



Escuela superior politécnica de Chimborazo
Facultad de informática y electrónica



Escuela de software

Nombre:

Jeferson Vargas

Codigo:

6473

Asignatura:

Aplicaciones informáticas II

Docente:

Dr. Julio Santillan

Tema:

Manual técnico

Periodo Académico:

Marzo - Julio 2025

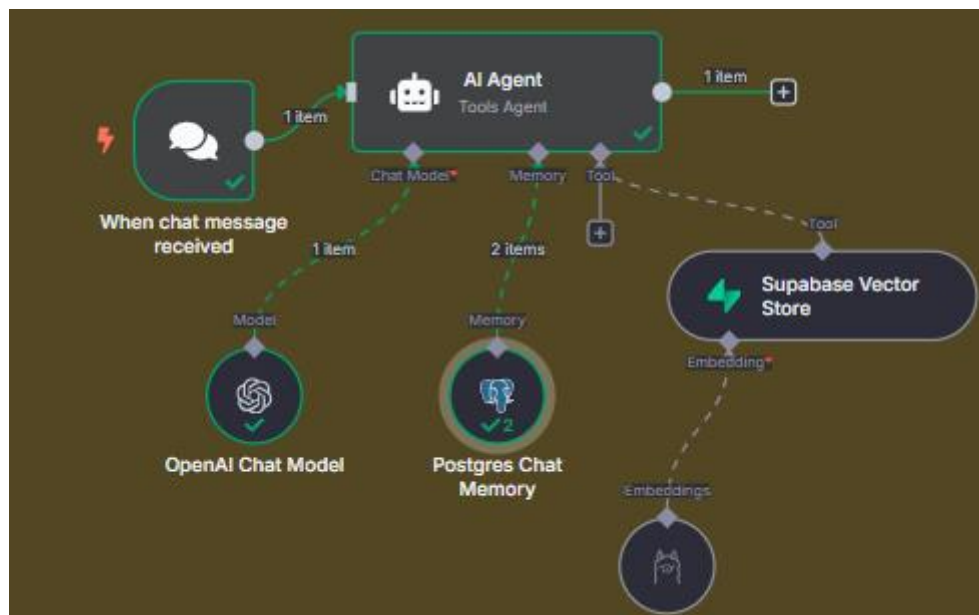
Manual Técnico del Flujo de Agente RAG y Actualización de Documentos

1. Introducción

Este documento describe dos flujos de trabajo interconectados en n8n:

- **Agente RAG (Retrieval-Augmented Generation):** Un chatbot que responde preguntas basándose en una base de conocimiento vectorial alojada en Supabase, utilizando un modelo de lenguaje grande (LLM) y un modelo de *embeddings* local.
- **Carga y Actualización de Documentos:** Un flujo automatizado para ingestar y actualizar documentos en la base de conocimiento vectorial, lo que permite al Agente RAG acceder a información relevante y actualizada.

2. Flujo del Agente RAG

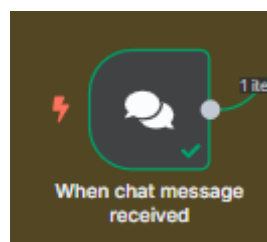


Este flujo gestiona las interacciones del chatbot con los usuarios, recuperando información de una base de datos vectorial para proporcionar respuestas.

2.1. Nodos y su Configuración

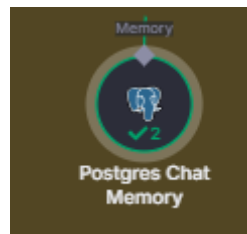
A continuación se detalla cada nodo del flujo del Agente RAG:

2.1.1. When chat message



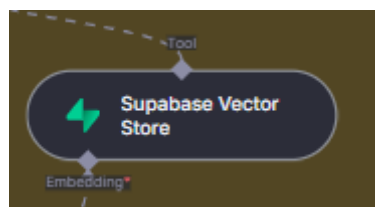
- **Tipo:** @n8n/n8n-nodes-langchain.chatTrigger
- **Descripción:** Este nodo actúa como el punto de entrada del chatbot. Escucha los mensajes de chat entrantes.
- **Parámetros:**
 - public: true (El chatbot es accesible públicamente).
 - initialMessages: "Hola soy Polito. ¿En qué puedo ayudarte hoy?" (Mensaje de bienvenida inicial del chatbot).
 - webhookId: 55796d5f-a64d-40e0-b614-db9ae5be5d1a (Identificador del webhook para la comunicación).
- **Credenciales:** No aplica (es un trigger).

2.1.2. Postgres Chat Memory



- **Tipo:** @n8n/n8n-nodes-langchain.memoryPostgresChat
- **Descripción:** Almacena el historial de conversación en una base de datos PostgreSQL, permitiendo al chatbot recordar interacciones previas con el usuario.
- **Credenciales:** Agente_RAG (ID: 7VZttiYrjMesrD2J) - Credenciales para la conexión a la base de datos PostgreSQL.

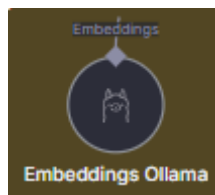
2.1.3. Supabase Vector Store



- **Tipo:** @n8n/n8n-nodes-langchain.vectorStoreSupabase
- **Descripción:** Este nodo es una herramienta de recuperación de documentos. Busca y recupera fragmentos de documentos de la base de conocimiento de la ESPOL basada en la similitud semántica con la consulta del usuario.

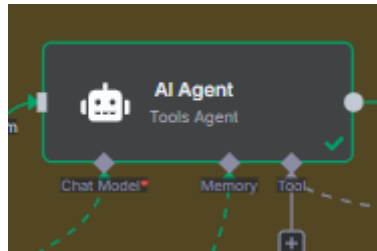
- **Parámetros:**
 - mode: retrieve-as-tool (Configurado para ser usado como una herramienta dentro del agente).
 - toolName: "Epoch" (Nombre de la herramienta que el Agente AI utilizará para la recuperación).
 - toolDescription: "Útil para buscar y recuperar fragmentos de documentos específicos y relevantes de la base de conocimiento de la ESPO, basándose en la similitud semántica con la consulta del usuario. Proporciona el contenido de los documentos más pertinentes para responder preguntas." (Descripción de la función de la herramienta).
 - tableName: documents (Tabla en Supabase donde se almacenan los vectores de los documentos).
- **Credenciales:** Agente_RAG (ID: E92sXoJhcVYQLZBC) - Credenciales para la conexión a la API de Supabase.

2.1.4. Embeddings Ollama



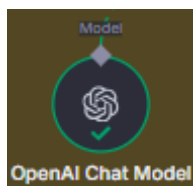
- **Tipo:** @n8n/n8n-nodes-langchain.embeddingsOllama
- **Descripción:** Genera *embeddings* (representaciones vectoriales numéricas) de texto utilizando el modelo nomic-embed-text:latest de Ollama. Estos *embeddings* son utilizados para la búsqueda por similitud en la base de datos vectorial.
- **Parámetros:**
 - model: nomic-embed-text:latest (Modelo de *embeddings* a utilizar).
- **Credenciales:** Local Ollama service (ID: xHuYe0MDGOs9lpBW) - Credenciales para la conexión al servicio local de Ollama.

2.1.5. AI Agent

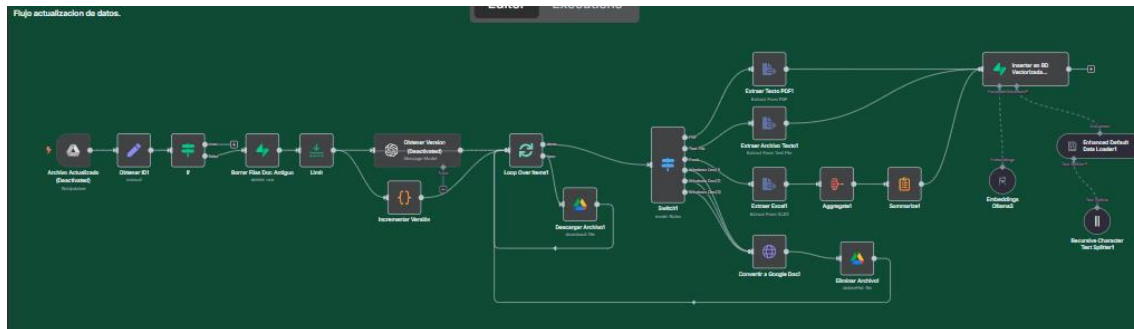


- **Tipo:** @n8n/n8n-nodes-langchain.agent
- **Descripción:** Este nodo es el cerebro del chatbot. Utiliza el modelo de lenguaje (LLM) y las herramientas disponibles (como Espoch para la recuperación de documentos) para comprender las consultas del usuario y generar respuestas.
- **Parámetros:**
 - **systemMessage:** Define el rol y las directrices estrictas del agente: "Eres un asistente de investigación especializado en documentos de la ESPOCH. Tu objetivo principal es proporcionar respuestas precisas y concisas **EXCLUSIVAMENTE basadas en la información consultada en la base de datos de documentos proporcionada por la herramienta 'Espoch'**. Si la pregunta no puede ser respondida con la información disponible, indica claramente: 'Lo siento, no tengo información sobre eso en los documentos que puedo consultar.' Nunca inventes información. Si el usuario solicita un enlace o descarga y la herramienta 'Espoch' recupera un documento con file_id, proporciona el enlace en el formato [https://drive.google.com/uc?export=download&id=\[ID_DEL_ARCHIVO_DE_GOOGLE_DRIVE\]](https://drive.google.com/uc?export=download&id=[ID_DEL_ARCHIVO_DE_GOOGLE_DRIVE])."

2.1.6. OpenAI Chat Model



- **Tipo:** @n8n/n8n-nodes-langchain.lmChatOpenAi
- **Descripción:** Proporciona el modelo de lenguaje grande (LLM) que el Agente AI utiliza para procesar el lenguaje natural y generar respuestas.
- **Parámetros:**
 - **model:** deepseek/deepseek-chat-v3-0324:free (Modelo de chat utilizado).



Este flujo automatiza la ingesta inicial y la posterior actualización de documentos en la base de datos vectorial de Supabase, asegurando que el Agente RAG siempre tenga acceso a la información más reciente.

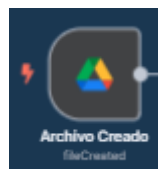
3.1. Visión General

El flujo se divide en dos secciones principales:

- **Carga de archivos primera vez:** Procesa documentos recién creados en Google Drive.
- **Actualización de datos:** Monitorea y procesa documentos modificados en Google Drive.

3.2. Nodos y su Configuración (Carga de Archivos Primera Vez)

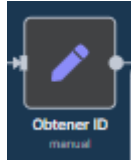
3.2.1. Archivo Creado



- **Tipo:** n8n-nodes-base.googleDriveTrigger
- **Descripción:** Este nodo es el punto de inicio para la ingesta de nuevos documentos. Se dispara cada minuto si se habilita, y detecta cuando un archivo es *creado* en la carpeta especificada de Google Drive.
- **Parámetros:**
 - pollTimes: everyMinute
 - triggerOn: specificFolder
 - folderToWatch: DocumentacionP1 (ID: 1AmnBHTwTXR1FFykXNZKyyZ0j4Abx-lj0)
 - event: fileCreated

- disabled: true (Actualmente deshabilitado; debe habilitarse para su funcionamiento).
- **Credenciales:** AgenteRAG (ID: 5BLbqJdcAeFAnoL6) - Credenciales OAuth2 para Google Drive.

3.2.2. Obtener ID



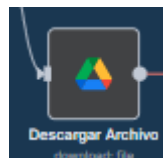
- **Tipo:** n8n-nodes-base.set
- **Descripción:** Extrae el file_id y file_type del archivo recién creado, y también su name y webViewLink, para usarlos en nodos posteriores.
- **Parámetros:**
 - file_id:={{ \$json.id }}
 - file_type:={{ \$('Archivo Creado').item.json.mimeType }}
 - name:={{ \$json.name }}
 - webViewLink:={{ \$json.webViewLink }}

3.2.3. Loop Over Items



- **Tipo:** n8n-nodes-base.splitInBatches
- **Descripción:** Divide la entrada en lotes para procesar cada archivo individualmente.

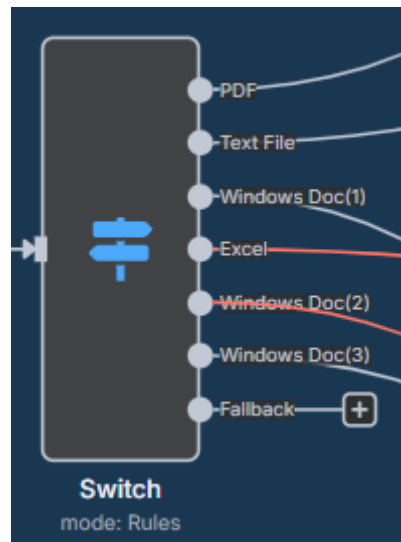
3.2.4. Descargar Archivo



- **Tipo:** n8n-nodes-base.googleDrive
- **Descripción:** Descarga el contenido del archivo desde Google Drive. Los archivos de Google Docs se convierten automáticamente a texto plano.

- **Parámetros:**
 - operation: download
 - fileId: `={{ $json.file_id }}`
 - googleFileConversion: docsToFormat: text/plain
- **Credenciales:** AgenteRAG (ID: 5BLbqJdcAeFAnoL6)

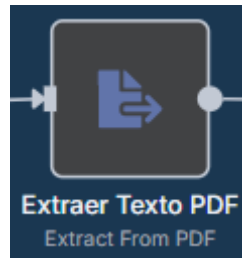
3.2.5. Switch



- **Tipo:** n8n-nodes-base.switch
- **Descripción:** Redirige el flujo basándose en el tipo MIME (file_type) del archivo.
- **Reglas:**
 - **PDF:** Si file_type es application/pdf.
 - **Text File:** Si file_type es application/vnd.google-apps.document (Google Docs).
 - **Excel:** Si file_type es application/vnd.openxmlformats-officedocument.spreadsheetml.sheet (Excel XLSX).
 - **Windows Doc(1):** Si file_type es application/vnd.openxmlformats-officedocument.wordprocessingml.document (Word DOCX).
 - **Windows Doc(2):** Si file_type es application/msword (Word DOC).
 - **Windows Doc(3):** Si file_type es application/vnd.ms-word (Word antiguo).

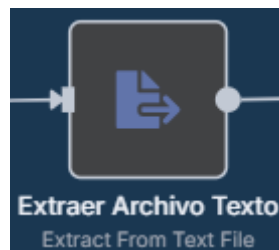
- **Fallback:** La salida 2 se usa como fallback (para tipos no especificados o para "Windows Doc(2)").

3.2.6. Extraer Texto PDF



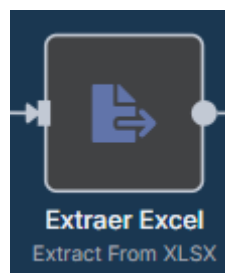
- **Tipo:** n8n-nodes-base.extractFromFile
- **Descripción:** Extrae texto de archivos PDF.
- **Parámetros:** operation: pdf

3.2.7. Extraer Archivo Texto



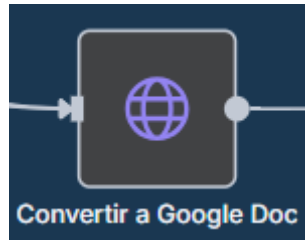
- **Tipo:** n8n-nodes-base.extractFromFile
- **Descripción:** Extrae texto de archivos de texto plano o Google Docs convertidos.
- **Parámetros:** operation: text

3.2.8. Extraer Excel



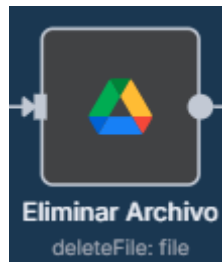
- **Tipo:** n8n-nodes-base.extractFromFile
- **Descripción:** Extrae datos de archivos Excel.
- **Parámetros:** operation: xlsx

3.2.9. Convertir a Google Doc



- **Tipo:** n8n-nodes-base.httpRequest
- **Descripción:** Convierte el archivo original (si no es Google Doc) en un Google Doc para facilitar la extracción de texto.
- **Parámetros:**
 - method: POST
 - url: `https://www.googleapis.com/drive/v3/files/{{ $('Obtener ID').item.json.file_id }}/copy`
 - mimeType: `application/vnd.google-apps.document`
- **Credenciales:** AgenteRAG (ID: 5BLbqJdcAeFAnoL6)

3.2.10. Eliminar Archivo



- **Tipo:** n8n-nodes-base.googleDrive
- **Descripción:** Elimina el archivo original después de la conversión a Google Doc para evitar duplicados o mantener el espacio.
- **Parámetros:**
 - operation: `deleteFile`
 - fileId: `={{ $('Obtener ID').item.json.file_id }}`
- **Credenciales:** AgenteRAG (ID: 5BLbqJdcAeFAnoL6)

3.2.11. Aggregate



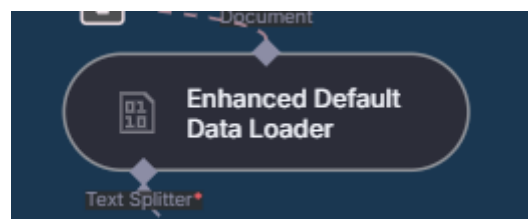
- **Tipo:** n8n-nodes-base.aggregate
- **Descripción:** Agrupa todos los datos de los ítems para una posterior concatenación.

3.2.12. Summarize



- **Tipo:** n8n-nodes-base.summarize
- **Descripción:** Concatena el contenido extraído de los documentos en un solo campo data (o similar) para el siguiente paso.
- **Parámetros:** aggregation: concatenate field: data

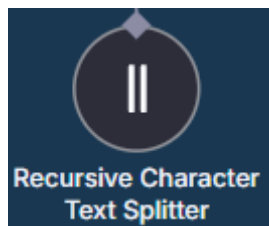
3.2.13. Enhanced Default Data Loader



- **Tipo:** @n8n/n8n-nodes-langchain.documentDefaultDataLoader
- **Descripción:** Prepara los datos del documento para ser cargados en la base de datos vectorial, añadiendo metadatos relevantes.
- **Parámetros:**
 - jsonData: ={{ \$json.data || \$json.text || \$json.concatenated_data }}
 - metadataValues:
 - file_id: ={{ \$('Obtener ID').item.json.file_id }}
 - version: v1 (Versión inicial del documento).

- `creator:={{ $('Archivo Creado').item.json.owners[0].displayName }}`
- `created_at:={{ $('Archivo Creado').item.json.createdTime }}`
- `last_modified:={{ $('Archivo Creado').item.json.modifiedTime }}`
- `folder_path: projects`
- `file_name:={{ $('Archivo Creado').item.json.name }}`
- `file_extension:={{ $('Archivo Creado').item.json.mimeType }}`

3.2.14. Recursive Character Text Splitter



- **Tipo:** @n8n/n8n-nodes-langchain.textSplitterRecursiveCharacterTextSplitter
- **Descripción:** Divide el texto largo de los documentos en fragmentos más pequeños (chunks) para facilitar la vectorización y la búsqueda.
- **Parámetros:**
 - `chunkSize`: 2000
 - `chunkOverlap`: 200
 - `splitCode`: markdown

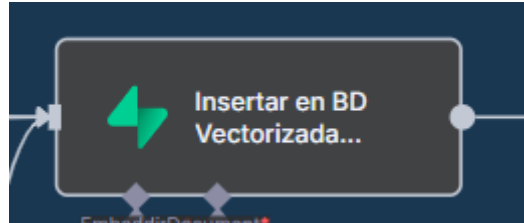
3.2.15. Embeddings Ollama2



- **Tipo:** @n8n/n8n-nodes-langchain.embeddingsOllama
- **Descripción:** Genera los *embeddings* para los fragmentos de texto, utilizando el mismo modelo `nomic-embed-text:latest` que el Agente RAG.
- **Parámetros:**

- model: nomic-embed-text:latest
- **Credenciales:** Local Ollama service (ID: xHuYe0MDGOs9IpBW)

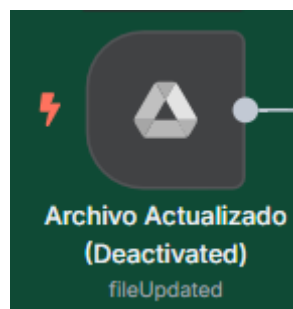
3.2.16. Insertar en BD Vectorizada Supabase



- **Tipo:** @n8n/n8n-nodes-langchain.vectorStoreSupabase
- **Descripción:** Inserta los *chunks* vectorizados y sus metadatos en la tabla documents de Supabase.
- **Parámetros:**
 - mode: insert
 - tableName: documents
 - queryName: match_documents
- **Credenciales:** RAG_agent (ID: K54wKWCWt6vluLzi)

3.3. Nodos y su Configuración (Actualización de Datos)

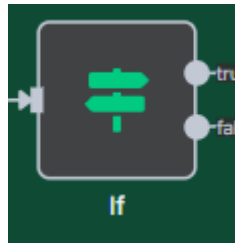
3.3.1. Archivo Actualizado



- **Tipo:** n8n-nodes-base.googleDriveTrigger
- **Descripción:** Este nodo es el punto de inicio para las actualizaciones de documentos. Se dispara cuando un archivo es *modificado* en la carpeta especificada de Google Drive.
- **Parámetros:**
 - pollTimes: everyMinute
 - triggerOn: specificFolder

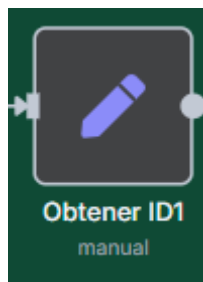
- folderToWatch: DocumentacionP1 (ID: 1AmnBHTwTXR1FFykXNZKyyZ0j4Abx-lj0)
- event: fileUpdated
- disabled: true (Actualmente deshabilitado; debe habilitarse para su funcionamiento).
- **Credenciales:** AgenteRAG (ID: 5BLbqJdcAeFAnoL6)

3.3.2. If



- **Tipo:** n8n-nodes-base.if
- **Descripción:** Condiciona el flujo para asegurar que solo se procesen documentos de Google Docs (application/vnd.google-apps.document) y que hayan sido creados o modificados en los últimos 60 segundos (lo cual parece un período muy corto para modificaciones, revisar si es el comportamiento deseado).
- **Condiciones:**
 - mimeType es igual a application/vnd.google-apps.document
 - El tiempo de creación (o modificación, según cómo se interprete createTime en este contexto tras la actualización) es menor a 60 segundos desde ahora.
- **onError:** continueRegularOutput

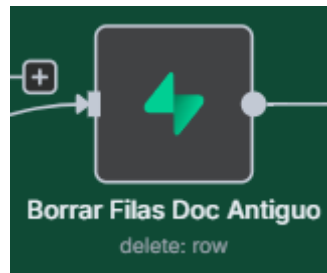
3.3.3. Obtener ID1



- **Tipo:** n8n-nodes-base.set

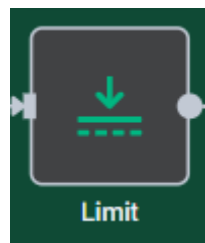
- **Descripción:** Extrae el file_id y file_type del archivo actualizado, similar al nodo "Obtener ID" pero para el flujo de actualización.
- **Parámetros:**
 - file_id: `={{ $('Archivo Actualizado').item.json.id }}`
 - file_type: `={{ $('Archivo Actualizado').item.json.mimeType }}`

3.3.4. Borrar Filas Doc Antiguo



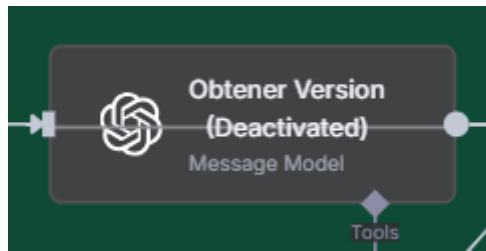
- **Tipo:** n8n-nodes-base.supabase
- **Descripción:** Elimina las entradas antiguas del documento en la base de datos de Supabase antes de insertar la nueva versión, basándose en el file_id.
- **Parámetros:**
 - operation: delete
 - tableId: documents
 - filterString: `=metadata->>file_id=like.*{{ $('Obtener ID1').item.json.file_id }}`*
- **Credenciales:** RAG_agent (ID: K54wKWCWt6vluLzi)

3.3.5. Limit



- **Tipo:** n8n-nodes-base.limit
- **Descripción:** Limita el número de ítems que pasan a los siguientes nodos. Puede ser útil para pruebas o para evitar sobrecargar el sistema.

3.3.6. Obtener Version



- **Tipo:** @n8n/n8n-nodes-langchain.openAi
- **Descripción:** Utiliza un LLM (gpt-4o-mini) para incrementar la versión del documento.
- **Parámetros:**
 - modelId: gpt-4o-mini
 - systemMessage: "Tu objetivo es tomar el número de versión entrante y sumarle 1. [Ejemplos] Input: v1 Output: v2. El output debe llamarse siempre 'version'".
 - content: =numero de version entrante: {{ \$json.metadata.version }}
 - jsonOutput: true
- **Credenciales:** OpenAi account (ID: xqERdUgeq2W1RqXO)
- **Estado:** disabled (Actualmente deshabilitado).

3.3.7. Incrementar Versión



- **Tipo:** n8n-nodes-base.code
- **Descripción:** Un nodo de código que incrementa una variable de versión (myNewField) en 1. Parece un nodo temporal o alternativo a "Obtener Version" ya que el "Obtener Versión" está deshabilitado.
- **Código JS:**

JavaScript

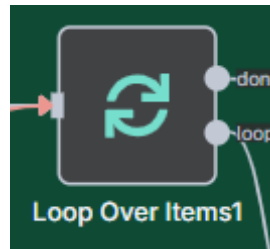
```
for (const item of $input.all()) {
```

```
item.json.myNewField = 1; // Este código establece myNewField a 1, no lo incrementa basándose en una versión anterior. Es probable que necesite ajustarse.
```

```
}
```

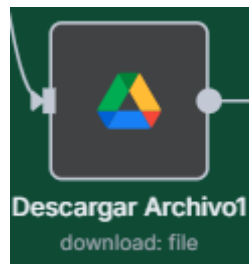
```
return $input.all();
```

3.3.8. Loop Over Items1



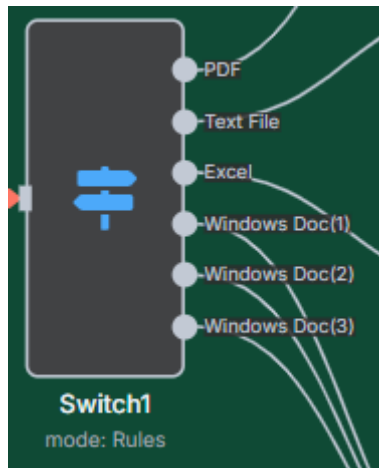
- **Tipo:** n8n-nodes-base.splitInBatches
- **Descripción:** Divide los ítems en lotes para procesamiento.

3.3.9. Descargar Archivo1



- **Tipo:** n8n-nodes-base.googleDrive
- **Descripción:** Descarga el contenido del archivo actualizado desde Google Drive. Los archivos de Google Docs se convierten automáticamente a texto plano.
- **Parámetros:**
 - operation: download
 - fileId: ={{ \$('Obtener ID1').item.json.file_id }}
 - googleFileConversion: docsToFormat: text/plain
- **Credenciales:** AgenteRAG (ID: 5BLbqJdcAeFAnoL6)

3.3.10. Switch1



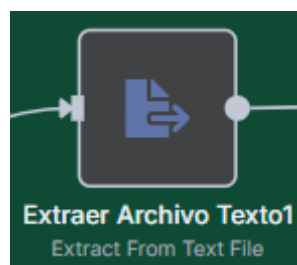
- **Tipo:** n8n-nodes-base.switch
- **Descripción:** Redirige el flujo basándose en el tipo MIME (file_type) del archivo actualizado, similar a "Switch" en el flujo de carga inicial.
- **Reglas:** Idénticas a "Switch" para PDF, Text File, Excel y Windows Docs.

3.3.11. Extraer Texto PDF1



- **Tipo:** n8n-nodes-base.extractFromFile
- **Descripción:** Extrae texto de archivos PDF en el flujo de actualización.
- **Parámetros:** operation: pdf

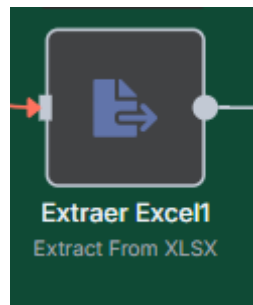
3.3.12. Extraer Archivo Texto1



- **Tipo:** n8n-nodes-base.extractFromFile
- **Descripción:** Extrae texto de archivos de texto plano o Google Docs convertidos en el flujo de actualización.

- **Parámetros:** operation: text

3.3.13. Extraer Excel1



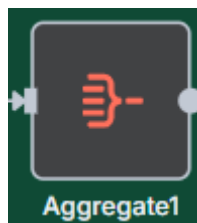
- **Tipo:** n8n-nodes-base.extractFromFile
- **Descripción:** Extrae datos de archivos Excel en el flujo de actualización.
- **Parámetros:** operation: xlsx

3.3.14. Convertir a Google Doc1



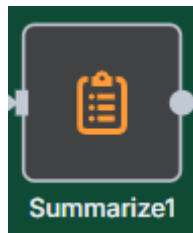
- **Tipo:** n8n-nodes-base.httpRequest
- **Descripción:** Convierte el archivo original (si no es Google Doc) en un Google Doc para facilitar la extracción de texto en el flujo de actualización.
- **Parámetros:** Idénticas a "Convertir a Google Doc".
- **Credenciales:** AgenteRAG (ID: 5BLbqJdcAeFAnoL6)

3.3.15. Aggregate1



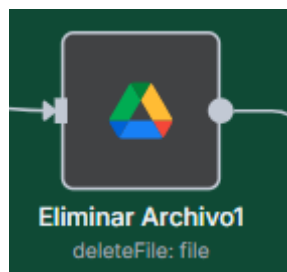
- **Tipo:** n8n-nodes-base.aggregate
- **Descripción:** Agrupa todos los datos de los ítems para una posterior concatenación en el flujo de actualización.

3.3.16. Summarize1



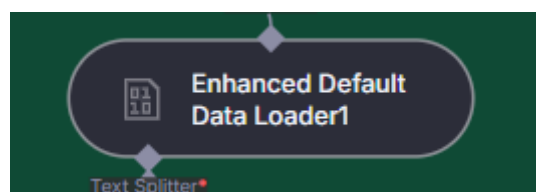
- **Tipo:** n8n-nodes-base.summarize
- **Descripción:** Concatena el contenido extraído de los documentos en un solo campo data (o similar) para el siguiente paso en el flujo de actualización.
- **Parámetros:** aggregation: concatenate field: data

3.3.17. Eliminar Archivo1



- **Tipo:** n8n-nodes-base.googleDrive
- **Descripción:** Elimina el archivo original después de la conversión a Google Doc en el flujo de actualización.
- **Parámetros:** Idénticas a "Eliminar Archivo".
- **Credenciales:** AgenteRAG (ID: 5BLbqJdcAeFAnoL6)

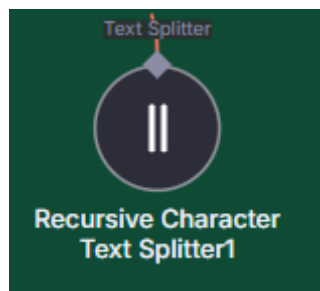
3.3.18. Enhanced Default Data Loader1



- **Tipo:** @n8n/n8n-nodes-langchain.documentDefaultDataLoader
- **Descripción:** Prepara los datos del documento para ser cargados en la base de datos vectorial, añadiendo metadatos relevantes para la actualización.
- **Parámetros:**
 - jsonData: ={{ \$json.data || \$json.text || \$json.concatenated_data }}

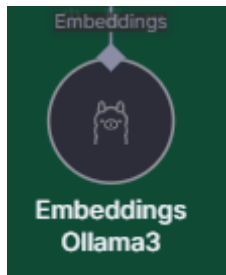
- metadataValues:
 - file_id: ={{ \$('Obtener ID1').item.json.file_id }}
 - version: ={{ \$('Incrementar Versión').item.json.myNewField }}
(Usa la versión incrementada).
 - creator: ={{ \$('Archivo Actualizado').item.json.owners[0].displayName }}
 - created_at: ={{ \$('Archivo Actualizado').item.json.createdTime }}
 - last_modified: ={{ \$('Archivo Actualizado').item.json.modifiedTime }}
 - folder_path: projects
 - file_name: ={{ \$('Archivo Actualizado').item.json.name }}
 - file_extension: ={{ \$('Archivo Actualizado').item.json.mimeType }}

3.3.19. Recursive Character Text Splitter1



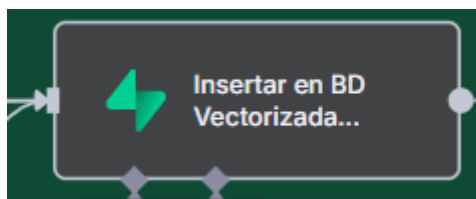
- **Tipo:** @n8n/n8n-nodes-langchain.textSplitterRecursiveCharacterTextSplitter
- **Descripción:** Divide el texto largo de los documentos actualizados en fragmentos más pequeños.
- **Parámetros:**
 - chunkSize: 2000
 - chunkOverlap: 200

3.3.20. Embeddings Ollama3



- **Tipo:** @n8n/n8n-nodes-langchain.embeddingsOllama
- **Descripción:** Genera los *embeddings* para los fragmentos de texto actualizados.
- **Parámetros:**
 - model: nomic-embed-text:latest
- **Credenciales:** Local Ollama service (ID: xHuYe0MDGOs9IpBW)

3.3.21. Insertar en BD Vectorizada Supabase1



- **Tipo:** @n8n/n8n-nodes-langchain.vectorStoreSupabase
- **Descripción:** Inserta los nuevos *chunks* vectorizados y sus metadatos actualizados en la tabla documents de Supabase.
- **Parámetros:**
 - mode: insert
 - tableName: documents
 - queryName: match_documents
- **Credenciales:** RAG_agent (ID: K54wKWCWt6vluLzi)

3.4. Flujo de Operación de Carga y Actualización

1. Detección de Archivos:

- **Carga Inicial:** El nodo "Archivo Creado" detecta nuevos documentos en la carpeta configurada de Google Drive.
- **Actualización:** El nodo "Archivo Actualizado" detecta modificaciones en los documentos existentes de Google Drive.

2. **Filtrado de Actualizaciones:** Para el flujo de actualización, el nodo "If" filtra los archivos, procesando solo los Google Docs actualizados recientemente.
3. **Extracción de Metadatos:** Los nodos "Obtener ID" y "Obtener ID1" extraen información clave del archivo (ID, tipo, nombre, etc.).
4. **Gestión de Versiones (Solo Actualización):**
 - "Borrar Filas Doc Antiguo" elimina las entradas vectoriales anteriores del documento en Supabase para evitar duplicados.
 - "Obtener Version" (si está habilitado) o "Incrementar Versión" calcula la nueva versión del documento.
5. **Descarga y Conversión:** "Descargar Archivo" y "Descargar Archivo1" obtienen el contenido del documento. "Convertir a Google Doc" y "Convertir a Google Doc1" (si aplica) transforman el archivo a un formato legible para la extracción de texto.
6. **Extracción de Contenido:** Los nodos "Extraer Texto PDF", "Extraer Archivo Texto" y "Extraer Excel" (y sus contrapartes con "1" para el flujo de actualización) procesan el contenido del documento según su tipo.
7. **Agregación y Resumen:** "Aggregate" y "Summarize" (y sus contrapartes con "1") combinan y consolidan el texto extraído.
8. **Preparación de Datos y Metadatos:** "Enhanced Default Data Loader" y "Enhanced Default Data Loader1" estructuran el texto y añaden metadatos esenciales (ID de archivo, versión, creador, fechas, etc.).
9. **Fragmentación de Texto:** "Recursive Character Text Splitter" y "Recursive Character Text Splitter1" dividen el texto en chunks manejables.
10. **Vectorización:** "Embeddings Ollama2" y "Embeddings Ollama3" generan *embeddings* para cada chunk de texto.
11. **Carga en Base de Datos Vectorial:** "Insertar en BD Vectorizada Supabase" y "Insertar en BD Vectorizada Supabase1" almacenan los *chunks* vectorizados y sus metadatos en la tabla documents de Supabase.
12. **Limpieza (Solo Carga Inicial/Conversión):** "Eliminar Archivo" y "Eliminar Archivo1" eliminan los archivos temporales o convertidos si es necesario.

4. Dependencias y Credenciales

Para el correcto funcionamiento de estos flujos, se requieren las siguientes configuraciones de credenciales en n8n:

- **Google Drive OAuth2 API (AgenteRAG)** Necesario para acceder y gestionar archivos en Google Drive (descarga, detección de eventos, eliminación, copia).
- **PostgreSQL (Agente_RAG):** Para la memoria del chat.
- **Supabase API (Agente_RAG y RAG_agent):** Para la gestión de la base de datos vectorial (inserción, eliminación, recuperación). Es importante notar que hay dos IDs de credenciales Supabase distintos nombrados Agente_RAG y RAG_agent, que apuntan a diferentes IDs de credenciales. Es fundamental que ambos estén correctamente configurados para las operaciones respectivas.
- **Ollama API (Local Ollama service):** Para los modelos de *embeddings* locales.
- **OpenAI API (OpenAi accoun):** Aunque el modelo principal es DeepSeek, las credenciales son nombradas para OpenAI y se usan para el "OpenAI Chat Model" y "Obtener Version".

5. Mantenimiento y Consideraciones

- **Monitorización:** Es crucial monitorizar las ejecuciones de ambos flujos en n8n para identificar errores o cuellos de botella.
- **Versión de Documentos:** El manejo de versiones en el flujo de actualización (Incrementar Versión y Obtener Version) debe ser revisado para asegurar que la lógica de incremento se alinee con los requisitos. El nodo de código actualmente establece la versión en 1, lo cual no es un incremento. El nodo "Obtener Version" (basado en LLM) está deshabilitado y podría ser una alternativa más robusta si se habilita y configura correctamente.
- **Tipos de Archivo:** El nodo "Switch" es fundamental para la correcta extracción de texto. Si se van a añadir nuevos tipos de documentos, se deben configurar nuevas ramas en el "Switch" y los nodos de extracción correspondientes.
- **Chunking:** Los parámetros chunkSize y chunkOverlap en los nodos "Recursive Character Text Splitter" son críticos para el rendimiento del RAG. Ajustarlos puede mejorar la calidad de las respuestas y la eficiencia de la búsqueda.
- **Credenciales:** Asegurarse de que todas las credenciales estén actualizadas y sean válidas para sus respectivos servicios.

- **Rendimiento de Ollama:** El uso de Ollama local puede requerir recursos significativos en el servidor donde se ejecuta n8n, especialmente si se procesan muchos documentos o se tienen muchas consultas concurrentes.

6. Solución de Problemas

- **Errores en la recuperación del Agente RAG:** Verificar la conexión a Supabase, el estado del servicio Ollama y el formato de los *embeddings*.
- **Documentos no actualizados:** Comprobar que los triggers de Google Drive ("Archivo Creado" y "Archivo Actualizado") estén habilitados y apunten a la carpeta correcta. Revisar las condiciones del nodo "If" para asegurarse de que no estén filtrando documentos incorrectamente.
- **Errores de conversión/extracción:** Verificar los nodos "Convertir a Google Doc" y "Extraer Texto/Excel" para asegurar que los formatos de archivo son compatibles y las conversiones se realizan correctamente.
- **Problemas de permisos:** Asegurarse de que las credenciales de Google Drive y Supabase tienen los permisos necesarios para leer, escribir y eliminar archivos/datos.