Documento de requerimientos de software

Asistente Digital para Consulta de Normativa y Procesos

Fecha: 21/04/2025

Tabla de contenido

Historial de Versiones

Fecha	Versión	Autor	Organización	Descripción
21/04/2025	**1.0**	**Jeferson Vargas A.**	**ESPOCH**	**Versión inicial del documento**
05/06/2025	**2.0**	**Jeferson Vargas A.**	**ESPOCH**	**Ajuste de requisitos**

Información del Proyecto

Empresa / Organización	Facultad de Informática y Electrónica - ESPOCH
Proyecto	Asistente Digital para Consulta de Normativa y Procesos Institucionales
Fecha de preparación	21/04/2025
Cliente	Facultad de Informática y Electrónica
Patrocinador principal	Dirección de Carrera de Ingeniería en Software
Gerente / Líder de Proyecto	Jeferson Vargas A.
Gerente / Líder de Análisis de negocio y requerimientos	

Aprobaciones

Nombre y Apellido	Cargo	Departamento u Organización	Fecha	Firma
Jeferson Vargas	**Desarrollador**			

1. Propósito

.El propósito de este Documento de Requisitos de Software (DRS) es detallar de manera exhaustiva las funcionalidades, características y restricciones que debe cumplir el sistema "Polito". Este documento servirá como una guía fundamental para el diseño, desarrollo, pruebas y validación del sistema, asegurando que cumpla con las expectativas de los usuarios y los objetivos de la ESPOL para la gestión del conocimiento institucional.

2. Alcance del producto / Software

El sistema "Polito" abarca dos componentes principales:
Agente Conversacional (Chatbot): Un asistente virtual capaz de interactuar con usuarios a través de una interfaz de chat, procesar sus preguntas y generar respuestas informadas y precisas basadas en una base de conocimiento documental.
Módulo de Gestión de Documentos: Un sistema automatizado para la ingesta, procesamiento (extracción de texto, fragmentación, vectorización) y actualización de documentos desde Google Drive hacia una base de datos vectorial, manteniendo así la base de conocimiento del Agente RAG siempre actualizada.
El sistema se enfoca en la consulta de documentos y no incluye funcionalidades para la creación, edición o flujo de aprobación de documentos externos al ámbito de Google Drive.

3. Referencias

- Manual Técnico del Flujo de Agente RAG y Actualización de Documentos (Versión 1.0)
- Manual de Usuario: Agente RAG y Gestión de Documentos (Versión 1.0)

- Documentación de n8n (https://n8n.io/)
- Documentación de Supabase (https://supabase.com/)
- Documentación de Ollama (https://ollama.ai/)
- Documentación de Google Drive API (https://developers.google.com/drive/api/v3/reference)
- Documentación de PostgreSQL (https://www.postgresql.org/docs/)

4. Funcionalidades del producto

El producto "Polito" ofrecerá las siguientes funcionalidades clave:

- **Interacción Conversacional Inteligente:** Permitir a los usuarios realizar preguntas en lenguaje natural y recibir respuestas precisas.
- **Recuperación de Información Relevante:** Buscar y obtener contenido relevante de la base de datos documental para fundamentar las respuestas.
- **Gestión Automática de Documentos:** Automatizar la detección, carga, procesamiento y actualización de documentos en la base de conocimiento.
- **Soporte Multi-Formato de Documentos:** Procesar documentos en diversos formatos comunes de oficina y PDF.
- **Mantenimiento de Contexto de Conversación:** Recordar el historial de chat para conversaciones más fluidas y coherentes.

5. Clases y características de usuarios

El sistema está diseñado para ser utilizado por dos clases principales de usuarios:

- **Usuario Final (Estudiantes, Personal Administrativo, Docentes de ESPOH):**
  - **Características:** Usuarios que necesitan acceder rápidamente a información oficial de la ESPOH. No requieren conocimientos técnicos sobre el funcionamiento interno del Agente RAG o la base de datos.
  - **Necesidades:** Obtener respuestas claras, concisas y fiables a sus preguntas sobre documentos de la ESPOL; recibir enlaces directos a los documentos fuente cuando sea aplicable.
- **Administrador de Contenido / Gestor Documental (Personal Designado de ESPOH):**

- **Características:** Responsables de mantener la base de conocimiento de Polito actualizada. Tienen acceso a la carpeta de Google Drive designada para los documentos. No requieren conocimientos de programación, pero deben entender el flujo básico de carga de documentos.
- **Necesidades:** Un método sencillo y automatizado para añadir o actualizar documentos; un sistema que refleje rápidamente los cambios en la base de conocimiento del chatbot.

6. Entorno operativo

El sistema "Polito" operará en el siguiente entorno:

- **Plataforma de Orquestación:** n8n (instalación en servidor, preferiblemente en contenedores Docker).
- **Base de Datos Vectorial:** Supabase (PostgreSQL con extensión pgvector).
- **Servicio de Embeddings Local:** Ollama (ejecutándose en el mismo servidor o en un servidor accesible por n8n).
- **Almacenamiento de Documentos Fuente:** Google Drive.
- **Base de Datos para Memoria de Chat:** PostgreSQL.
- **Modelo de Lenguaje Grande (LLM):** deepseek/deepseek-chat-v3-0324:free (accesible a través de una API).

El servidor que aloja n8n y Ollama debe cumplir con los requisitos mínimos de hardware para ejecutar estos componentes de manera eficiente (RAM, CPU, espacio en disco).

7. Requerimientos funcionales

7.1. Interacción con el Agente RAG

RF-RAG-001: Interacción de Chat

- **Descripción:** El sistema debe proporcionar una interfaz de chat para que los usuarios puedan enviar preguntas al Agente RAG.
- **Entrada:** Mensaje de texto del usuario.
- **Salida:** Respuesta del Agente RAG en la interfaz de chat.

RF-RAG-002: Generación de Respuestas Basadas en Documentos

- **Descripción:** El Agente RAG debe generar respuestas a las preguntas del usuario utilizando exclusivamente la información recuperada de la base de conocimiento vectorial de documentos de la ESPOL.
- **Entrada:** Consulta del usuario, contexto de conversación.
- **Salida:** Respuesta coherente y fundamentada en los documentos.

RF-RAG-003: Indicación de Información No Encontrada

- **Descripción:** Si la consulta del usuario no puede ser respondida con la información disponible en los documentos, el sistema debe informar explícitamente al usuario sobre la falta de información.
- **Entrada:** Consulta del usuario.
- **Salida:** Mensaje predefinido: "Lo siento, no tengo información sobre eso en los documentos que puedo consultar."

RF-RAG-004: Recuperación Semántica de Documentos

- **Descripción:** El sistema debe ser capaz de buscar y recuperar los fragmentos de documentos más relevantes de la base de conocimiento vectorial basándose en la similitud semántica con la pregunta del usuario.
- **Entrada:** Consulta del usuario (vectorizada).
- **Salida:** Conjunto de fragmentos de documentos relevantes con sus metadatos.

RF-RAG-005: Mantenimiento del Historial de Conversación

- **Descripción:** El sistema debe almacenar y recuperar el historial de chat de cada usuario para mantener el contexto en conversaciones consecutivas.
- **Entrada:** Mensaje del usuario, respuesta del sistema.
- **Salida:** Historial de conversación actualizado en la base de datos de memoria.

RF-RAG-006: Provisión de Enlaces a Documentos

- **Descripción:** Si un fragmento de documento recuperado incluye un file_id asociado a Google Drive, el sistema debe presentar al usuario un enlace de descarga en el formato https://drive.google.com/uc?export=download&id=[ID_DEL_ARCHIVO_DE_GOOGLE_DRIVE].
- **Entrada:** Fragmento de documento recuperado con file_id.

- **Salida:** Enlace de Google Drive en la respuesta del chat.

- **REQ-010:** habilitar mecanismos para retroalimentar y ajustar las respuestas del asistente, ya sea de manera manual o automática mediante entrenamiento incremental.

7.2. Gestión y Actualización de Documentos

RF-GD-001: Detección de Creación de Archivos

- **Descripción:** El sistema debe monitorear la carpeta de Google Drive DocumentacionP1 y detectar la creación de nuevos archivos en ella.
- **Frecuencia:** Cada minuto (configurable).
- **Entrada:** Evento de creación de archivo en Google Drive.
- **Salida:** Metadatos del nuevo archivo (ID, nombre, tipo MIME, etc.).

RF-GD-002: Detección de Modificación de Archivos

- **Descripción:** El sistema debe monitorear la carpeta de Google Drive DocumentacionP1 y detectar las modificaciones en archivos existentes.
- **Frecuencia:** Cada minuto (configurable).
- **Entrada:** Evento de modificación de archivo en Google Drive.
- **Salida:** Metadatos del archivo modificado (ID, nombre, tipo MIME, etc.).

RF-GD-003: Descarga de Archivos

- **Descripción:** El sistema debe ser capaz de descargar el contenido de los archivos detectados desde Google Drive. Los archivos de Google Docs deben ser descargados como texto plano.
- **Entrada:** file_id del archivo de Google Drive.
- **Salida:** Contenido del archivo.

RF-GD-004: Conversión Interna de Documentos (si aplica)

- **Descripción:** Para archivos no nativos de Google Docs (e.g., DOCX, DOC), el sistema debe realizar una copia temporal del archivo a formato Google Doc para facilitar la extracción de texto.
- **Entrada:** Archivo en formato Word (.docx, .doc).

- **Salida:** ID del archivo convertido temporalmente en Google Docs.

RF-GD-005: Extracción de Texto de PDF

- **Descripción:** El sistema debe poder extraer todo el texto de los archivos PDF.
- **Entrada:** Archivo PDF.
- **Salida:** Texto extraído del PDF.

RF-GD-006: Extracción de Texto de Archivos de Texto/Google Docs

- **Descripción:** El sistema debe poder extraer el texto de archivos de texto plano y de documentos de Google Docs.
- **Entrada:** Archivo de texto plano o Google Doc.
- **Salida:** Texto extraído.

RF-GD-007: Extracción de Datos de Excel

- **Descripción:** El sistema debe poder extraer los datos (texto) de las celdas de archivos Excel (XLSX).
- **Entrada:** Archivo XLSX.
- **Salida:** Texto concatenado de las celdas de Excel.

RF-GD-008: Eliminación de Archivos Convertidos/Temporales

- **Descripción:** Tras la conversión y procesamiento de un archivo no nativo de Google Docs, el sistema debe eliminar la copia temporal creada en Google Drive para evitar duplicados.
- **Entrada:** file_id de la copia temporal.
- **Salida:** Confirmación de eliminación.

RF-GD-009: Fragmentación de Texto

- **Descripción:** El sistema debe dividir el texto extraído de los documentos en fragmentos (chunks) de un tamaño configurable (chunkSize) con una superposición (chunkOverlap) configurable para mantener el contexto.
- **Entrada:** Texto completo del documento.
- **Salida:** Lista de fragmentos de texto.

RF-GD-010: Generación de Embeddings

- **Descripción:** Para cada fragmento de texto, el sistema debe generar una representación vectorial numérica (*embedding*) utilizando el modelo nomic-embed-text:latest de Ollama.
- **Entrada:** Fragmento de texto.
- **Salida:** Vector numérico (embedding) del fragmento.

RF-GD-011: Inserción de Datos Vectoriales

- **Descripción:** El sistema debe insertar los *embeddings* generados, junto con los metadatos relevantes del documento (file_id, nombre, versión, autor, fechas, etc.), en la tabla documents de la base de datos vectorial de Supabase.
- **Entrada:** Embedding, fragmento de texto, metadatos.
- **Salida:** Confirmación de inserción en Supabase.

RF-GD-012: Eliminación de Entradas Antiguas para Actualización

- **Descripción:** Cuando un documento existente es modificado, el sistema debe eliminar todas las entradas vectoriales previamente asociadas con el file_id de ese documento en la base de datos de Supabase antes de insertar la nueva versión.
- **Entrada:** file_id del documento modificado.
- **Salida:** Confirmación de eliminación de registros antiguos.

RF-GD-013: Gestión de Versionado de Documentos (Revisión)

- **Descripción:** El sistema debe ser capaz de determinar una nueva versión para un documento actualizado, preferiblemente incrementando la versión anterior (e.g., v1 a v2).
- **Entrada:** Metadatos del documento (incluyendo versión anterior si existe).
- **Salida:** Número de versión actualizado.
    - *Nota de Implementación: La implementación actual de n8n para el versionado necesita ser revisada para un incremento real y no una asignación fija.*

7.1 Funcionalidad: Consulta de normativa por lenguaje natural

Descripción: Permite que cualquier usuario del sistema (estudiante, docente o administrativo) pueda realizar preguntas en lenguaje natural para consultar normativas y procesos institucionales.

Prioridad: Alta

Acciones iniciadoras y comportamiento esperado: El usuario accede al sistema, escribe una consulta como "¿Cuál es el procedimiento para solicitar homologación?" y el sistema procesa el texto, identifica palabras clave y devuelve la información correspondiente. Si no encuentra resultados, sugiere términos relacionados.

Requerimientos funcionales:

REQ-001: El sistema debe contar con una caja de texto para ingresar preguntas en lenguaje natural.

REQ-002: El sistema debe procesar la consulta utilizando el motor Ollama y generar una intención de búsqueda

REQ-003: El sistema debe buscar respuestas relevantes en la base vectorial Qdrant.

REQ-004: El sistema debe presentar la respuesta en pantalla junto con la referencia del documento fuente.

REQ-005: En caso de error o consulta ambigua, el sistema debe ofrecer mensajes comprensibles y sugerencias alternativas.

7.2 . Funcionalidad: Gestión documental por el administrador

Descripción: Brinda al administrador la capacidad de subir, editar y clasificar documentos normativos dentro del sistema.

Prioridad: Alta

Acciones iniciadoras y comportamiento esperado: El administrador accede al panel de gestión, sube un archivo PDF o DOCX, selecciona etiquetas y lo guarda. El sistema indexa automáticamente el contenido en MongoDB y en la base vectorial.

Requerimientos funcionales:

REQ-006: El sistema debe permitir subir documentos en formatos DOCX y PDF.

REQ-007: El sistema debe indexar automáticamente los documentos cargados.

REQ-008: El administrador podrá asignar etiquetas personalizadas a cada documento.

REQ-009: El sistema debe permitir editar y eliminar documentos desde el panel.

7.3 . Funcionalidad: Gestión de usuarios y roles

Descripción: Permite administrar el acceso al sistema según el perfil del usuario.

Prioridad: Media

Acciones iniciadoras y comportamiento esperado: Al iniciar sesión, el sistema identifica el perfil del usuario y restringe funcionalidades según su rol (consulta, administración, supervisión).

Requerimientos funcionales:

REQ-010: El sistema debe autenticar usuarios y registrar su perfil (estudiante, docente, administrativo, administrador).

REQ-011: El sistema debe mostrar funcionalidades disponibles según el rol.

7.4 . Funcionalidad: Retroalimentación y mejora del asistente

Descripción: Permite que los usuarios evalúen las respuestas y que el administrador actualice el entrenamiento del sistema.

Prioridad: Media

Acciones iniciadoras y comportamiento esperado: Tras recibir una respuesta, el usuario puede valorarla. El administrador visualiza la retroalimentación y, si es necesario, realiza ajustes al motor de búsqueda o documentación.

Requerimientos funcionales:

- **REQ-014:** El sistema debe ofrecer la opción de valorar la calidad de la respuesta.

- **REQ-015:** El administrador podrá revisar el feedback y ajustar el sistema de forma manual o automática.

8. Reglas de negocio

Las reglas de negocio que rigen el comportamiento del sistema son:

9. **RB-001: Fuente Única de Verdad para RAG:** La base de conocimiento de Polito para generar respuestas se limita estrictamente a los documentos procesados y almacenados en la base de datos vectorial de Supabase. No se permite la consulta de fuentes externas en tiempo real (e.g., internet abierto).
10. **RB-002: Prioridad de Información:** En caso de múltiples fragmentos relevantes, el LLM debe priorizar la información para construir la respuesta más concisa y directa.
11. **RB-003: Autenticidad de Enlaces:** Los enlaces proporcionados por Polito a los documentos de Google Drive deben ser enlaces de descarga directos y válidos.
12. **RB-004: Proceso de Ingesta Automatizado:** La ingesta y actualización de documentos debe ser un proceso automatizado. No se requiere intervención manual directa después de colocar el archivo en la carpeta de Google Drive.
13. **RB-005: Tratamiento de Tipos de Archivo:** Solo los tipos de archivo especificados (PDF, Google Docs, DOCX, DOC, XLSX) serán procesados para extracción de texto. Otros tipos serán ignorados o generarán un error controlable.

14. Requerimientos de interfaces externas

9.1. Interfaces de usuario

- **IU-001: Interfaz de Chat Web:** El Agente RAG debe tener una interfaz de chat accesible vía web, intuitiva y fácil de usar, con un campo de entrada de texto y un área de visualización de mensajes.

9.2. Interfaces de hardware

- **IH-001: Servidor de Ejecución de n8n/Ollama:** El sistema requiere un servidor con capacidad de cómputo y memoria suficientes para ejecutar n8n y el servicio Ollama de manera eficiente.

9.3. Interfaces de software

- **IS-001: API de Google Drive:** El sistema interactuará con la API de Google Drive para la detección de eventos (creación/modificación de archivos), descarga de archivos y, opcionalmente, copia/eliminación de archivos.
- **IS-002: API de Supabase:** El sistema interactuará con la API de Supabase para operaciones de inserción y eliminación de datos en la base de datos vectorial.
- **IS-003: API de Ollama:** El sistema interactuará con la API de Ollama para la generación de *embeddings* a partir de los fragmentos de texto.
- **IS-004: API del Modelo de Chat (LLM):** El Agente RAG interactuará con la API del modelo de lenguaje (deepseek/deepseek-chat-v3-0324:free u otro configurado) para la generación de respuestas.
- **IS-005: Conexión PostgreSQL:** El sistema requerirá una conexión a una base de datos PostgreSQL para el almacenamiento del historial de chat.

9.4. Interfaces de comunicación

- **IC-001: HTTPS:** Todas las comunicaciones con servicios externos (Google Drive, Supabase, LLM API) deben realizarse a través de HTTPS para garantizar la seguridad de los datos.
- **IC-002: Protocolo Webhook:** El trigger de chat de n8n para Polito utilizará un webhook HTTP para recibir mensajes.
- **IC-003: Protocolo Interno Ollama:** La comunicación con el servicio Ollama se realizará a través del protocolo API de Ollama.

10. Requerimientos no funcionales

10.1. Rendimiento

- **RNF-PERF-001: Tiempo de Respuesta de Consulta:** El 90% de las respuestas del Agente RAG deben ser generadas y presentadas al usuario en menos de 5 segundos.
- **RNF-PERF-002: Tasa de Procesamiento de Documentos:** El sistema debe ser capaz de procesar un documento promedio (e.g., 500 KB, 10 páginas) en un máximo de 10 minutos desde su detección hasta su disponibilidad en la base de conocimiento vectorial.
- **RNF-PERF-003: Concurrencia del Chatbot:** El Agente RAG debe soportar un mínimo de 10 usuarios concurrentes sin degradación perceptible del rendimiento.
- **RNF-PERF-004: Frecuencia de Actualización de Documentos:** El proceso de detección y procesamiento de documentos nuevos/modificados debe ejecutarse al menos cada 60 segundos.

10.2. Disponibilidad

- **RNF-DISP-001: Disponibilidad del Chatbot:** El Agente RAG debe estar disponible para consultas el 99% del tiempo durante las horas de operación (Lunes a Viernes, 08:00 - 18:00).
- **RNF-DISP-002: Disponibilidad del Proceso de Ingesta:** El flujo de carga y actualización de documentos debe tener una disponibilidad del 95% para su ejecución programada.

10.3. Usabilidad

- **RNF-USAB-001: Simplicidad de Interfaz:** La interfaz de usuario del chatbot debe ser intuitiva, con un diseño limpio y sin elementos complejos que distraigan la interacción conversacional.
- **RNF-USAB-002: Claridad de Mensajes:** Todos los mensajes de Polito (respuestas, indicaciones de error, etc.) deben ser claros, concisos y libres de jerga técnica.
- **RNF-USAB-003: Facilidad de Gestión Documental:** El proceso de añadir o actualizar documentos debe limitarse a operaciones estándar de Google Drive (subir, editar) sin requerir pasos adicionales complejos.

10.4. Seguridad

- **RNF-SEG-001: Autenticación Segura:** El acceso a los servicios de Google Drive, Supabase y las APIs de LLM/Ollama debe realizarse mediante mecanismos de autenticación robustos (OAuth2, claves API).
- **RNF-SEG-002: Comunicación Cifrada:** Todas las comunicaciones entre los componentes del sistema (n8n, Supabase, Ollama, Google Drive API, LLM API) deben estar cifradas (HTTPS).
- **RNF-SEG-003: Control de Acceso:** El acceso a la carpeta DocumentacionP1 en Google Drive debe estar restringido solo a los usuarios o cuentas de servicio autorizadas.
- **RNF-SEG-004: Protección contra Inyección de Prompts:** El diseño del agente debe considerar mecanismos para mitigar riesgos de inyección de prompts, aunque esto es principalmente una característica del LLM subyacente y la configuración del systemMessage.

10.5. Mantenibilidad

- **RNF-MANT-001: Modularidad del Flujo:** El diseño del flujo en n8n debe ser modular, permitiendo la fácil modificación o adición de nodos para soportar nuevos tipos de documentos o funcionalidades sin afectar el resto del sistema.
- **RNF-MANT-002: Configurabilidad Externa:** Los parámetros críticos (rutas de carpetas, tamaños de chunk, modelos LLM/embeddings) deben ser configurables mediante variables de entorno o parámetros de n8n, sin requerir cambios en el código de los nodos.
- **RNF-MANT-003: Registro y Auditoría:** El sistema debe generar registros (logs) de sus operaciones (ejecuciones de flujos, errores, interacciones del chatbot) para facilitar la depuración y el monitoreo.

10.6. Confiabilidad

- **RNF-CONF-001: Manejo de Errores:** El sistema debe manejar los errores de manera elegante, registrándolos y, si es posible, reintentando operaciones fallidas (e.g., descarga de archivos, conexión a API) o notificando al administrador.
- **RNF-CONF-002: Consistencia de Datos:** La base de datos vectorial debe mantener la consistencia de los datos, asegurando que las actualizaciones reflejen con precisión el contenido actual de los documentos y que no haya duplicados obsoletos.

15. Requerimientos no funcionales

- **NF-001: Como usuario, quiero que el sistema responda rápidamente a mis consultas**
  Para no perder tiempo esperando, el sistema debe mostrar resultados de búsqueda en un máximo de 3 segundos.
- **NF-002: Como usuario frecuente, quiero que el sistema esté disponible la mayor parte del tiempo**
  Para poder acceder al sistema cuando lo necesite, este debe tener una disponibilidad mínima del 99.5% mensual.
- **NF-003: Como usuario, quiero sentirme seguro al usar el sistema**
  Todas mis interacciones deben estar protegidas mediante conexiones seguras (HTTPS), y mis datos personales, como contraseñas, deben estar cifrados y protegidos ante accesos no autorizados.
- **NF-004: Como administrador, quiero poder ampliar el sistema si aumentan los documentos o los usuarios**
  El sistema debe permitir escalar sus capacidades sin necesidad de rediseñar la arquitectura.
- **NF-005: Como usuario con discapacidad visual, quiero poder usar el sistema sin barreras**
  La interfaz debe ser accesible, cumplir con estándares internacionales (WCAG 2.1 nivel AA), y permitir el uso de lectores de pantalla y ajustes de contraste y tamaño de fuente.
- **NF-006: Como usuario, quiero que las acciones críticas estén protegidas**
  Para evitar errores, las acciones como borrar documentos o usuarios deben requerir confirmación previa.
- **NF-007: Como personal técnico, quiero que el sistema sea fácil de mantener y actualizar**
  El sistema debe tener una estructura modular, documentación clara y usar buenas prácticas de desarrollo para facilitar futuras mejoras.
- **NF-008: Como institución, queremos poder usar el sistema en distintas plataformas**
  El sistema debe funcionar correctamente tanto en servidores locales como en la nube, usando contenedores (Docker) para facilitar su despliegue.

16. Otros requerimientos

OT-001: Como administrador, quiero que el sistema almacene de forma organizada la información documental y de usuarios

El sistema debe utilizar una base de datos eficiente (MongoDB) que permita relacionar fácilmente usuarios, documentos, roles, consultas y retroalimentaciones.

OT-002: Como usuario, quiero que el sistema entienda mis consultas con precisión

Para ofrecer mejores respuestas, el sistema debe usar una base vectorial semántica (Qdrant) que interprete el significado de mis preguntas y no solo palabras clave.

OT-003: Como institución, queremos que el sistema respete las leyes de protección de datos personales

El sistema debe cumplir con la Ley Orgánica de Protección de Datos Personales de Ecuador (LOPD), informando a los usuarios sobre el uso de su información y asegurando su consentimiento.

OT-004: Como institución, queremos que el sistema pueda ser traducido a otros idiomas

Para facilitar su uso por parte de otros departamentos o universidades, el sistema debe estar preparado para ser traducido, separando todos los textos visibles en archivos multilanguage (i18n).

OT-005: Como desarrollador o técnico del sistema, quiero reutilizar componentes ya existentes y confiables

17. Glosario

 Agente RAG (Retrieval-Augmented Generation): Un sistema de inteligencia artificial que combina la capacidad de un modelo de lenguaje grande (LLM) con la recuperación de información de una base de datos externa para generar respuestas más precisas y fundamentadas.

 Chatbot: Un programa de computadora diseñado para simular una conversación humana, especialmente a través de interacciones de texto.

 Chunk: Pequeño fragmento de texto extraído de un documento. Los documentos grandes se dividen en chunks para facilitar su procesamiento y búsqueda vectorial.

- **Embedding:** Una representación numérica (vector) de una pieza de texto (palabra, frase, chunk) que captura su significado semántico. Textos con significados similares tienen embeddings "cercanos" en el espacio vectorial.

- **ESPOL:** Escuela Superior Politécnica del Litoral.

- **Google Drive:** Servicio de almacenamiento en la nube de Google, utilizado como la fuente de los documentos para el sistema.

- **LLM (Large Language Model):** Un modelo de inteligencia artificial entrenado en grandes cantidades de texto para comprender, generar y responder en lenguaje natural.

- **n8n:** Herramienta de automatización de flujos de trabajo de código abierto (workflow automation tool), utilizada para orquestar la lógica del Agente RAG y el proceso de gestión de documentos.

- **Ollama:** Una plataforma para ejecutar modelos de lenguaje grandes (LLMs) y modelos de *embeddings* localmente.

- **PostgreSQL:** Un sistema de gestión de bases de datos relacionales de código abierto, utilizado para almacenar el historial de chat.

- **Supabase:** Una alternativa de código abierto a Firebase, que proporciona una base de datos PostgreSQL, APIs, y extensiones como pgvector para bases de datos vectoriales.

- **Vector Store / Base de Datos Vectorial:** Un tipo de base de datos optimizada para almacenar y buscar *embeddings* (vectores numéricos) basándose en su similitud.

- **Webhook:** Un mecanismo que permite que una aplicación envíe información a otra aplicación en tiempo real cuando ocurre un evento específico.