### **Projeto 04) Criando um Sistema RAG com Google Drive & Supabase**

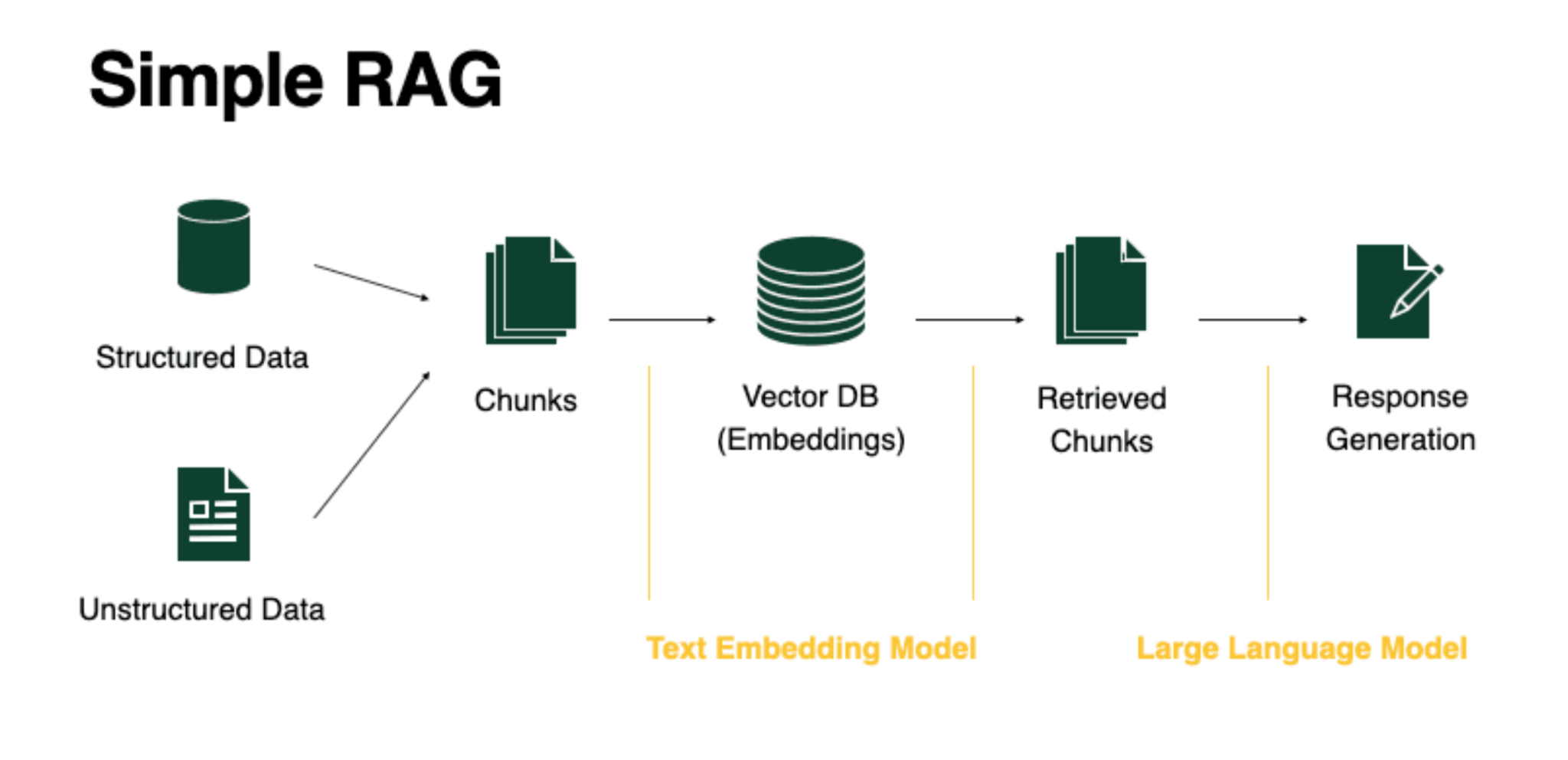
Arquivo:

Ingestão:

<https://www.dropbox.com/scl/fi/o56sa1gg54sggksy8zs7i/SaveFileToVectorStore.json?rlkey=zovjyaacf2vm0v8d6kycv471d&st=ver98cqy&dl=0>

Conversacional:

<https://www.dropbox.com/scl/fi/34s99bacoeo8xga5fm6wh/Agentes-de-IA.json?rlkey=84wg628z0ipr88jwigu1rp2sx&st=0pqlg2pj&dl=0>



**O que você vai aprender**

1. Integrar *Google Drive* com n8n
2. Usar *Supabase* como banco vetorial e memória Postgres
3. Criar um agente de IA que responde com *base nos seus documentos*
4. Implementar o conceito de *RAG (Retrieval-Augmented Generation)* do zero

**Etapa 1: Estrutura e Requisitos**

Ferramentas utilizadas:

* Supabase (PostgreSQL + Vector Store)
* Conta na OpenAI
* Documentos (Drive, Atas e planilhas)

Conecte sua conta do Google

1. Vá em Credenciais > OAuth2 no n8n

2. Preencha:

Authorization URL: https://accounts.google.com/o/oauth2/v2/auth

Token URL: https://oauth2.googleapis.com/token

**Etapa 2: Infraestrutura Inicial**

1. Criar projeto no Supabase (https://supabase.com/)

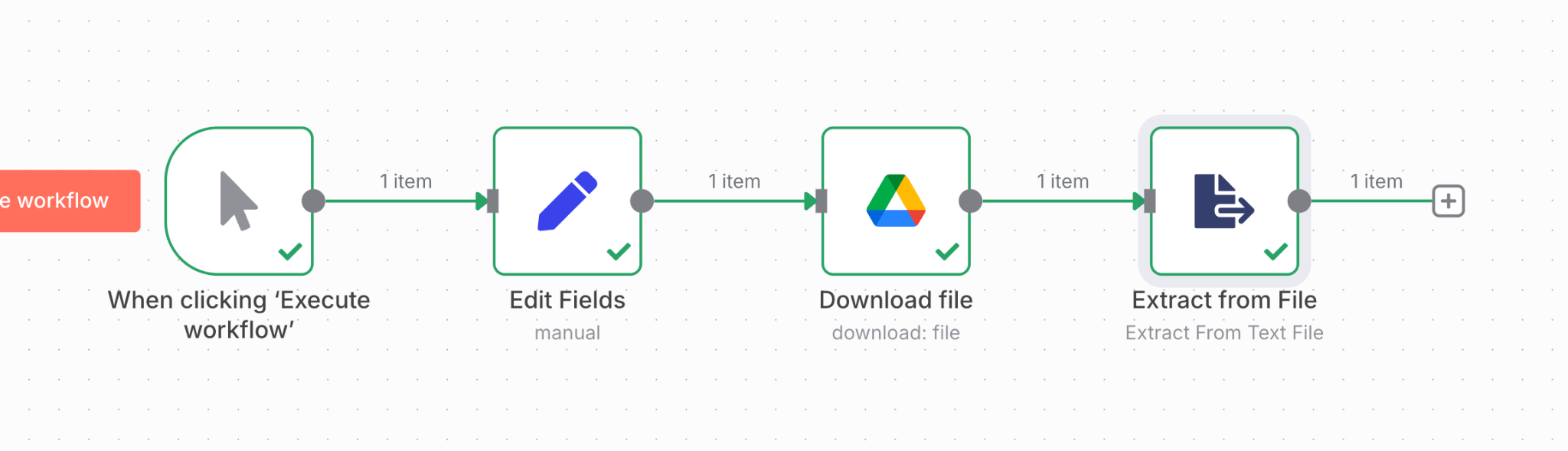
Crie um projeto chamado *rag-jornada*

Defina uma senha segura para o Postgres

**Etapa 3: Iniciando o Workflow para carregar Arquivos do Google Drive**

Crie um Workflow chamado **atualizacao\_banco\_vetorial**

Ele vai se parecer dessa forma depois de finalizado

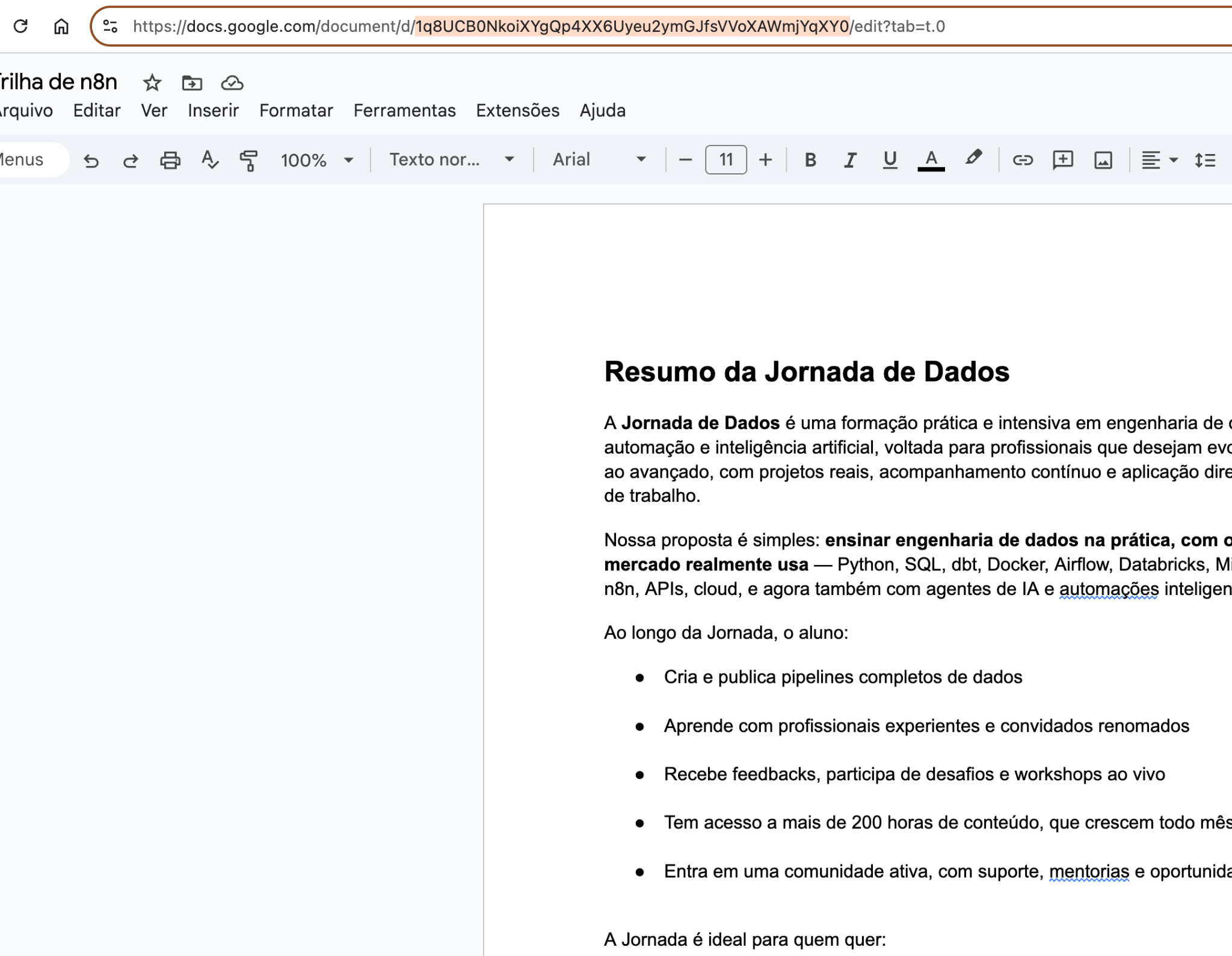


### **Node 1: Manual Trigger**

* **Nome do nó:** Manual Trigger
* **Função:** iniciar manualmente o fluxo para testes.

### **Node 2: Set File ID**

* **Tipo:** Set
* **Nome do nó:** Set File ID
* **Parâmetros:**
  + Mode: Manual Mapping
  + Values > file\_id → Ex: "1q8UCB0NkoiXYgQp4XX6Uye..."



### **Node 3: Google Drive > Download File**

* **Nome do nó:** Download Google Doc
* **Resource:** File
* **Operation:** Download
* **Parâmetros:**
  + **File ID:** ={{ $json["file\_id"] }}
  + **Options > Google File Conversion**
  + **Convert To:**
    - Google Docs → .txt
    - Google Sheets → .csv
    - Slides → .jpg
* **Função:** baixa o conteúdo do arquivo já convertido para texto.

### **Node 4: Extract File (Binary to Text)**

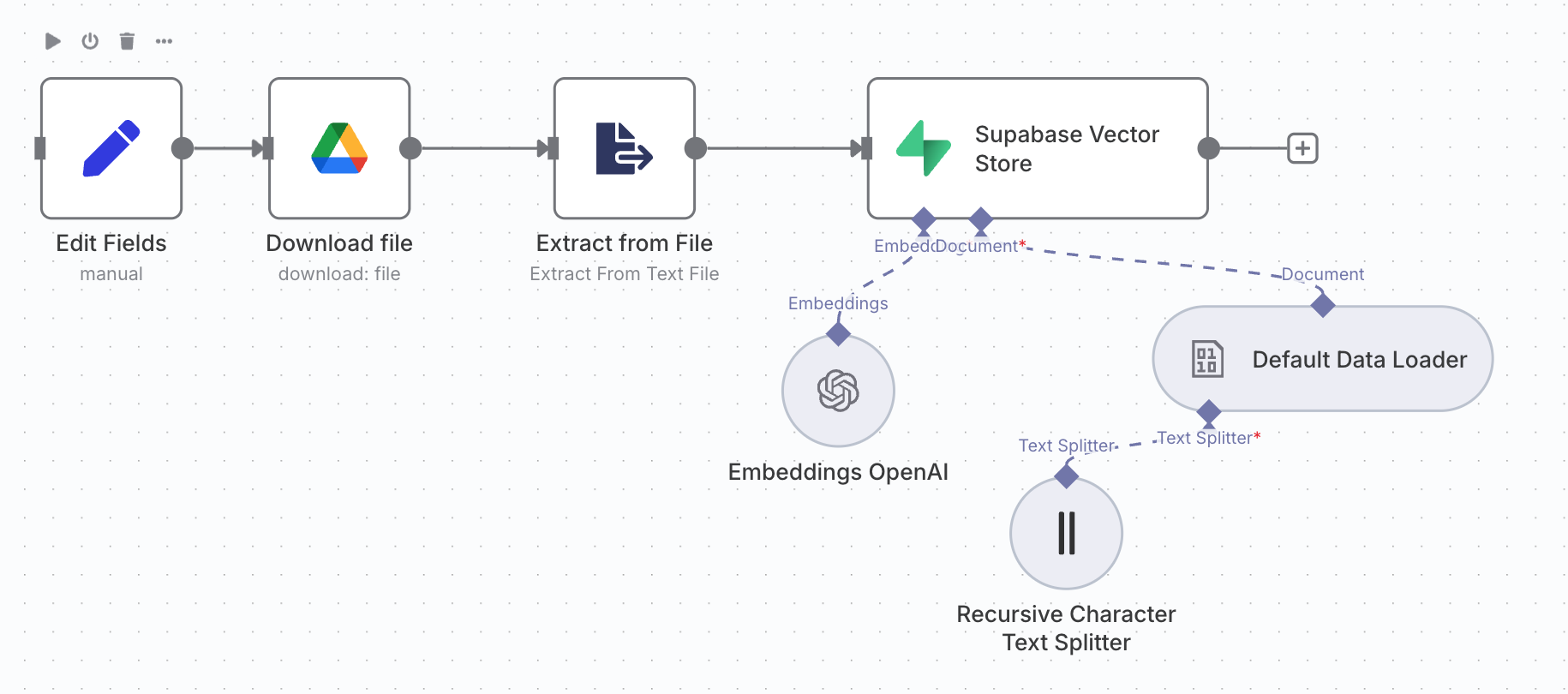
* **Nome do nó:** Extract Text from File
* **Tipo:** Move Binary Data
* **Input:** Data
* **Destination Output Field:** Data
* **Parâmetros:**
  + Mode: Binary to Text
* **Função:** transforma binário em texto plano para manipulação.

### **Agora queremos adicionar nosso processo de salvar no banco vetorial**

### 

### **Node 5: Supabase - Add Document to Vector Store**

Necessário criar um banco vetorial: [Banco vetorial](https://supabase.com/docs/guides/ai/langchain?queryGroups=database-method&database-method=sql)



Necessário criar um banco vetorial: [Banco vetorial](https://supabase.com/docs/guides/ai/langchain?queryGroups=database-method&database-method=sql)

* **Nome do nó:** Add Document to Vector Store
* **Tipo:** Supabase Vector Store
* **Parâmetros:**
  + **Operations Mode:** Insert Documents
  + **Table name:** Documents
  + **Options -> Query Name:** Match\_documents
* **Embeddings:** text-embedding-3-small

**Default Data Loader**

* **Mode:** Load Specific Data
* **Data:** {{ $json.data }}
* **Text Splitting**: Custom
* **Options -> Metadata**

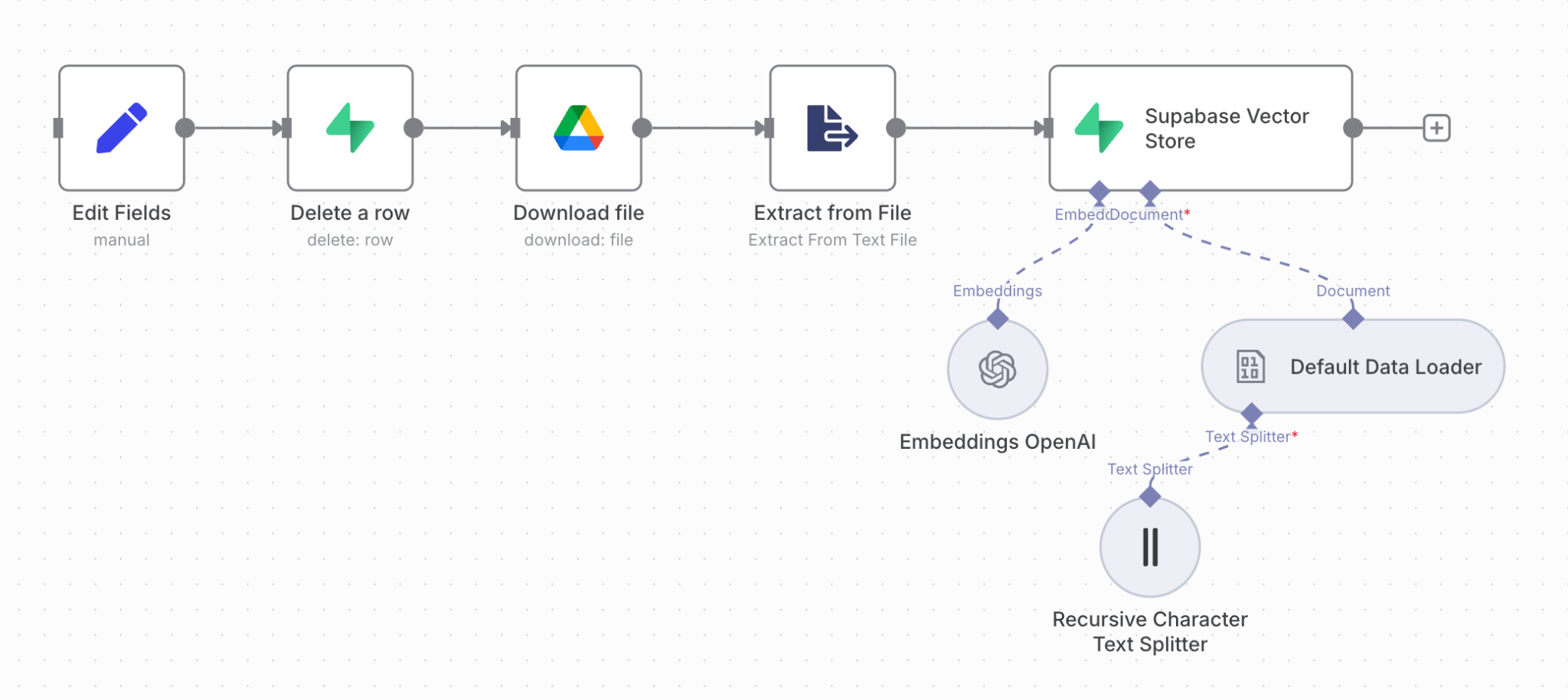
**name:** file\_id

**value:** {{ $('Edit Fields').item.json.file\_id }}

**Recursive Character Text Splitter** 1000 , 200

### 

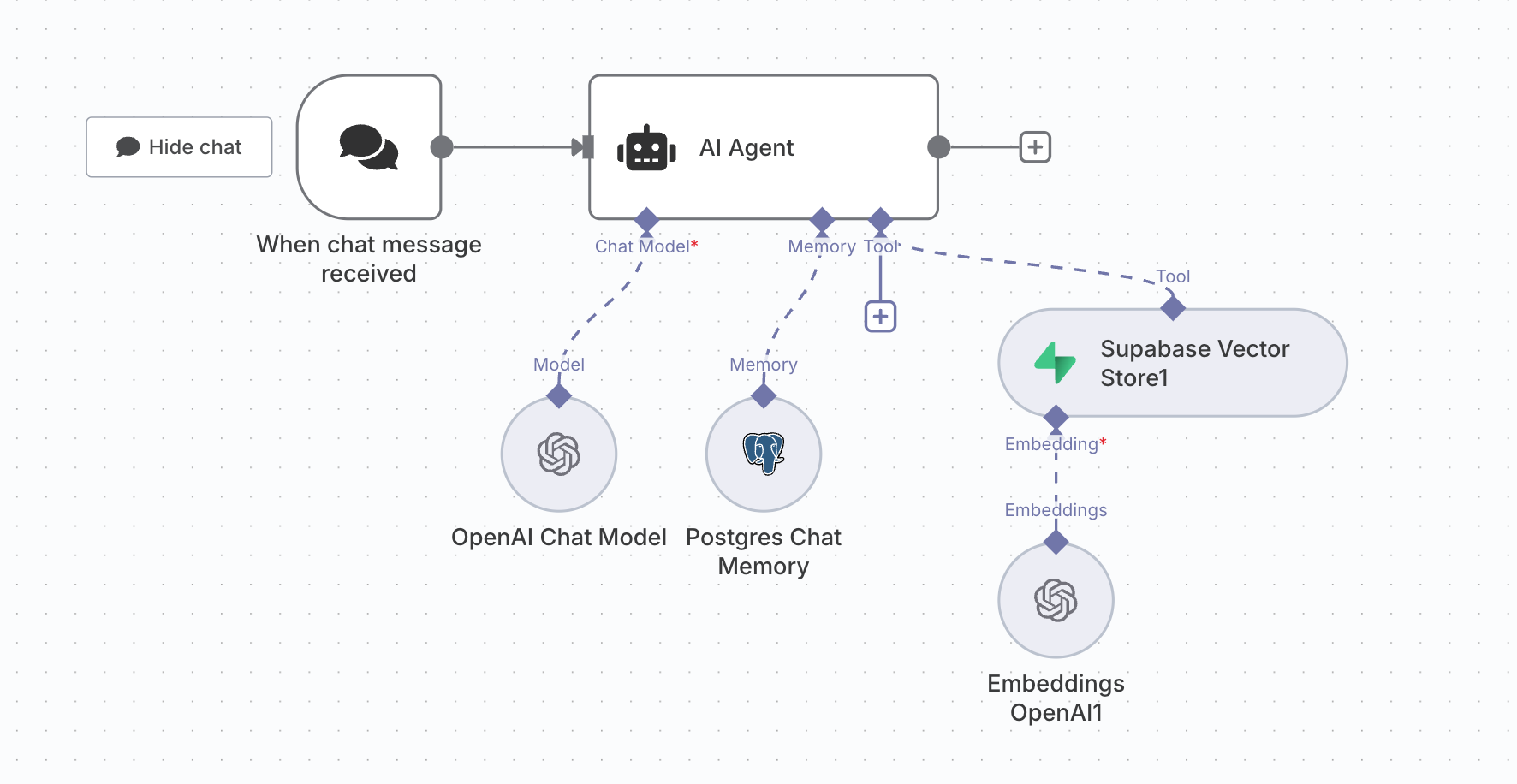
### **Node 6: Supabase - Delete Rows**



* **Nome do nó:** Delete Existing File from Vector Store
* **Resource:** Row
* **Operations:** Delete
* **Table name:** Documents
* **Select Type:** String
* **Filters:** metadata->>file\_id=eq.{{ $json.file\_id }}

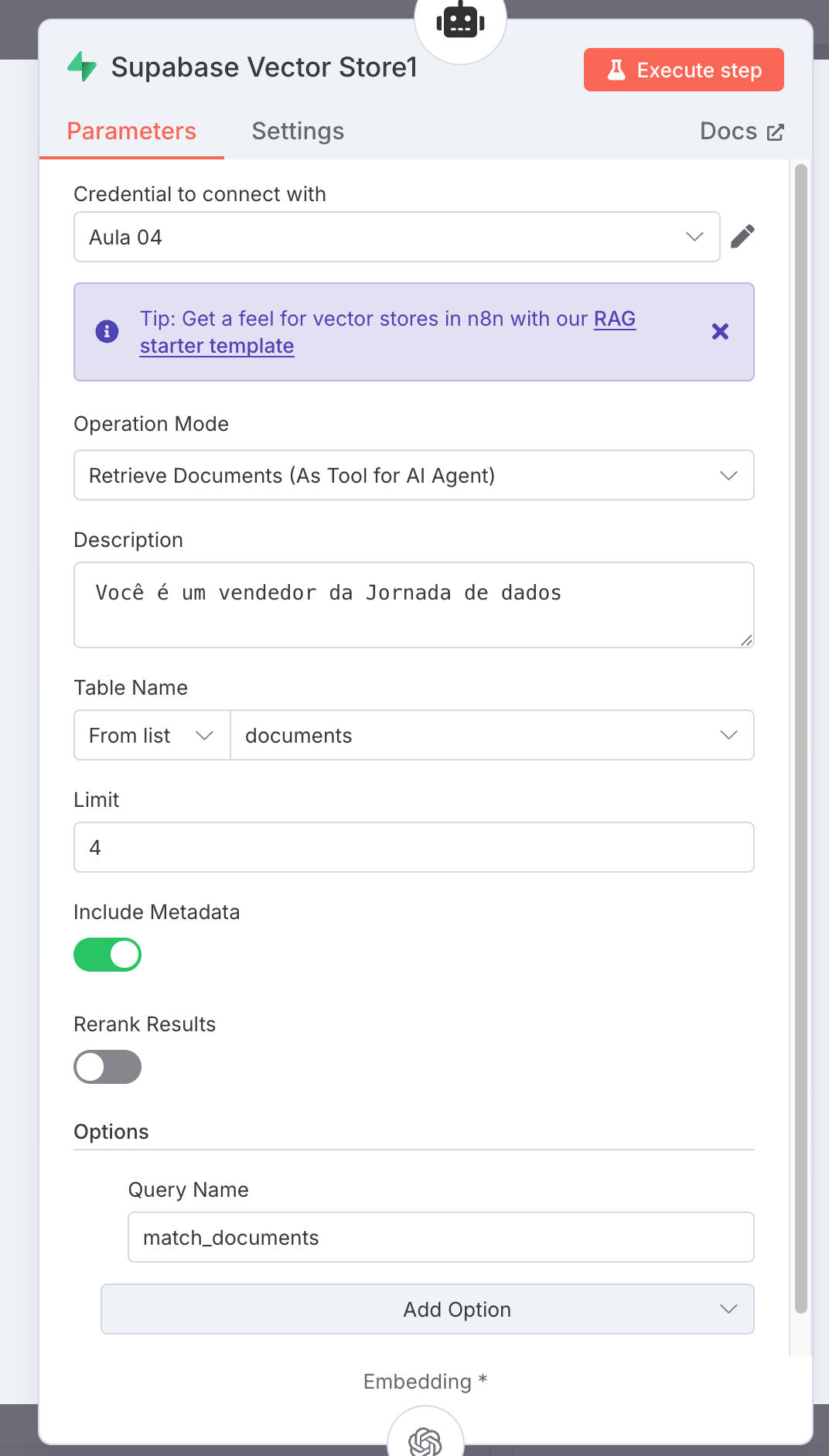
**Etapa 4: Criar o Agente Conversacional**

Crie um Workflow chamado **agente\_conversacional**

****

### **Node 7: Conversational Agent**

* **Nome do nó:** Chat with RAG Agent
* **Tipo:** AI Agent
  + **Parâmetros principais:**
  + **Model Provider:** OpenAI
  + **Model:** gpt-4
  + **Memory Type:** PostgreSQL (Supabase)
  + **Session ID: ={{ $json["session\_id"] || $executionId }}**
* **Supabase Vector Store**

****

## 

## **TESTE DE CONSULTA**

### **No chat integrado:**

* Teste com prompts:  
  + "O que é a Jornada de dados?"
  + "Quais as tecnologias tem na Jornada de dados?"

## **MONITORAR PASTA NO GOOGLE DRIVE**

### **Node 9: Google Drive Trigger**

* **Nome do nó: Files Updated Trigger**
* **Evento: File Updated**
* **Pasta: Anotações**
* **Função: dispara fluxo automaticamente ao atualizar arquivos.**