# **Projeto 01) Agente de E-mails com n8n**

## **Objetivo**

Neste guia, vamos criar um **agente de IA para gestão de e-mails** que classifica automaticamente mensagens recebidas no Gmail usando o n8n + modelo da OpenAI. Ao final, você terá um workflow capaz de:

* Categorizar e rotular e-mails com base em conteúdo
* Aplicar rótulos do Gmail automaticamente
* Preparar o caminho para respostas automáticas ou triagem inteligente

## **Etapa 1: Criar o gatilho de novos e-mails no Gmail**

### **1.1 Adicione o nó Gmail Trigger no n8n**

* Tipo: Gmail Trigger
* Evento: on message received
* Poll time: everyMinute
* Marque: **"Include only unread messages"** (ou adicione o filtro readStatus: unread)

### **1.2 Conecte sua conta do Gmail**

* Clique em **"Create New Credential"**
* No Google Cloud Console:
  + Crie um projeto
  + Ative a **Gmail API**
  + Configure a tela de consentimento OAuth (adicione seu e-mail como test user)
  + Gere um **OAuth Client ID** com tipo Web Application
  + No campo redirect URI, cole o link fornecido pelo n8n
  + Copie o Client ID e Secret e cole no n8n

Dica: dê permissão total ao Gmail na autenticação

### 

### **1.3 Desmarque a opção "Simplify Response"**

* Isso garante que você veja todos os dados brutos do e-mail (headers, corpo, ID, etc)

## **Etapa 2: Adicionar o classificador de texto**

### **2.1 Adicione o nó Text Classifier**

* Texto a classificar: ={{ $json.text }} ou outro campo com o corpo do e-mail
* Adicione **4 categorias**:

### **Text Classifier (nó nativo do n8n)**

Esse node utiliza modelos de linguagem para classificar um texto em categorias predefinidas.  
 No seu caso, é usado para **ler e classificar o conteúdo de um e-mail** (vindo do Gmail Trigger) em categorias como:

* Potencial Cliente
* Aluno da Jornada
* Spam ou Irrelevante
* E-mail Interno ou Importante

💡 *Por trás dos panos*, ele provavelmente utiliza embeddings (como text-embedding-3-small) e compara a similaridade entre o texto recebido e as descrições das categorias fornecidas.

#### **Categoria 1: Potencial Cliente**

Descrição: E-mails com interesse comercial, propostas de parceria, orçamento, agendamento de reunião ou dúvidas sobre serviços da Red Rex ou ZapFlow. Palavras-chave que ajudam na classificação: orçamento, proposta, parceria, reunião, consultoria, projeto, serviço, empresa

#### **Categoria 2: Aluno da Jornada**

Mensagens de alunos com dúvidas sobre conteúdo, acesso à plataforma, problemas com pagamento, ou suporte geral da Jornada de Dados. Palavras-chave: acesso, curso, jornada, plataforma, pagamento, aula, inscrição, certificado

#### **Categoria 3: Spam ou Irrelevante**

Descrição: E-mails genéricos, promoções automáticas, listas de e-mails ou conteúdo não solicitado. Palavras-chave: oferta, promoção, black friday, ganhe dinheiro, última chance, sorteio, unsubscribe

#### **Categoria 4: E-mail Interno ou Importante**

Descrição: E-mails com informações críticas, avisos do time, problemas técnicos ou alertas de sistemas internos. Palavras-chave: urgente, falha, erro, sistema, produção, equipe, reunião, alerta, interno

### 

### **2.2 Adicione um campo no Options: System Prompt Template**

* Pode usar o prompt padrão ou customizado com instruções claras para a IA

## **Etapa 3: Adicionar Label**

### **1.1 Adicione o nó Gmail Labbel no n8n**

* Resource: Message
* Operation: Add Label
* Message ID: {{ $('Gmail Trigger').item.json.id }}
* Label Names or IDs (ex: Potencial Cliente)

Faça isso 4 vezes.

### **1.2 Adicione o nó Mark as read**

Faça isso 4 vezes.

### **Categoria 1: Potencial Cliente**

**Assunto:** Solicitação de orçamento para projeto de automação de dados  
**Corpo do e-mail:**

Olá, equipe da RedRex!

Estamos avaliando consultorias para um novo projeto de integração de dados aqui na empresa e gostaríamos de receber uma proposta comercial de vocês.

Vocês poderiam enviar um orçamento estimado e, se possível, agendar uma reunião ainda esta semana?

Aguardo o retorno,

Carlos Mendes – Diretor de Tecnologia

### 

### **Categoria 2: Aluno da Jornada**

**Assunto:** Dúvida sobre certificado do curso  
**Corpo do e-mail:**

Bom dia!

Finalizei o curso de Python para Dados na Jornada e ainda não recebi o certificado. Também estou com dificuldade para acessar o conteúdo da aula bônus que vocês comentaram na última monitoria.

Poderiam me ajudar com isso?

Obrigado,

Fernanda Nunes

### 

### **Categoria 3: Spam ou Irrelevante**

**Assunto:** Oferta imperdível só hoje! 70% de desconto  
**Corpo do e-mail:**

Ei Luciano!

Aproveite agora a nossa SUPER PROMOÇÃO de Black Friday! Descontos de até 70% em todos os nossos produtos de inteligência artificial.

Clique aqui para garantir: ganhedinheiroagora.ai/promocao

Não perca essa última chance do ano!

Descadastre-se aqui

### 

### **Categoria 4: E-mail Interno ou Importante**

**Assunto:** ⚠️ Falha no agendamento dos dashboards de produção  
**Corpo do e-mail:**

Luciano, bom dia.

Detectamos uma falha na rotina de atualização dos dashboards de produção na madrugada de hoje. O Airflow não conseguiu rodar o DAG das 2h e os dados estão desatualizados no painel do cliente.

A equipe de engenharia já está analisando. Avisamos assim que estiver normalizado.

Abraço,

Gustavo (time de operações)

## 

## **Etapa 5: Agente de Pré-Vendas (IA que transforma e-mails em propostas estruturadas)**

### **5.1 Adicione o nó Agent – Agente Pré-vendas**

Tipo: AI Agent (Langchain)

Prompt:

Meu título:{{ $('Gmail Trigger').item.json.subject }}

Meu texto: {{ $('Gmail Trigger').item.json.text }}

Email de quem enviou: {{ $('Gmail Trigger').item.json.from.value[0].address }}

Nome de quem enviou: {{ $('Gmail Trigger').item.json.from.value[0].name }}

Data de envio: {{ $('Gmail Trigger').item.json.date }}

System Message:

Você é um engenheiro de dados responsável por analisar demandas técnicas recebidas por e-mail, WhatsApp ou formulário. Sua função é interpretar essas solicitações e gerar um JSON com informações técnicas e comerciais para uso em CRM e pré-vendas.

Você deve gerar um único objeto JSON, com todos os campos em formato texto corrido, sem listas ou arrays, mesmo quando houver múltiplos itens.

Além dos campos técnicos e comerciais, inclua três perguntas importantes que devem ser feitas na primeira reunião com o cliente, para melhor entendimento do projeto.

---

O JSON deve conter os seguintes campos:

- resumo: descrição técnica resumida da demanda

- escopo: etapas técnicas propostas, descritas em texto corrido

- estimativa\_de\_esforco: tempo estimado por fase do projeto (também em texto corrido)

- stack\_sugerida: ferramentas/tecnologias recomendadas, separadas por vírgula

- nome: nome do solicitante

- email: e-mail do solicitante

- data\_de\_recebimento: data da solicitação no formato ISO

- motivo\_do\_contato: qual foi o motivo inicial da solicitação

- escopo\_resumido: escopo técnico em uma frase

- estimativa\_resumida: tempo total estimado do projeto

- riscos\_resumidos: possíveis riscos ou obstáculos

- perguntas\_para\_reuniao: três perguntas que devem ser feitas na primeira conversa, em uma única string com pontuação adequada

Adicionalmente, gere dois novos campos no final do JSON:

- subject: assunto do e-mail de resposta automática

- mensagem\_email: corpo do e-mail de resposta (em texto corrido), utilizando o nome do solicitante e o resumo do projeto

Use um tom profissional e acolhedor.

Exemplo esperado:

"subject": "Sobre sua solicitação – Projeto de Dados",

"mensagem\_email": "Olá João da Silva,\n\nRecebemos sua solicitação e já estamos avaliando os detalhes técnicos do seu projeto. Nossa equipe de engenharia de dados preparou uma proposta inicial com base nas informações enviadas. Em breve, entraremos em contato para alinhar próximos passos.\n\nAtenciosamente,\nEquipe Red Rex\n\n---\nResumo do pedido: [resumo aqui]"

---

\*\*Formato de resposta esperado: apenas um JSON. Não inclua explicações ou quebras de linha.\*\*

* Marque a opção Require Specific Output Format

### **5.2 Conecte ao modelo OpenAI Chat Model**

* Modelo: gpt-4o-mini
* Crie a credencial com a API Key do OpenAI

### **5.3 Conecte o Tool: Think e o OutputParser Structured**

### **Think (Tool de Agente de IA)**

O Think é uma **ferramenta auxiliar (tool)** disponível para o agente de IA.  
 Ele permite que o agente "pense" ou processe lógica adicional, simule raciocínios ou recupere informações antes de tomar uma ação.

Exemplo de uso:

Se o agente precisa gerar um plano com base em uma solicitação, ele pode usar o Think para simular a quebra da demanda em etapas antes de responder ou gerar um JSON de escopo.

É como um "plugin" de reflexão dentro do cérebro do agente.

### 

### **Structured Output Parser (Tool de Agente de IA)**

Esse node é essencial quando queremos que o agente de IA **gere uma resposta em formato JSON estruturado**.  
 Você fornece um **exemplo de JSON (schema)**, e o Output Parser valida se a resposta do agente bate com a estrutura.

Ele:

* Força o agente a responder exatamente naquele formato
* Facilita o uso posterior dos dados (ex: salvar no Google Sheets, CRM, etc)

💡 *Fundamental em casos de pré-vendas, atendimento, CRM, etc.*

* O parser garante que a saída siga o formato JSON esperado

No OutputParser Structured colocar

{

"resumo": "Cliente deseja automatizar relatórios de vendas diários, integrando os dados do ERP interno a um sistema de dashboards atualizados automaticamente.",

"escopo": "Coleta de dados via API REST, armazenamento em BigQuery, transformação com dbt e visualização com Looker Studio.",

"estimativa\_de\_esforco": "Coleta: 5 dias, Transformação: 3 dias, Visualização: 2 dias.",

"stack\_sugerida": "Airbyte, dbt, BigQuery, Looker Studio.",

"nome": "João da Silva",

"email": "joao@empresa.com",

"data\_de\_recebimento": "2025-07-03T14:00:00Z",

"motivo\_do\_contato": "Solicitou automação de relatórios de vendas recorrentes.",

"escopo\_resumido": "Pipeline automatizado com dashboard integrado.",

"estimativa\_resumida": "3 semanas de duração total estimada.",

"riscos\_resumidos": "Instabilidade da API, ausência de documentação do ERP e possíveis limitações de acesso.",

"perguntas\_para\_reuniao": "1. O ERP atual possui documentação da API e acesso técnico disponível? 2. Existe um padrão de atualização e limpeza dos dados que devemos seguir? 3. Qual ferramenta de visualização é preferida pelo time de tomada de decisão?",

"subject": "Sobre sua solicitação – Projeto de Dados",

"mensagem\_email": "Olá João da Silva,\n\nRecebemos sua solicitação e já estamos avaliando os detalhes técnicos do seu projeto. Nossa equipe de engenharia de dados preparou uma proposta inicial com base nas informações enviadas. Em breve, entraremos em contato para alinhar próximos passos.\n\nAtenciosamente,\nEquipe Red Rex\n\n---\nResumo do pedido: [resumo aqui]"

}

### **5.4 Armazene os dados no Google Sheets (CRM RedRex)**

* Use o node Append row in sheet
* Colunas: resumo, escopo, nome, email, data de recebimento, stack, estimativa, perguntas, etc.

### **5.5 Geração automática de resposta (opcional)**

* Use o nó Create Draft (Gmail)
* Use os campos subject e mensagem\_email gerados pelo agente
* A resposta vai com nome personalizado, tom acolhedor e resumo técnico automático

## **Fluxo: Agente que responde automaticamente dúvidas de alunos da Jornada de Dados**

### **🔹1. Adicionar Label de Alunos (nó Gmail)**

**Função:** Marca o e-mail recebido com uma label no Gmail (ex: "Aluno Jornada").

* **Entrada:** Um e-mail novo (gatilho externo).
* **Ação:** Usa messageId do email para adicionar uma Label específica.
* **Objetivo:** Organizar os e-mails e garantir que esse e-mail será processado como uma dúvida de aluno.

### **🔹2. Node de Agente de IA**

**Função:** Fornece o modelo de linguagem (neste caso, o gpt-4o-mini) que será usado pelo agente.

* **Prompt:**

From: {{ $('Gmail Trigger').item.json.headers.from }}

Assunto: {{ $('Gmail Trigger').item.json.headers.subject }}

Mensagem: {{ $('Gmail Trigger').item.json.text }}

* System Message
* Structured Output

{

"subject": "Sobre sua solicitação – Projeto de Dados",

"mensagem\_email": "Olá João da Silva,\n\nRecebemos sua solicitação e já estamos avaliando os detalhes técnicos do seu projeto. Nossa equipe de engenharia de dados preparou uma proposta inicial com base nas informações enviadas. Em breve, entraremos em contato para alinhar próximos passos.\n\nAtenciosamente,\nEquipe Red Rex\n\n---\nResumo do pedido: [resumo aqui]"

}

### **🔹3. Think2 (Tool)**

**Função:** Permite ao agente “raciocinar” antes de responder.  
 Pode ser útil para construir respostas mais detalhadas ou validar a intenção do usuário.

* **Uso complementar**, especialmente se houver múltiplos caminhos de resposta.

### **🔹4. Structured Output Parser1**

**Função:** Força a **saída da IA** a obedecer um JSON padronizado.

* Garante que a IA produza dois campos principais:  
  + subject: título do e-mail de resposta
  + mensagem\_email: corpo do e-mail com a resposta para o aluno
* **Importante:** evita que a IA responda com texto solto ou fora do padrão.

### **🔹5. Agente Respondendo Aluno (Agente de IA)**

**Função:** É quem realmente **interpreta a dúvida do aluno e gera a resposta automatizada**.

* **Prompt:** Ele lê os campos do e-mail original (remetente, assunto, corpo) e segue um *system message* simples:  
    
    
   "Você responde a dúvida do aluno e responde seguindo o JSON especificado"
* **Integrações internas:**
  + Usa o modelo do OpenAI Chat Model3
  + Usa a ferramenta de raciocínio Think2
  + Usa o parser Structured Output Parser1

### **🔹6. Send a message (nó Gmail)**

**Função:** Envia automaticamente a resposta para o aluno.

* **Para:** o mesmo remetente do e-mail original
* **Assunto:** gerado pela IA
* **Mensagem:** gerada pela IA no campo mensagem\_email

**Fluxo: Agente que responde no whatsapp mensagens que são urgente**

1. Criar um fluxo de Agente

Prompt:

Data de recebimento: {{ $json.headers.date }}

From: {{ $json.headers.from }}

Mensagem: {{ $json.text }}

System Message:

Você é um agente inteligente que ajuda a classificar e resumir e-mails.

Sua tarefa é:

1. Ler o conteúdo do e-mail fornecido.

2. Analisar se a mensagem é urgente, com base em termos como: "erro", "cancelamento", "problema", "atraso", "urgente", "reclamação", "falha", "responder hoje", "última tentativa".

3. Classificar a mensagem como:

- URGENTE: exige atenção imediata–

- NÃO URGENTE: pode ser lida depois

- IRRELEVANTE: spam, promoções, mensagens automáticas

4. Criar uma resposta curta para ser enviada via WhatsApp, contendo:

- A classificação de urgência

- Um resumo breve (1 a 2 frases) do e-mail

- Um campo “ação sugerida” caso for URGENTE

Não usar campos especiais ou quebras como \n

1. Criar um node de HTTP request Post

**Desafios:**

### **1. Dashboard de Indicadores de Suporte**

**Objetivo:** mostrar como o agente não só responde, mas gera inteligência.

**Como fazer:**

* Use Google Sheets para salvar:  
  + Categoria do e-mail
  + Data
  + Nome do aluno
  + Se foi respondido automaticamente
* Crie um gráfico no Google Sheets ou Data Studio:  
  + Volume por categoria
  + Tempo médio de resposta
  + % de automação vs manual

**Aprendizado:** Mostra o impacto real dos agentes no atendimento e na gestão da operação.

### **2. Revisão de Mensagens Human-in-the-Loop**

**Objetivo:** introduzir o conceito de revisão e aprendizado contínuo.

**Como fazer:**

* Para e-mails que o agente não tem confiança alta ou que contenham certas palavras-chave, redirecione para revisão humana (via label ou Slack).
* Mostre como usar um nó “If” com base na confiança do modelo (pode ser hardcoded no início).

**Aprendizado:** Fala sobre responsabilidade, controle de qualidade e como combinar IA + humano de forma estratégica.

### **3. Como trabalhar com Anexos**

### **4. Como enviar mensagens para grupo**

### **5. Mandar mensagem para Microsoft Teams**

### **6. Consigo com o meu servidor de email local smtp**

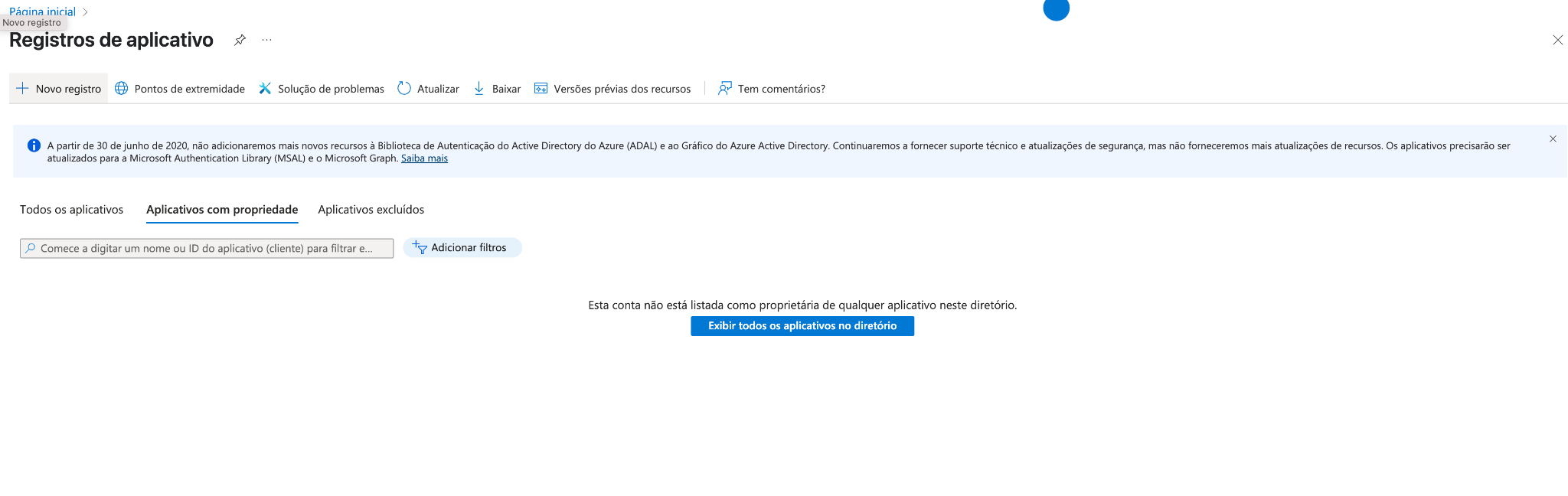
### **7. Fazer o Excel funcionar (Luciano que fez)**

### 

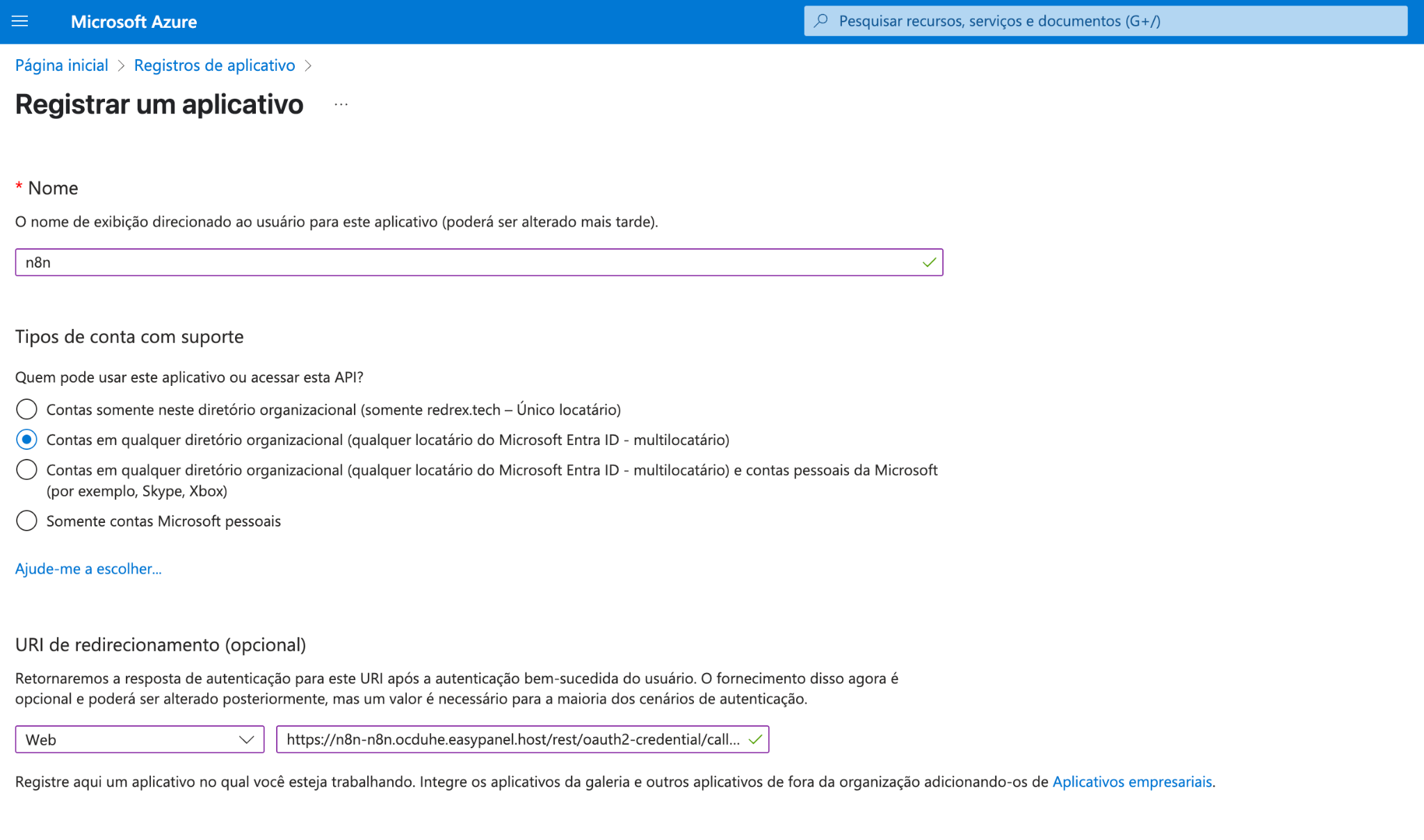
Extra:

Registro na Microsoft

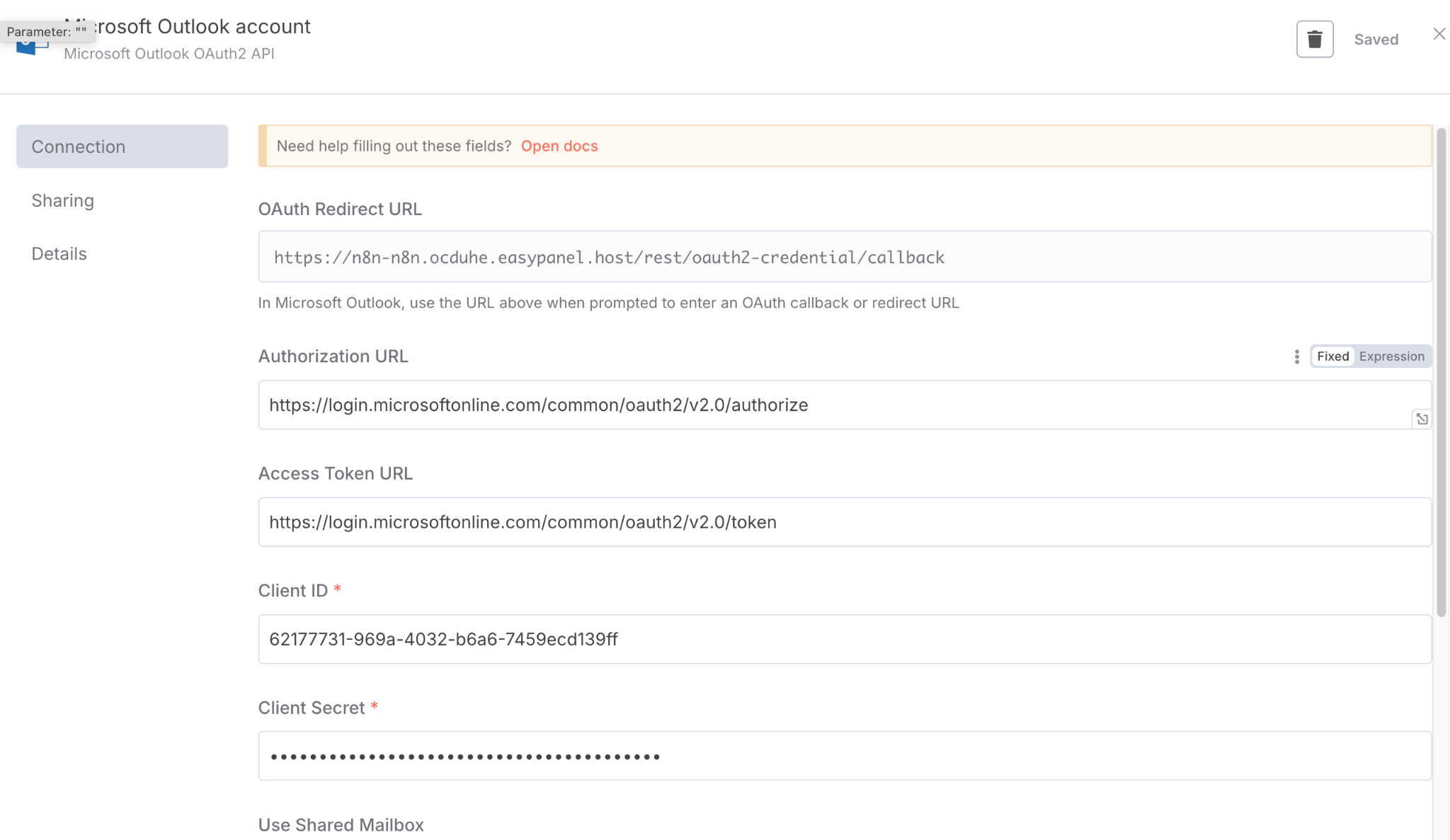
1. **Acesse o** [**Portal de Registro de Aplicativos da Microsoft.**](https://portal.azure.com/#view/Microsoft_AAD_RegisteredApps/ApplicationsListBlade)

****

1. **Clique em "Registrar um aplicativo".**



O valor do call callback vai pegar aqui



1. **Insira um nome para o seu aplicativo.**
2. Em **"Tipos de contas compatíveis"**, selecione:  
    **"Contas em qualquer diretório organizacional (Qualquer diretório do Azure AD - Multi-inquilino) e contas pessoais da Microsoft (por exemplo, Skype, Xbox)".**
3. Em **"Registrar um aplicativo"**:  
   * Copie a **URL de Callback OAuth** da sua credencial no n8n.
   * Cole no campo **URI de redirecionamento (opcional)**.
   * Selecione **"Selecionar uma plataforma" > "Web"**.
4. **Clique em "Registrar"** para finalizar a criação do aplicativo.
5. Copie o **ID do aplicativo (cliente)** e cole no n8n como o **Client ID**.

