DOCUMENTO DE REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES

PROYECTO:

DESARROLLO DE UN CHATBOT CON IA GENERATIVA

[REQUERIMIENTOS FUNCIONALES (RF) 3](#_Toc200040176)

[RF01: Almacenamiento de mensajes 3](#_Toc200040177)

[RF02: Identificación de preguntas y respuestas 3](#_Toc200040178)

[RF03: Asociación entre preguntas y respuestas 3](#_Toc200040179)

[RF04: Identificación manual de roles 3](#_Toc200040180)

[RF05: Filtrado de mensajes irrelevantes 3](#_Toc200040181)

[RF06: Segmentación de preguntas extensas 3](#_Toc200040182)

[RF07: Generación de embeddings y metadatos 3](#_Toc200040183)

[RF08: Búsqueda semántica y recuperación de respuestas 3](#_Toc200040184)

[RF09: Carga manual de docentes 4](#_Toc200040185)

[RF10: Generación de logs para trazabilidad 4](#_Toc200040186)

[RF11: Actualización de la base de conocimiento 4](#_Toc200040187)

[RF12: Generación automática de respuestas 4](#_Toc200040188)

[ESTADO DE LOS REQUERIMIENTOS FUNCIONALES (RF) 4](#_Toc200040189)

[REQUERIMIENTOS NO FUNCIONALES (RNF) 5](#_Toc200040190)

[RNF01: Portabilidad 5](#_Toc200040191)

[RNF02: Mantenibilidad 5](#_Toc200040192)

[RNF03: Preparación para concurrencia 5](#_Toc200040193)

[RNF04: Trazabilidad 5](#_Toc200040194)

[RNF05: Usabilidad técnica (para desarrolladores y evaluadores) 6](#_Toc200040195)

[RNF06: Buenas prácticas de seguridad y privacidad 6](#_Toc200040196)

[RNF07: Reproducibilidad 6](#_Toc200040197)

[RNF08: Robustez ante errores comunes 6](#_Toc200040198)

[ESTADO DE LOS REQUERIMIENTOS NO FUNCIONALES (RNF) 7](#_Toc200040199)

[GLOSARIO 7](#_Toc200040200)

[BIBLIOGRAFÍA 8](#_Toc200040201)

# REQUERIMIENTOS FUNCIONALES (RF)

## RF01: Almacenamiento de mensajes

**Descripción:** El sistema debe almacenar los mensajes del canal de Discord en una base de datos relacional.  
**Justificación:** Permite organizar y persistir la información en una estructura consultable.

## RF02: Identificación de preguntas y respuestas

**Descripción:** El sistema debe identificar preguntas realizadas por estudiantes y las respuestas asociadas, sin importar el rol del autor.  
**Justificación:** Permite estructurar las interacciones en pares pregunta-respuesta para su posterior indexación y recuperación.

## RF03: Asociación entre preguntas y respuestas

**Descripción:** El sistema debe aplicar reglas específicas para asociar respuestas a preguntas detectadas, considerando factores como tiempo, validaciones, saludos de cierre y cantidad de respuestas.  
**Justificación:** Permite automatizar la relación entre preguntas y respuestas sin intervención manual.

## RF04: Identificación manual de roles

**Descripción:** El sistema debe permitir definir manualmente el rol de los autores (docente o alumno), en base a observación del comportamiento en los mensajes.  
**Justificación:** Permite mejorar la clasificación y trazabilidad de interacciones según el contexto educativo.

## RF05: Filtrado de mensajes irrelevantes

**Descripción:** El sistema debe eliminar mensajes irrelevantes aplicando reglas definidas: mensajes vacíos, que contienen solo emojis, stickers, GIFs, símbolos u otras expresiones sin contenido textual útil.  
**Justificación:** Mejora la calidad de los datos procesados y evita ruido en el análisis.

## RF06: Segmentación de preguntas extensas

**Descripción:** El sistema debe dividir preguntas largas en fragmentos si exceden el tamaño máximo admitido por el modelo de embeddings.  
**Justificación:** Asegura que todas las preguntas puedan ser vectorizadas, manteniendo su significado.

## RF07: Generación de embeddings y metadatos

**Descripción:** El sistema debe generar embeddings de las preguntas (o sus fragmentos) y almacenarlos en una base de datos vectorial, junto con metadatos (ID de pregunta).

**Justificación:** Permite realizar búsquedas semánticas eficientes y mantener trazabilidad.

## RF08: Búsqueda semántica y recuperación de respuestas

**Descripción:** El sistema debe permitir la búsqueda semántica de preguntas mediante embeddings y recuperar las respuestas asociadas utilizando los metadatos.  
**Justificación:** Mejora la capacidad del chatbot de brindar respuestas precisas ante consultas similares.

## RF09: Carga manual de docentes

**Descripción:** El sistema debe permitir la carga manual de docentes conocidos. **Justificación:** Facilita una clasificación más precisa de las respuestas y su validación.

## RF10: Generación de logs para trazabilidad

**Descripción:** El sistema debe permitir la generación de archivos de log que almacenen las preguntas cerradas y persistidas en la base de datos.  
**Justificación:** Brinda trazabilidad del conocimiento consolidado y permite auditorías del proceso.

## RF11: Actualización de la base de conocimiento

**Descripción:** El sistema debe permitir que las nuevas interacciones relevantes en el canal de consultas (no chatbot) sean evaluadas y, si contienen al menos una respuesta validada por docente, se incorporen a la base de conocimiento.  
**Justificación:** Permite mantener la base de datos actualizada con aportes reales y validados.

## RF12: Generación automática de respuestas

**Descripción:** El sistema debe permitir generar una respuesta automática a partir de la pregunta original del alumno, la pregunta encontrada en la búsqueda semántica y las respuestas asociadas, utilizando un modelo generativo.  
**Justificación:** Es el paso final del pipeline, donde se espera que el chatbot brinde respuestas útiles basadas en conocimiento ya consolidado.

# ESTADO DE LOS REQUERIMIENTOS FUNCIONALES (RF)

|  |  |  |  |
| --- | --- | --- | --- |
| Requerimientos Funcionales (RF) | PLANIFICADOS | EN IMPLEMENTACIÓN | CUMPLIDOS |
| RF01: Almacenamiento de mensajes |  |  | X |
| RF02: Identificación de preguntas y respuestas |  |  | X |
| RF03: Asociación entre preguntas y respuestas |  |  | X |
| RF04: Identificación manual de roles |  |  | X |
| RF05: Filtrado de mensajes irrelevantes |  |  | X |
| RF06: Segmentación de preguntas extensas |  | X |  |
| RF07: Generación de embeddings y metadatos |  |  | X |
| RF08: Búsqueda semántica y recuperación de respuestas |  |  | X |
| RF09: Carga manual de docentes |  |  | X |
| RF10: Generación de logs para trazabilidad |  |  | X |
| RF11: Actualización de la base de conocimiento | X |  |  |
| RF12: Generación automática de respuestas | X |  |  |

Definidos para las PPS (Prácticas Profesionales Supervisadas): RF1, RF2, RF3, RF4, RF5, RF7, RF8, RF9 Y RF10.

# REQUERIMIENTOS NO FUNCIONALES (RNF)

## RNF01: Portabilidad

**Descripción:** El sistema debe poder ejecutarse en cualquier entorno con Python 3.10 o superior, y con las dependencias instaladas mediante un entorno virtual (venv), utilizando el archivo requirements.txt.

**Justificación:** Asegura que pueda ser reutilizado o mantenido fácilmente en distintos equipos o entornos académicos.

**Nota:** El proyecto se instala con:

python -m venv venv

source venv/bin/activate

pip install -r requirements.txt

## RNF02: Mantenibilidad

**Descripción:** El sistema debe estar desarrollado con una estructura modular que facilite la incorporación de nuevas reglas o funcionalidades, como filtros adicionales, criterios de cierre de interacción, o mejoras en la identificación de preguntas y respuestas.

**Justificación:** Permite actualizar el sistema fácilmente a medida que cambien las necesidades del curso o se amplíe el alcance del chatbot.

**Nota:** Se utilizó un diseño basado en funciones independientes y potencialmente extensible a través de patrones como Strategy (para filtros), y se prevé incorporar otros como Chain of Responsibility en etapas futuras.

## RNF03: Preparación para concurrencia

**Descripción:** Aunque actualmente el sistema no gestiona múltiples consultas simultáneas, se contempla la futura incorporación de técnicas de concurrencia mediante hilos, procesos o programación asíncrona para permitir que el chatbot responda eficientemente ante múltiples interacciones o realice tareas en paralelo (por ejemplo, consultas a la base de datos mientras analiza nuevos mensajes).

**Justificación:** En situaciones como fechas de parciales o finales, podrían producirse picos de actividad con múltiples alumnos haciendo preguntas o interactuando con el chatbot. Preparar el sistema para estos escenarios mejora su escalabilidad futura.

**Nota:** Se prevé capacitación futura en manejo de concurrencia con asyncio, threading, o librerías específicas de bots en Discord (como discord.py con soporte asíncrono).

## RNF04: Trazabilidad

**Descripción:** El sistema debe permitir rastrear el origen de cada elemento procesado, incluyendo:

* Mensajes filtrados (irrelevantes)
* Preguntas identificadas
* Relación con posibles respuestas
* Vectorización y recuperación semántica

Esto se logra mediante el uso de logs y almacenamiento de metadatos (ID de mensaje, autor, timestamp, entre otros.).

**Justificación:** Es fundamental para auditar el comportamiento del sistema, corregir errores y comprender el flujo completo de procesamiento.

## RNF05: Usabilidad técnica (para desarrolladores y evaluadores)

**Descripción:** El sistema debe incluir documentación técnica clara que describa:

* El flujo general de trabajo
* La estructura del repositorio
* Cómo ejecutar cada etapa del sistema (filtrado, extracción de conocimiento, vectorización, búsqueda semántica)

**Justificación:** Facilita la comprensión por parte de evaluadores docentes y posibles futuros desarrolladores o mantenedores del proyecto.

## RNF06: Buenas prácticas de seguridad y privacidad

**Descripción:** El sistema no debe exponer públicamente claves de acceso a servicios externos (como Groq), contraseñas de bases de datos o rutas sensibles en los scripts del repositorio. Los datos de Discord pueden incluirse por tratarse de interacciones académicas sin contenido privado sensible, pero debe evaluarse cada caso.

**Justificación:** Aunque el sistema opera en entornos académicos y locales, es importante aplicar principios básicos de seguridad para evitar filtraciones accidentales y promover buenas prácticas de desarrollo.

**Nota:** Se identificó que, durante el desarrollo, se incluyeron sin intención archivos con datos sensibles (como claves de conexión), lo cual será corregido en futuras versiones.

## RNF07: Reproducibilidad

**Descripción:** El sistema debe producir resultados consistentes si se ejecuta múltiples veces con los mismos datos de entrada y las mismas reglas de procesamiento.

**Justificación:** La reproducibilidad es fundamental para validar el funcionamiento del sistema, especialmente durante el desarrollo y evaluación del proyecto.

**Nota:** La generación de archivos intermedios (como logs o CSV con mensajes descartados) y el uso de reglas definidas para filtrado y detección de preguntas permiten comprobar esta propiedad.

## RNF08: Robustez ante errores comunes

**Descripción:** El sistema debe ser capaz de detectar y manejar errores típicos como:

* Archivos de entrada faltantes o mal formateados
* Conexiones inválidas a la base de datos
* Estructuras inesperadas en los JSON
* Fallas al vectorizar o al aplicar filtros

**Justificación:** Previene interrupciones innecesarias del flujo de trabajo y mejora la experiencia de desarrollo, especialmente en etapas de prueba o validación.

# ESTADO DE LOS REQUERIMIENTOS NO FUNCIONALES (RNF)

|  |  |  |  |
| --- | --- | --- | --- |
| REQUERIMIENTOS NO FUNCIONALES (RNF) | PLANIFICADOS | EN IMPLEMENTACIÓN | CUMPLIDOS |
| RNF01: Portabilidad |  |  | X |
| RNF02: Mantenibilidad |  |  | X |
| RNF03: Preparación para concurrencia | X |  |  |
| RNF04: Trazabilidad |  |  | X |
| RNF05: Usabilidad técnica (para desarrolladores y evaluadores) |  |  | X |
| RNF06: Buenas prácticas de seguridad y privacidad |  |  | X |
| RNF07: Reproducibilidad |  |  | X |
| RNF08: Robustez ante errores comunes |  | X |  |

Definidos para las PPS (Prácticas Profesionales Supervisadas): RNF01, RNF02, RNF04, RNF05, RNF06, RNF07.

# GLOSARIO

|  |  |
| --- | --- |
| Término | Significado |
| Asyncio | Es una biblioteca de Python que permite gestionar múltiples tareas al mismo tiempo sin necesidad de usar múltiples hilos o procesos. |
| Base de datos relacional | Tipo de base de datos que organiza la información en tablas relacionadas entre sí por claves. Se utiliza SQL para su gestión. |
| Base de datos vectorial | Almacena vectores que representan información, comúnmente usada para realizar búsquedas semánticas. Los datos se consultan en función de su similitud en el espacio vectorial. |
| Bots | Programas automatizados que ejecutan tareas específicas, como responder mensajes o procesar información, comúnmente usados en plataformas como Discord. |
| Búsqueda semántica | Método de recuperación de información basado en el significado del texto, no solo en palabras clave exactas. Se apoya en embeddings y modelos de lenguaje. |
| Chatbot | Programa que simula una conversación humana, capaz de responder preguntas o realizar tareas a través de mensajes de texto. |
| CSV | Formato de archivo de texto plano donde los datos están separados por comas. Es ampliamente usado para exportar o importar información en tablas. |
| Discord | Plataforma de comunicación que permite crear servidores con canales de texto y voz, usada frecuentemente por comunidades de estudio o desarrollo. |
| Embeddings | Representaciones numéricas (vectores) de palabras, frases o documentos que preservan el significado semántico. Usados para búsqueda y clasificación, entre otros. |
| Groq | Plataforma de hardware y software optimizada para ejecutar modelos de inteligencia artificial con alto rendimiento. |
| Indexación | Proceso de organizar y estructurar datos para hacerlos más fácilmente recuperables. |
| JSON | Formato de texto ligero para el intercambio de datos estructurados. Es fácil de leer para humanos y procesar por máquinas. |
| Logs | Registros automáticos de eventos o acciones del sistema que permiten monitorear y depurar el funcionamiento de aplicaciones. |
| Metadatos | Datos que describen otros datos, como autor, fecha, tipo de archivo, etc. Son útiles para organización y búsqueda. |
| Patrón Chain of Responsibility | Patrón de diseño que permite pasar una solicitud a través de una cadena de manejadores hasta que alguno la procese. Se usa para separar el emisor del receptor. |
| Patrón Strategy | Patrón de diseño que permite definir una familia de algoritmos, encapsularlos y hacerlos intercambiables. Se usa para cambiar el comportamiento de un objeto en tiempo de ejecución. |
| Procesos | Instancias independientes de programas en ejecución. Cada proceso tiene su propia memoria, y se usa cuando se necesita ejecutar tareas pesadas de forma paralela. |
| Programación asincrónica | Modelo de programación que permite ejecutar tareas sin bloquear el flujo del programa. Es útil para mejorar el rendimiento en operaciones de entrada/salida. |
| Python | Lenguaje de programación de alto nivel, fácil de leer y aprender. Es muy usado en ciencia de datos, inteligencia artificial y automatización. |
| Threading (Hilos) | Mecanismo que permite ejecutar varias tareas en paralelo dentro de un mismo proceso, compartiendo memoria. Es útil para operaciones concurrentes. |
| Venv | Herramienta de Python para crear entornos virtuales. Permite gestionar dependencias de proyectos de forma aislada. |

# BIBLIOGRAFÍA

Crockford, D. (2006). *Introducing JSON*. Recuperado el 9 de mayo de 2025 de <https://www.json.org/>

Discord Inc. (s.f.). *Documentación para desarrolladores*. Recuperado el 9 de mayo de 2025 de <https://discord.com/developers/docs/intro>

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.

Gilliland, A. J. (2016). *Setting the Stage*. En Baca, M. (Ed.), *Introduction to Metadata*. Getty Research Institute. Recuperado el 9 de mayo de 2025 de <https://www.getty.edu/publications/intrometadata/>

Grinberg, M. (2018). *Flask Web Development* (2ª ed.). O’Reilly Media.

Groq Inc. (2025, 9 de mayo). *Groq Console*. Recuperado de <https://console.groq.com/home>

Python Software Foundation. (s.f.). *venv — Creation of virtual environments*. Recuperado el 9 de mayo de 2025 de <https://docs.python.org/3/library/venv.html>

RFC 4180. (2005). *Common Format and MIME Type for Comma-Separated Values (CSV) Files*. Recuperado el 9 de mayo de 2025 de <https://datatracker.ietf.org/doc/html/rfc4180>

Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4ª ed.). Pearson.

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). *Database System Concepts* (7ª ed.). McGraw-Hill.

Sweigart, A. (2015). *Automate the Boring Stuff with Python: Practical Programming for Total Beginners*. No Starch Press. Recuperado de <https://automatetheboringstuff.com/>

Tanenbaum, A. S. (2015). *Modern Operating Systems* (4ª ed.). Pearson.

Van Rossum, G., & Drake, F. L. (2023). *The Python Language Reference Manual*. Python Software Foundation. Recuperado el 9 de mayo de 2025 de <https://docs.python.org/3/reference/>

Van Rossum, G. et al. (2023). *asyncio — Asynchronous I/O*. Python Software Foundation. Recuperado el 9 de mayo de 2025 de <https://docs.python.org/3/library/asyncio.html>