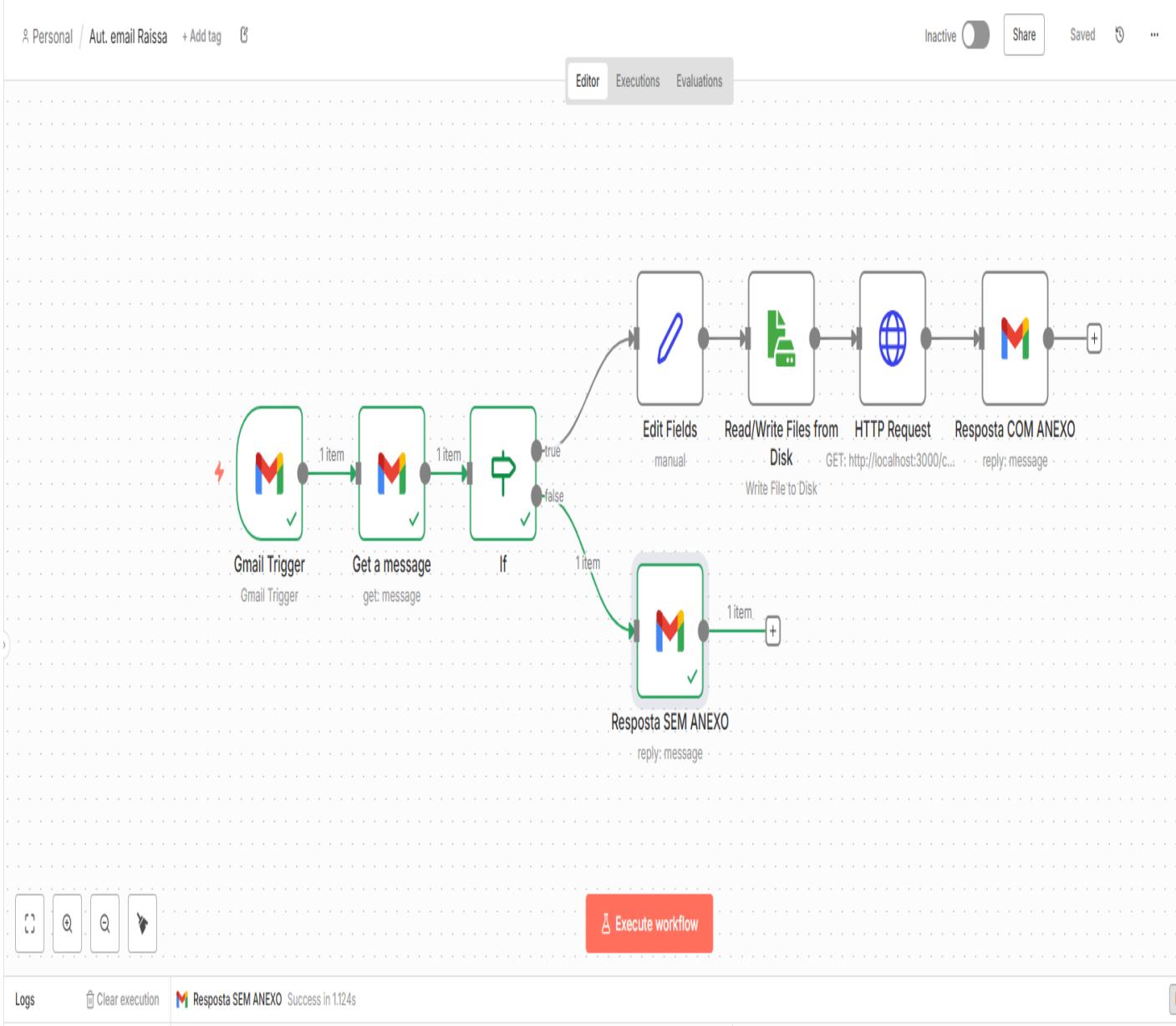




Teste Técnico – Automação no n8n

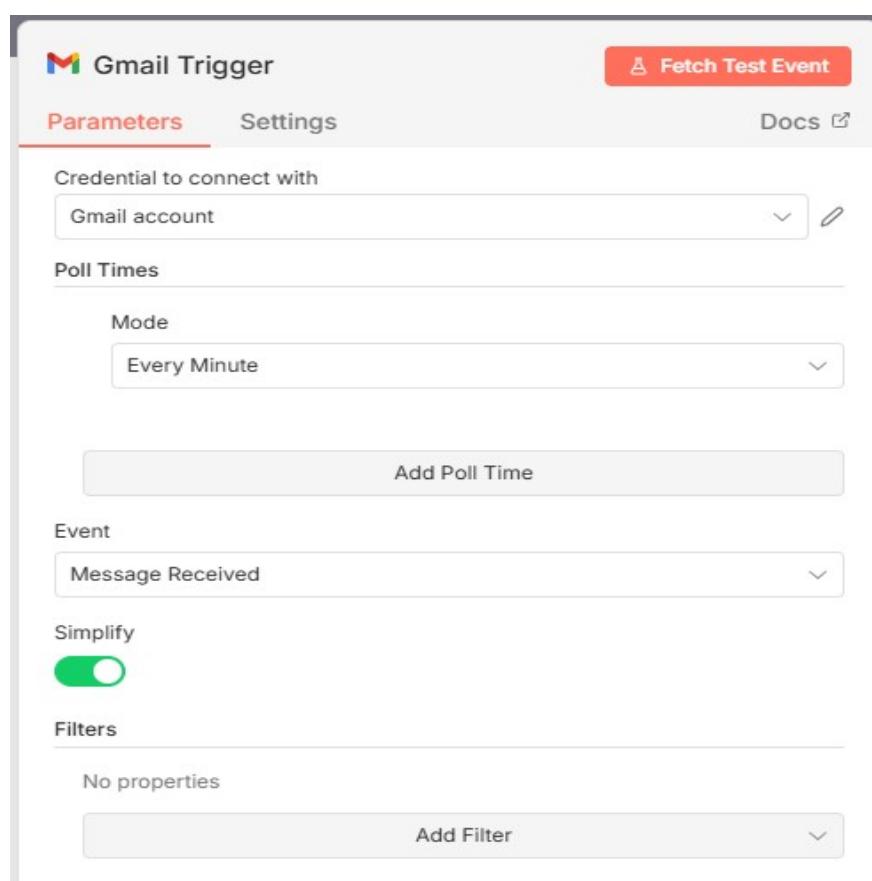
Raíssa

n8n – Fluxograma geral



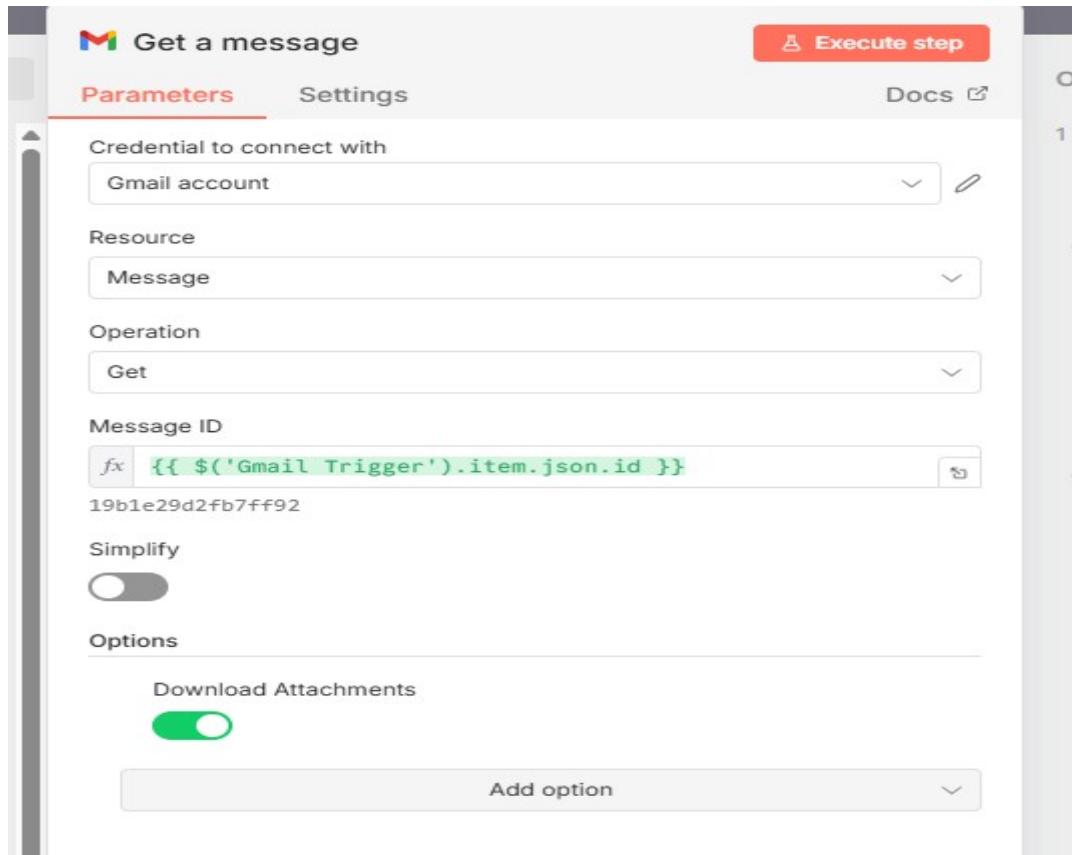
Primeiro node Gmail Trigger:

Foi configurado com a conta Gmail (é o e-mail que gera gatilho para a automação rodar ao receber, coloquei um e-mail de teste que criei, e realizei as autenticações necessárias do google, como client id, secret etc. Essa credencial pode ser ajustada e mudar para outro e-mail, que será o e-mail origem, dentro de Credential to connect with, onde todas as mensagens que chegam nele serão automatizados. Foi colocada para rodar a cada minuto, porém isso fica aberto de acordo com a necessidade. No teste vi que é um e-mail por dia que recebe, logo, não precisaria ser todo minuto.



Segundo node Get a message:

Esse node foi criado para identificar se o e-mail tem anexo ou não, habilitar a opção de baixar, e para fazer a conversão do binário e consequentemente conseguir usar na cláusula da lógica.



Terceiro node IF:

Esse node é onde fica a lógica de que se o e-mail tem anexo, faz X ação e caso não tenha, faça Y ação. Então se o anexo é diferente de indefinido (não existe) ele é verdadeiro, logo existe anexo e vai para a regra do TRUE.

The screenshot shows the configuration interface for an 'If' node. At the top, there are tabs for 'Parameters' (which is selected), 'Settings', and 'Docs'. A red 'Execute step' button is located in the top right corner. The main area is titled 'Conditions' and contains the following configuration:

- A condition input field with the expression `fx {{ $binary.attachment_0 !== undefined }}`. To its right is a checkbox labeled 'is true' with a checked status.
- An 'AND' dropdown menu.
- A second condition input field with 'value1' and 'value2' fields, and a dropdown menu showing 'T is equal to'.
- A 'Convert types where required' toggle switch, which is turned on (green).
- An 'Options' section with a note 'No properties' and a 'Add option' button.

Quarto node Edit Fields (Se for TRUE, com anexo):

Esse node é para fazer o tratamento da planilha, editando para trazer o nome dela ao salvar o anexo, para não ficar um nome que não é o do arquivo. O fileName do próprio.

Edit Fields

[Execute step](#)

[Parameters](#) [Settings](#) [Docs](#)

Mode: Manual Mapping

Fields to Set:

filename	Type: String
= {{ \$binary.attachment_0.fileName }}	Edit

Drag input fields here or [Add Field](#)

Include Other Input Fields: [Toggle](#)

Input Fields to Include: All

Options: No properties

Add option

Quinto node Read/Write Files from disk:

Esse node foi usado para salvar LOCALMENTE o anexo processado, definindo um destino (podendo editar, caso tenha necessidade), no caso, foi criado uma pasta no [C:/](#).

Read/Write Files from Disk

[Execute step](#)

[Parameters](#) [Settings](#) [Docs](#)

Use this node to read and write files on the same computer running n8n. To handle files between different computers please use other nodes (e.g. FTP, HTTP Request, AWS).

Operation

Write File to Disk

File Path and Name

fx C:/n8n/csv/{{\$json.filename}}

Input Binary Field

attachment_0

The name of the input binary field containing the file to be written

Options

No properties

Add option

Sexto node HTTP request:

Esse node foi criado com o intuito de consumir a API, para trazer a cotação USD e BRL em tempo real e usar na resposta do e-mail.

HTTP Request

Execute step

Parameters Settings Docs

Import cURL

Method: GET

URL: http://localhost:3000/cotacao_raissa/usd-brl

Authentication: None

Send Query Parameters:

Send Headers:

Send Body:

Options: No properties

Sétimo node Gmail resposta:

Esse node foi criado para responder de forma automática, quando o e-mail tem o anexo csv. Nesse caso, assim como o e-mail de origem pode ser alterado, o e-mail de resposta também pode, na configuração de credencias (eu usei o mesmo de teste de origem para o destino recebimento/resposta).

Obs.: envio um vídeo em anexo para ver a automação funcionando

The screenshot shows the configuration interface for the 'Resposta COM ANEXO' node. At the top right is a red 'Execute step' button. Below it, there are three tabs: 'Parameters' (which is selected), 'Settings', and 'Docs'. The 'Parameters' tab contains several dropdown menus and input fields:

- Credential to connect with:** Set to 'Gmail account'.
- Resource:** Set to 'Message'.
- Operation:** Set to 'Reply'.
- Message ID:** An input field containing the expression `fx {{ $('Gmail Trigger').item.json.id }}`.
- Email Type:** Set to 'Text'.
- Message:** A large text area containing three messages:
 - 'Olá!'
 - 'Espero que esteja bem.'
 - 'Seu anexo foi recebido e o processo foi concluído.'
- Options:** A section at the bottom with a single input field.

Quarto (ou oitavo) node para caso a regra do IF seja FALSA:

Foi criado para dar uma resposta ao e-mail informado que o mesmo está sem anexo csv.

Resposta SEM ANEXO

Parameters **Settings** **Docs**

Credential to connect with: Gmail account

Resource: Message

Operation: Reply

Message ID: `fx {{ $('Gmail Trigger').item.json.id }}`

Email Type: Text

Message: Olá! O e-mail foi processado, porém informamos que não há nenhum ar

Options: No properties

Add option

Print de requisição da API no Postman, sem parameters, body e autenticação passados, apenas uma API Rest simples:

GET http://localhost:3000/cotacao_raissa/usd-brl

Params

Key	Value
Key	Value

Body

```
{ "origem": "USD", "baseUSD": 1, "destino": "BRL", "valorBRL": 5.402488, "description": "1 USD = 5.402488 BRL", "fonte": "open-er-api" }
```

Obs1.: envio dentro do repositório o vídeo da automação em funcionamento, além desse material em PDF, export do workflow n8n em json.

Obs2.: Para mim, ficou subjetivo o entendimento se era para consumir a API da cotação no open er ou se era para criar uma API consumindo ela (terceirizando), dito isso, eu criei uma API Rest simples com NestJS direcionando para a API do open er.

```
{"origem": "USD", "baseUSD": 1, "destino": "BRL", "valorBRL": 5.402488, "description": "1 USD = 5.402488 BRL", "fonte": "open-er-api"}
```