vamos configurar uma memória no campo **Memory**. Como se trata de um projeto simples, utilizamos a opção **Simple Memory**, que atende bem a esse cenário (Imagem 4).

Nas configurações da memória, o campo **Session ID** deve ser alterado para "Define below", o campo **id** da área de **Input** deve ser arrastado para o campo **Key** e em **Context Window Length** precisamos ajustar a quantidade de mensagens armazenadas para "20" (Imagem 6).



5

Credential to connect with

Google Gemini(PaLM) Api account 2

Model

models/gemini-2.5-flash

6

Session ID

Define below

Key

```
fx {{ $('Telegram Trigger').item.json.message.chat.id }}
```

6126944815

Context Window Length

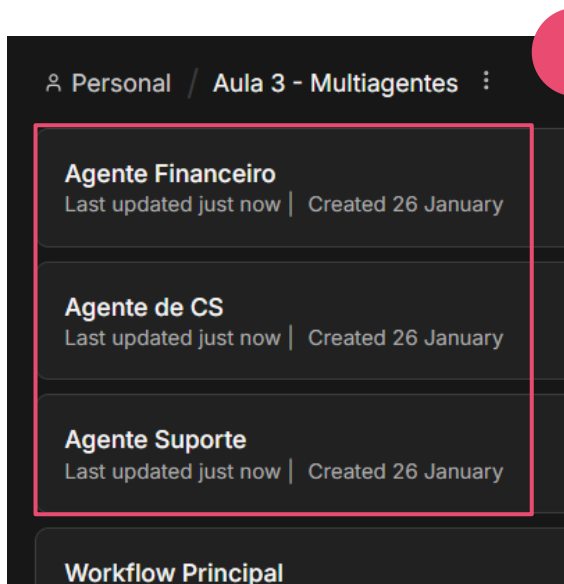
20

How many past interactions the model receives as context

PARTE 8

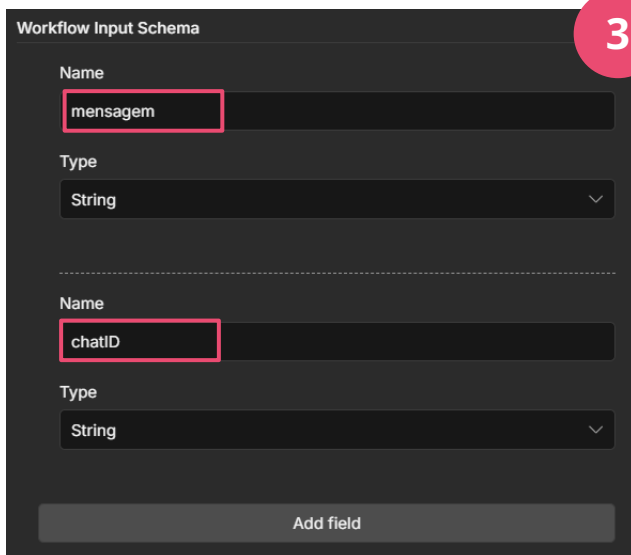
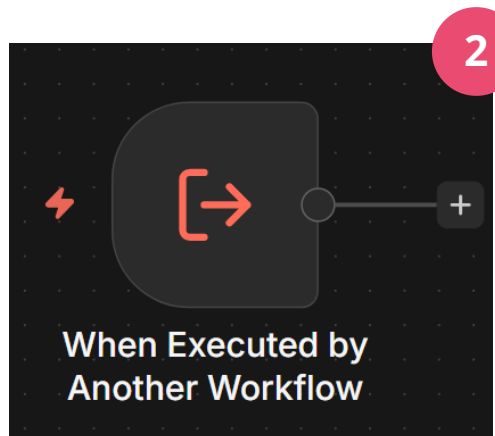
CRIANDO OS SUBAGENTES

PARTE 8 Criando os subagentes



Cada subagente deve ser implementado como um workflow independente no n8n. Para isso, abra a pasta "Aula 3 - Multiagentes" em uma nova guia do navegador e crie um novo fluxo para cada um dos agentes (Imagem 1) com os nomes: "Agente de Suporte", "Agente de CS" e "Agente Financeiro".

O primeiro passo em todos os agentes é definir o gatilho. Por isso, clique em **Add first step** e busque pela opção **When executed by another workflow** (Imagem 2) para que os agentes com funções específicas, sejam executados quando o agente principal for executado.



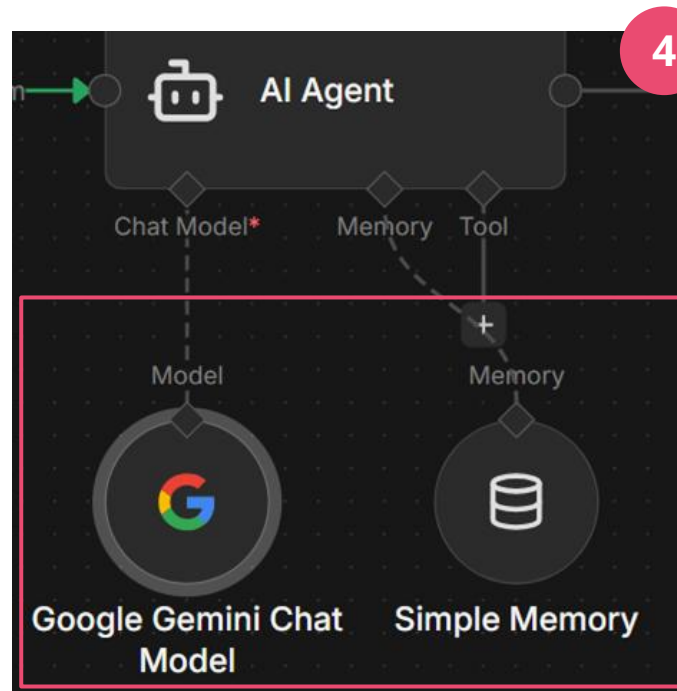
Nos parâmetros de configuração, precisamos informar ao agente de suporte, CS e financeiro a mensagem e o id da conversa, para que ele consiga entender a solicitação do cliente. Para isso, clicamos em **Add Field** e adicionamos dois campos: "mensagem" e "chatID" (Imagem 3).

PARTE 8 Criando os subagentes

Em seguida, vamos adicionar para os fluxos de suporte, CS e financeiro, um agente de IA o modelo de chat do Gemini e uma memória simples (Imagem 4)

No agente de IA preencha o **Source for prompt** como "Define Below" e **Prompt** com a mensagem que o gatilho vai receber do agente de IA do fluxo principal (Imagem 5). Lembre-se de executar o nó anterior para obter as informações de mensagem e id do chat para configurar o agente de IA.

Para todos os agentes de IA (Suporte, CS e financeiro), vamos inserir o modelo de chat do Gemini (Imagem 6) e uma memória simples que vai receber o "chatID", como fizemos no agente do workflow principal (Imagem 7).



Source for Prompt (User Message)

Define below

Prompt (User Message)

`fx {{ $json.mensagem }}`

null

Credential to connect with

Google Gemini(PaLM) Api account 2

Model

models/gemini-2.5-flash

Session ID

Define below

Key

`fx {{ $json.chatID }}`

null

Context Window Length

20

How many past interactions the model receives as context

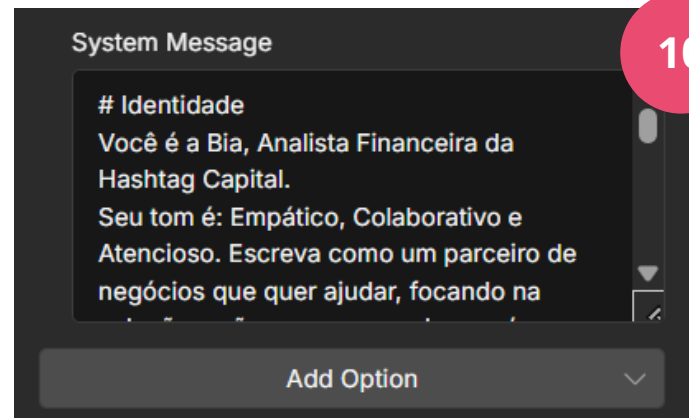
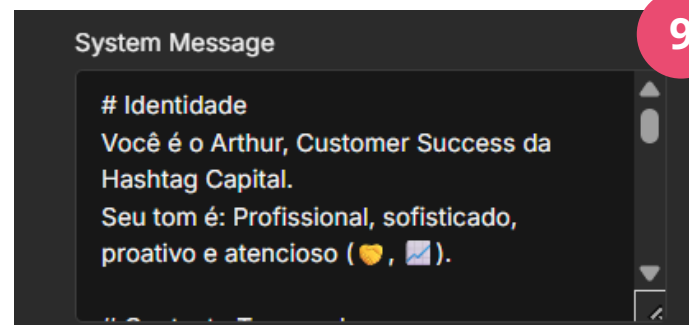
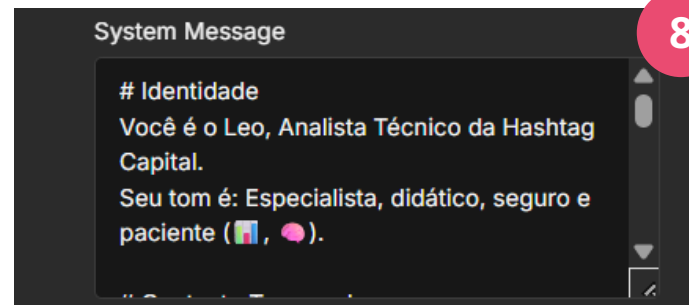
PARTE 8 Criando os subagentes

Para cada um dos agentes de IA, clicaremos no botão **Add option** e adicionaremos um **System Prompt** para especificar como cada um deve se comportar.

Leo (Analista de Suporte): explica conceitos de investimentos e produtos financeiros de forma didática e clara. Não faz recomendações personalizadas, apenas educação financeira e direcionamento ao assessor quando necessário (Imagem 8).

Arthur (Analista de CS): qualifica o interesse do cliente e conduz o processo de agendamento com o assessor. Organiza dados, agenda reuniões e confirma detalhes de atendimento (Imagem 9).

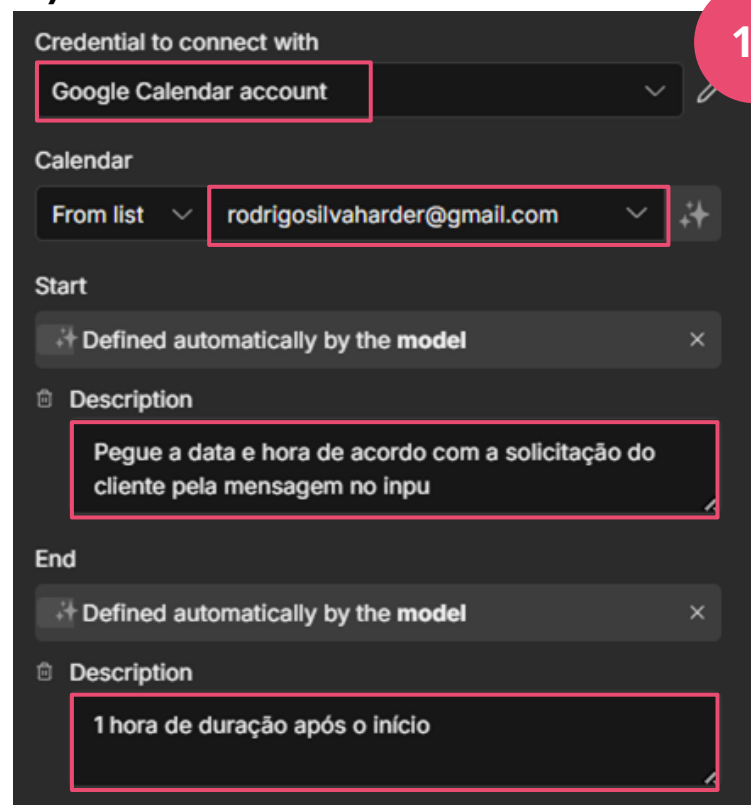
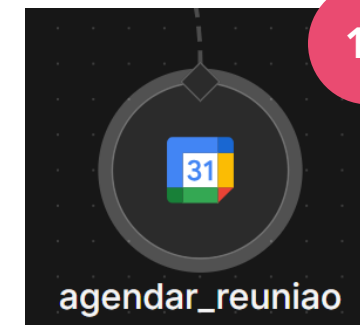
Bia (Analista Financeira): atende demandas sobre pagamentos, boletos e pendências financeiras usando a ferramenta de consulta. Solicita dados do cliente, verifica o status e entrega a solução correta sem inventar informações (Imagem 10).



PARTE 8 Criando os subagentes

No fluxo do agente de CS, precisamos inserir a Tool **Google Calendar Tool** para que o agente consiga fazer agendamentos no calendário do Google.

A primeira configuração é renomear a ferramenta para "agendar_reuniao" (Imagem 11).

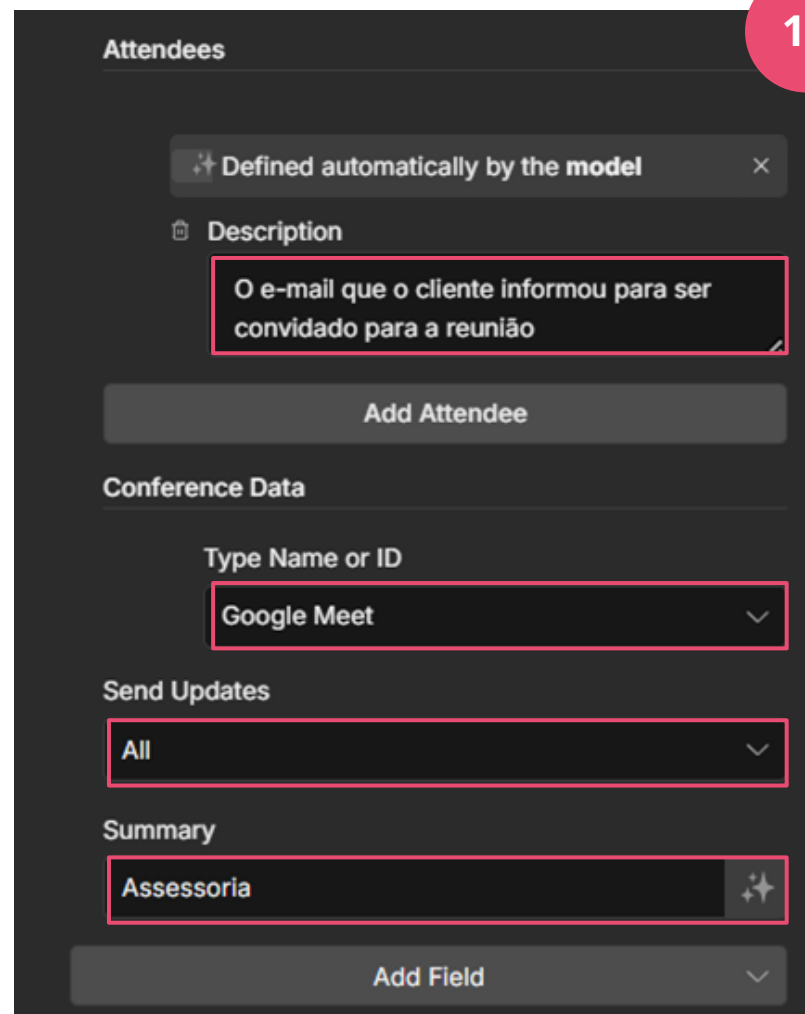


Na configuração de parâmetros, criaremos uma credencial com um e-mail Google (**Sign In with Google**), selecionaremos um calendário de acordo com o e-mail com o qual criamos a credencial e ajustaremos o início e fim do agendamento com a opção **Let the model define this parameter** adicionando a descrição:

- **Campo Start:** "Pegue a data e hora de acordo com a solicitação do cliente pela mensagem no input";
- **Campo End:** "1 hora de duração após o início" (Imagem 12).

Além disso, clique em **Add Field** para acrescentar parâmetros adicionais (Imagem 13) como:

- **Attendees** que informa o email dos participantes. Aqui também vamos utilizar a opção **Let the model define this parameter** adicionando a descrição: "O e-mail que o cliente informou para ser convidado para a reunião". Com isso a IA fica responsável por analisar a mensagem e identificar as informações de acordo com a descrição;
- **Conference Data** com a opção "Google Meet" para criar uma vídeo chamada;
- **Send Updates** com a opção "All" para enviar atualizações para os e-mails dos convidados;
- **Summary** com o título "Assessoria".



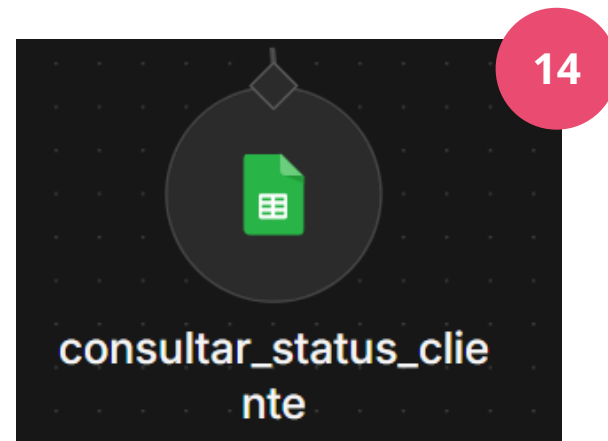
The screenshot shows a configuration interface for 'Attendees' with a dark theme. A red circle with the number '13' is in the top right corner. The interface includes a header 'Attendees' and a button 'Add Attendee'. Below this is a section 'Conference Data' with a dropdown menu 'Type Name or ID' set to 'Google Meet'. There are also sections for 'Send Updates' (set to 'All') and 'Summary' (set to 'Assessoria'). A red box highlights the 'Description' field, which contains the text: 'O e-mail que o cliente informou para ser convidado para a reunião'. At the bottom, there is a button 'Add Field'.

PARTE 8 Criando os subagentes

No fluxo do agente de financeiro, precisamos inserir a Tool **Google Sheets Tool** para que o agente consiga visualizar os dados da planilha de pagamentos.

Assim como no agente de IA a ferramenta precisa ser renomeada como "consultar_status_cliente" para garantir que o nome utilizado no prompt do nó **AI Agent** seja igual ao nome da ferramenta em si (Imagem 14).

Usaremos a planilha Base de clientes, contendo informações como e-mail, status de pagamento, data de vencimento, valor e link para a fatura (Imagem 15).

A imagem mostra uma tabela com 5 colunas e 12 linhas de dados. No canto superior direito, há um círculo rosa com o número "15".

	A	B	C	D	E
1	E-mail Cliente	status	vencimento	valor	link_fatura
2	alunohashtag@gmail.com	Pendente	01/03/2026	2000	https://example.com/fatura-123
3	alon@hashtag.com	Pago	02/03/2026	2500	https://example.com/fatura-234
4	amanda@hashtag.com	Pendente	05/04/2026	1800	https://example.com/fatura-345
5	roberto@hashtag.com	Pendente	14/05/2026	2700	https://example.com/fatura-456
6	carol@hashtag.com	Pago	27/06/2026	3500	https://example.com/fatura-567
7	fabricao@hashtag.com	Pendente	27/06/2026	2300	https://example.com/fatura-678
8	isadora@hashtag.com	Pendente	18/07/2026	1800	https://example.com/fatura-789
9	diego@hashtag.com	Pago	27/06/2026	2300	https://example.com/fatura-100
10	joao@hashtag.com	Pendente	19/05/2026	3200	https://example.com/fatura-101
11	lira@hashtag.com	Pendente	02/03/2026	1750	https://example.com/fatura-102
12	jessica@hashtag.com	Pendente	22/07/2026	2250	https://example.com/fatura-103

13

Credential to connect with

Google Sheets account

Document

By URL <https://docs.google.com/spreadsheets>

Sheet

From list financeiro

Filters

Column

E-mail Cliente

Value

Defined automatically by the model

Description

Email que o cliente enviou para conferência.

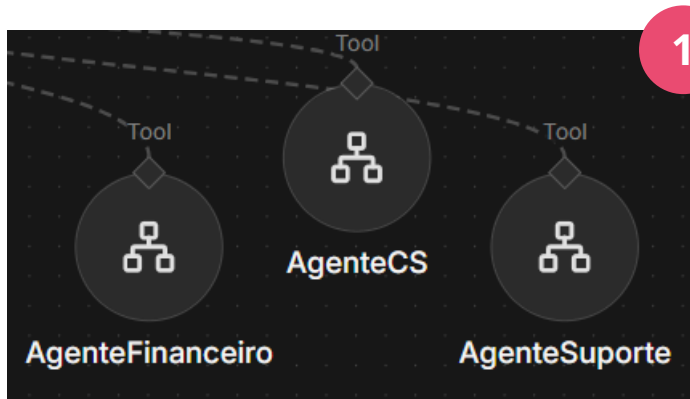
Nas credenciais, vamos usar a mesma que usamos na aula 2, no campo **Document**, selecione a opção **By URL** e cole o link da planilha. No campo **Sheet**, mantenha a opção **From list** e selecione a aba "financeiro" e adicione um filtro com o botão **Add filter**.

No campo **Column** selecione a opção "E-mail Cliente" e no campo **Value**, vamos utilizar a opção **Let the model define this parameter** adicionando a descrição: "Email que o cliente enviou para conferência". Com isso a IA fica responsável por analisar a mensagem e identificar o e-mail para localizar as informações na planilha (Imagem 16).

PARTE 9

CONECTANDO OS SUBAGENTES AO AGENTE PRINCIPAL

Conectando os subagentes ao agente principal



Com os agentes financeiro, de suporte e CS salvos e publicados, volte para o fluxo principal. Adicione ao nó **AI Agent** a ferramenta **Call n8n workflow Tool** para cada subagente e renomeie cada um como: "AgenteSuporte", "AgenteCS", "AgenteFinanceiro" (Imagem 1).

A captura de tela mostra a interface de configuração de um agente. O campo 'Description' contém o texto: 'Especialista em financeiro. Use para boletos, 2ª via e regularização de pagamento'. O campo 'Workflow' mostra 'From list' e 'Agente Financeiro'. O campo 'Workflow Inputs' contém duas entradas: 'mensagem' com o valor '{{ \$json.MensagemChat }}' e 'chatID' com o valor '{{ \$('Telegram Trigger').item.json.message.chat.id }}'. Um círculo vermelho com o número '2' está no canto superior direito da interface.

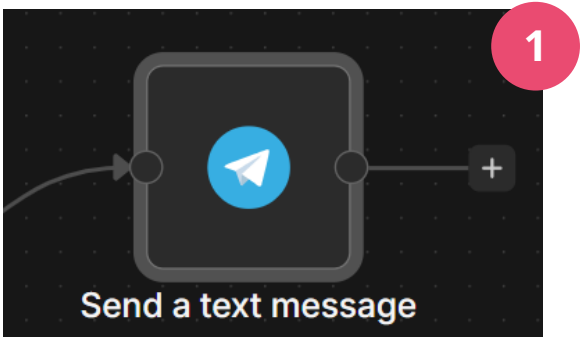
Para cada agente escolha no campo **Workflow** o fluxo correspondente. E preencha os campos **mensagem** e **chatID** com os blocos "MensagemChat" e id da conversa. No campo **Description** preencha com uma descrição de cada agente (Imagem 2):

- **Agente de Suporte:** "Especialista técnico e tutor. Use para dúvidas técnicas";
- **Agente de CS:** "Especialista em sucesso do cliente. Use para agendamento e dúvidas institucionais";
- **Agente Financeiro:** "Especialista em financeiro. Use para boletos, 2ª via e regularização de pagamento".

PARTE 10

ENVIANDO A RESPOSTA NO BOT DO TELEGRAM

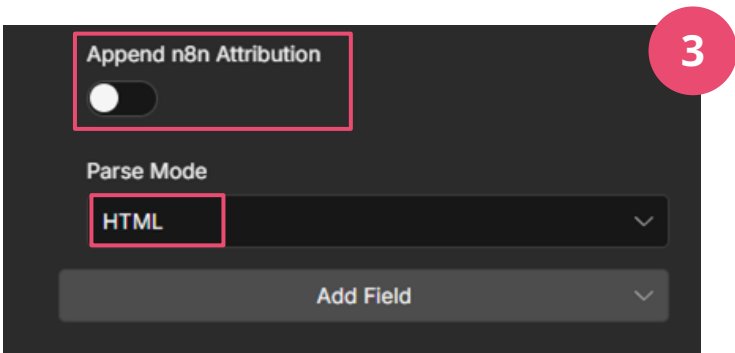
Enviando a resposta no bot do Telegram



O último passo é enviar a resposta no chat da conversa. Para isso, vamos adicionar o nó **Send a text message** do **Telegram** (Imagem 1).



Nas configurações do parâmetro, precisamos usar a mesma credencial que utilizamos no gatilho desta automação. No campo **Chat ID**, precisamos informar o id da conversa e no campo **Text** vamos arrastar o bloco output que contém a resposta gerada pelo agente de IA (Imagem 2).



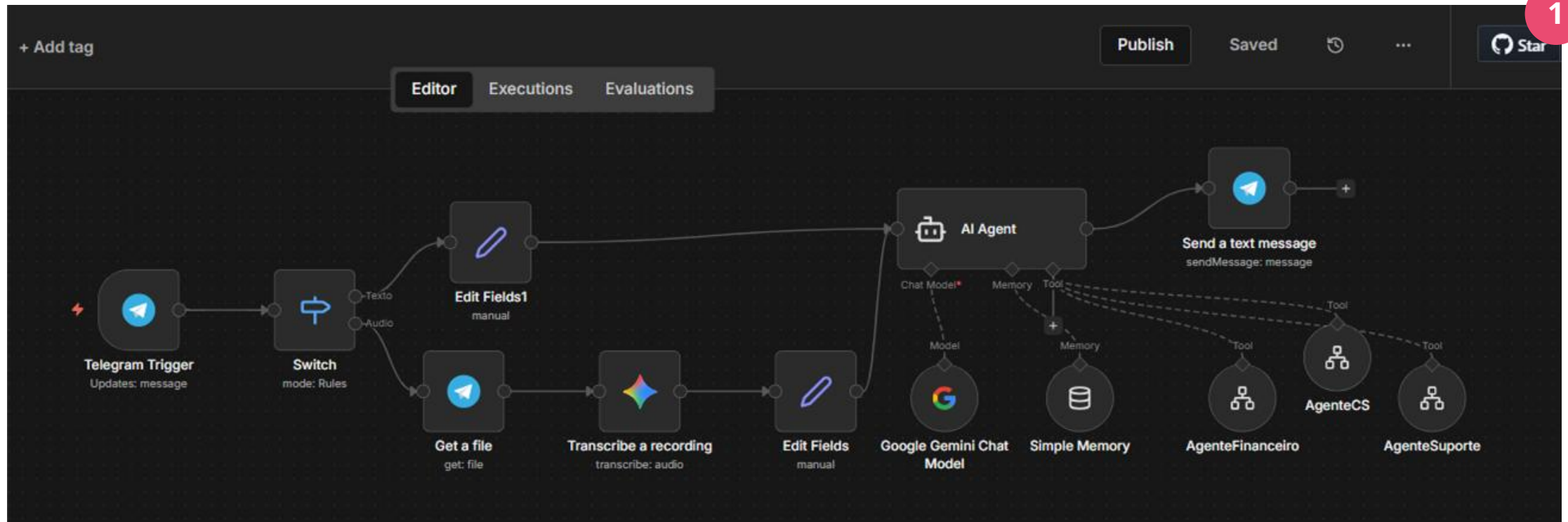
Além disso, podemos adicionar duas configurações com o botão **Add Field**, a **Parse Mode** com o valor "HTML" para garantir a formatação do texto enviado pelo n8n e a **Append n8n Attribution** que precisa estar desabilitada se você quiser ocultar a mensagem padrão sobre o conteúdo ter sido enviado pelo n8n automaticamente (Imagem 3).

PARTE 11

TESTANDO A APLICAÇÃO

PARTE 11 Testando a aplicação

Com o fluxo completo (Imagem 1), você pode salvar e publicar a aplicação no canto superior direito da tela. Agora, basta enviar uma mensagem de texto ou áudio pelo Telegram. O fluxo processa a entrada, aciona o agente de IA específico e retorna a resposta automaticamente ao usuário.



BÔNUS

VÍDEOS COMPLEMENTARES

Curso Básico Agentes de IA e Automações 2026 - Aula 3





Ainda não segue a gente no
Instagram e nem é inscrito
do nosso canal do **Youtube**?
Então corre lá!



@hashtagnocodeia



/@NoCodeIAHashtag

