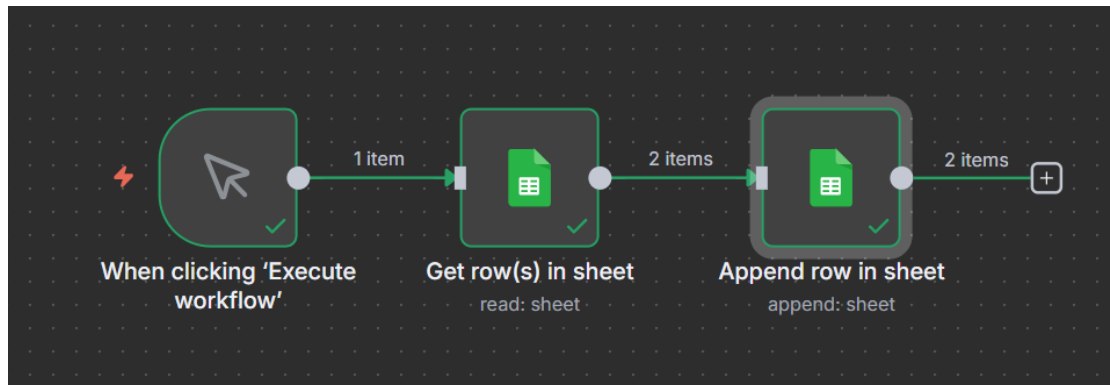


Automações com N8N

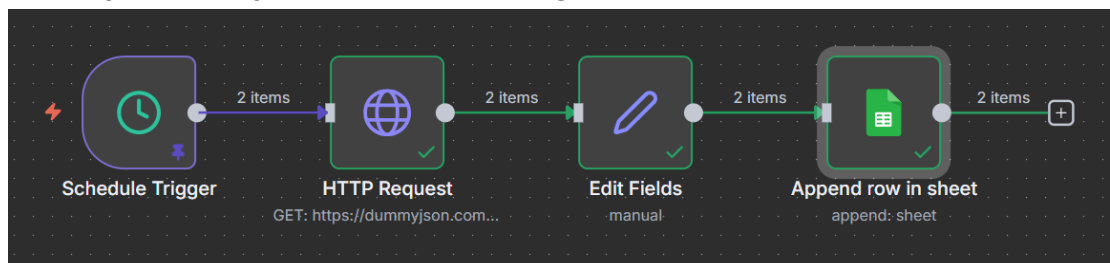
Criando primeiro fluxo

- Criei primeiramente uma automação simples de leitura e escrita em arquivos **Google Sheets**, para conhecer a interface da aplicação;



Lendo arquivos CSV automaticamente de uma pasta no OneDrive_SharePoint

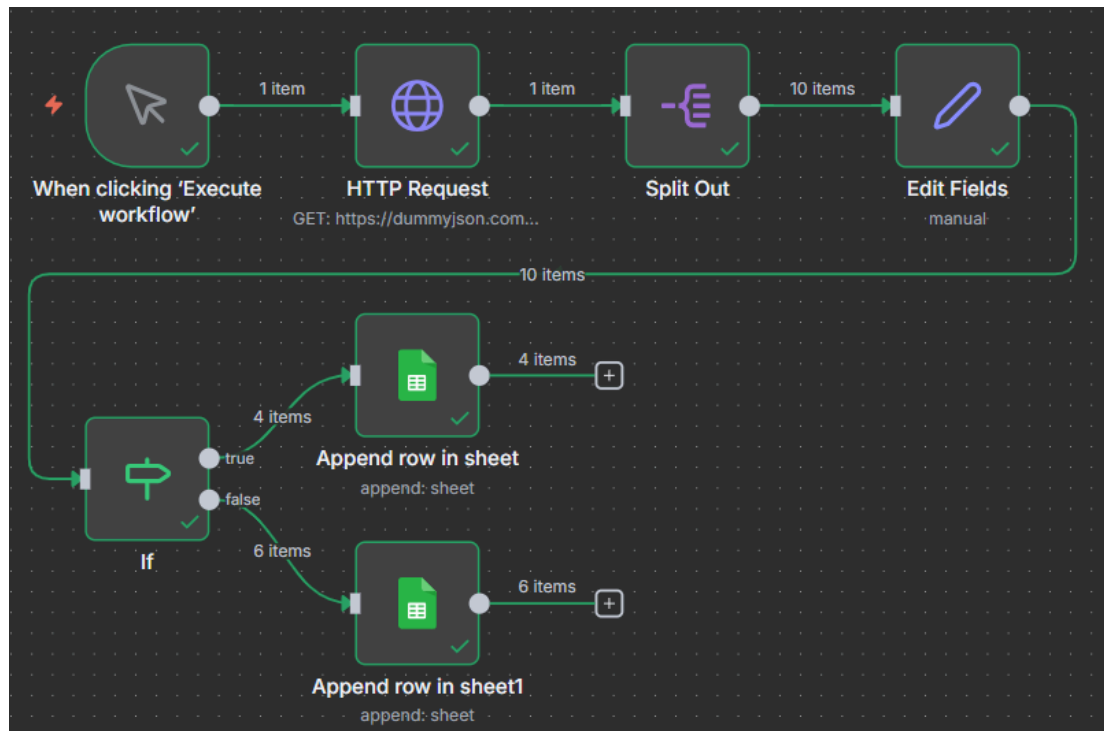
- Criei um fluxo que pega informações de uma API pública via **HTTP** método GET, em um horário específico utilizando o **Schedule Trigger** e configurando diariamente à meia noite, filtrar quais colunas quero pelo **Edit Fields** e por fim adicionar as informações na Página DadosAPI do **Google Sheet**;



Próximos passos abaixo

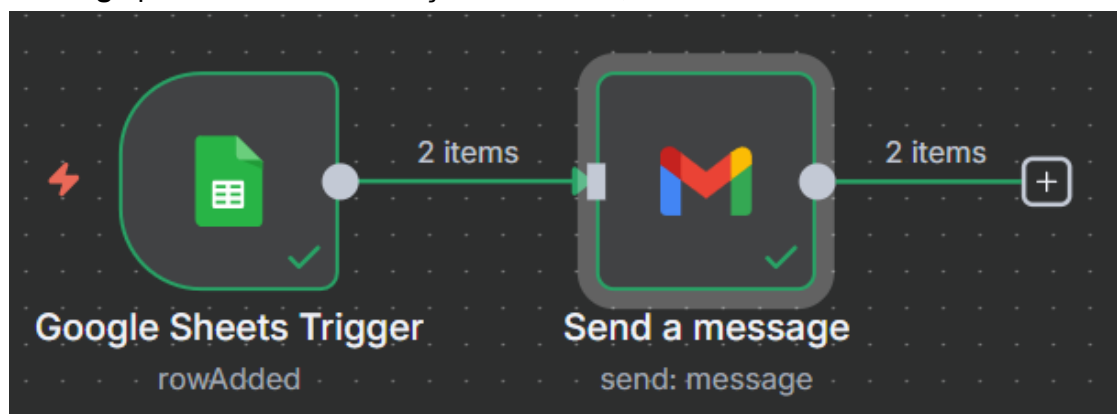
Salvando os dados tratados em um repositório

- Agora recebemos a missão de criar um fluxo que monitore se os produtos estão sendo vendidos pelo preço certo durante uma campanha de marketing. Começo criando um acionamento manual, em seguida crio um request **HTTP** da API, utilizo o **Split Out** para transformar uma lista em itens separados, em seguida utilizo o **Edit Fields**, adiciono um condicional **IF** para itens com preço acima de 50, em seguida criamos uma página de ProdutosVIP para armazenar os produtos com preço acima de 50 e uma página ProdutosBaratos para armazenar os produtos com preço abaixo de 50;



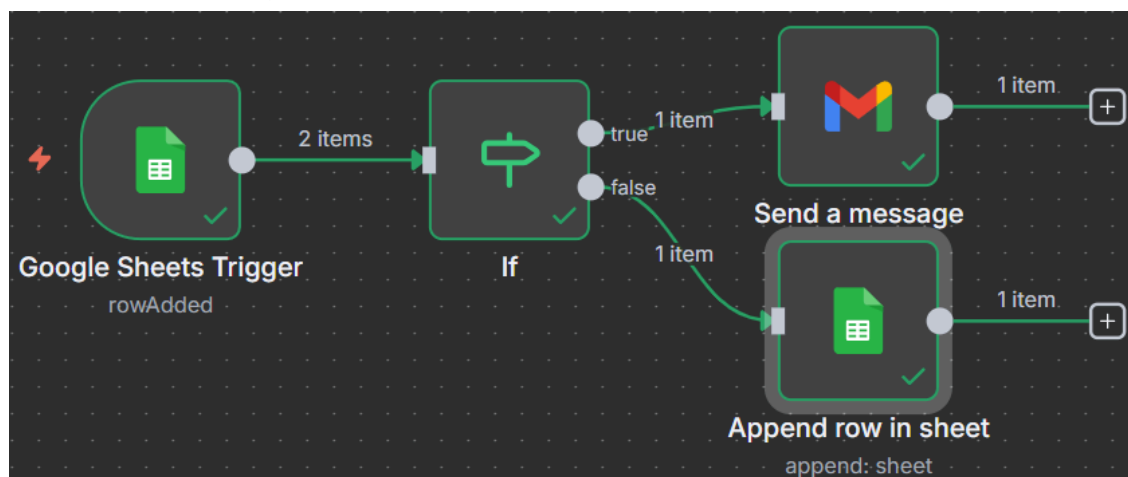
Monitorando planilhas no Google Sheets

- Agora recebemos uma missão do RH de controlar as faltas de funcionários por meio de uma planilha **Google Sheets** que será atualizada automaticamente, e quando uma nova linha for adicionada o gerente será notificado. Então começo adicionando um **Google Sheets Trigger** que irá ser ativado quando uma nova linha for adicionada a página FaltasFuncionarios, em seguida adiciono o **Gmail Send a Message** para enviar as notificações via e-mail;



Detectando mensagens críticas (ex nota 3) e tomando decisão condicional

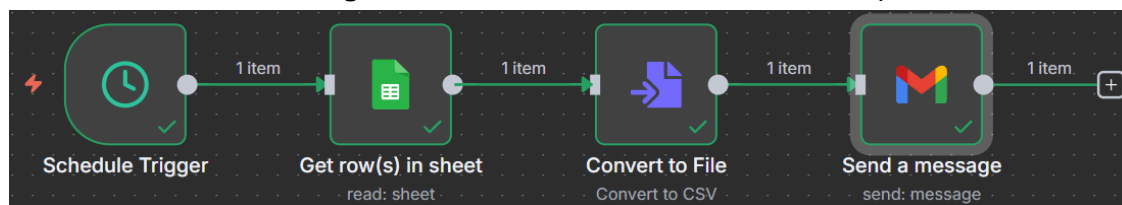
- Recebemos a missão do time de atendimento, utilizando a página AvalicoesClientes analisar quais avaliações estão com nota abaixo de 3 e notificar a gestão. Primeiramente adiciono um **Google Sheets Trigger** que será ativado quando uma nova linha for adicionada a página, adiciono o condicional **IF** para verificar se a nota está abaixo de 3, caso a nota esteja abaixo de 3 adiciono o **Gmail Send a Message** que enviará um e-mail ao gestor e por fim crio um **Append row in a sheet** para o caso do feedback ser positivo adicionando uma nova linha na página FeedbacksPositivos;



- Agora adiciono os dados no **PowerBI**, clicando em Obter Dados, seleciono a opção mais... e seleciono **Planilhas Google**, então faço login e tenho acesso as planilhas do Google no PowerBI;

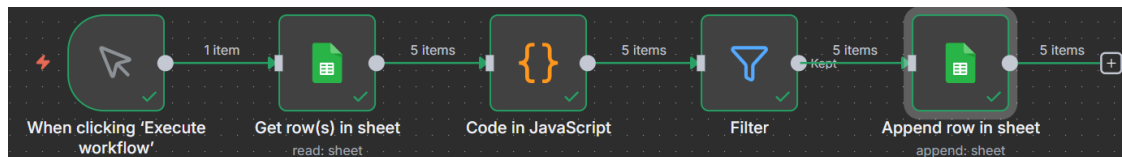
Automatizando exportações de planilhas para alimentar o Power BI

- Agora recebemos a tarefa de criar uma automação para o setor de BI, que quer que todas as manhãs um arquivo CSV consolidado das vendas seja enviado por e-mail. Primeiramente adiciono um **Schedule Trigger** para que seja disparado todo dia de manhã, então adiciono o **Get row(s) in sheet** para puxar as informações da planilha ConsolidacaoVendas, adiciono o **Convert to CSV** para converter o arquivo, então adiciono o **Send a Message** com o attachment do CSV convertido para enviá-lo;



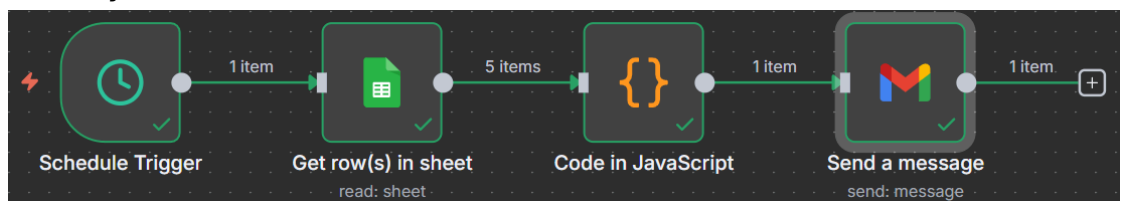
Atualizando dados em bases intermediárias

- A próxima missão é padronizar dados crus, vindos de diversas fontes de dados, foi especificado um problema de notas formatadas como texto (“três”) e linhas vazias. A primeira coisa que faço é criar um **Trigger** manual para iniciar o **workflow**, em seguida adiciono **Get row(s) in sheet** para selecionar a página DadosBrutos, então adiciono um **Code** em **JavaScript** para fazer a correção dos números das notas, agora adiciono um **Filter** para filtrar no campo NomeCliente os nomes vazios e por fim adiciono **Append row in sheet** selecionando a página DadosLimpos para registrar os dados agora filtrados e corrigidos;



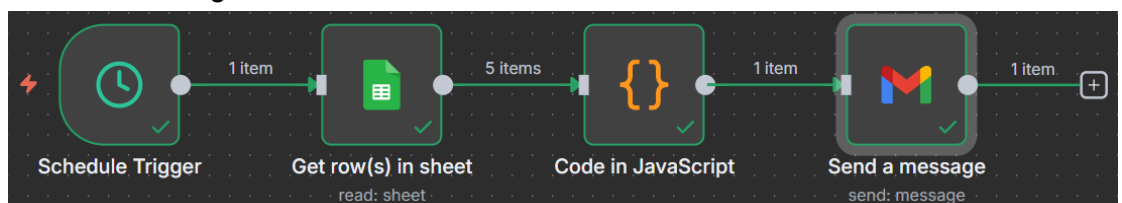
Enviando alertas semanais com link

- Agora precisamos fazer verificações se a base intermediária de vendas foi atualizada corretamente toda semana. Começo adicionando um **Schedule Trigger** sendo ativado semanalmente, então adiciono um **Get row(s) in sheet** para ler a página de DadosLimpos, adiciono **Code** em **JavaScript** para gerar uma mensagem com informações sobre o momento da execução e a quantidade de itens processados, então crio um **Gmail Send a Message** para enviar um e-mail de notificação de atualização dos dados;



Agendando rotinas com Cron no N8N (ex toda madrugada)

- A diretoria da empresa fez uma solicitação para que toda madrugada às 2h da manhã o sistema rode um processo que coleta as vendas do dia anterior, atualiza a base intermediária usada pelo PowerBI e envia um aviso rápido sobre a conclusão da rotina. Início adicionando um **Schedule Trigger** e selecionando a opção **Custom(Cron)** escrevendo '0 2 * * *' que representa todo dia às 2h da manhã, adiciono **Get row(s) in sheet** para selecionar a página DadosLimpos, em seguida crio um **Code** em **JavaScript** que cria uma mensagem dizendo o total de registros processados na data, adiciono **Gmail Send a Message** para enviar a notificação da rotina da madrugada;



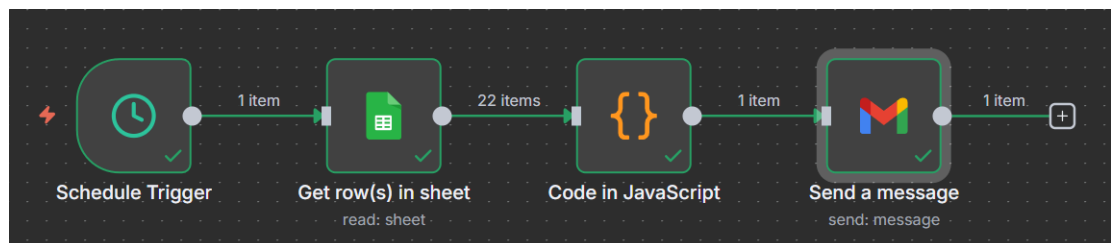
Enviando relatórios diários com dados filtrados e formatados

- A solicitação agora é de um relatório diário de vendas apenas do dia anterior, formatadas em uma tabela e que devem ser enviados ao email do gestor. Começo adicionando um **Schedule Trigger** todos os dias às 8am, em seguida adiciono o **Get row(s) in sheet** para pegar as informações da planilha em que todas as vendas estão agrupadas (VendasHistorico), adiciono o bloco **Code** em **JavaScript** para executar um código.

O que o código faz:

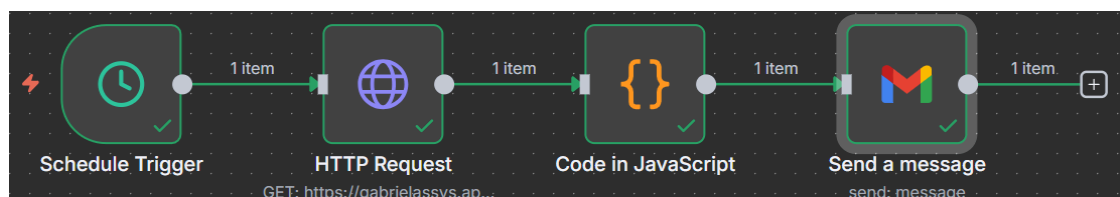
1. Obtém a data de hoje e a data de ontem;
2. Formata a data de ontem para o formato YYYY-MM-DD;
3. Filtra as vendas de ontem;
4. Calcula o total das vendas de ontem;
5. Cria o corpo do email em HTML;

Então adiciono o bloco **Gmail Send a Message** para enviar o relatório via email.



Monitorando falhas do próprio N8N e criando alertas de erro

- Agora resolveremos um problema, em um dia um dos fluxos falharam durante a madrugada e ninguém percebeu, e o gerente apenas descobriu o problema durante uma reunião, quando o dashboard estava desatualizado. Crio um **Schedule Trigger** todo dia às 8am, adiciono um bloco **HTTP** que faz a conexão com a API de onde os dados estão vindo, em seguida adiciono um **Code** em **JavaScript** que verifica se existe algum erro nos dados e retorna uma mensagem e por fim adiciono o bloco **Gmail Send a Message** para enviar as mensagens geradas pelo código via email.



Próximos passos abaixo

Criando a simulação de chatbot com planilha (entrada e resposta)

- Agora criarei um fluxo demonstrativo de como utilizar inteligência artificial na plataforma n8n. Começo adicionando o bloco **When chat message received**, em seguida adiciono um **AI Agent** e este bloco precisa estar conectado a um modelo, a um local de memória e a uma ferramenta, adiciono o **Groq chat model** (que precisa de uma chave de API criada no Groq), então testamos com uma mensagem.

