



IMERSÃO Agentes de IA

APOSTILA COMPLETA

AULA 4

Como ganhar mais dinheiro
com agentes de IA

PARTE

1

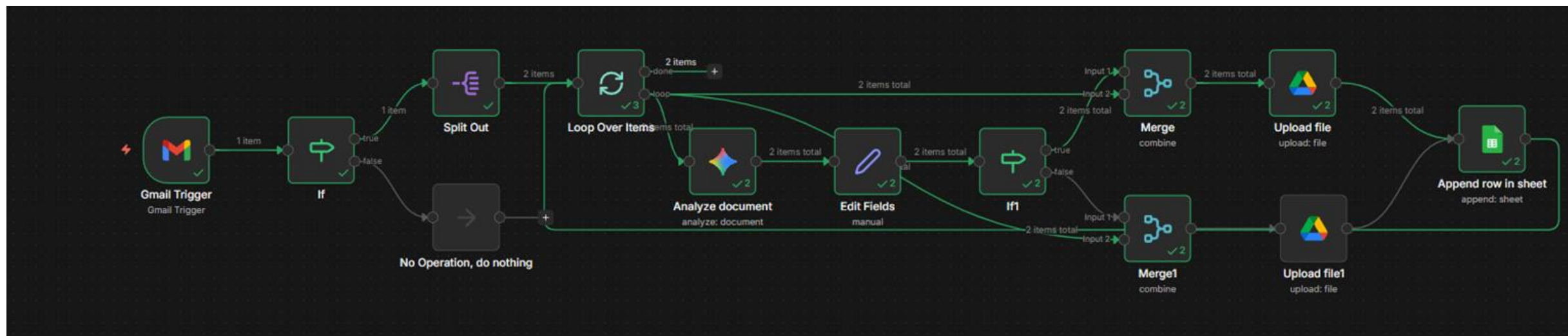
CONHECENDO A AUTOMAÇÃO

PARTE 1 **Conhecendo a automação**

Fala, **Impressionador!**

Nesta aula, vamos construir um projeto real, que encontramos na plataforma [upwork](#). Nele, você aprenderá a criar um agente de IA capaz de monitorar e processar automaticamente e-mails com faturas em anexo.

O fluxo utiliza inteligência artificial para extrair os dados das faturas, classificar cada documento como pessoal ou corporativo, organizar os arquivos no Google Drive e registrar automaticamente os dados extraídos em uma planilha do Google Sheets.



PARTE 2

Inserindo o gatilho

PARTE 2

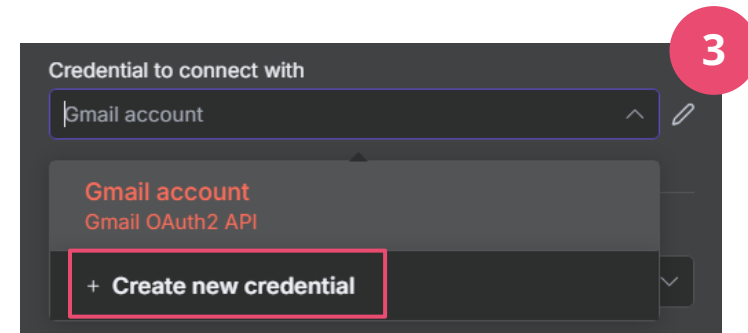
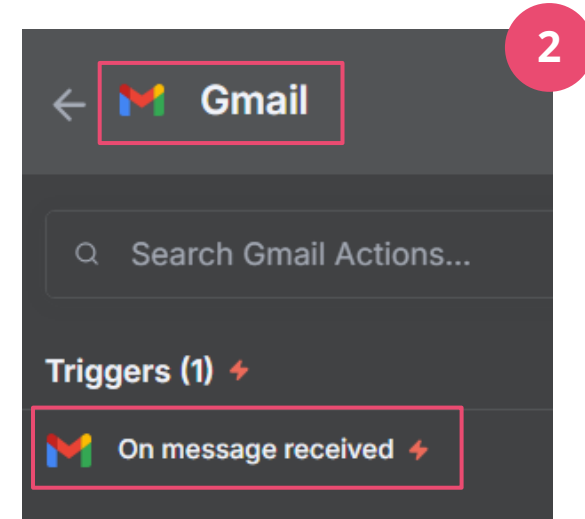
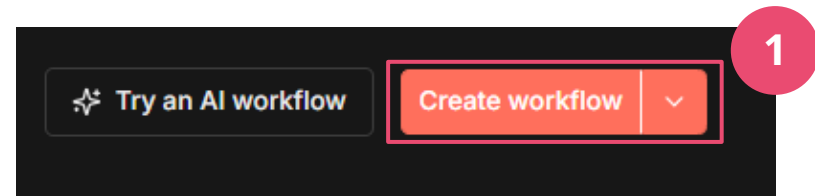
Inserindo o gatilho

Na tela inicial do n8n, vamos criar um novo workflow com o botão **Create Workflow** (Imagem 1) localizado no canto superior direito da tela.

Para isso, vamos clicar no botão **Add first step**, buscar na lateral direita por **Gmail** e selecionar a opção **On message Received**, que dispara a automação sempre que um novo e-mail for recebido (Imagem 2).

Em seguida, é necessário configurar os parâmetros desse gatilho.

No campo **Credential to Connect With**, deve-se clicar em **Create new credential** (Imagem 3), selecionar a opção **Sign in with Google** e conceder todas as permissões solicitadas para que o n8n tenha acesso à conta de e-mail. Caso tenha dúvidas, você pode consultar o vídeo [Configurando OAuth Google \(Entrar com Gmail\)](#) com o passo a passo detalhado.

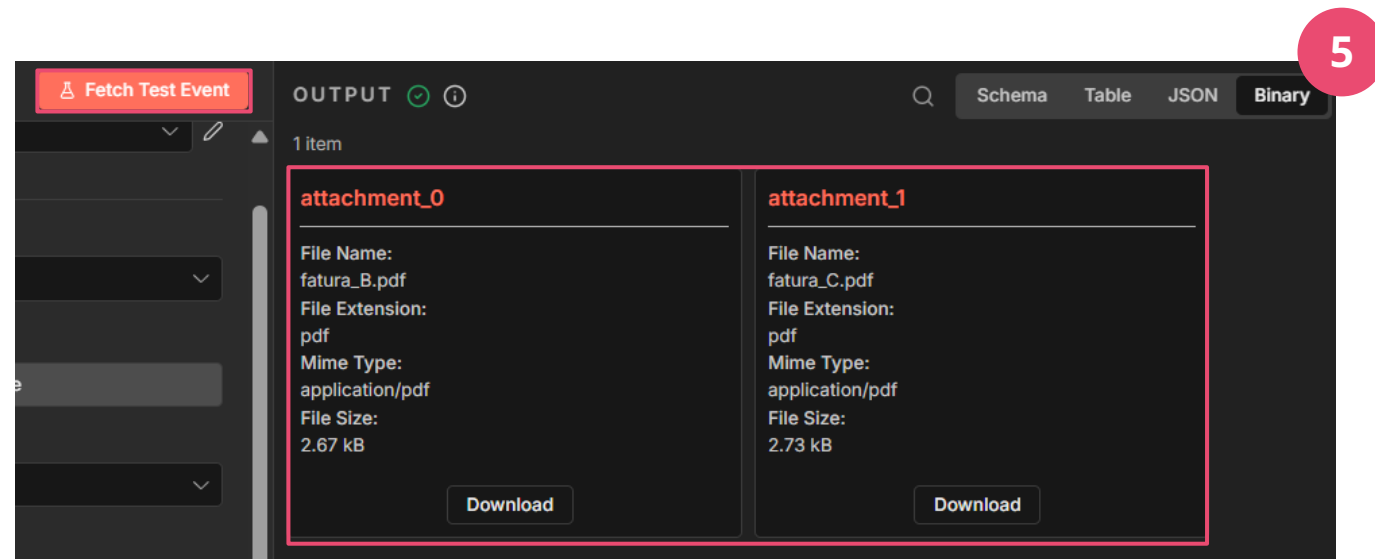
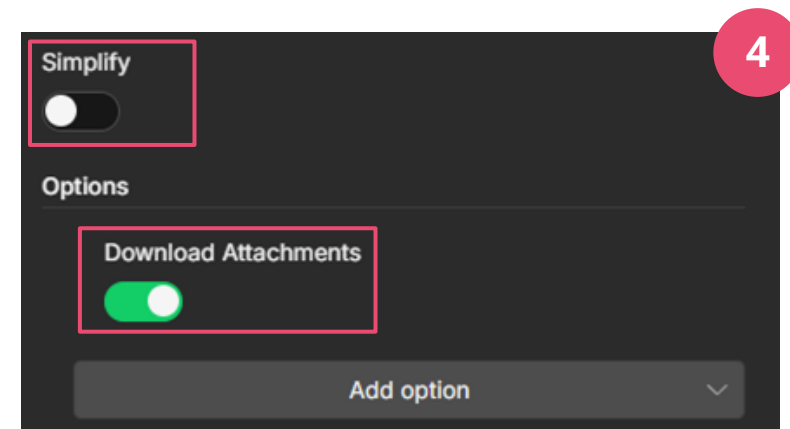


PARTE 2 Inserindo o gatilho

É importante desabilitar a opção **Simplify**, para que todas os anexos sejam exibidos na área de Output. Na seção de **Option**, clique no botão **Add option** e habilite a opção **Download Attachments** para que o gatilho baixe todos os anexos enviados na mensagem (Imagem 4).

Após essa configuração, vamos realizar um teste enviando um e-mail com duas faturas para o endereço utilizado na criação da credencial. Para acessar as faturas você pode consultar a pasta "Invoices" que está disponível [neste link](#).

Depois de aguardar alguns instantes, basta clicar no botão **Fetch Test Event** para confirmar se o gatilho está funcionando corretamente e se os dados do e-mail aparecem no Output. Os arquivos devem aparecer na aba **Binary** no área de **Output** na lateral direita dentro das configurações dos parâmetros (Imagem 5).

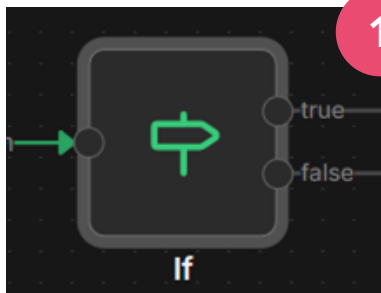


PARTE 3

VERIFICANDO A PRESENÇA DE ARQUIVOS

Verificando a presença de arquivos

1



O próximo passo é criar uma condição com um nó **If** para verificar se o existe arquivos no e-mail recebido (Imagem 1).

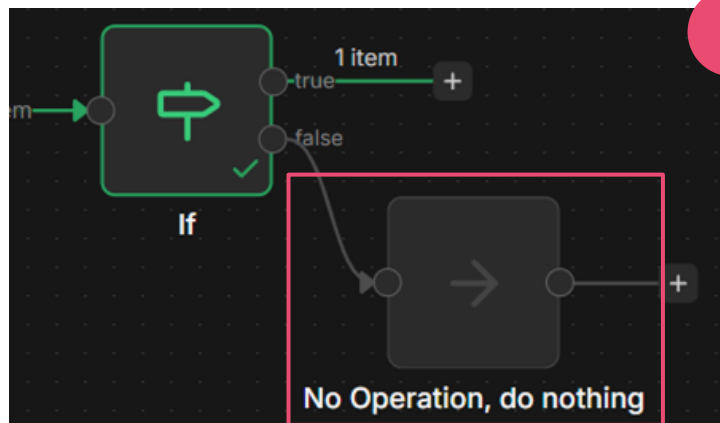
Na configuração dos parâmetros do nó, preencha o campo **value1**, digite o código `{{ $binary.attachment_0 }}`.

2



Lembre-se de mudar o campo para **Expression** para garantir que a expressão seja interpretada corretamente. Além disso, mude a condição para **exists** que se encontra dentro do menu de **Objects** (Imagem 2).

3



Não iremos fazer nada caso a condição que implementamos no nó **If** seja falsa. Por isso, podemos deixar esse caminho em branco ou adicionar um nó de **nothing** para garantir compreensão do fluxo e evitar confusões (Imagem 3).

Após finalizar as configurações deste nó, clique no botão **Execute step** para testar esse passo da automação.

PARTE 4

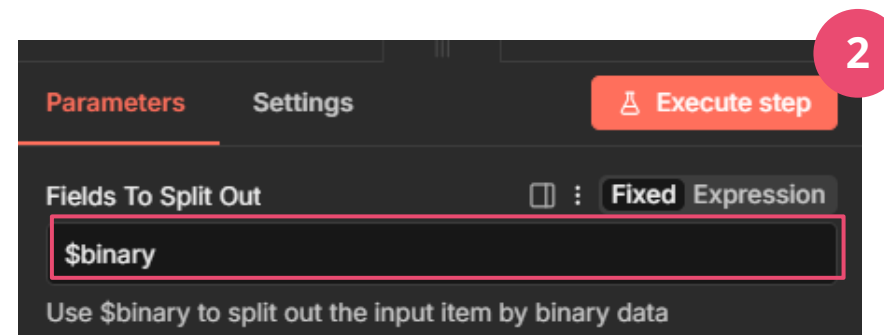
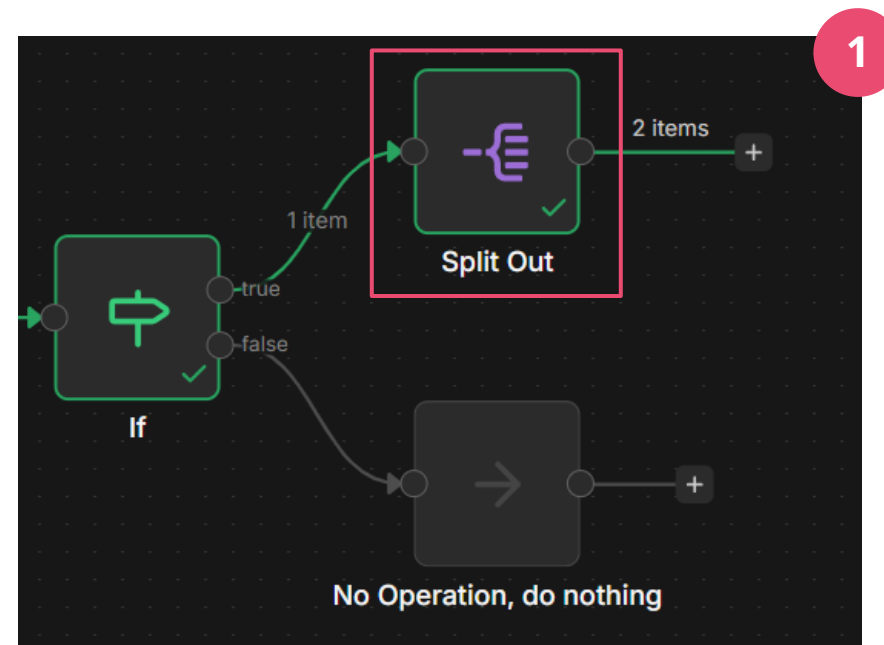
SEPARANDO OS ARQUIVOS

PARTE 4 Separando os arquivos

Ao enviar mais de uma fatura, como fizemos no e-mail de teste, elas ficam agrupadas. Precisamos analisar cada arquivo individualmente. Para isso, vamos usar o nó **Split Out** no caminho **true** do nó **If** (Imagem 1).

Na configuração dos parâmetros, dentro do campo **Fields To Split Out** preencha com a expressão `$binary`. Dessa forma, conseguimos analisar todos os arquivos na aba **Binary** na área de **Input** à esquerda (Imagem 2).

Clique no botão **Execute step** para testar esse passo da automação. Perceba que o nó **Split Out** recebe 1 item e devolve 2 itens, separando as duas faturas.



PARTE 5

EXECUTANDO AÇÕES EM LOOP

PARTE 5 Executando ações em loop

Para cada arquivo que separamos, precisamos executar uma sequência de ações para extrair os dados, preencher a planilha e guardar o arquivo em uma pasta do Google Drive.

Essa sequência precisa ser repetida várias vezes, por isso usaremos o **Loop Over Items** (Imagem 1).

Nas configurações do loop, vamos manter o campo **Batch Size** com o valor padrão "1", para que a execução das etapas seja feita item por item, um de cada vez (Imagem 2).

Além disso, vamos remover o nó **Replace me** para inserir no caminho de loop os passos que queremos.

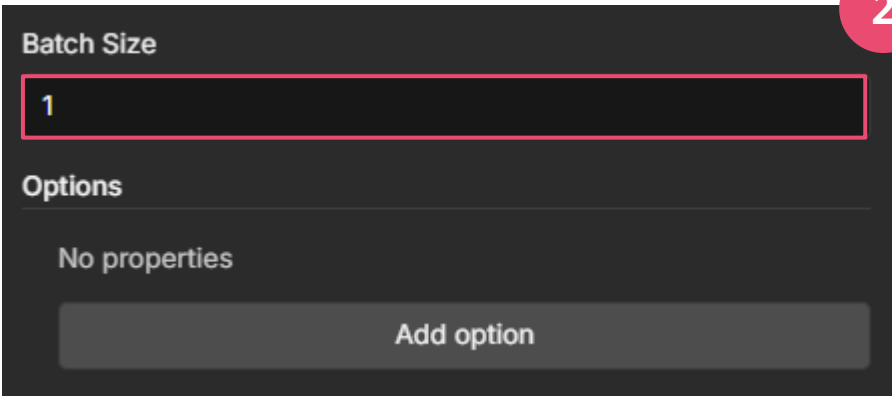
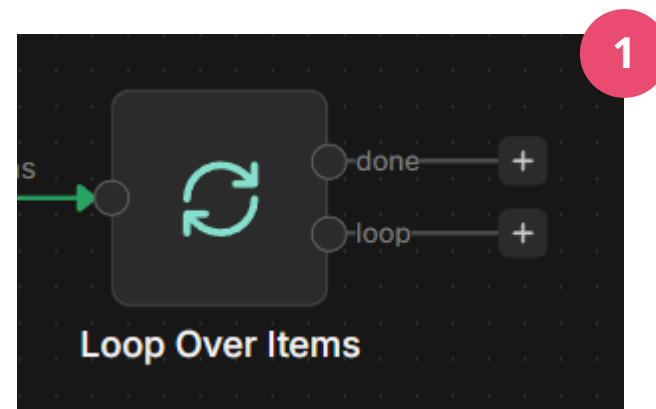


Imagem 2: Configuração do nó "Loop Over Items" no Zapier. O formulário mostra o campo "Batch Size" com o valor "1" selecionado. Abaixo, há a seção "Options" com o texto "No properties" e um botão "Add option". Um círculo vermelho com o número "2" está no canto superior direito da imagem.

PARTE 6

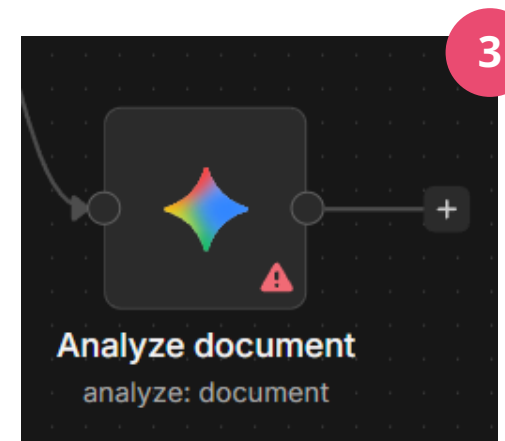
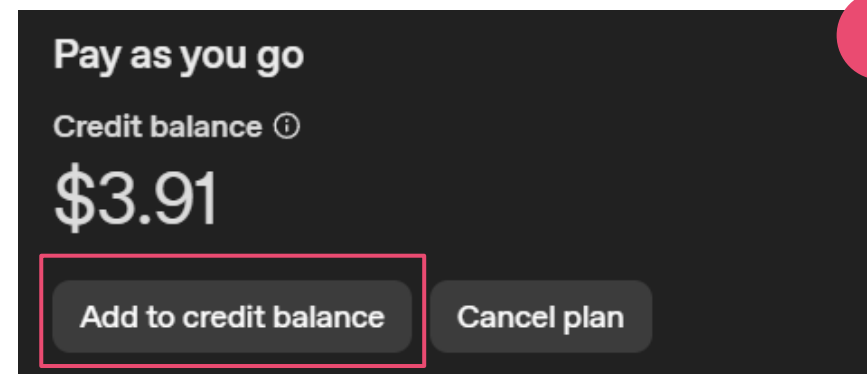
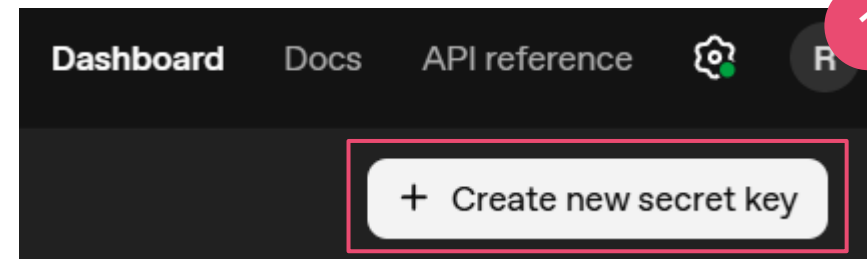
ANALISANDO O DOCUMENTO COM IA

O primeiro passo do Loop é analisar o documento com inteligência artificial. No projeto que pegamos no upwork, o briefing pede para utilizarmos o ChatGPT. Para isso, precisaremos acessar a plataforma da OpenAI na área de [API Key](#) e clicar no botão **Create new secret key** no canto superior direito da tela (Imagem 1).

Em seguida, precisamos adicionar créditos na seção [Billings](#) ao clicar no botão **Add to credit balance** (Imagem 2).

Por fim, podemos criar um nó de **AI Agent** e utilizar o modelo de linguagem do ChatGPT com a chave de API que criamos para realizar a análise do anexo.

Contudo, para testar, podemos usar o nó **Analyze document** do **Google Gemini** que é gratuito e já possui uma configuração para analisar o documento (Imagem 3).



Nas configurações do parâmetro do nó **Analyze document**, vamos preencher o campo **Credential to connect with** com a credencial que já criamos nas aulas anteriores.

Vamos escolher o modelo de linguagem que será utilizado no campo **Model**. Por exemplo, "models/gemini-2.5-flash".

Na sequência vamos inserir no campo **Text Input** o prompt que vai instruir o modelo a atuar como um extrator de dados de faturas/recibos em PDF, retornando apenas um JSON válido com campos padronizados para planilha. Vamos utilizar o prompt que está no arquivo "Prompt.docx" que está [neste link](#).

Em **Input Type** selecione a opção "Binary File(s)" para lidar com os arquivos anexados no e-mail. Já em **Input Data Field Name(s)** adicione a expressão `{{ Object.keys($binary)[0] }}` (Imagem 4).

4

Credential to connect with
Google Gemini(PaLM) Api account 2

Resource
Document

Operation
Analyze Document

Model
From list models/gemini-2.5-flash

Text Input
Você é um assistente de extração de dados.

Input Type
Binary File(s)

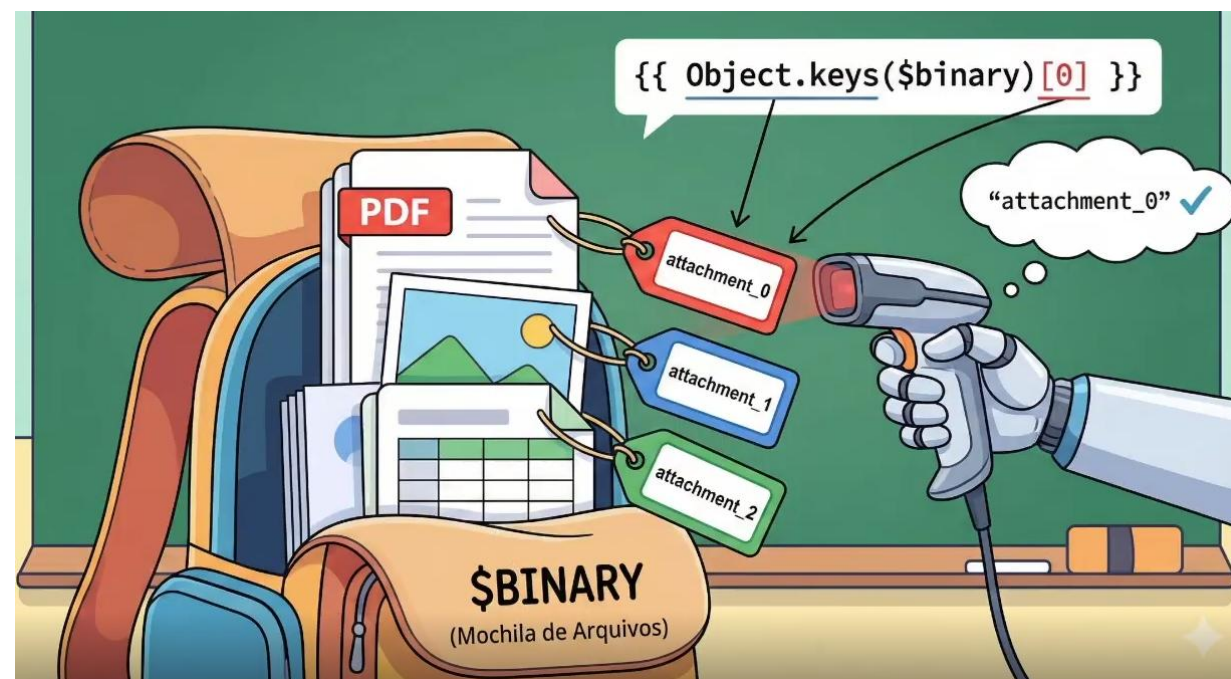
Input Data Field Name(s)
`{{ Object.keys($binary)[0] }}`
attachment_0

A expressão `{{ Object.keys($binary)[0] }}` é usada no n8n para identificar o nome do campo que contém um arquivo recebido por um nó.

No n8n, o objeto `$binary` armazena todos os arquivos binários que chegam ao nó, como PDFs, imagens ou documentos. Cada arquivo fica associado a uma chave, por exemplo, "attachment_0".

A função `Object.keys($binary)` é um padrão do JavaScript que retorna uma lista com os nomes das chaves existentes dentro do objeto `$binary`. Já o trecho `[0]` no final da expressão serve para acessar o primeiro elemento da lista.

Quando tudo é combinado, a expressão retorna dinamicamente o nome do campo binário que contém o arquivo. Na prática, ela diz ao n8n para usar automaticamente o primeiro arquivo recebido como entrada, sem que você precise definir manualmente o nome do campo.



PARTE 7

SEPARANDO AS INFORMAÇÕES

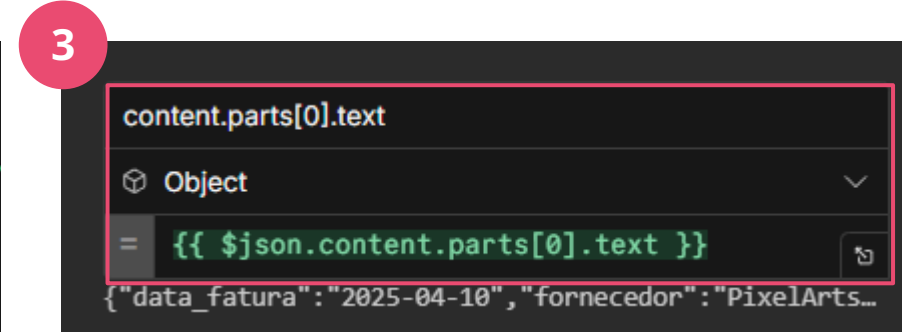
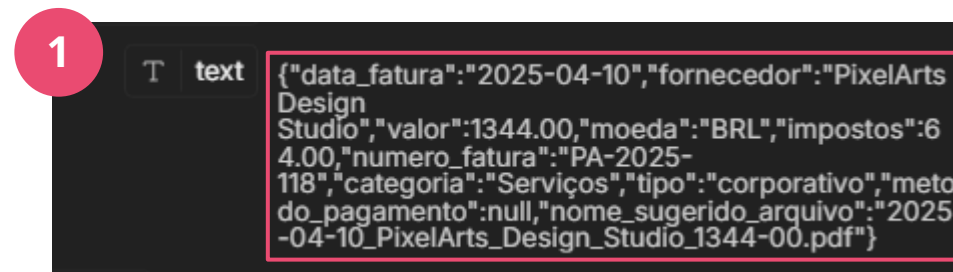
PARTE 7 Separando as informações

Ao executar o nó **Analyze document** obtemos um único objeto com todas as informações que pedimos no prompt (Imagem 1).

Precisamos separar esses dados para usar individualmente. Para isso, vamos adicionar o nó **Edit Fields (Set)** (Imagem 2).

Na configuração dos parâmetros, arraste a informação **text** do nó **Analyze Document** da seção de **Input** para o campo **Field to set**, selecione a opção **Object** para garantir a separação do texto em partes como indicado pelo formato JSON (Imagem 3).

Para testar o nó, clique no botão **Execute step** e veja o resultado na seção de **Output** (Imagem 4).



PARTE 8

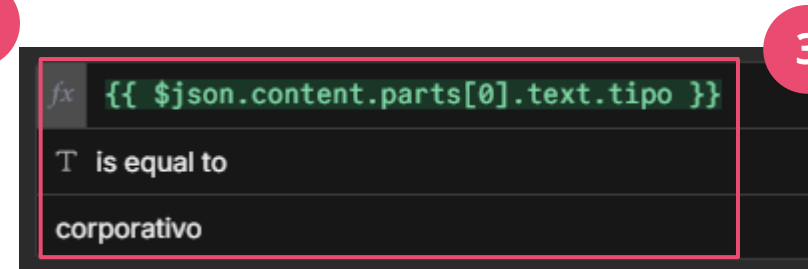
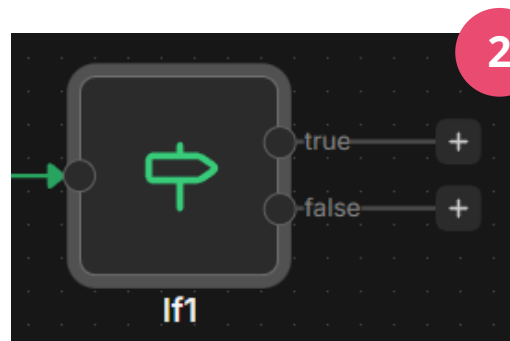
SALVANDO NO GOOGLE DRIVE

PARTE 8 Salvando no Google Drive

Após analisar o documento, vamos salvar os arquivos no Google Drive separando as faturas pessoais das corporativas. Para isso precisamos: criar as pastas Drive, verificar o tipo de arquivo, mesclar as informações dos nós para acessar o arquivo binário e as informações coletadas e fazer o upload do arquivo dentro das pastas.

A criação das pastas deve ser feita diretamente no [Google Drive](#) através do botão **Novo** no canto superior esquerdo. Criaremos uma pasta principal, que chamaremos de "Análise de faturas", por exemplo. Dentro dela, criaremos outras duas pastas chamadas: "Pessoal" e "Corporativa" (Imagem 1).

O próximo passo é identificar o tipo de fatura utilizando um nó **If**. Na configuração, arraste a variável **tipo** do nó **Edit Fields** para o campo **value1**, mantenha a condição como **is equal to** e preencha o campo **value2** com o valor "corporativo". Dessa forma, o fluxo conseguirá diferenciar automaticamente faturas corporativas das pessoais.

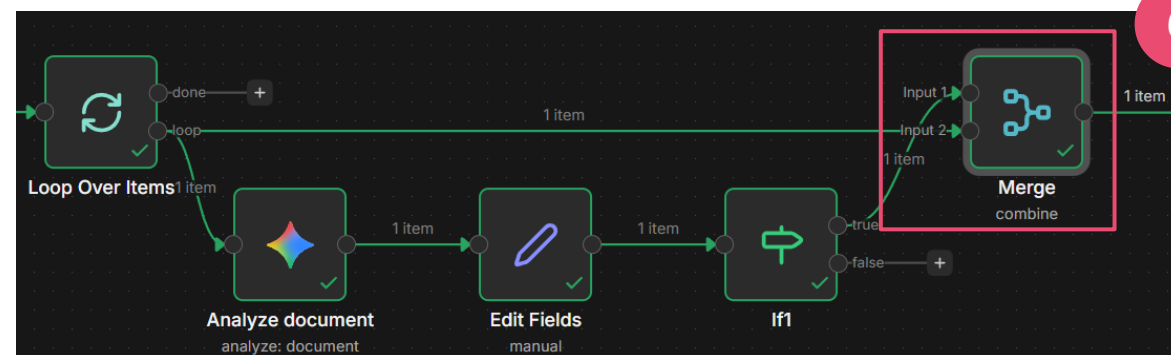
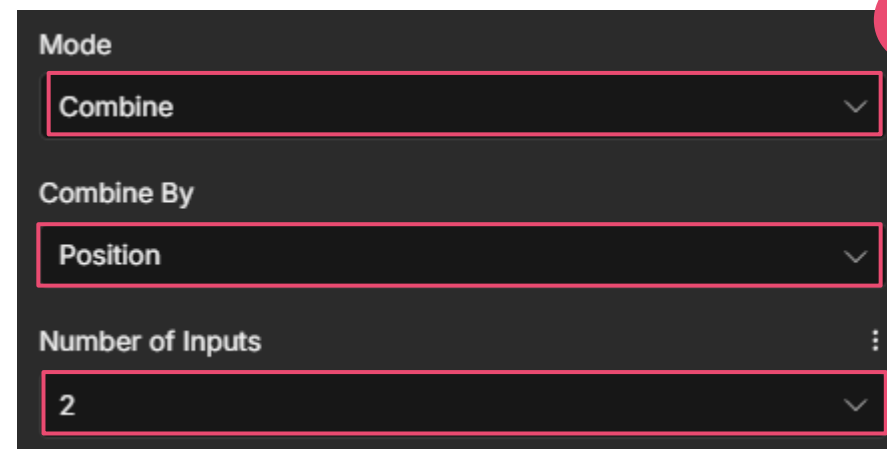
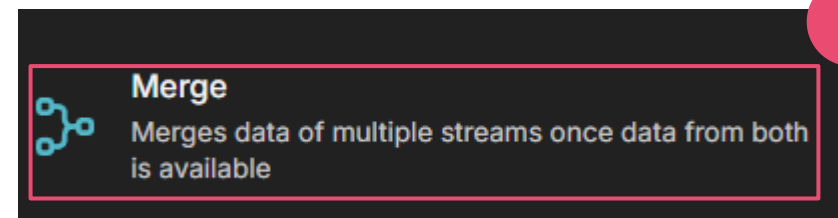


PARTE 8 Salvando no Google Drive

Após a condição, é necessário agrupar os dados coletados no campo **Edit Field (Set)** com o arquivo da fatura que está no nó **Loop Over Items**. Por isso, no caminho **true** do nó **If1**, adicione o nó **Merge** (Imagem 4).

Configure o campo **Mode** como "Combine", para combinar os dados o **Combine By** como "Position" e defina o **Number of Inputs** como "2". Com isso, você combina os dados através da posição de dois nós que serão plugados como fontes de entrada para o nó **Merge** (Imagem 5).

O primeiro input do **Merge** já receberá os dados do **If1**, e o segundo deve ser conectado manualmente ao nó **Loop Over Items**, garantindo acesso aos arquivos anexados ao e-mail (Imagem 6).

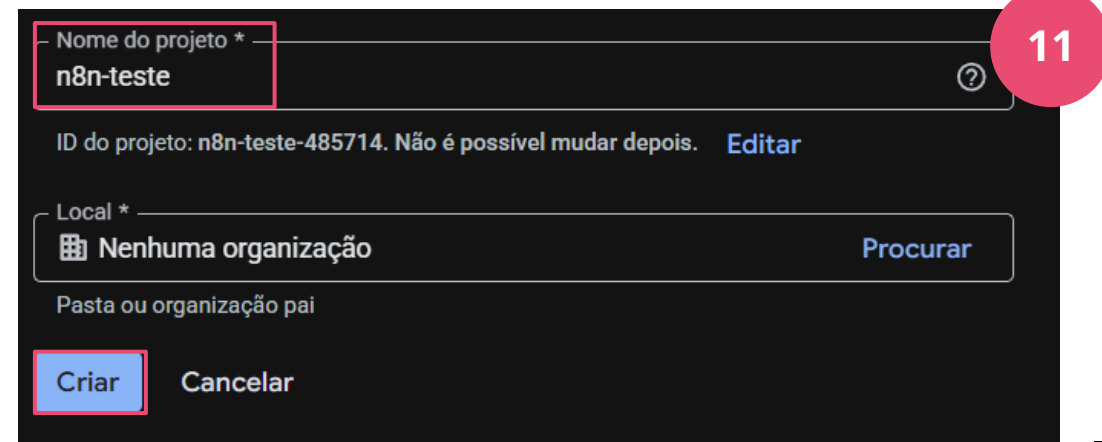
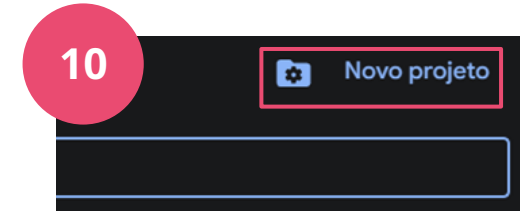
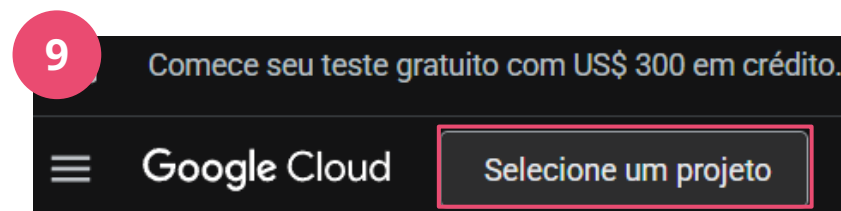
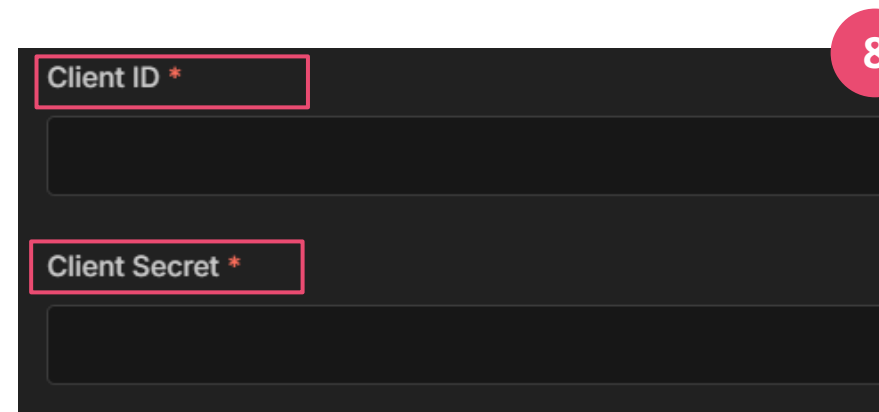
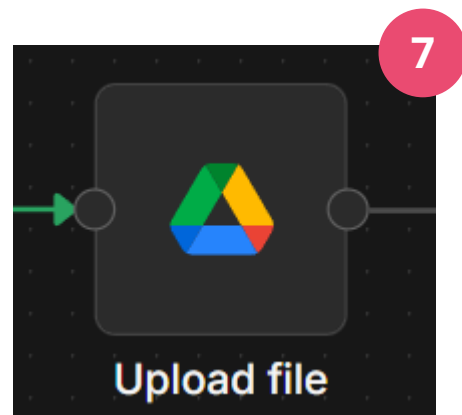


PARTE 8 Salvando no Google Drive

Com os dados e o arquivo unidos, o último passo é realizar o upload no Google Drive com o nó **Upload file** (Imagem 7).

Nas configurações deste nó, a criação da credencial é um pouco diferente. Precisamos preencher os campos **Client ID** e **Client Secret** (Imagem 8), que serão criados no [Google Cloud](#).

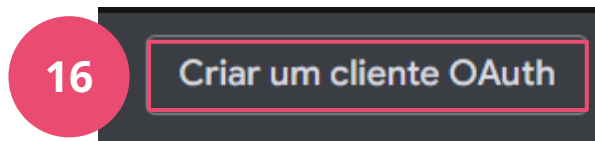
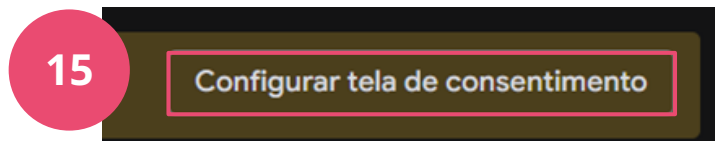
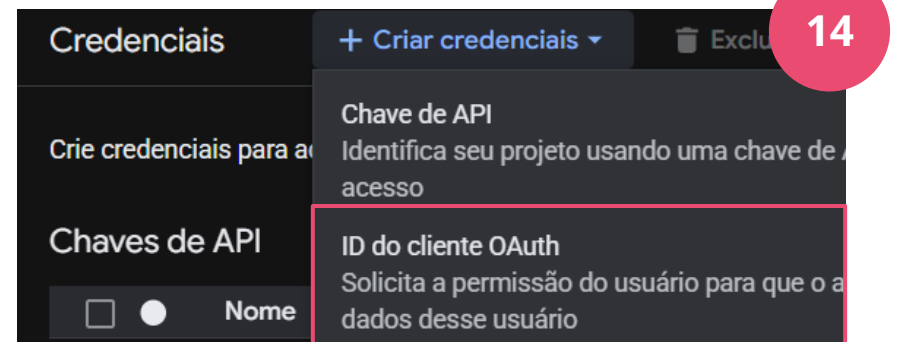
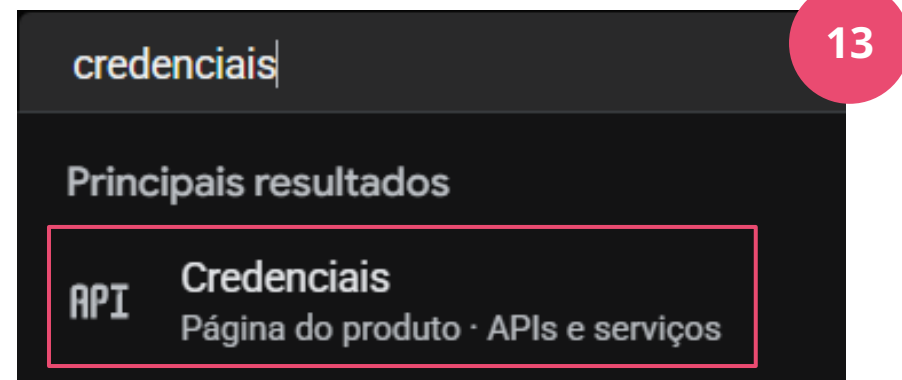
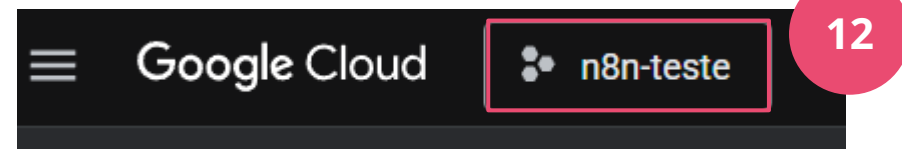
O processo começa com a criação de um projeto. Para isso, clique em **Selecionar um projeto** (Imagem 9) no canto superior esquerdo, depois em **Novo projeto** (Imagem 10), defina um nome (por exemplo, “n8n-teste”) e conclua clicando em **Criar** (Imagem 11). Esse projeto será o container onde todas as configurações de autenticação e permissões do n8n ficarão centralizadas.



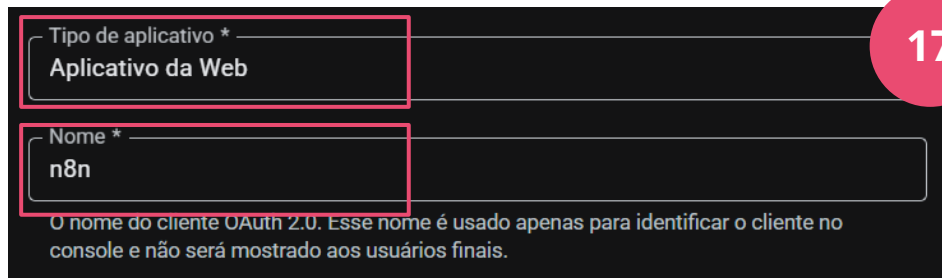
PARTE 8 Salvando no Google Drive

Clique novamente no botão **Selecione um projeto** e escolha o projeto que acabamos de criar (Imagem 12). Utilize a barra de pesquisa do Google Cloud para buscar por Credenciais (Imagem 13), selecione o projeto recém-criado, clique em Criar credenciais e escolha a opção ID do cliente OAuth (Imagem 14).

Antes de gerar o ID, o Google exigirá a configuração da tela de consentimento, que define como o aplicativo será apresentado ao usuário durante a autenticação (Imagem 15). Preencha as informações básicas do aplicativo, como nome do app e e-mail de suporte. Em seguida, escolha o tipo de público como "Externo", informe um e-mail de contato na seção de dados de contrato, aceite os termos e finalize clicando em Criar. Após essa etapa, o console liberará a opção para criar o cliente OAuth propriamente dito (Imagem 16).



PARTE 8 Salvando no Google Drive



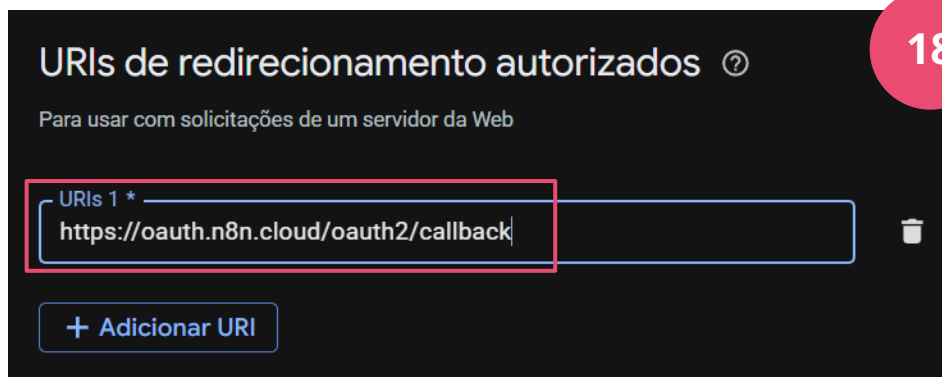
Tipo de aplicativo *

Aplicativo da Web

Nome *

n8n

O nome do cliente OAuth 2.0. Esse nome é usado apenas para identificar o cliente no console e não será mostrado aos usuários finais.



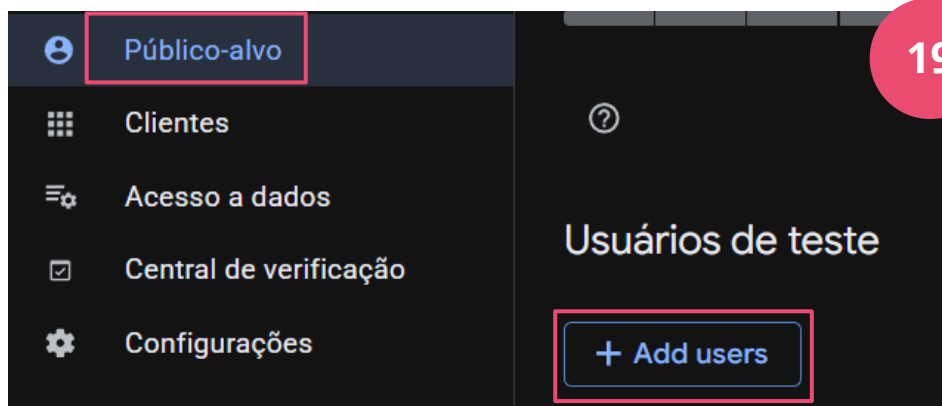
URIs de redirecionamento autorizados ?

Para usar com solicitações de um servidor da Web

URIs 1 *

https://oauth.n8n.cloud/oauth2/callback

+ Adicionar URI



Público-alvo

Clientes

Acesso a dados

Central de verificação

Configurações

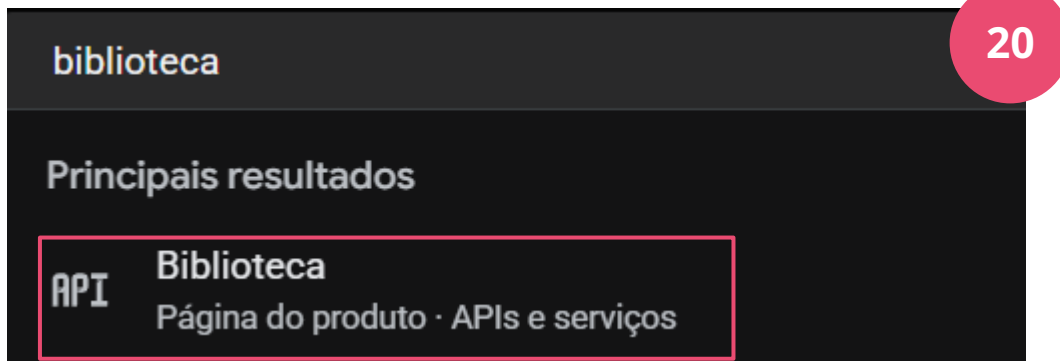
Usuários de teste

+ Add users

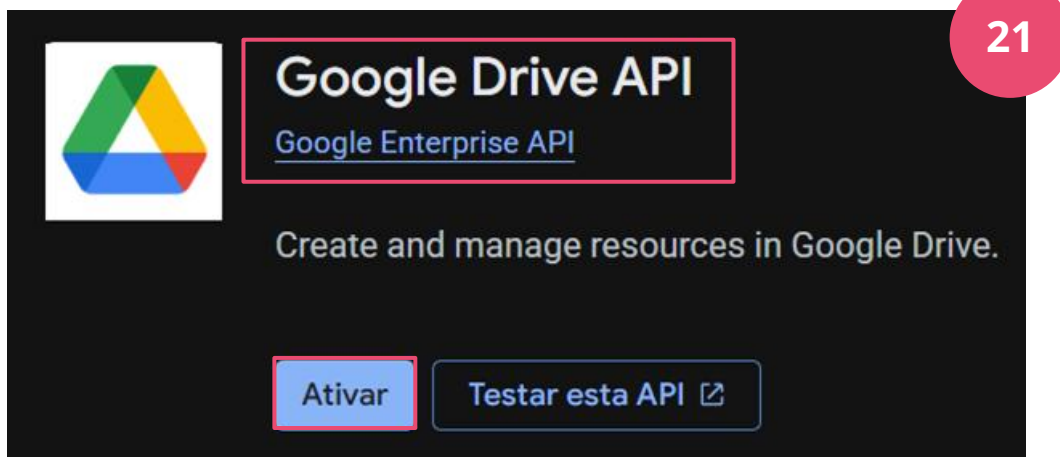
Ao criar o cliente OAuth, selecione o **Tipo de aplicativo** como "Aplicativo da Web", defina um nome identificável (por exemplo, "n8n") (Imagem 17). Na seção de **URIs de redirecionamento autorizados**, insira exatamente a URL exibida no campo **OAuth Redirect URL** da tela de credenciais do n8n (Imagem 18). Esse passo é essencial para que o Google permita a autenticação corretamente.

Após concluir a criação, o Google Cloud exibirá o Client ID e o Client Secret. Essas informações devem ser copiadas e salvas, pois serão usadas diretamente no n8n. Em seguida, é necessário definir quem poderá usar essa credencial enquanto ela estiver em modo de teste, adicionando os e-mails autorizados na seção **Público-alvo**, dentro de **Usuários de teste** (Imagem 19).

PARTE 8 Salvando no Google Drive



Com a credencial criada, o próximo passo é ativar as APIs necessárias. Na **Biblioteca** do Google Cloud (Imagem 20), busque e ative os serviços que o n8n precisará acessar, como Google Drive (Imagem 21). Sem essa ativação, mesmo com a autenticação correta, o acesso às funcionalidades não será permitido.



Por fim, volte ao n8n e preencha os campos **Client ID** e **Client Secret** com os dados gerados no Google Cloud. Clique em **Sign in with Google**, escolha o e-mail autorizado, revise e confirme todas as permissões solicitadas. Se tudo estiver configurado corretamente, o n8n exibirá a mensagem de conta conectada, indicando que a credencial foi criada com sucesso e já pode ser selecionada nos nós do Google Drive (Imagem 22).



PARTE 8 Salvando no Google Drive

17

Input Data Field Name

`{{Object.keys($binary)[0]}}`

attachment_0

File Name

`{{ $json.content.parts[0].text.nome_sugerido_arquivo }}`

2025-04-10_PixelArts_Design_Studio_1344-00.pdf

Parent Drive

From list My Drive

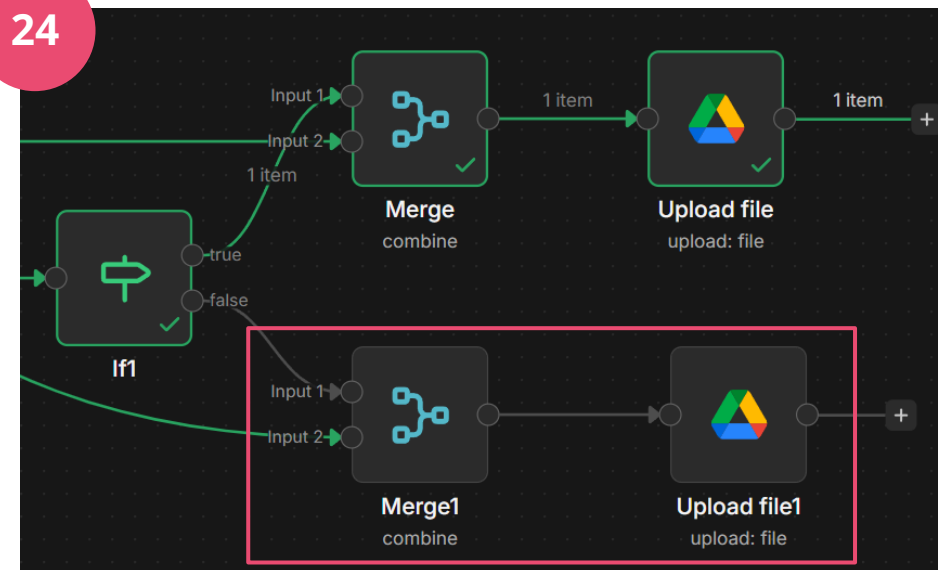
Parent Folder

From list Corporativas

Por fim, o mesmo processo de **Merge** e **Upload file** deve ser repetido no caminho **False** do nó **If1**. A única diferença será a seleção da pasta "Pessoal" no campo **Parent Folder**. Assim, sempre que a fatura não for classificada como corporativa, o fluxo salvará o arquivo na pasta pessoal (Imagem 24).

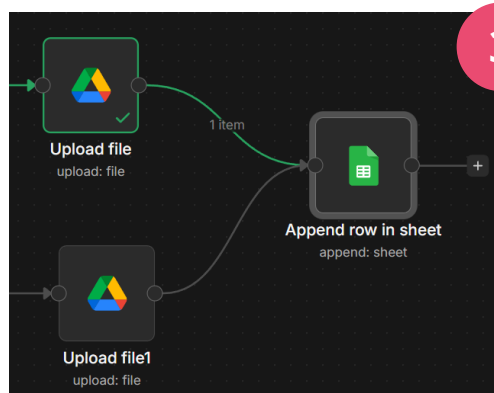
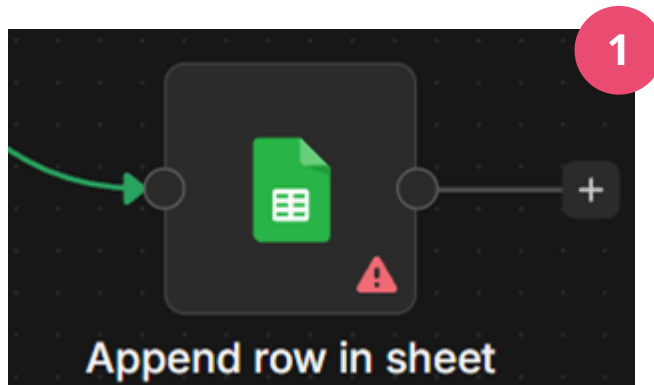
Com a credencial configurada, vamos ajustar os demais campos do nó **Upload file**. Preencha o campo **Input Data Field Name** com a expressão `{{ Object.keys($binary)[0] }}` para que a cada vez que o loop seja executado ele buscar pelo nome correto do arquivo. No campo **File Name**, arraste o bloco "nome_sugerido_arquivo" do nó anterior. Em **Parent Folder**, selecione a pasta "Corporativa", já que esse fluxo corresponde às faturas corporativas. Após isso, execute o passo para confirmar que o arquivo foi enviado corretamente (Imagem 23).

24



PARTE 9

INSERINDO OS DADOS NA PLANILHA DO GOOGLE



Na etapa final da automação criaremos a planilha onde os dados serão armazenados. No navegador, digite *sheets.new*, dê um nome à planilha, por exemplo, “Análise de faturas” e crie as colunas que irão organizar as informações: “data da fatura”, “fornecedor”, “valor”, “moeda”, “impostos”, “número da fatura”, “categoria”, “tipo” (corporativo ou pessoal), “método de pagamento”, “URL do arquivo” e “e-mail de origem”.

Com a planilha pronta, o próximo passo é configurar o nó **Append row in sheet**, responsável por enviar os dados para o Google Sheets (Imagem 1)

Na configuração do nó, selecione a credencial de acesso ao Google Sheets. Depois, no campo **Document**, escolha a planilha “Análise de faturas” e, em Sheet, selecione a aba onde os dados serão adicionados (Imagem 2).

Em seguida, volte ao fluxo completo e conecte o nó **Upload file1** ao nó **Append row in sheet**, garantindo que o envio dos dados aconteça após o upload do arquivo de contas corporativas ou pessoais.

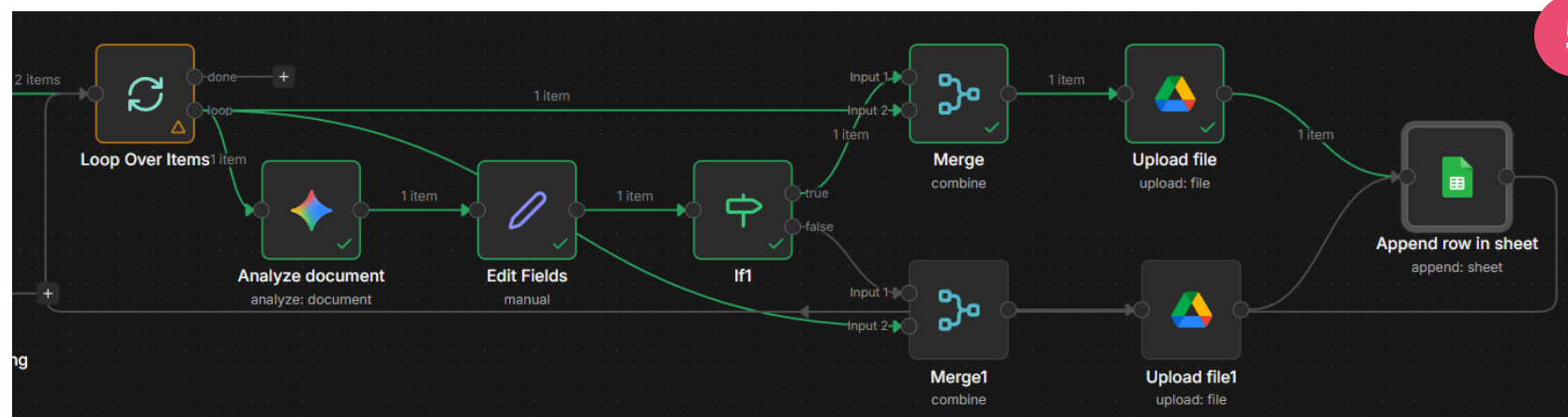
4



Após a conexão, é necessário mapear cada coluna da planilha com as informações correspondentes do fluxo. Os campos como "data da fatura", "fornecedor", "valor", "moeda", "impostos", "número da fatura", "categoria", "tipo" e "método de pagamento" devem ser preenchidos com os dados vindos do nó **Edit Fields**. Para a coluna **URL do Arquivo**, arrastaremos o bloco "webViewLink" do nó **Upload File**. Já o campo **E-mail de Origem** deve ser preenchido com o endereço capturado pelo **Gmail Trigger** (Imagem 4)

Por fim, conecte a saída do nó **Append row in sheet** à entrada do nó **Loop Over Items** para que o processo seja repetido para todos os arquivos (Imagem 5).

5

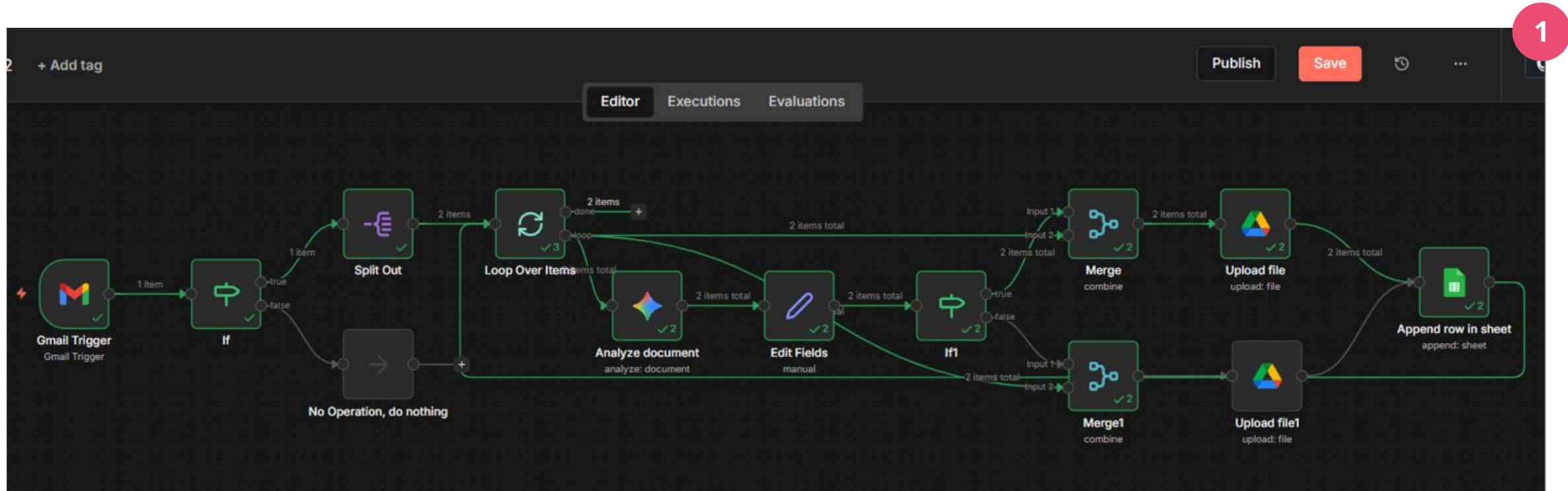


PARTE 10

TESTANDO A APLICAÇÃO

PARTE 10 Testando a aplicação

Com isso, a automação é finalizada com sucesso. A partir desse ponto, é possível realizar novos testes com faturas da pasta "Invoices" e utilizar o botão **Execute workflow** para rodar todo o fluxo de ponta a ponta. Além disso, você pode salvar e publicar a aplicação no canto superior direito da tela.



BÔNUS

VÍDEOS COMPLEMENTARES

Curso Básico Agentes de IA e Automações 2026 - Aula 4





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



@hashtagnocodeia



/@NoCodeIAHashtag

