

TRABAJO DE FIN DE MÁSTER

# Chatbot Basado en Retrieval para la Plataforma Sabentis

Máster Executive en Advanced Analytics & Data Science

**Autores:** Brandon Maldonado Alonso,  
Victor Aranda Belmonte ,  
Verónica Sánchez Muñoz

**Tutores:** Tetiana Klymchuk ,  
Albert Puntí

**Fecha:** 21 de Junio del 2024

1. Resumen Ejecutivo	3
1.1 Justificación del proyecto	3
1.1.1 Objetivos	3
1.1.2 Problema de Estudio	4
1.1.3 Equipo	4
1.1.4 Material	4
1.1.5 Entrega	4
1.1.6 Librerías utilizadas	5
1.1.7 Herramientas utilizadas	5
1.1.8 Metodología	6
1.1.9 Beneficios	6
2. Descripción de Sabentis	7
3. Planificación del Reto	8
3.1 Cronograma del Proyecto	8
4. State of Art	9
4.1 Revisión de la Literatura	9
4.2 Aplicación de Resultados al Proyecto	10
4.2.1 Maximización de la Eficiencia Operativa	10
4.2.2 Mejora de la Satisfacción del Usuario	10
4.2.3 Adaptabilidad a las Necesidades Específicas de Sabentis	10
4.2.4 Fortalezas y Limitaciones de los Modelos de Lenguaje	10
5. Análisis de Datos	11
5.1 Visión General de los Datos	11
5.2 Preprocesamiento de Datos Específicos	12
5.3 Análisis Exploratorio de Datos	13
6. Selección de Solución Técnica	16
6.1 Investigación Inicial	16
6.2 Opciones Evaluadas	16
6.2.1 ChatGPT LLM sin Información Directa	17
6.2.2 LM con Información de los Manuales	18
6.2.3 LLM Finetuned con Información de los Manuales	19
6.3 Solución técnica escogida	20
6.3.1 Retrieval-Augmented Generation (RAG)	20

7. Embedding Model Selection	23
7.1 ¿Qué es un embedding?	23
7.2 Diferentes modelos de generación de embeddings	24
7.2.1 BERT	24
7.2.2 Word2Vec	24
7.2.3 TF-IDF	25
7.2.4 OpenAI (ADA)	25
7.3 Comparación de métricas de incrustación	26
7.3.1 Similitud de Coseno	28
7.3.2 Distancia Euclidiana	29
7.3.3 Métrica Elegida	29
7.4 Retrieval Evaluation	30
7.4.1 Resultados BERT	31
7.4.2 Resultados Word2Vec	33
7.4.3 Resultados TF-IDF	35
7.4.4 Resultados OpenAI (ADA)	36
7.5 Selección del Mejor Modelo	36
7.5.1 ¿Qué Modelo?	36
7.5.2 ¿Por Qué?	36
7.5.3 Modelo Seleccionado	38
7.5.4 Cálculo del Threshold Óptimo	38
8. Retrieval	39
8.1 Retrieval Setup	39
8.2 Retrieval Results for FAQ	40
9. ChatGPT API	42
9.1 Entorno de trabajo y decisiones técnicas	42
9.2 Ejemplos	43
10. Solución de Validación	43
10.1 ¿Cómo hemos validado los resultados?	43
11. Análisis de Calidad	44
12. User Interface	44
13. Próximos pasos	45
14. Referencias Bibliográficas	46
15. Anexos	47

# 1. Resumen Ejecutivo

## 1.1 Justificación del proyecto

El desarrollo de un chatbot avanzado para Sabentis surge como una necesidad imperante para optimizar la gestión de la seguridad y salud en el trabajo (SST), un aspecto crítico en cualquier entorno laboral. Sabentis proporciona un conjunto integral de 43 módulos diseñados específicamente para la gestión de SST, los cuales están categorizados en cuatro áreas principales:

1. Módulos transversales
2. Gestión organizativa y planificación
3. Seguridad, salud y bienestar
4. Comunicación, capacitación y cumplimiento

Estos módulos están respaldados por una extensa documentación que incluye más de 100 manuales explicativos. Para mejorar la gestión de estos recursos y facilitar la interacción de los usuarios con el sistema, es fundamental integrar un chatbot en el software de Sabentis. Este chatbot está diseñado para ofrecer acceso rápido y eficiente a la información contenida en los manuales, permitiendo a los usuarios recibir asistencia en tiempo real y gestionar de manera eficiente las tareas relacionadas con la SST.

La implementación de esta tecnología en la plataforma de Sabentis no solo optimiza la eficiencia operativa y proporciona soporte continuo a los usuarios, especialmente en situaciones de emergencia o riesgo, sino que también se alinea con las tendencias contemporáneas en inteligencia artificial y automatización. Este proyecto otorga a Sabentis una ventaja competitiva significativa en la transformación digital del sector SST, reforzando su compromiso con la innovación y la excelencia operativa. La adopción de un chatbot avanzado no solo moderniza la gestión de SST, sino que también posiciona a Sabentis como un líder en la integración de tecnologías avanzadas para mejorar la seguridad y el bienestar en el entorno laboral.

### 1.1.1 Objetivos

El objetivo principal es el desarrollo e implementación de un chatbot avanzado para la gestión de la seguridad y salud en el trabajo en el software de Sabentis.

1. **Mejorar la eficiencia operativa:** Facilitar el acceso rápido y preciso a la información sobre SST, permitiendo a los usuarios la tomar decisiones informadas de manera ágil.
2. **Automatizar respuestas y gestiones:** Implementar un sistema de respuestas automáticas a consultas frecuentes, liberando al personal para enfocarse en tareas más complejas y estratégicas.
3. **Reducir riesgos laborales:** Permitir una comunicación más fluida y efectiva entre los usuarios y el sistema de gestión de SST, contribuyendo a una identificación y gestión más rápida de los riesgos.

4. **Incrementar la satisfacción del cliente:** Proporcionar respuestas rápidas y personalizadas, mejorando la experiencia del usuario y la percepción del servicio ofrecido por Sabentis.
5. **Impacto a nivel de negocio:** Aumentar la competitividad de Sabentis mediante la adopción de tecnologías avanzadas que mejoren la eficiencia y la capacidad de respuesta del servicio, generando ahorros significativos en costos operativos y mejorando el rendimiento global del negocio.

### 1.1.2 Problema de Estudio

Sabentis cuenta con 43 módulos diseñados específicamente para la gestión de SST, respaldados por una extensa documentación que incluye más de 100 manuales explicativos. Los usuarios no pueden acceder a dicha información de forma rápida y ordenada. La gestión de consultas frecuentes y la necesidad de soporte continuo en el ámbito de SST presentan un desafío significativo. La capacidad de ofrecer respuestas precisas y personalizadas en tiempo real es crucial para la eficacia operativa y la seguridad en el trabajo. Actualmente, la dependencia de respuestas manuales y la falta de automatización pueden llevar a retrasos y errores en la gestión de la información, afectando negativamente la seguridad y la eficiencia operativa.

### 1.1.3 Equipo

El equipo del proyecto, compuesto por Brandon Maldonado Alonso, Victor Aranda Belmonte y Verónica Sánchez Muñoz, ha adoptado una metodología ágil, utilizando herramientas como Trello para la organización del trabajo y sprints semanales para asegurar un avance estructurado y eficiente.

A lo largo del proyecto, se ha prestado especial atención a la optimización del código y la gestión eficiente de recursos, mitigando riesgos asociados como la incapacidad del chatbot para encontrar respuestas adecuadas y el alto costo computacional y económico de implementación. Esta estrategia ha sido crucial para garantizar que el proyecto se mantenga en línea con los objetivos de Sabentis y que la solución desarrollada sea robusta, escalable y alineada con las mejores prácticas de la industria.

### 1.1.4 Material

**Documentación:** Seis manuales de Seguridad y Salud en el Trabajo (SST) y un documento de Preguntas Frecuentes (FAQ) proporcionados por Sabentis, que constituyen la base de datos principal para el entrenamiento del chatbot.

### 1.1.5 Entrega

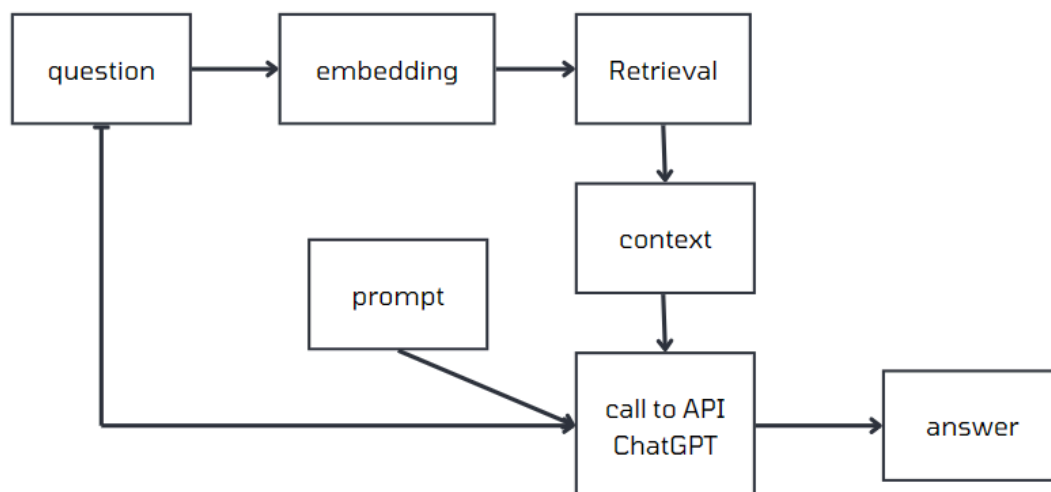
El objetivo de este proyecto es crear una *Proof of Concept* para demostrar la viabilidad de un chatbot usando un pequeño conjunto de información contenida en PDFs proporcionados por Sabentis y por lo tanto poder evaluar la efectividad de las respuestas del mismo, una vez se pueda demostrar la viabilidad del proyecto dando respuestas acertadas, se puede escalar añadiendo todos los PDFs o los que la empresa requiera ya que el modelo será capaz de dar respuestas a las preguntas que el usuario requiera sin importar la cantidad de PDF que

haya en en el conjunto de datos, siempre y cuando la respuesta se encuentre en los PDFs suministrados al modelo e integrar en el entorno web de Sabentis.

Los puntos acordados con Sabentis fueron los siguientes:

- Encontrar el modelo idóneo comparando con el documento de preguntas frecuentes.
- Crear una UI que permita a Sabentis interactuar con este modelo de forma fácil.

## THE FINAL USER FLOW DIAGRAM



\*Imagen 1: Diagrama de flujo usuario final [Creación propia]

### 1.1.6 Librerías Utilizadas

**Python:** Lenguaje de programación utilizado para el desarrollo del chatbot.

**PyMuPDF:** Biblioteca para la extracción y procesamiento de texto de documentos PDF.

**NLTK** (Natural Language Toolkit): Biblioteca para el procesamiento de lenguaje natural.

**SpaCy:** Biblioteca avanzada para el procesamiento de lenguaje natural.

**Hugging Face Transformers:** Plataforma que proporciona modelos de lenguaje preentrenados como BERT.

### 1.1.7 Herramientas Utilizadas:

**Trello:** Herramienta de gestión de proyectos utilizada para la organización del trabajo y la planificación de sprints.

**Docker:** Herramienta para la creación de contenedores que facilita el despliegue y la gestión del entorno de desarrollo.

### 1.1.8 Metodología

Decidimos implementar la metodología SCRUM, esto implica hacer sprints de 2-3 semanas de duración, cada uno de estos sprints ha estado auditado por los tutores por tal de marcar objetivos y tareas a conseguir al finalizar el sprint, siendo conscientes del proyecto total y como los diferentes sprints ayudaban a conseguir llegar al resultado final.

Ya que el proyecto de TFM consta de 2 presentaciones de nuestros resultados y progresos, separamos el desarrollo del proyecto en 2 partes.

La primera parte se centró en crear un pipeline funcional, desde la entrada de los datos y el proceso hasta llegar al final, que incluía una UI para presentar los resultados del proyecto.  
[imagen1]

La segunda parte se centró en entender cada uno de los modelos de generación de embeddings y retrieval, e ir aplicándolos al flujo de trabajo creado en la primera parte, ya que al final una vez teníamos el esqueleto del proyecto es fácil ir cambiando de modelo. Podíamos ver las respuestas al haber ya creado la UI, con esto podíamos ir haciendo pruebas de cada uno de los modelos, comparando sus respuestas con el documento de preguntas frecuentes y entonces entendiendo si un modelo estaba devolviendo respuestas correctas, si debíamos seguir ajustando sus diferentes parámetros o si lo descartamos ya que no aportaba respuestas idóneas para la necesidad de este proyecto.

### 1.1.9 Beneficios

El desarrollo de este chatbot aporta una solución innovadora que mejora significativamente la gestión de la seguridad y salud en el trabajo (SST) en Sabentis. Al automatizar respuestas y gestionar múltiples interacciones simultáneamente, el chatbot libera al personal para que se enfoque en tareas más complejas y estratégicas, incrementando así la productividad y reduciendo los costos operativos.

La disponibilidad continua del chatbot garantiza que los usuarios reciban soporte en tiempo real, lo cual aumenta la satisfacción y la lealtad del cliente, fortaleciendo la posición de Sabentis en el mercado. Además, la adopción de esta tecnología optimiza la gestión de consultas relacionadas con SST, permitiendo una mayor eficiencia operativa y una mejor asignación de recursos humanos.

Este proyecto representa un avance importante en el campo de la inteligencia artificial aplicada a la gestión de SST. La implementación y evaluación de modelos avanzados como RAG en un entorno práctico proporcionan insights valiosos sobre la eficacia de estas tecnologías en aplicaciones reales.

Además, el uso de técnicas de procesamiento de lenguaje natural para mejorar la precisión y personalización de las respuestas ofrece un caso de estudio relevante para futuras investigaciones en el área de chatbots y automatización. Los resultados y metodologías desarrolladas en este proyecto pueden servir como referencia para la integración de tecnologías similares en otros sectores, ampliando su impacto y relevancia.

La implementación del chatbot tiene un impacto significativo en el negocio de Sabentis. Estudios recientes indican que en 2024, los chatbots podrían generar ahorros anuales de

hasta 11 mil millones de dólares en sectores clave como el retail, la banca y la salud, gracias a la automatización de tareas repetitivas y la mejora en la eficiencia del servicio al cliente. [\(Tidio\)](#) [\(Business Wire\)](#).

La capacidad del chatbot para ofrecer soporte continuo 24/7 es esencial para manejar emergencias y consultas urgentes de SST, mejorando la respuesta y mitigación de riesgos. En consecuencia, se espera que Sabentis obtenga beneficios económicos significativos, optimizando sus operaciones y fortaleciendo su competitividad en el mercado. [\(Ubique Digital Solutions\)](#).

Como hemos mencionado anteriormente la automatización de las respuestas a consultas frecuentes mediante el chatbot tiene un impacto positivo en los empleados de Sabentis que previamente realizaban estas tareas manualmente. Al liberar a estos empleados de tareas repetitivas y rutinarias, se les permite concentrarse en actividades más complejas y enriquecedoras, lo que puede aumentar su satisfacción laboral y su desarrollo profesional. Esta transición promueve un ambiente de trabajo más motivador y menos estresante, contribuyendo a la retención de talento y mejorando el bienestar general de los empleados.

## 2. Descripción de Sabentis



Sabentis es una empresa líder en la transformación digital de la gestión de la seguridad y salud en el trabajo (SST). Su software optimiza la prevención de riesgos laborales mediante la automatización y centralización de información clave, utilizando inteligencia artificial generativa y Power BI. Esto permite a las organizaciones predecir riesgos, mejorar protocolos de seguridad y tomar decisiones informadas.

Sabentis ofrece un conjunto integral de 43 módulos diseñados para la gestión de la seguridad y salud en el trabajo, agrupados en cuatro categorías principales:

1. Módulos transversales
2. Gestión organizativa y planificación
3. Seguridad, salud y bienestar
4. Comunicación, capacitación y cumplimiento

De los cuales Sabentis dispone de más de 100 manuales relacionados con la Seguridad y la Salud en el trabajo. Para mejorar aún más la gestión y facilitar la interacción de los usuarios con el sistema, surge la necesidad de integrar un chatbot en su software.

Este chatbot permitirá a los usuarios acceder rápidamente a la información, recibir asistencia en tiempo real y gestionar eficientemente las tareas relacionadas con la seguridad y salud en el trabajo.



### 3. Planificación del Reto

El proyecto se centra en el desarrollo de un chatbot personalizado, diseñado específicamente para integrarse con la plataforma de Sabentis, proporcionando un servicio de atención y asistencia directa, optimizado para las necesidades específicas de la empresa y sus usuarios. Este sistema no está concebido como un chatbot de propósito general, enfocado en mantener conversaciones genéricas, sino como una herramienta especializada, capaz de ofrecer respuestas y soluciones pertinentes y de alto valor añadido a las consultas específicas de los usuarios.

Aún siendo un chatbot personalizado, este chatbot utiliza Retrieval Augmented Generation (RAG), con lo cual no es un chatbot entrenado exclusivamente con los PDFs otorgados, sino que es un chatbot escalable. Sabentis le puede introducir más PDFs y seguirá funcionando eficientemente, y a su vez, cualquier otra empresa que utilice una documentación en PDF similar a Sabentis puede utilizar el código fuente, solo basta con ajustar ciertos parámetros dependiendo de cómo la información esté clasificada y sería capaz de responder a las preguntas de otra empresa.

#### 3.1 Cronograma del Proyecto

El cronograma del proyecto se desarrolló meticulosamente para asegurar un avance organizado y eficiente en cada fase del desarrollo del chatbot para Sabentis. A continuación, se detalla de forma extensa y clara cómo transcurrieron los trabajos a lo largo de varios meses, describiendo cada hito importante y las actividades realizadas por el equipo.

- **Fase de Planificación y Configuración Inicial | 14/02/24:** Definición de roles y configuración inicial del entorno de trabajo.
- **Fase de Análisis y Comprensión de Necesidades | 19/02/24:** Kickoff meeting con Sabentis.
- **Fase de Lectura y Análisis de la Documentación | Semana del 03/03/24:** Comienzo de la fase de lectura y análisis de los manuales recibidos.
- **Fase de Limpieza de Datos y Desarrollo Inicial | Semana del 11/03/24:** Reunión de equipo para discutir la metodología óptima de limpieza de documentos.
- **Fase de Creación de la UI y Investigación de Modelos | Semana del 25/03 - 13/04:** Creación de la UI e Investigación de primeros modelos: Word2Vec, Word2Vec pre-entrenado, TF-IDF y BERT.
- **Fase de Pruebas Comparativas y Ajustes Finales | Semana del 14/04 - 07/05:** Investigación de Modelos e Integración de ChatGPT a los resultados de los modelos.
- **Fase de Revisión y Análisis de Modelos | Semana del 07/05 - 21/05 :** Continuación de investigación de Modelos e Integración de un nuevo modelo OpenAI.
- **Fase Final Elección del Modelo Final | Semana del 21/05 - 04/06 :** Elección del modelo OpenAI como solución final y retoques y testeo de la UI.

## 4. State of Art

En esta sección, se realiza un análisis exhaustivo de los resultados obtenidos en estudios previos y la literatura existente relacionada con el uso de chatbots y modelos de procesamiento de lenguaje natural (NLP) en el ámbito de la seguridad y salud en el trabajo (SST). Este estudio proporciona un contexto teórico y práctico que fundamenta las decisiones metodológicas y técnicas adoptadas en el desarrollo del chatbot para Sabentis.

### 4.1 Revisión de la Literatura

Se llevó a cabo una revisión sistemática de la literatura para identificar investigaciones y estudios relevantes que hayan abordado la implementación de chatbots en diversos sectores, con un enfoque particular en la SST. Las principales fuentes consultadas incluyeron artículos académicos, conferencias, tesis de maestría y estudios de caso publicados en bases de datos reconocidas como IEEE Xplore, Springer, Elsevier y Google Scholar.

Principales hallazgos:

- **Eficiencia Operativa:** Diversos estudios han demostrado que los chatbots pueden mejorar significativamente la eficiencia operativa al automatizar respuestas a preguntas frecuentes y gestionar múltiples interacciones simultáneamente. Por ejemplo, el trabajo de Jain et al. (2018) destaca cómo los chatbots pueden reducir la carga de trabajo administrativo en el sector salud, permitiendo a los profesionales de la salud centrarse en tareas más complejas y críticas. Similarmente, un estudio de Adamopoulou y Moussiades (2020) muestra cómo la implementación de chatbots en empresas puede liberar recursos humanos para actividades estratégicas, mejorando así la eficiencia general de la organización.
- **Reducción de Costos:** La implementación de chatbots ha sido asociada con una reducción notable en los costos operativos, particularmente en áreas de atención al cliente y soporte técnico. Un estudio realizado por Deloitte (2019) en el sector financiero muestra que los chatbots pueden gestionar hasta el 80% de las consultas de clientes sin intervención humana, resultando en ahorros significativos en costos operativos. Otro estudio de Gartner (2020) proyecta que para 2024, los chatbots podrían generar ahorros anuales de hasta 11 mil millones de dólares en sectores clave como el retail, la banca y la salud.
- **Mejora en la Satisfacción del Usuario:** Los chatbots que utilizan modelos avanzados de NLP han mostrado mejorar la satisfacción del usuario al proporcionar respuestas rápidas y precisas, ajustadas a las consultas específicas de los usuarios. Un estudio de McKinsey (2021) en el sector de servicios muestra que los clientes valoran la rapidez y precisión de las respuestas proporcionadas por chatbots, lo que resulta en una mejora significativa en la satisfacción del cliente y en su lealtad.

## 4.2 Aplicación de Resultados al Proyecto

Los resultados de los estudios analizados proporcionan una base sólida para las decisiones técnicas y metodológicas adoptadas en este proyecto. Al comprender las fortalezas y limitaciones de cada modelo de lenguaje y las experiencias previas de implementación de chatbots en diferentes sectores, se ha diseñado un sistema que maximiza la eficiencia operativa, mejora la satisfacción del usuario y se adapta a las necesidades específicas de Sabentis.

### 4.2.1 Maximización de la Eficiencia Operativa

Los casos de Tesla, Alcoa y DuPont demuestran cómo la implementación de chatbots puede reducir significativamente el tiempo de respuesta a consultas repetitivas y mejorar la precisión de la información proporcionada. En el contexto de Sabentis, estas mejoras se traducirán en una gestión más ágil y precisa de las consultas sobre seguridad y salud en el trabajo (SST). La automatización de respuestas a preguntas frecuentes permitirá al personal enfocarse en tareas más complejas y estratégicas, mejorando la productividad general de la empresa ([Tesla](#)) ([Alcoa](#)) ([Alcoa](#)).

### 4.2.2 Mejora de la Satisfacción del Usuario

Los chatbots, al proporcionar respuestas rápidas y precisas, han demostrado ser efectivos en mejorar la satisfacción del usuario. Los estudios de McKinsey (2021) y otros indican que los usuarios valoran la rapidez y exactitud en las respuestas, lo cual es crucial para mantener una alta satisfacción y lealtad del cliente. En el caso de Sabentis, esto se traducirá en una mejor experiencia para los usuarios, quienes podrán acceder a información crítica de SST en tiempo real y recibir soporte continuo, especialmente en situaciones de emergencia .

### 4.2.3 Adaptabilidad a las Necesidades Específicas de Sabentis

Los estudios revisados destacan la importancia de personalizar los chatbots para adaptarse a las necesidades específicas de cada organización. En Sabentis, el chatbot será diseñado para manejar consultas relacionadas con más de 100 manuales explicativos de SST y 43 módulos especializados. Esta personalización asegurará que el chatbot pueda proporcionar información precisa y relevante, mejorando la efectividad de las respuestas y facilitando una cultura de seguridad proactiva dentro de la empresa.

### 4.2.4 Fortalezas y Limitaciones de los Modelos de Lenguaje

La revisión de modelos como Word2vec, TF-IDF, BERT y los modelos de OpenAI ha sido fundamental para seleccionar la tecnología más adecuada para Sabentis. Cada modelo tiene sus propias fortalezas y limitaciones:

- **Word2vec:** Eficiente para generar vectores de palabras, pero con limitaciones en capturar relaciones contextuales complejas .
- **TF-IDF:** Simple y efectivo para recuperar información basada en la relevancia de términos, pero limitado en su capacidad para captar relaciones semánticas .
- **BERT:** Avanzado en la comprensión del contexto de palabras, pero con alta demanda computacional .

- **Modelos de OpenAI:** Capaces de generar texto coherente y relevante en contextos complejos, aunque requieren significativos recursos computacionales y pueden generar respuestas inexactas o irrelevantes .

## 5. Análisis de Datos

### 5.1 Visión General de los Datos

Sabentis proporcionó seis manuales en PDF y un documento de preguntas frecuentes (FAQ) que sirven como base para el desarrollo del chatbot. Estos documentos incluyen los siguientes temas:

1. Auditorías
2. Ausentismo
3. Estructura Organizativa
4. Identificación y Evaluación de Riesgos
5. Información Documentada
6. Planes de Emergencia

Estos documentos constituyen el corpus principal sobre el cual se entrenará y evaluará el chatbot para garantizar que pueda proporcionar respuestas precisas y útiles a las consultas relacionadas con la seguridad y salud en el trabajo.

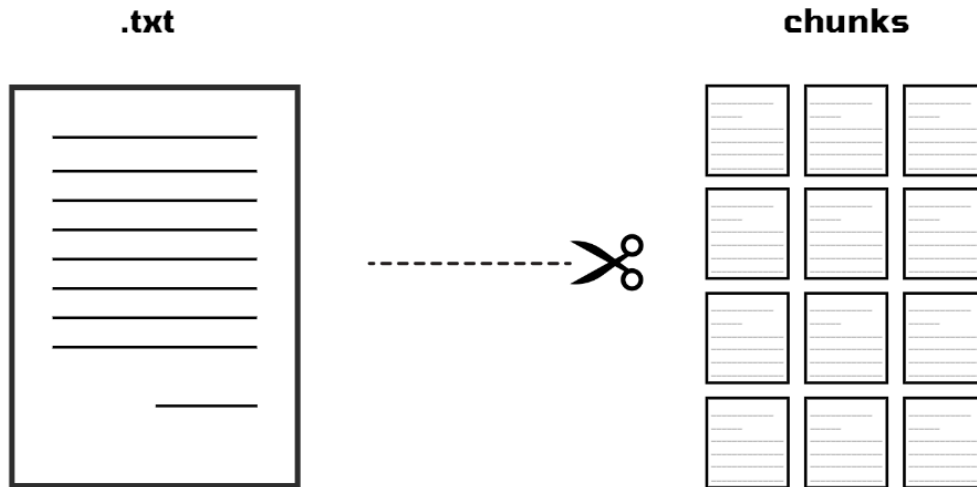
### 5.2 Preprocesamiento de Datos Específicos

El preprocesamiento de los datos se llevó a cabo utilizando Python y la biblioteca PyMuPDF. Los principales pasos del preprocesamiento incluyen:

1. **Extracción de Texto:** Se extrajo el texto de los documentos PDF, eliminando elementos no textuales como encabezados, pies de página e imágenes que no aportan valor a la información requerida.



2. **Limpieza de Datos:** Se eliminaron datos redundantes y se corrigieron errores en el texto extraído para asegurar la coherencia y precisión de la información.
3. **Chunking:** Los documentos fueron divididos en fragmentos (chunks) manejables. Este proceso permite una mejor gestión y análisis de la información contenida en los documentos.



Para dividir los documentos en fragmentos, se utilizó la librería **RecursiveCharacterTextSplitter** de Python, con un tamaño de fragmento (chunk size) de 500 caracteres y un solapamiento de 60 caracteres entre fragmentos para mantener el contexto.

**.txt**

Sebenitis plataforma gestión especializado ambito Seguridad Salud Trabajo adaptado tecnologia modelo gestión prevención . plataforma distinto tipo auditoria objetivo , figura : evaluación estándares Internacionales ( ISO ) Metodología So , responder conjunto norma , requisito procedimiento reflejar imagen sistema . prevención riesgo laboral

sistema : prevención riesgo laboral empresa .

usuario acceder módulo Auditorías : • entrada menú lateral plataforma : • dashboard " Auditorías " panel principal , requerir empresa : contexto : i • Diagnósticos ISO • Metodología So . • Planificación ejecución medida . • Comparativa estudio . , visualizar dashboard contener datos estadístico , gráfico : descargar mostrar información :

Gráfica Descripción : conformidad identificado : cumplir vs cumplir : Gráfica circular indicar cumplimiento conformidad : identificado . medida : Gráfica dato circular determinar número medida : gestión . vencimiento medida : Gráfica línea vencimiento fecha medida 6 mes , área tipo vencimiento ( aprobación , validación , implementación , verificación , etcétera ) .

usuario acceder dashboard " Diagnósticos ISO " visualizar pantalla conjunto : estándar reconocimiento internacional objetivo ayudar empresa : establecer nivel homogeneidad relación gestión , prestación : servicio desarrollo producto industria . • ISO 45001 : Sistema gestión SST . • ISO 14001 : Sistema gestión ambiente . • ISO 9001 : Sistema gestión calidad . • ISO 39001 : Sistema

Sistema gestión calidad . • ISO 39001 : Sistema gestión seguridad vial . • ISO 37001 : Sistema gestión antisoborno . • ISO 22301 : Sistema gestión continuidad negocio . • ISO 26000 : Guía responsabilidad social . • ISO 50001 : Sistema gestión energía .

listado estudio ISO crear estudio acceder : botón " evaluación " . campo formulario marcado " carácter " obligatorio caso completir crear / editar registro evaluación . formulario seleccionar cuestionario : querer evaluar . defecto aparecer marcado selección concreto : finalidad estudio parcial

finalidad estudio parcial específico . crear estudio : existir vigente abierto . caso plataforma mostrar : mensaje indicar crear estudio .

creado estudio pestaña " Datos evaluación normativa " habilitar : botón " Sistema " evaluación normativa " permitir listar estudio : ficha dato : listado : eliminar estudio . adicionalmente , habilitar pestaña : Evaluación aspecto legal , Plan acción , Gráficas resultado , Resumen , Generar documentación : Documentos ( pestaña habilitar cerrar : evaluación ) .

evaluación aspecto legales :

pestaña evaluación aspecto legal permitir usuario evaluar cuestionario : aplicable evaluación . aparecer cuestionario usuario seleccionar : previamente : pestaña " Datos evaluación normativa " , finalidad : estudio parcial específico . completar evaluación aspecto legal , ítem estándar disponer opción : respuesta : cumplir / cumplir / aplicar .

variable respuesta vínculo " " habilitar ventana : lateral contener información asociado cuestión seleccionado . opción " Añadir archivo " habilitar : conforme completar evaluación permitir : cargar pregunta 5 archivo evidencia .

pestaña barra mostrar : porcentaje avance evaluación : aspecto legal actualizar conforme guardar dato . descargar : permitir guardado parcial significar : " salir funcionalidad " : finalizar evaluación , guardar avance previamente guardado .

13 : evaluar totalidad ítem estándar , pantalla sustituir : botón " Guardar " botón " Finalizar evaluación " permitir guardar respuesta , bloquear : cuestionario respuesta definido .

generalmente , configuración establecido , ítem evaluación marcado : opción " cumplir " , generar conformidad , visualizar pestaña " Plan acción " . visualizar finalizar evaluación ; realizar : 14 : guardado parcial mostrar conformidad independientemente : respuesta " cumplir " . PLAN ACCIÓN

PLAN ACCIÓN : pestaña mostrar listado conformidad resultado evaluación : aspecto legal previamente finalizado : ítem estándar marcado respuesta : " cumplir " : pestaña previo . listado opción búsqueda , añadir : filtro , configurar columna , paginado , botón Volver acción editar redirigir ficha : conformidad .

ficha conformidad mostrar apartado : • Datos conformidad : muestra dato referente conformidad , único campo editar apartado " Cumple " , defecto , mostrar marcado " cumplir " , campo editable marcar : " Cumple " haber implementar medida asociado : conformidad .

• medida correctiva : apartado mostrar medida correctiva añadido : usuario configurada automáticamente permitir : gestión medida llegar Cerrada . campo formulario : marcado " carácter obligatorio " , caso completario : permitir añadir medida . ( Manual gestión medida ) . gráfico resultado

gráfica resultado : pestaña " Gráficas resultado " mostrar tipo gráfico , precisa : resultado porcentaje cumplimiento estándar : referencia porcentaje cumplimiento ciclo . gráfica superior disponer componente transversal nivel : plataforma permitir usuario siguiente acción :

17 • Refreshar : herramienta permitir actualizar gráfico . • descargar imagen : permitir descargar información visualizado gráfica . • Exportar : hoja cálculo : permitir descargar información hoja cálculo .

pestaña mostrar : sintetis información correspondiente proceso : evaluación aspecto legal estudio . pestaña cerrar : evaluación : botón " cerrar : evaluación " : habilitar únicamente usuario : establecer , medida correctiva conformidad . GENERAR DOCUMENTACIÓN

pestaña generar documentación habilitar cerrar estudio pestaña : resumen , pantalla generar informe evaluación normativa : formato PDF adjuntar archivo caso necesario . informe : evaluación generar : forma ilimitado . descargar informe : clicar directamente nombre : autoevaluación ( link azul ) . listado opción : búsqueda , añadir filtro .

opción búsqueda , añadir filtro , configurar columna , paginado botón Guardar .

pestaña Documentos habilitar cerrar estudio pestaña : resumen mostrar pantalla informe evaluación .

usuario acceder dashboard " Metodología So " visualizar listado auditoria : realizado Metodología So . evaluación basado 5 principio facilitar : dinámica

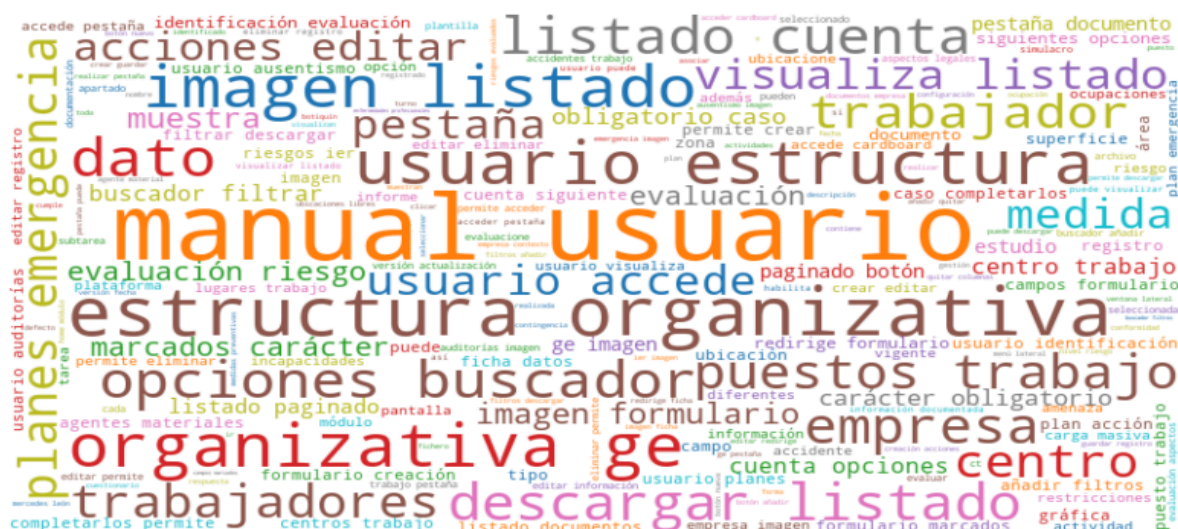
**60 overlap**

**500 chunk size**

**60 overlap**

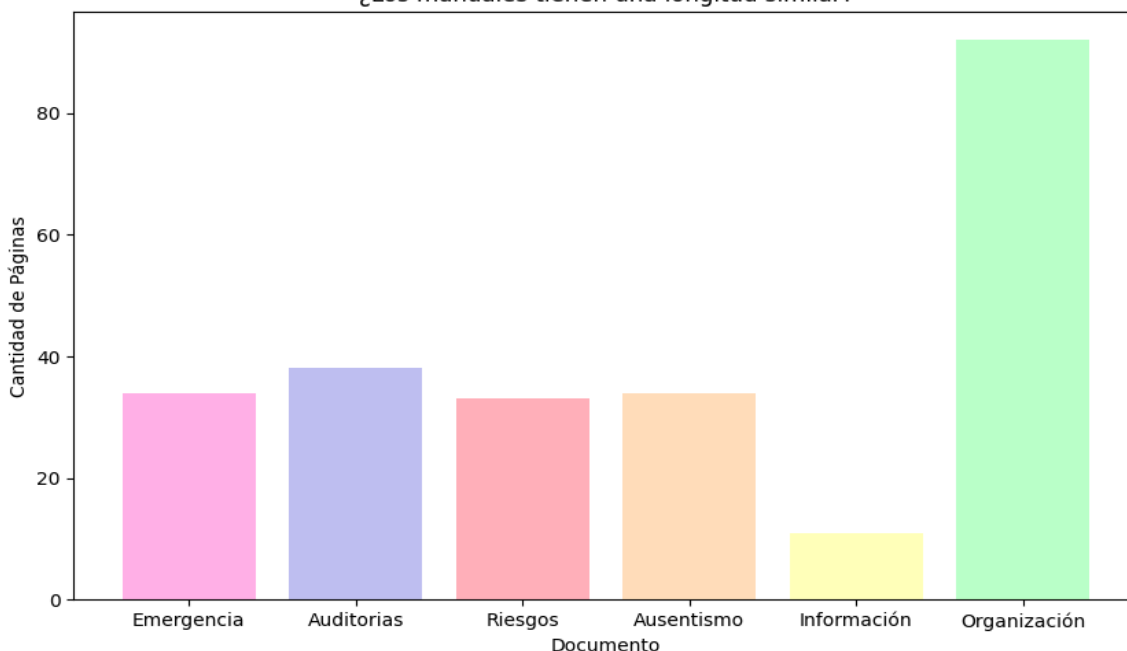
El análisis exploratorio de los datos incluyó varias técnicas para entender mejor la estructura y contenido de los documentos. Algunas de las técnicas utilizadas fueron:

- ¿Qué palabras predominan en los documentos?



2. **Análisis de la longitud de los documentos:** Se analizaron las longitudes de los documentos para detectar si todos tenían las mismas características.

¿Los manuales tienen una longitud similar?



3. **Análisis de Patrones:** Se exploraron los documentos visualmente para detectar patrones, información innecesaria y secciones clave que podrían influir en el rendimiento del chatbot.

Durante el análisis, se observó que los documentos tenían una cantidad de páginas similar, con la excepción de dos documentos que tenían una cantidad de páginas significativamente diferente. Esta observación ayudó a determinar si esos documentos podrían considerarse outliers en el conjunto de datos.

Analizamos cada manual por separado y en conjunto, que palabras son mas y menos habituales, que unión de 2 y 3 palabras son mas y menos habituales, por tal de entender mejor el contenido de los documentos y sacar insights a la hora de poder mejorar modelos como TF-IDF. Se muestran palabras de la unión de todos los documentos.

Detectamos palabras residuales que aparecen o en un solo documento o muy pocas veces, y palabras muy generales que aparecen en todos los documentos.

	Palabra	Frecuencia
0	listado	288
1	trabajo	196
2	usuario	179
3	pestaña	143
4	permitir	141
5	editar	138
6	evaluación	127
7	acción	120
8	formulario	116
9	visualizar	110
10	acceder	105
11	trabajador	96
12	centro	89
13	descargar	88
14	empresa	86
15	eliminar	78
16	puesto	73
17	opción	73
18	añadir	70
19	medida	65

Palabras más frecuentes

	Palabra	Frecuencia
0	nombre	6
1	cardboard	6
2	generado	6
3	archivo	7
4	documentos	8
5	información	8
6	módulo	8
7	plataforma	8
8	informe	9
9	medidas	10
10	nivel	10
11	evaluar	10
12	plantilla	11
13	riesgos	11
14	columna	12
15	quitar	12
16	filtro	12
17	iso	13
18	cerrar	14
19	legal	14

Palabras menos frecuentes

	<b>Bigram</b>	<b>Frecuencia</b>
0	(visualizar, listado)	71
1	(opción, buscador)	70
2	(descargar, listado)	67
3	(puesto, trabajo)	61
4	(acción, editar)	56
5	(centro, trabajo)	52
6	(usuario, acceder)	51
7	(siguiente, opción)	44
8	(buscador, filtrar)	44
9	(obligatorio, caso)	43
10	(listado, paginado)	42
11	(redirigir, formulario)	39
12	(listado, opción)	36
13	(formulario, creación)	34
14	(carácter, obligatorio)	32
15	(filtrar, descargar)	32
16	(marcado, carácter)	31
17	(permitir, crear)	29
18	(caso, completarlo)	28
19	(completarlo, permitir)	27

Unión de 2 palabras más frecuentes

	<b>Trigram</b>	<b>Frecuencia</b>
0	(marcado, carácter, obligatorio)	46
1	(siguiente, opción, buscador)	44
2	(opción, buscador, filtrar)	44
3	(carácter, obligatorio, caso)	43
4	(descargar, listado, paginado)	42
5	(obligatorio, caso, completarlo)	40
6	(caso, completarlo, permitir)	38
7	(formulario, marcado, carácter)	36
8	(campo, formulario, marcado)	35
9	(redirigir, formulario, creación)	33
10	(buscador, filtrar, descargar)	32
11	(filtrar, descargar, listado)	32
12	(completarlo, permitir, crear)	32
13	(listado, siguiente, opción)	26
14	(listado, opción, buscador)	26
15	(pestaña, visualizar, listado)	19
16	(permitir, crear, editar)	19
17	(crear, editar, registro)	19
18	(acción, editar, eliminar)	19
19	(usuario, acceder, pestaña)	19

Unión de 3 palabras más frecuentes

Al tratarse de manuales de uso de la plataforma hay muchas referencias a acciones, elementos de la propia UI de la web de Sabentis, ya que muchos de los documentos aportados no solo muestran información de reglas y normas generales en el entorno de la seguridad y salud, si no ayuda al usuario para poder realizar trámites sobre la seguridad y salud en su centro de trabajo mediante el entorno de Sabentis.

Entendemos pues que la complejidad no está en la seguridad y salud, si no en darle al usuario la capacidad de aprender a usar la plataforma de una forma más fácil gracias a contar con una inteligencia artificial que desde cualquier lugar de la página web pueda responder a cualquier pregunta.



## 6. Selección de Solución Técnica

### 6.1 Investigación Inicial

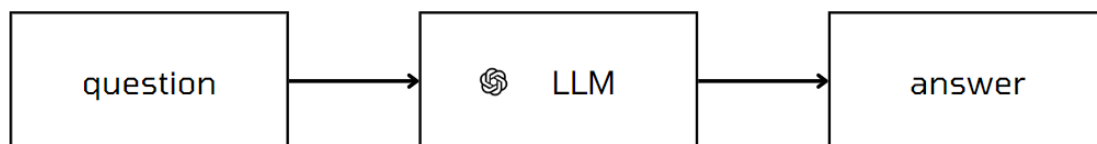
Para determinar el proceso más adecuado para el desarrollo del chatbot, comenzamos investigando diversos artículos académicos y documentación técnica disponible en internet.

Buscamos identificar las mejores prácticas y las metodologías más eficaces para implementar un chatbot capaz de manejar información específica contenida en los manuales de Sabentis.

### 6.2 Opciones Evaluadas

#### 6.2.1 ChatGPT LLM sin Información Directa:

Nuestra primera opción fue implementar directamente ChatGPT LLM. Sin embargo, descartamos esta solución de inmediato ya que, sin la información directa de los manuales, el sistema tomaba datos de internet, lo cual resultaba en respuestas inexactas o inventadas ("alucinaciones").



Realizamos la prueba preguntando a ChatGPT sin contexto. Como era previsible, ChatGPT alucinaba en muchas respuestas, aunque algunas respuestas oficiales como normativas ISO eran correctas. No podíamos confiar en esta aproximación para proporcionar información correcta y exclusiva de Sabentis.

Ejemplo de una pregunta de Sabentis a ChatGPT:

**Pregunta:** ¿Qué elementos específicos se incluyen en un plan de emergencia generado a través de este módulo?

Respuesta de ChatGPT:

Un plan de emergencia generado a través de un módulo integral generalmente incluye los siguientes elementos específicos:

1. Identificación de riesgos:

- Análisis de posibles emergencias (incendios, terremotos, inundaciones, etc.)
- Evaluación de las áreas de mayor riesgo dentro de las instalaciones

2. Establecimiento de responsabilidades:

- Definición del equipo de respuesta a emergencias y sus roles específicos

- Asignación de responsabilidades a los empleados y coordinación con las autoridades locales

3. Procedimientos de respuesta:

- Protocolos de evacuación (rutas de escape, puntos de reunión)
- Procedimientos específicos para diferentes tipos de emergencias (cómo actuar en caso de incendio, derrames químicos, etc.)
- Instrucciones para primeros auxilios y atención médica

4. Recursos y equipos:

- Inventario de equipos de emergencia (extintores, botiquines de primeros auxilios, sistemas de alarma, etc.)
- Ubicación y mantenimiento de equipos de seguridad
- Listado de contactos de emergencia (bomberos, policía, servicios médicos, etc.)

5. Capacitación y simulacros:

- Programas de capacitación regular para todos los empleados sobre procedimientos de emergencia
- Realización periódica de simulacros para ensayar la respuesta ante emergencias

6. Comunicación:

- Protocolos de comunicación interna y externa durante una emergencia
- Métodos para informar rápidamente a todos los empleados y a las autoridades pertinentes

7. Documentación y evaluación:

- Registro de todas las incidencias y respuestas ante emergencias
- Evaluación y actualización regular del plan de emergencia en base a experiencias previas y cambios en las instalaciones o el entorno

8. Plan de recuperación:

- Estrategias para la restauración de las operaciones normales después de una emergencia
- Procedimientos para la evaluación de daños y la coordinación de reparaciones

9. Consideraciones especiales:

- Adaptaciones para personas con discapacidades
- Planes específicos para la protección de datos y tecnología

Estos elementos se integran para asegurar una respuesta coordinada, eficiente y efectiva ante cualquier situación de emergencia, minimizando los riesgos y daños para las personas y la propiedad.

**Como podemos comprobar en la respuesta, si bien parece tener sentido, es un sentido muy general, y no especifica elementos concretos de la plataforma Sabentis al no tener contexto.**

### 6.2.2 LLM con Información de los Manuales:

La siguiente opción fue utilizar ChatGPT LLM añadiendo los manuales de Sabentis. Aunque esta solución mejoró ligeramente la precisión, seguía generando respuestas alucinadas, lo que la hizo inadecuada para nuestras necesidades.



Cargamos un entorno OpenWebUI usando Docker, donde se le pueden añadir diferentes LLM locales y cargar documentos.

Subimos distintos PDFs a Ollama, y a Mixtral, y le hicimos preguntas sobre ellos. Aunque la diferencia con ChatGPT sin contexto fue significativa, Ambos modelos aunque Mixtral era más eficiente que Ollama al estar entrenado en español aún se inventaban respuestas.

**Pregunta:** ¿Qué elementos específicos se incluyen en un plan de emergencia generado a través de este módulo?

Respuesta de LLM + Documents:

Un plan de emergencia generado a través del módulo descrito en el documento incluye varios elementos específicos que son esenciales para garantizar una respuesta adecuada ante situaciones de emergencia. A continuación, se detallan estos elementos según el manual:

1. Datos del Plan de Emergencia: Información general sobre el plan, incluyendo las fechas de inicio y fin de vigencia, así como datos específicos del centro de trabajo y su ocupación.
2. Medios de Protección: Lista y detalles de los recursos disponibles para responder ante emergencias, tales como instalaciones centralizadas, instalaciones de emergencia, señalización, sistemas de detección automática y sistemas de extinción.
3. Turnos y Equipos: Información sobre los turnos de trabajo y los equipos de respuesta e intervención asignados a cada turno, con detalles sobre los miembros titulares y suplentes de cada equipo.
4. Directorio Externo: Directorio de servicios externos con los que se puede contactar en caso de emergencia, incluyendo entidades de socorro, planes de ayuda mutua o cadenas de llamadas.
5. Puntos de Reunión: Listado y ubicación de los puntos de reunión e instalaciones de emergencia dentro del plan, visualizados en un mapa.

6. Evaluación de Amenazas: Identificación y evaluación de las amenazas presentes en el centro de trabajo para determinar el nivel de riesgo y tomar medidas adecuadas para mitigarlo.

7. Análisis de Riesgo: Resumen de la evaluación de amenazas con puntuaciones y nivel de riesgo, disponible una vez que el plan se ha declarado vigente.

8. Plan de Acción: Listado de las medidas de control que se deben implementar para reducir el riesgo asociado a las amenazas evaluadas, con estados de las medidas (pendiente de planificar, pendiente de evidencia).

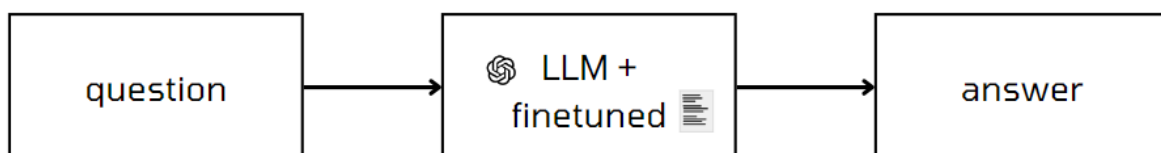
9. Generación de Documentos: Funcionalidad para generar y descargar informes del plan de emergencia en formatos PDF y Word, permitiendo la documentación y registro de todas las acciones y evaluaciones realizadas.

10. Gestión de Botiquines: Registro y mantenimiento de los botiquines existentes en la empresa, asegurando que todos los productos estén vigentes y disponibles para su uso en emergencias.

Estos elementos forman parte integral del plan de emergencia y permiten a las organizaciones planificar, organizar, dirigir y controlar eficazmente la respuesta ante situaciones de emergencia, minimizando los riesgos y daños potenciales.

**Las respuestas de este método son más específicas ya que empieza a tener contexto de las preguntas al pasarle los PDF, pero aún así tiende a alucinar, es decir querer extender la respuesta más allá del contenido del PDF. Con lo que aunque la información parece correcta ya que se le ve contexto esto puede hacer confiar al usuario de que toda la información es correcta cuando está inventándose información o añadiendo información genérica que no tiene que ver con Sabentis.**

### 6.2.3 LLM Finetuned con Información de los Manuales:



LLM finetuned se refiere a entrenar un modelo específico con la información contenida en los PDF. Este es el siguiente paso de las pruebas propuestas. Sin embargo, descartamos este método debido a varias razones. Aunque es cierto que, al estar entrenado con todo el contexto de Sabentis, el modelo podrá proporcionar respuestas más precisas, esto no evitará que ocasionalmente dé respuestas incorrectas. Esto se debe a que, además del entrenamiento específico de Sabentis, el modelo también cuenta con un entrenamiento más general.

Además, a nivel de gestión de recursos, este método es costoso. Cada vez que Sabentis genere un nuevo PDF o modifique la información de un PDF existente, será necesario reentrenar el modelo. Este proceso implica una pérdida de tiempo y recursos hasta que el modelo vuelva a estar operativo con toda la información actualizada. Hemos observado que los PDFs han tenido varias modificaciones en su historial de cambios, lo que indica que la información varía con el tiempo. Cuando se implemente una nueva mejora en la web de Sabentis, se añada un nuevo módulo, etc., será necesario empezar desde cero y reentrenar el modelo.

Para concluir, este método no es sostenible a largo plazo en términos de seguridad en las respuestas y gestión de recursos técnicos.

## 6.3 Solución técnica escogida

### 6.3.1 Retrieval-Augmented Generation (RAG)

Finalmente, después de un análisis exhaustivo y de la revisión de varios artículos clave, decidimos utilizar la metodología de Retrieval-Augmented Generation (RAG). Los siguientes artículos fueron particularmente influyentes en nuestra decisión:

**Una encuesta sobre RAG encuentra modelos de lenguaje grandes: hacia modelos de lenguaje grandes aumentados por recuperación** por Yajuan Ding y otros. Este documento ofrece una visión completa de cómo las técnicas aumentadas por recuperación se integran con modelos de lenguaje grandes, destacando su impacto y aplicación en diversas tareas de IA.

**Generación aumentada por recuperación (RAG): de la teoría a LangChain.** Este artículo proporciona información sobre las aplicaciones prácticas de RAG, explorando su integración con cadenas de lenguaje para mejorar las tareas de PNL.

Utilizar RAG para desarrollar un chatbot que busque información en un conjunto de documentos tiene varias ventajas importantes:

- **Acceso a Información actualizada y relevante:** RAG permite al chatbot recuperar información de una base de datos o un conjunto de documentos en tiempo real, asegurando que las respuestas sean actuales y relevantes para las consultas del usuario. Esto es especialmente útil en dominios donde la información cambia rápidamente.
- **Personalización de respuestas:** Gracias al componente de recuperación, RAG puede ajustar sus respuestas basándose en los datos más pertinentes extraídos de documentos específicos, ofreciendo una personalización que los modelos de generación pura no pueden alcanzar.
- **Eficiencia en el procesamiento:** En vez de generar respuestas desde cero, el modelo RAG busca primero los fragmentos más relevantes de texto, que pueden servir como base para generar la respuesta final. Esto es más eficiente desde el punto de vista computacional.

- **Mejora continua del sistema:** Incorporando un mecanismo de retroalimentación, los sistemas basados en RAG pueden mejorar con el tiempo, aprendiendo qué documentos son más útiles para diferentes tipos de preguntas y afinando sus métodos de recuperación y generación.
- **Reducción de sesgos y errores:** Al basar las respuestas en información recuperada de fuentes confiables y verificadas, se reduce el riesgo de generar respuestas incorrectas o sesgadas, comunes en modelos entrenados exclusivamente en datos de internet.

Para aplicar la metodología de RAG en nuestro proyecto, seguimos estos pasos y consideraciones:

### Indexación:

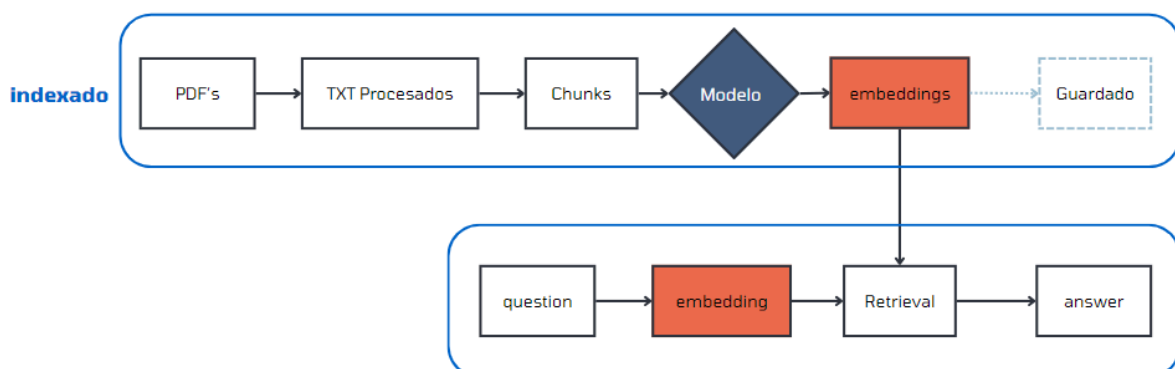
Recogimos y organizamos los datos en un formato que sea fácilmente accesible. Utilizamos herramientas como vectorizadores de documentos para transformar el texto en representaciones numéricas que pueden ser indexadas y recuperadas rápidamente.

### Recuperación y Generación:

Durante la ejecución, el sistema recupera la información relevante basada en la entrada del usuario y luego utiliza un modelo de lenguaje para generar una respuesta basada en esta información.

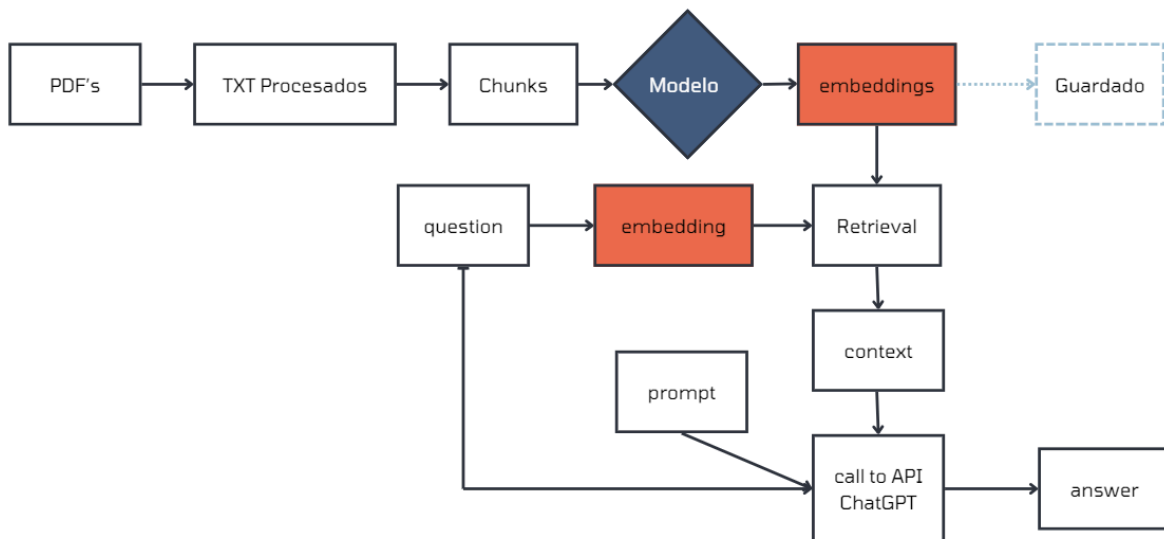
Observamos que las respuestas iniciales del sistema eran poco “humanas”. Para mejorar esto, añadimos un prompt y realizamos llamadas a la API de ChatGPT.

## RAG



El uso de RAG ofrece un enfoque robusto y adaptable para la creación de chatbots que necesitan buscar y utilizar información de un conjunto específico de documentos. Este método maximiza la relevancia, precisión y utilidad de las respuestas generadas, mejorando la interacción con los usuarios y la eficiencia operativa del sistema de gestión de seguridad y salud en el trabajo de Sabentis.

# RAG



Dado que la información contenida en los PDF podría ser demasiado técnica o no tener un tono conversacional, una vez obtenidos los chunks de texto más relevantes gracias al proceso de Retrieval, es decir, los que contienen la respuesta a la pregunta del usuario, pasamos esos chunks a ChatGPT utilizando su API. Junto con los chunks, también incluimos la pregunta realizada por el usuario y un prompt que proporciona instrucciones sobre el tipo de respuesta que debe generarse. Con esto, buscamos que ChatGPT, enfocándose exclusivamente en los chunks de texto, sea capaz de estructurar una respuesta de manera mucho más conversacional y legible.

El motivo de esta solución técnica no es solo porque la información de los manuales puede ser técnica, sino porque en el documento de Preguntas Frecuentes que Sabentis nos proporcionó para validar que el modelo funcionaba adecuadamente, muchas de las respuestas estaban redactadas a mano. Estas respuestas contenían la información de la página y el documento al que hacían referencia, pero estructuraban la información de manera diferente. Por ello, reemplazamos esta elaboración manual con ChatGPT.

**Pregunta:** ¿Qué elementos específicos se incluyen en un plan de emergencia generado a través de este módulo?

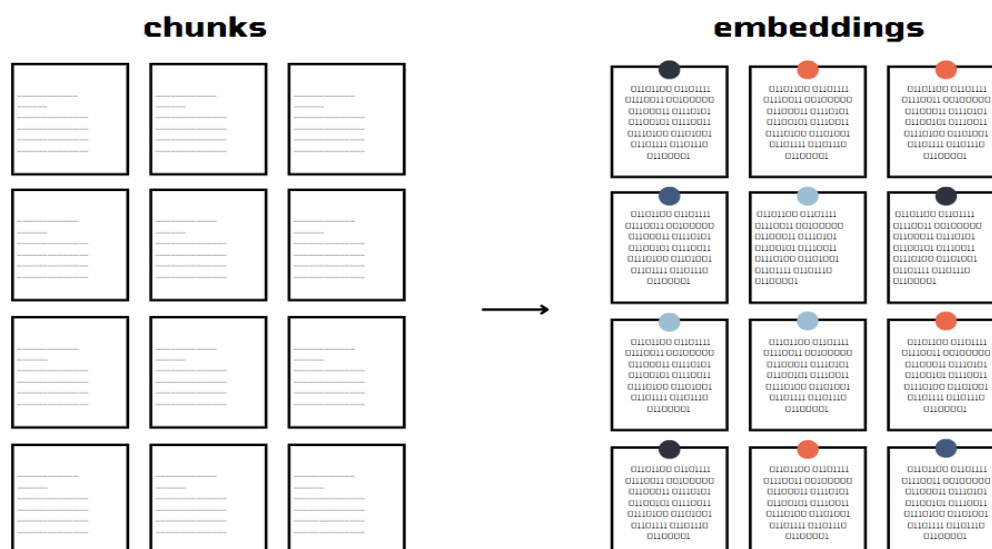
**Respuesta de RAG:** ¡Hola! Claro que puedo ayudarte con eso. En un plan de emergencia generado a través de este módulo, se incluyen elementos como la identificación de peligros y evaluación de riesgos, informes de auditorías, políticas de seguridad y salud en el trabajo, informes de equipos de protección personal, e informes de inspecciones de seguridad. ¡Espero que esta información te sea útil! ¿Hay algo más en lo que pueda ayudarte?

## 7. Embedding Model Selection

### 7.1 ¿Qué es un embedding?

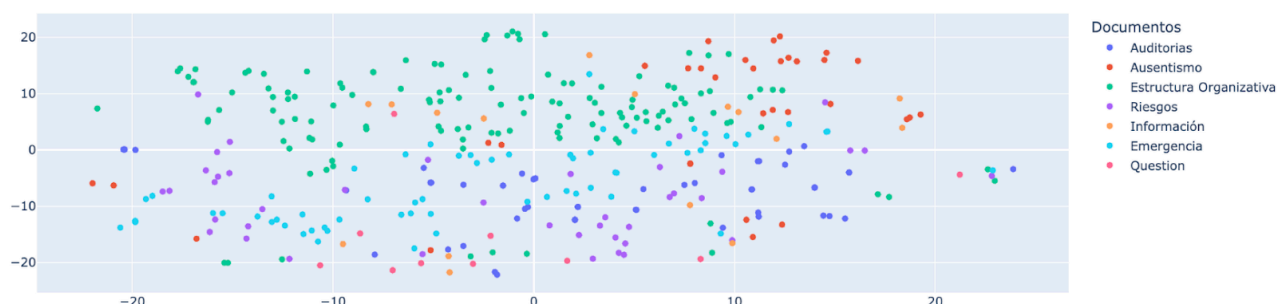
Es una representación matemática de objetos (como palabras, imágenes o documentos) en un espacio vectorial de dimensiones fijas. Estos embeddings permiten que los objetos sean representados de manera que relaciones semánticas y contextuales sean preservadas. En el contexto del procesamiento del lenguaje natural y aprendizaje automático, los embeddings son comúnmente utilizados para transformar datos categóricos o texto en una forma que las máquinas pueden interpretar y procesar.

En PLN, los embeddings de palabras (word embeddings) son vectores de números que representan palabras individuales. Modelos como Word2Vec, TF-IDF y más recientemente BERT y GPT generan estos embeddings. La idea es que palabras que tienen significados similares tendrán representaciones vectoriales cercanas en el espacio de embedding. Por ejemplo:



En la siguiente gráfica se puede ver la interpretación de los chunks convertidos en embeddings como están clasificados mediante t-SNE los manuales de Sabentis.

Clasificación de los manuales mediante t-SNE





## 7.2 Diferentes modelos de generación de embeddings

Se revisaron estudios específicos sobre los modelos de lenguaje utilizados en el desarrollo de chatbots, como Word2vec, TF-IDF, BERT y modelos de OpenAI, para evaluar su desempeño en tareas de comprensión y generación de texto.

Bag of words → Words Embedding's → RNN based models → LSTM based models → Attention based → Transformers

Cronología de los modelos de procesamiento del lenguaje natural. Fuente: Wikipedia

### 7.2.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) es una técnica avanzada de modelado de lenguaje basada en la arquitectura de transformers, que se distingue por utilizar mecanismos de atención para captar el contexto de una palabra en todas las posiciones de un texto ingresado.

A diferencia de modelos anteriores, BERT se entrena simultáneamente en dos tareas: predicción de la siguiente oración y enmascaramiento de palabras (MLM). Esta capacidad para comprender el contexto bidireccional de las palabras es fundamental para generar respuestas coherentes y contextualmente adecuadas en aplicaciones como los chatbots.

- **Ventajas:** Devlin et al. (2018) mostraron que BERT tiene una capacidad avanzada para comprender el contexto de palabras en una oración, lo que mejora significativamente la precisión de las respuestas generadas.
- **Limitaciones:** BERT requiere mayor capacidad computacional y tiempo de procesamiento, lo que puede ser una barrera para su implementación en entornos con recursos limitados.

### 7.2.2 WORD2VEC

Word2vec es una metodología avanzada dentro del campo del procesamiento del lenguaje natural (NLP) diseñada para transformar palabras en vectores numéricos. Estos vectores, conocidos como vectores de palabras, están diseñados para encapsular el significado semántico de las palabras de manera que palabras con significados similares presenten representaciones vectoriales próximas entre sí.

En el contexto de nuestro proyecto, el modelo Word2vec se implementó para analizar y vectorizar textos extraídos de los manuales de Sabentis. La aplicación de esta técnica transforma cada palabra de los textos en un vector que refleja su significado y relevancia contextual. Esto facilita que el sistema pueda evaluar y responder a las consultas de los usuarios con gran precisión, basándose en la proximidad semántica entre los términos de la consulta y los vectores correspondientes a los fragmentos de los manuales. Este enfoque no solo mejora la capacidad del sistema para entender las consultas, sino que también permite generar respuestas más pertinentes y contextualmente apropiadas.

- **Ventajas:** La investigación de Mikolov et al. (2013) destaca la eficiencia de Word2vec en la generación de vectores de palabras, lo que permite una rápida similitud semántica.
- **Limitaciones:** Estudios posteriores, como el de Levy y Goldberg (2014), señalan que Word2vec tiene una menor capacidad para capturar relaciones contextuales complejas en comparación con modelos más avanzados como BERT.

### 7.2.3 TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) es un modelo focalizado en la recuperación de información que evalúa la importancia de una palabra en un conjunto de documentos. Combina dos métricas:

**Frecuencia de términos (TF):** Cuántas veces aparece una palabra en el documento, dividido por la cantidad de palabras en el documento.

**Frecuencia inversa de documentos (IDF):** El número total de documentos dividido entre los documentos que contienen el término buscado en formato logarítmico.

- **Ventajas:** Salton y McGill (1986) demostraron la simplicidad y efectividad de TF-IDF en la recuperación de información basada en la relevancia de términos.
- **Limitaciones:** Sin embargo, TF-IDF no puede captar relaciones semánticas entre términos y contextos, lo que limita su aplicabilidad en tareas más complejas de NLP.

### 7.2.4 OpenAI (ADA)

OpenAI Embeddings (ADA) es un modelo avanzado entrenado para entender decenas de miles de palabras, ofreciendo una ventaja significativa frente a Word2vec y TF-IDF. Es capaz de entender la semántica, los diferentes significados de las palabras y sus sinónimos, generando una amplia versatilidad.

- **Ventajas:** Radford et al. (2019) demostraron que los modelos de OpenAI, como GPT-2, tienen una alta capacidad para generar texto coherente y relevante, incluso en contextos complejos.
- **Limitaciones:** Estos modelos requieren recursos computacionales significativos y pueden presentar problemas de "alucinación", donde el modelo genera respuestas inexactas o irrelevantes.

## 7.3 Comparación de métricas de incrustación

Para comparar la eficacia de los diferentes modelos de incrustación y algoritmos de recuperación de información, se configuraron pruebas utilizando dos métricas de similitud principales: la Similitud de Coseno y la Distancia Euclidiana. Estas pruebas son esenciales para evaluar cómo los diferentes modelos manejan las consultas y recuperan la información pertinente de los manuales.

Hacemos unos ejercicios sencillos para comprender el funcionamiento de las 2 métricas que queremos probar, que explicamos con más detalle en el siguiente punto.

Para hacer este ejercicio elegimos 3 frases sencillas, 2 de ellas son muy parecidas gramaticalmente, y 1 de ellas completamente diferente, así podemos ver los resultados y como una métrica interpreta los diferentes valores. Ponemos aquí los resultados obtenidos para TF-IDF y OpenAI embeddings..

```
import openai
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity,
euclidean_distances

# Frases de ejemplo
sentences = [
    "La casa es de color rojo",
    "La casa es de color azul",
    "Fui de vacaciones a Tokyo"
]

# Convertir frases a embeddings usando TF-IDF
def tfidf_embeddings(sentences):
    vectorizer = TfidfVectorizer()
    tfidf_matrix = vectorizer.fit_transform(sentences)
    return tfidf_matrix.toarray()

# Convertir frases a embeddings usando OpenAI
def openai_embeddings(sentences):
    openai.api_key = 'editada por seguridad'
    embeddings = []
    model = "text-embedding-ada-002"
    for sentence in sentences:
        response = openai.Embedding.create(input=sentence,
model=model)
        embeddings.append(response['data'][0]['embedding'])
    return np.array(embeddings)

# Calcular y mostrar las similitudes
def calculate_similarities(embeddings, method):
    cos_sim = cosine_similarity(embeddings)
    euc_dist = euclidean_distances(embeddings)

    print(f"Similitud coseno ({method}):")
    print(f"Entre 'La casa es de color rojo' y 'La casa es de color azul': {cos_sim[0, 1]}")
```

```

    print(f"Entre 'La casa es de color rojo' y 'Fui de vacaciones a Tokyo': {cos_sim[0, 2]}")
    print()

    print(f"Distancia euclídea ({method}):")
    print(f"Entre 'La casa es de color rojo' y 'La casa es de color azul': {euc_dist[0, 1]}")
    print(f"Entre 'La casa es de color rojo' y 'Fui de vacaciones a Tokyo': {euc_dist[0, 2]}")
    print()

# Embeddings con TF-IDF
tfidf_embs = tfidf_embeddings(sentences)
print("Embeddings TF-IDF:")
print(tfidf_embs)
print()
calculate_similarities(tfidf_embs, "TF-IDF")

# Embeddings con OpenAI
openai_embs = openai_embeddings(sentences)
print("Embeddings OpenAI:")
print(openai_embs)
print()
calculate_similarities(openai_embs, "OpenAI")
Similitud coseno (TF-IDF):
Entre 'La casa es de color rojo' y 'La casa es de color azul':
0.7269568815850795
Entre 'La casa es de color rojo' y 'Fui de vacaciones a Tokyo':
0.09960469563447014
Distancia euclídea (TF-IDF):
Entre 'La casa es de color rojo' y 'La casa es de color azul':
0.7389764792128647
Entre 'La casa es de color rojo' y 'Fui de vacaciones a Tokyo':
1.341935396630948
Similitud coseno (OpenAI):
Entre 'La casa es de color rojo' y 'La casa es de color azul':
0.9427289746594278
Entre 'La casa es de color rojo' y 'Fui de vacaciones a Tokyo':
0.7868335420496669
Distancia euclídea (OpenAI):
Entre 'La casa es de color rojo' y 'La casa es de color azul':
0.33844062500745853
Entre 'La casa es de color rojo' y 'Fui de vacaciones a Tokyo':
0.652941752495682

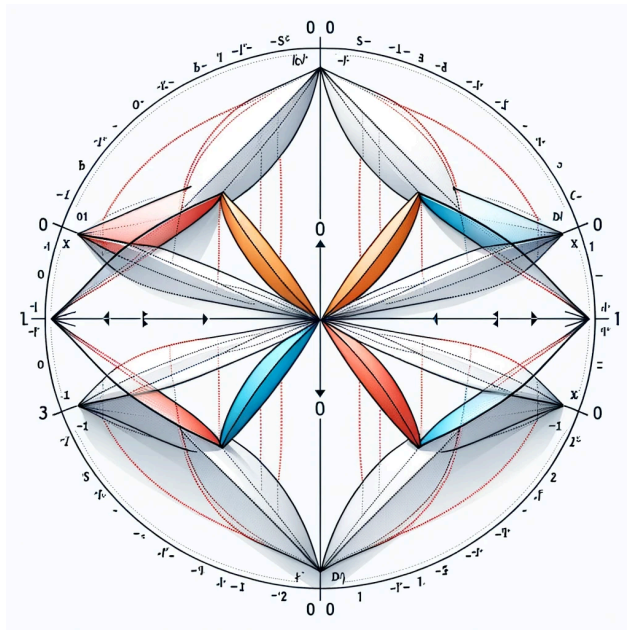
```

### 7.3.1 Similitud de Coseno

La Similitud de Coseno mide el coseno del ángulo entre dos vectores proyectados en un espacio vectorial. El valor resultante varía entre -1 y 1:

- **1** indica que los vectores están en la misma dirección (máxima similitud).
- **0** indica que los vectores son ortogonales (sin similitud).
- **-1** indica que los vectores están en direcciones opuestas (máxima disimilitud).

En este caso, al medir el ángulo entre dos vectores y determinar si están en la misma dirección o en otra, un valor de similitud mayor representa que el vector de la pregunta se parece más al vector de la posible respuesta (chunk de texto del manual).



Esta imagen ilustra la similitud de coseno entre dos vectores en un espacio vectorial. Fuente: Creada con IA a través de Chat GPT.

La fórmula de la similitud de coseno entre dos vectores **A** y **B** en un espacio vectorial es:

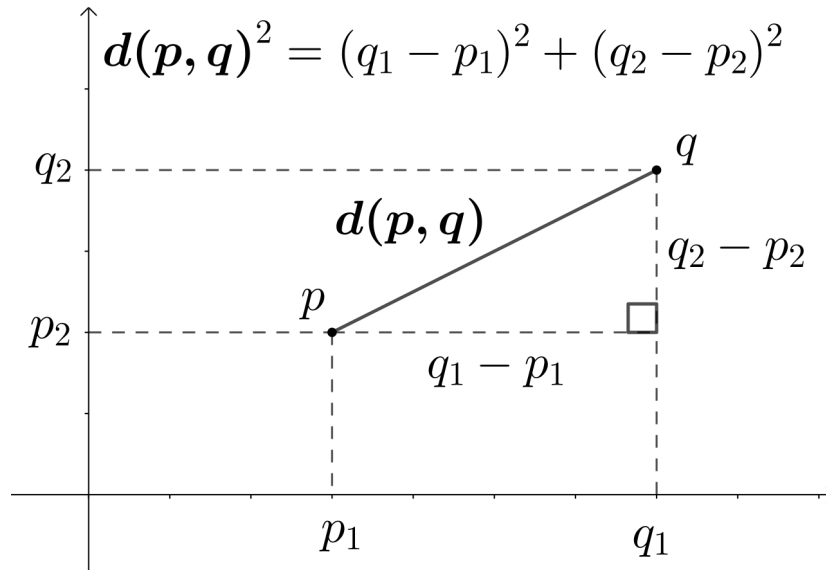
$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

Donde:

- **$\mathbf{A} \cdot \mathbf{B}$**  es el producto punto (o producto escalar) de los vectores **A** y **B**.
- **$\|\mathbf{A}\|$**  es la magnitud (o norma) del vector **A**.
- **$\|\mathbf{B}\|$**  es la magnitud (o norma) del vector **B**.

### 7.3.2 Distancia Euclidiana

La Distancia Euclidiana es una medida de la longitud del segmento de línea recta que conecta dos puntos en un espacio euclidiano. En términos matemáticos, es la distancia "ordinaria" entre dos puntos, calculada usando el teorema de Pitágoras.



\*En la imagen se usa el teorema de Pitágoras para calcular la distancia euclídea bidimensional. Fuente Wikipedia.

En este contexto, cómo mide la distancia entre dos puntos, un valor menor de distancia indica que los vectores están más cerca, resultando en que los vectores de posibles respuestas están más cerca del vector de la pregunta correspondiente. Así, un valor menor indica mayor similitud entre los vectores.

La fórmula de la distancia euclidiana entre dos vectores **A** y **B** en un espacio vectorial es:

$$d(\mathbf{A}, \mathbf{B}) = \sqrt{\sum_{i=1}^n (A_i - B_i)^2}$$

Para dos vectores en un espacio  $n$ -dimensional,  $\mathbf{A}=(A_1, A_2, \dots, A_n)$  y  $\mathbf{B}=(B_1, B_2, \dots, B_n)$ , la fórmula se puede expresar como:

$$d(\mathbf{A}, \mathbf{B}) = \sqrt{(A_1 - B_1)^2 + (A_2 - B_2)^2 + \dots + (A_n - B_n)^2}$$

### 7.3.3 Métrica Elegida

No se observó una diferencia significativa entre los resultados obtenidos con la Similitud de Coseno y la Distancia Euclidiana. Debido a este motivo, junto con la facilidad de interpretación y la amplia utilización de esta métrica en el procesamiento del lenguaje natural (NLP), optamos por utilizar la **Similitud de Coseno** para este proyecto.

## 7.4 Retrieval Evaluation

En esta sección, evaluaremos el rendimiento de distintos modelos de recuperación de información utilizados en nuestro proyecto. A lo largo del trabajo, hemos implementado y probado varios enfoques de procesamiento del lenguaje natural para analizar un corpus proporcionado por los manuales de SST de Sabentis. Estos manuales abordan temas variados como auditorías, ausentismo y planes de emergencia.

Para llevar a cabo la evaluación, hemos formulado y utilizado un conjunto de preguntas diseñadas para probar la capacidad de los modelos en recuperar información relevante y contextualizada. Cada modelo ha sido evaluado en términos de su precisión, eficiencia y capacidad para manejar el lenguaje técnico específico del corpus.

A continuación, se presentan los resultados obtenidos con los modelos BERT, Word2Vec, TF-IDF y OpenAI, detallando las técnicas de procesamiento utilizadas y las observaciones derivadas de las visualizaciones de embeddings.

### 7.4.1 RESULTADOS BERT

En primer lugar realizamos procedimientos de limpieza y preprocesamiento, mediante herramientas de Python como PyMuPDF, se extrajo y limpió el texto de estos manuales, eliminando elementos no textuales que podrían interferir con el procesamiento del lenguaje natural.

Para la implementación específica, se seleccionó el modelo *"dccuchile/bert-base-spanish-wwm-uncased"*, una adaptación de BERT preentrenada específicamente para el idioma español. Este modelo, conocido comúnmente como BETO, se ha entrenado utilizando la técnica de Enmascaramiento de Palabra Completa (Whole Word Masking), lo cual mejora la comprensión del modelo sobre las palabras completas en lugar de fragmentos de palabras. Este enfoque es particularmente beneficioso para el español debido a su rica morfología y variaciones conjugadas ([GitHub](#)) ([Spark NLP](#)).

BETO ha demostrado superar a otros modelos multilingües en tareas como el reconocimiento de entidades nombradas (NER) y el análisis de sentimientos, especialmente en contextos en español ([GitHub](#)). Esto se debe a su entrenamiento en un extenso corpus en español, permitiéndole captar una amplia gama de usos lingüísticos y contextos que son esenciales para entender y procesar el lenguaje técnico y específico encontrado en los manuales de SST.

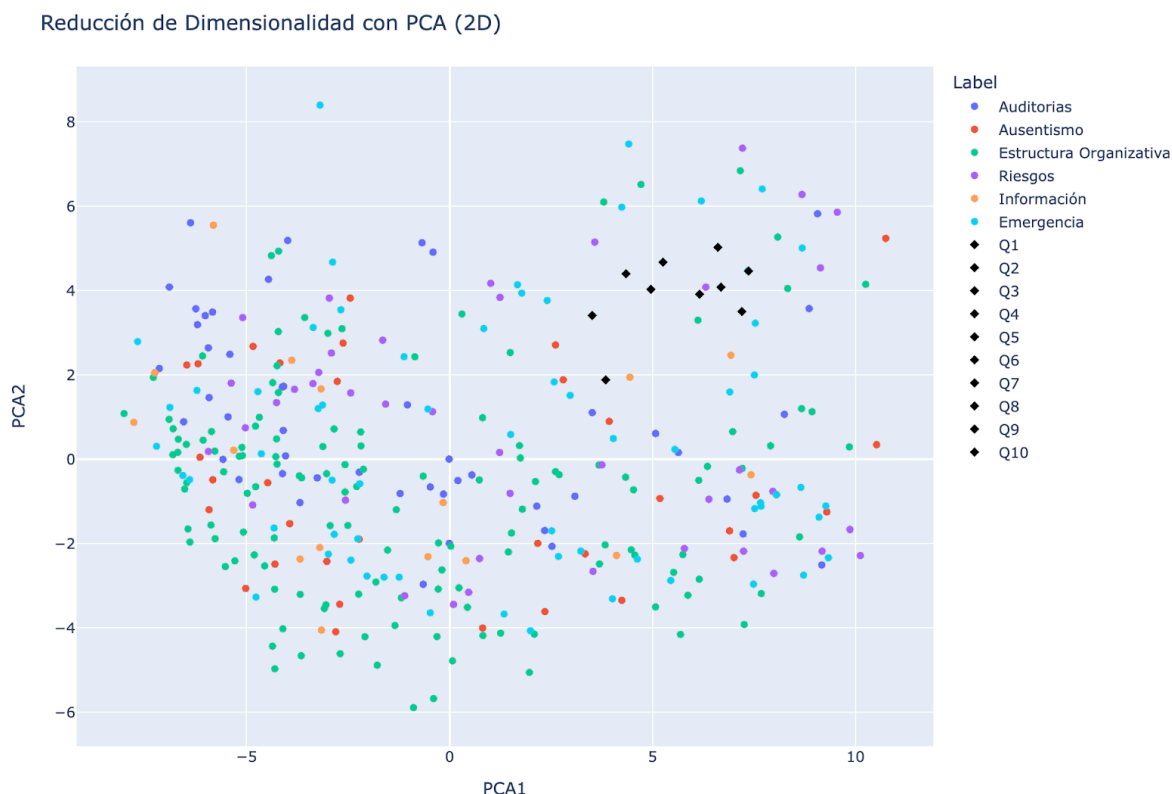
La integración de este modelo se facilita mediante la biblioteca de Transformers de Hugging Face, utilizando BertTokenizer para la tokenización y BertModel para la generación de embeddings. La elección de un modelo preentrenado se justifica por su probada capacidad para procesar y comprender el español a un nivel que es crucial para el análisis efectivo de textos técnicos y específicos.

### Visualización de Embeddings

Se generaron embeddings para segmentos de texto extraídos, los cuales luego se visualizaron utilizando técnicas de reducción de dimensionalidad como PCA y t-SNE,

implementadas a través de sklearn. Estas técnicas permiten una exploración visual de la distribución de los embeddings y ayudan a verificar la capacidad del modelo para agrupar información relacionada de manera significativa.

La gráfica presentada utiliza PCA para reducir la dimensionalidad de los embeddings generados por el modelo BERT, representando las preguntas realizadas en el espacio bidimensional.

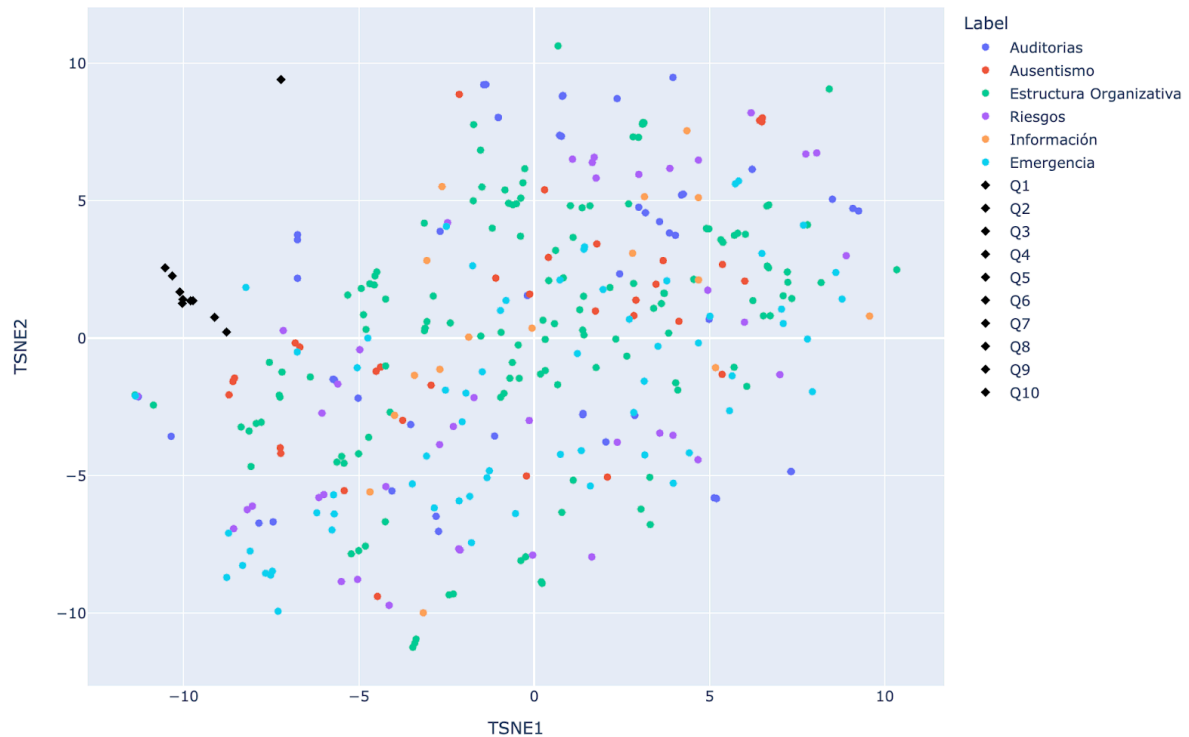


En esta visualización, cada punto representa un segmento de texto, con diferentes colores que indican diversas categorías de temas, tales como Auditorías, Ausentismo, Estructura Organizativa, Riesgos, Información y Emergencia. Las preguntas específicas (Q1 a Q10) se destacan con puntos negros.

Al observar la gráfica, se aprecia una dispersión considerable de los puntos, lo que sugiere que los embeddings generados están bien distribuidos en el espacio bidimensional. Esta dispersión permite visualizar cómo se relacionan y diferencian los segmentos de texto entre sí. Sin embargo, no se observa un agrupamiento claro por temas. Aunque algunos colores tienden a agruparse, indicando similitudes entre ciertos temas, en general, hay una considerable superposición de puntos de diferentes colores. Esto sugiere que los temas tratados en los manuales de SST tienen una interrelación significativa y no se separan claramente en categorías distintas en el espacio de embeddings.



Reducción de Dimensionalidad con TSNE (2D)



La segunda gráfica emplea la técnica de T-distributed Stochastic Neighbor Embedding (t-SNE) para reducir la dimensionalidad de los mismos embeddings a un espacio bidimensional. Al igual que en la gráfica de PCA, los diferentes temas se indican con colores y las preguntas específicas se representan con puntos negros.

En esta gráfica, los puntos también están bien dispersos, pero muestran una estructura diferente debido a la naturaleza de t-SNE, que preserva mejor las relaciones locales entre los datos. A diferencia de la gráfica de PCA, la técnica de t-SNE permite visualizar agrupamientos locales más definidos, lo que indica que t-SNE es más efectiva para resaltar las similitudes locales entre los embeddings. Se observa una mejor separación de algunos colores, lo que sugiere que t-SNE está capturando diferencias locales más claramente que PCA.

Ambas gráficas proporcionan información valiosa sobre cómo el modelo BETO procesa y representa textos técnicos en español. La gráfica de PCA ofrece una vista más global de la distribución de los embeddings, mostrando cómo los datos se dispersan en el espacio bidimensional y revelando una interrelación significativa entre los temas. Por otro lado, la gráfica de t-SNE proporciona una mejor visualización de la similitud local entre los segmentos de texto, destacando agrupamientos más definidos y diferencias locales más claras.

Estas visualizaciones confirman que el modelo BETO es capaz de capturar la complejidad y la variabilidad de los temas en español, lo cual es crucial para el análisis de textos técnicos. La superposición de colores y la dispersión de puntos en ambas gráficas reflejan la rica morfología y variaciones conjugadas del idioma español, así como la capacidad del modelo para manejar y comprender este lenguaje a un nivel avanzado.

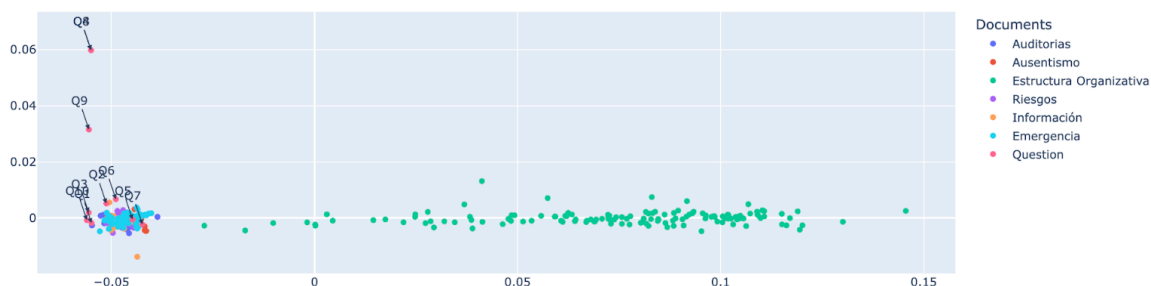
## 7.4.2 RESULTADOS WORD2VEC

Tras una revisión detallada de las gráficas generadas por el código actual, hemos concluido que la utilización de Word2vec como módulo principal para esta aplicación no es la más adecuada. Word2vec es fundamentalmente una herramienta diseñada para modelar el significado semántico de las palabras mediante vectores numéricos, pero tiene limitaciones significativas cuando se trata de representar relaciones más complejas entre términos, especialmente en contextos donde la interacción y la dependencia contextual entre palabras son cruciales.

Las gráficas resultantes, aunque técnicamente correctas, no logran capturar la multidimensionalidad y la dinámica de las relaciones semánticas de una manera que sea útil para nuestros objetivos de análisis. Esto se debe principalmente a que Word2vec, mientras es excelente en captar similitudes basadas en la co-ocurrencia en grandes corpus de texto, no ajusta sus vectores para reflejar la sintaxis o la semántica más profunda necesaria en aplicaciones más especializadas.

Por estas razones, recomendamos explorar alternativas más robustas y adaptadas específicamente para análisis semántico complejo, como pueden ser modelos basados en redes neuronales más profundas o técnicas de NLP que integren mejor el contexto y la estructura gramatical, como BERT o GPT. Estos modelos son más aptos para capturar las sutilezas del lenguaje y proporcionar una representación más rica y efectiva para los propósitos de nuestro proyecto.

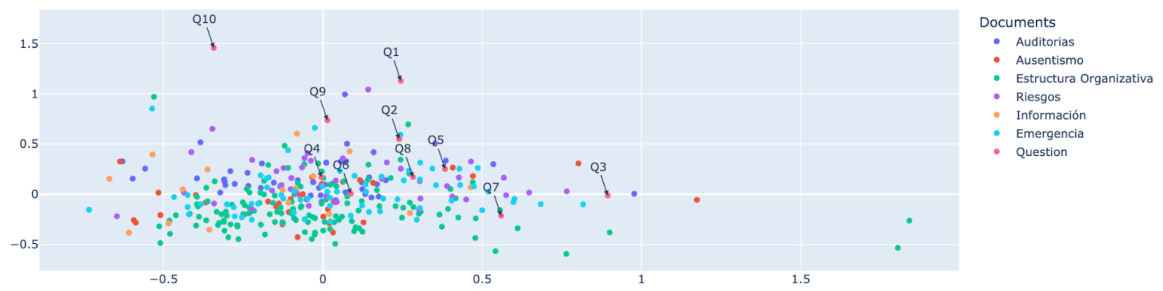
Embeddings Visualization with 2D PCA



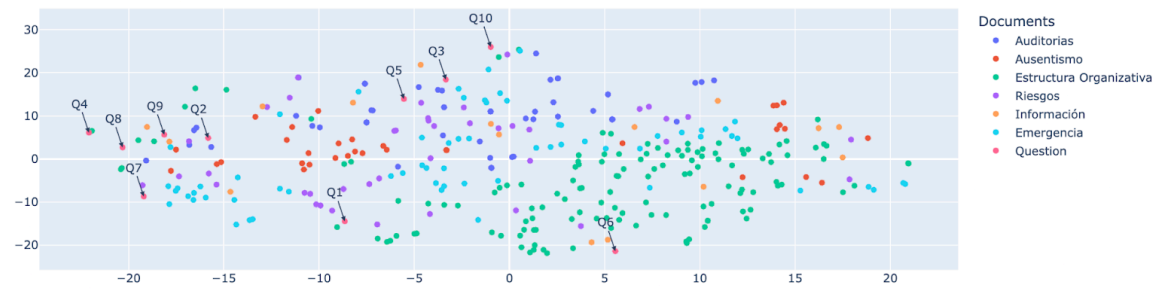
Embeddings Visualization with 2D t-SNE



Embeddings Visualization with 2D PCA



Embeddings Visualization with 2D t-SNE



### 7.4.3 RESULTADOS TFIDF

Para mejorar la eficiencia del modelo, se aplicaron técnicas adicionales como lemmatization (reducción de las palabras a su raíz) y la eliminación de stopwords (palabras comunes y frecuentes que no aportan valor a la frase). Este modelo, aunque sencillo de implementar, tiene limitaciones en cuanto a la semántica y el orden de las palabras.

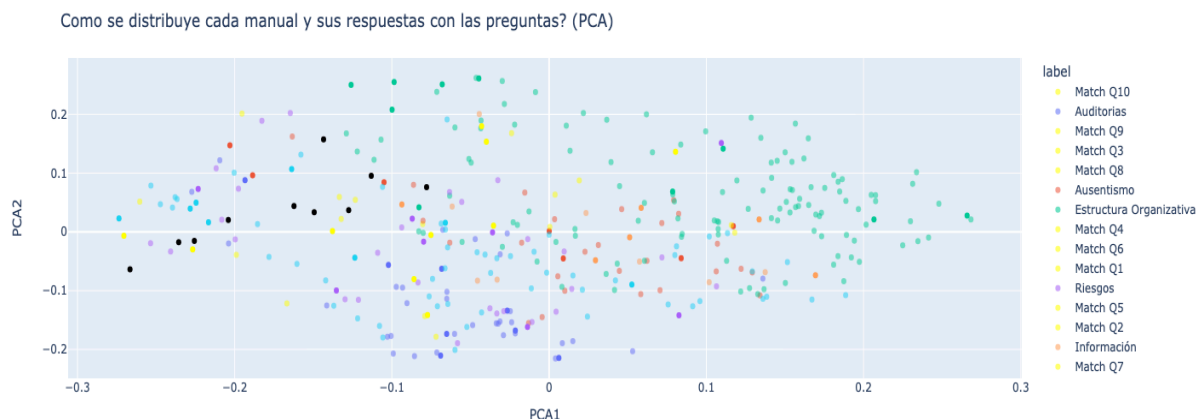


En las pruebas realizadas con TF-IDF vemos como el manual estructura organizativa abarca gran parte del espacio, incluso mezclados en el espacio junto con chunks de otros manuales, aun así se detectan diferentes áreas localizadas de manuales, vemos como las preguntas se van repartiendo en el espacio y las respuestas correspondientes a esas preguntas están en muchos casos aproximadas a esas preguntas.

Aun asi TF-IDF da relevancia a cuántas veces aparece una palabra en un chunk de texto con lo que interpreta que los chunks en los que aparece repetida múltiples veces una palabra contenida en la pregunta tiene más relevancia que aquel chunk que solo la tiene una vez, la exploración de estos chunks cuando pasamos el cursor por encima de ello nos lo confirma.

## 7.4.4 RESULTADOS OPENAI (ADA)

En las pruebas realizadas, OpenAI Embeddings mostró ser rápido y eficiente en la generación de embeddings de cada documento, proporcionando las respuestas más idóneas gracias a su capacidad para entender el contexto y los sinónimos.



Una exploración visual rápida nos deja ver como muchos manuales están separados los unos de los otros, creando así una diferencia entre temas, hay otros puntos que están mezclados en el espacio ya que los temas de los que tratan los chunks de texto pueden ser ligeramente similares en diferentes manuales, por ejemplo hay preguntas que para encontrar la respuesta completa hay que buscarla en dos manuales distintos, esto nos lo reflejó el documento de preguntas frecuentes.

En el notebook interactivo, vemos que cuando pasamos el cursor por delante de las distintas respuestas, su similitud es muy alta con la pregunta, y tiene mucho sentido a lo que debería responder.

## 7.5 Selección del Mejor Modelo

### 7.5.1 ¿Qué Modelo?

Para acabar de concluir cuál es el mejor modelo para el proyecto, decidimos hacer este análisis para identificar cuál de los modelos de procesamiento de lenguaje natural (NLP) disponibles proporciona respuestas más precisas y relevantes en el contexto de las preguntas que puedan realizar los usuarios de Sabentis. Los manuales proporcionados por Sabentis incluyen temas como estructura organizativa, evaluación de riesgos y auditorías, y queremos asegurarnos de que el modelo seleccionado pueda responder de manera confiable a preguntas basadas en esta información.

### 7.5.2 ¿Por Qué?

El objetivo principal de este análisis es validar la precisión y relevancia de las respuestas generadas por diferentes modelos NLP en comparación con las respuestas oficiales contenidas en los manuales de Sabentis. Queremos contrarrestar y verificar la información proporcionada por estos modelos para asegurar que el chatbot integrado en el software de Sabentis ofrezca respuestas confiables y útiles a los usuarios.

Como hemos comentado anteriormente los manuales se segmentaron en chunks o fragmentos de texto para facilitar el procesamiento por parte de los modelos. Se formularon las siguientes preguntas documentadas en los manuales, asegurando que cada pregunta tuviera una respuesta similar a la respuesta que proporcionaron en el documento FAQs:

#### Preguntas con la Respuesta en los Manuales

1. ¿Cómo se diferencia la ubicación física de la operativa?
2. ¿Qué información se incluye en el informe de la evaluación de riesgos que ofrece la plataforma y cuál es su utilidad para la gestión de riesgos en la empresa?
3. ¿Hay indicadores relativos al cumplimiento de normas?
4. ¿Qué papel juega la estructura organizativa en la funcionalidad general de la plataforma y cómo interactúa con otros módulos?
5. ¿Qué permite realizar la evaluación de riesgos?
6. ¿Qué tipo de documentos se pueden almacenar y compartir en el espacio denominado 'Documentos' y cuál es su importancia dentro del contexto de la gestión empresarial en la plataforma?
7. ¿Se pueden llevar a cabo auditorías internas?

#### Preguntas que No se Pueden Resolver con los Manuales

8. ¿Cuál es la tasa de adopción de esta plataforma en el mercado?
9. ¿Cómo se compara esta plataforma con otras soluciones de gestión de SST?

#### Pregunta Totalmente Fuera del Contexto

10. ¿Qué día hará mañana?

Se utilizó la métrica de similitud de coseno para comparar las respuestas generadas por los modelos con las respuestas oficiales del documento de preguntas frecuentes. Cada respuesta se puntuó en términos de relevancia y precisión, permitiendo una comparación objetiva entre los modelos.

La tabla a continuación resume los resultados obtenidos en la comparación de los modelos BERT, TF-IDF, OpenAI y Word2Vec:

	BERT	TF-IDF	Embeddings OpenAI	Word2Vec
Pregunta 1	//	//	✓	✗
Pregunta 2	//	✗	✓	✗
Pregunta 3	✗	//	✓	✗
Pregunta 4	//	✗	✓	✗
Pregunta 5	✗	//	✓	//
Pregunta 6	✓	✗	//	✗
Pregunta 7	//	✗	✓	✗

// = Parcialmente relevante ✗ = Irrelevante ✓ = Adecuada

### 7.5.3 Modelo Seleccionado

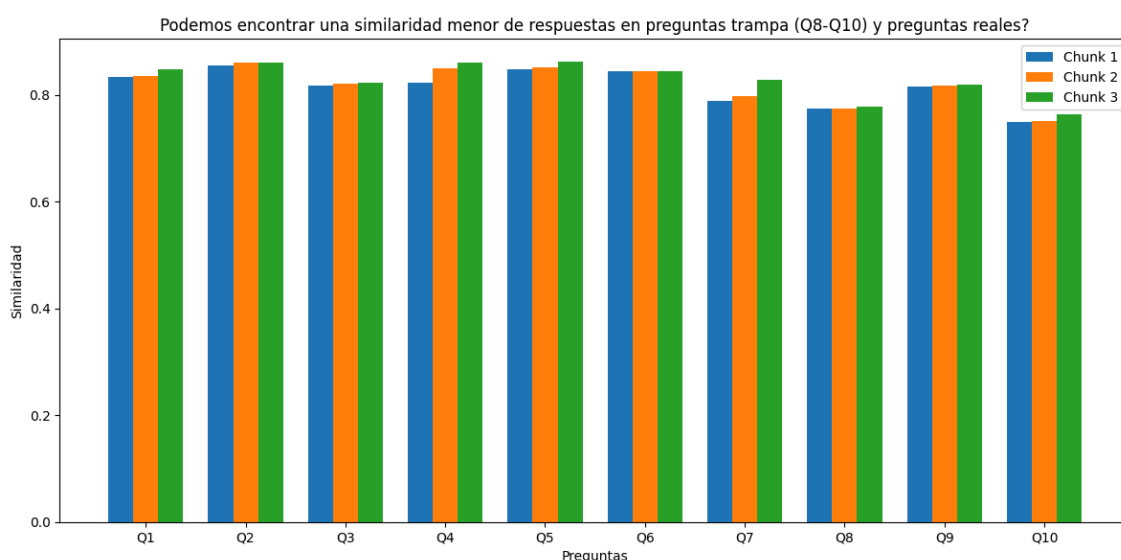
Con base en los resultados obtenidos, se concluye que el modelo **OpenAI Embeddings** es el más adecuado para el chatbot integrado en el software de Sabentis.

Este modelo mostró un rendimiento consistentemente adecuado en la mayoría de las preguntas clave, proporcionando respuestas precisas y relevantes. Al elegir OpenAI Embeddings, garantizamos que el chatbot ofrecerá una alta calidad y relevancia en las respuestas, mejorando significativamente la experiencia del usuario y asegurando que las respuestas proporcionadas estén alineadas con los manuales oficiales de Sabentis.

### 7.5.4 Cálculo del Threshold Óptimo

El proceso de Retrieval-Augmented Generation siempre devuelve información para cualquier pregunta, incluso si esta no está relacionada con los documentos. Por lo tanto, es crucial establecer un mecanismo de validación para determinar la relevancia de los chunks de texto.

Para ello, analizamos cómo el modelo OpenAI Embeddings (ADA) maneja los resultados positivos y falsos positivos. Realizamos preguntas al modelo, algunas extraídas del documento de Preguntas Frecuentes y otras inventadas, que no tenían relación con el ecosistema de Sabentis ni con la seguridad y salud en el entorno laboral.



Usamos la métrica de Similitud de Coseno para comparar las respuestas y observamos una ligera diferencia entre las respuestas a las preguntas con contexto real y las preguntas fuera de contexto. Aunque la diferencia no es notable, pudimos observar una tendencia a la baja en las respuestas a las preguntas fuera de contexto.

Trazamos un threshold de **0.80** de Similitud de Coseno. Este umbral nos permite, sin perder mucha información relevante, establecer una línea de corte:

- Si el modelo devuelve respuestas con una similitud inferior a 0.80, estas no serán presentadas al usuario.

- Si las respuestas superan el threshold de 0.80, continuarán con los siguientes pasos en el procesamiento y generación de respuestas.

Este threshold es crucial para asegurar que las respuestas ofrecidas por el modelo sean precisas y relevantes, mejorando así la experiencia del usuario y la eficacia del chatbot.

## 8. Retrieval

### 8.1 Retrieval Setup

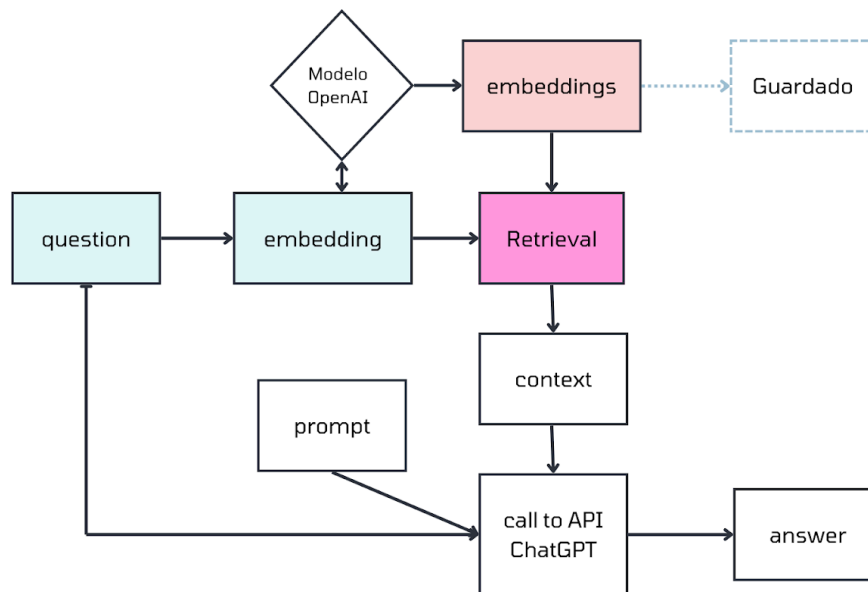
Para el proceso de retrieval, una vez tenemos los embeddings (explicamos el paso a paso más adelante) hemos utilizado la Similaridad de Coseno debido a su interpretabilidad, su uso extendido en la comunidad y que no hemos visto diferencias entre los resultados de esta similaridad y la distancia euclídea.

Como hemos explicado anteriormente un valor más alto de similaridad de coseno indica una mayor coincidencia entre los embeddings. Para determinar si una respuesta es relevante, hemos establecido un umbral (threshold) de 0.80. Este umbral se ha seleccionado con base en pruebas empíricas: las preguntas inventadas (no presentes en los manuales) típicamente generaban embeddings con una similaridad inferior a 0.80, mientras que las preguntas que estaban efectivamente en los manuales presentaban una similaridad superior a 0.80.

El flujo del proceso es el siguiente:

1. **Generación de Embeddings:** Utilizamos el modelo OpenAI ADA para generar los embeddings de varios chunks de texto extraídos de múltiples manuales.
2. **Embeddings de la Pregunta del Usuario:** Cuando un usuario plantea una pregunta, esta también se convierte a embedding usando el mismo modelo OpenAI ADA.
3. **Comparación de Embeddings:** El embedding de la pregunta del usuario se compara con todos los embeddings de los manuales utilizando la similaridad de coseno.
4. **Selección de Candidatos:** Se seleccionan los tres embeddings con la similaridad más alta (siempre que superen el umbral de 0.80) como los candidatos más relevantes.





Esta metodología asegura que las respuestas proporcionadas sean las más relevantes y precisas en función de la pregunta del usuario.

## 8.2 Retrieval Results for FAQ

Pregunta 1: ¿Cómo se diferencia la ubicación física de la operativa?

**Respuesta esperada:** Las ubicaciones físicas pretenden reflejar el espacio físico de la empresa, que va desde el centro de trabajo al lugar de trabajo, que va por debajo y es el conjunto de áreas del centro. En caso de querer detallar un nivel más específico a nivel físico, se dispone de la zona, área o superficie que son ubicaciones libres físicas. En cambio, las ubicaciones operativas pretenden definir las acciones concretas que se realizan en el desempeño del trabajo, con la Ocupación como codificación. En caso de querer detallar un nivel más específico a nivel operativo, se dispone de la actividad, tarea y subtarea, que son ubicaciones libres operativas.

### Chunks obtenidos:

#### OpenAI 1er chunk - Similitud: 0.86

Texto: "dependiendo de si provienen de ubicaciones físicas u operativas. En el primer caso, las herencias siguen una dirección de 'arriba a abajo'. Es decir, de los niveles más altos de la estructura organizativa hasta los niveles más bajos. Por el contrario, en el caso de las ubicaciones operativas, se consideran. Por ejemplo, al trabajador de una empresa le afectan los riesgos evaluados en: Estructura física: • Los riesgos evaluados en la empresa a la que pertenece. • Los riesgos..."

#### OpenAI 2o chunk - Similitud: 0.86

Texto: "está expuesto con motivo de las relaciones que tiene con las distintas ubicaciones de la empresa (ya sean ubicaciones físicas u operativas). Las herencias funcionan de manera distinta dependiendo de si provienen de ubicaciones físicas u operativas. En el primer caso, las herencias siguen una dirección de 'arriba a abajo'. Es decir, de los niveles más altos de la estructura organizativa hasta los niveles más bajos. Por el contrario, en el caso de las ubicaciones operativas, se..."

Pregunta 2: ¿Qué información se incluye en el informe de la evaluación de riesgos que ofrece la plataforma y cuál es su utilidad para la gestión de riesgos en la empresa?

**Respuesta esperada:** El informe de la evaluación de riesgos incluye la explicación de la metodología implementada en la evaluación, así como los datos básicos de la misma e información básica sobre las ubicaciones asociadas a dicho estudio. Además, aparecen los riesgos con su respectiva evaluación, así como las medidas que tiene asociadas cada riesgo. El informe de la evaluación de riesgos permite informar de manera fácil y ordenada sobre los riesgos de la empresa a los trabajadores.

**Chunks obtenidos:**

**OpenAI 1er chunk - Similitud: 0.87**

Texto: "Permite elegir la metodología que se aplicará en las evaluaciones de los riesgos identificados. • Vigencia de evaluación: La información definida en este campo establecerá automáticamente el periodo de vigencia de las evaluaciones de riesgo a partir de la fecha en la que se registran en la plataforma. • Responsable: Permite elegir el personal responsable del correcto desarrollo de las evaluaciones de riesgos. • Evidencia de Implementación de Medidas: Si se selecciona la opción..."

**OpenAI 2o chunk - Similitud: 0.86**

Texto: "La identificación y evaluación de riesgos es el proceso técnico inicial en materia de seguridad y salud en el trabajo, sobre el que se basa la gestión preventiva de una empresa. Esta actividad tiene por objetivo analizar tanto los elementos que componen el espacio físico, como las actividades que se desarrollan sobre el mismo de modo que se pueda estimar o valorar el daño que pueden producir un conjunto de factores en un tiempo determinado y, finalmente, determinar las medidas de acción..."

**OpenAI 3er chunk - Similitud: 0.86**

Texto: "Se visualiza el Informe de evaluación de riesgos. • Auditorías: Se visualiza el informe de auditoría. • Planes de emergencia: Se visualiza el informe de planes de emergencia. • Políticas de seguridad y salud en el trabajo. • Equipos de protección personal: Se visualiza el Informe de EPP. • Inspecciones de seguridad: Se visualiza el informe de inspección de los diferentes tipos de inspecciones definidas en la plataforma..."

Pregunta 3: ¿Hay indicadores relativos al cumplimiento de normas?

**Respuesta esperada:** Hay unas Gráficas de resultados donde se muestran dos tipos de gráficas en función del tipo de estudio. La primera precisa los resultados de porcentaje de cumplimiento por estándar, mientras que la segunda referencia el porcentaje de cumplimiento por ciclo. En cada gráfica se pueden realizar las siguientes acciones: Refrescar: Herramienta que permite actualizar las gráficas. Descargar imagen: Permite descargar la información visualizada en las gráficas en formato .png. Exportar a hoja de cálculo: Permite descargar la información en una hoja de cálculo.

**Chunks obtenidos:**

**OpenAI 1er chunk - Similitud: 0.84**

Texto: "La pestaña evaluación de aspectos legales permite al usuario evaluar los cuestionarios aplicables a la evaluación. Aparecerán aquellos cuestionarios que el usuario ha seleccionado previamente en la pestaña 'Datos de la evaluación de la normativa', con la finalidad de hacer estudios parciales o más específicos. Para completar la evaluación de

aspectos legales, debajo de cada ítem de estándar se dispone de las opciones de respuesta: Cumple/ No cumple/ No aplica."

**OpenAI 2o chunk** - Similitud: 0.83

Texto: "Gráfica Descripción No conformidades identificadas que no cumplen vs cumplen Gráfica circular que indica el cumplimiento de las no conformidades identificadas. Medidas por estado Gráfica de datos circular que determina el número de medidas por estado de gestión. Vencimiento de medidas Gráfica de líneas con vencimiento de fechas de medidas por 6 meses, un área para cada tipo de vencimiento (aprobación, validación, implementación, verificación, etcétera)."

**OpenAI 3er chunk** - Similitud: 0.82

Texto: "24 EVALUACIÓN DE ASPECTOS LEGALES. La pestaña evaluación de aspectos legales permite al usuario evaluar los cuestionarios aplicables. Aparecerán aquellos que se han seleccionado previamente en la pestaña 'Datos de la evaluación de la normativa'. Para completar la evaluación de aspectos legales, debajo de cada ítem de estándar se dispone de un campo que dispone las opciones de respuesta: Cumple/ No cumple/ No aplica.."

## 9. ChatGPT API

### 9.1 Entorno de trabajo y decisiones técnicas

El entorno de trabajo actual opera utilizando la API de OpenAI, que está integrada a través de Langchain. En primer lugar, se calculan los embeddings de la pregunta empleando los modelos de OpenAI. A partir de estos embeddings, se identifica la respuesta más cercana. Posteriormente, se utiliza nuevamente la API de OpenAI(Chat GPT) para refinar y mejorar esta respuesta, haciéndola más natural y coherente. Este proceso asegura que las respuestas generadas sean precisas, contextualmente relevantes y de alta calidad.

En cuanto al aspecto técnico de cómo consumimos esta API, contamos con un backend ligero desarrollado en Python utilizando Flask. Este backend se integra con una aplicación web construida con React, permitiendo una interacción fluida y eficiente entre el frontend y el backend. Esta arquitectura asegura una comunicación robusta y ágil, facilitando el consumo de la API desde la interfaz web.

Prompt actual que se pasa a Chat GPT:

```
Hola, eres un chatbot de Sabentis, una plataforma para la gestión de la seguridad y los riesgos laborales en el trabajo.

Necesito tu ayuda para responder a una pregunta basándote solamente en información específica que tengo aquí,

pero no digas en la respuesta que respondes en base a la información proporcionada, tienes que hacerlo lo mas natural y transparente posible para el usuario.

Aquí está el contexto:
{context}
---

Teniendo en cuenta solamente la información anterior, ¿podrías ayudarme a responder esto de manera clara y amigable?: {question}
```

## 9.2 Ejemplos

Pregunta técnica: **¿Qué elementos específicos se incluyen en un plan de emergencia generado a través de este módulo?**

Respuesta sin Chat GPT:

6 2. SABENTIS: MÓDULO PLANES DE EMERGENCIA\n\nEl usuario puede acceder al módulo “Planes de emergencia”, desde: • La entrada del menú lateral de la plataforma. • Desde el cardboard “Planes de emergencia”. \n\n7 Se visualiza un dashboard que contiene datos estadísticos, las gráficas muestran la siguiente información:

¿QUÉ ES UN PLAN DE EMERGENCIA?\n\nLa pestaña evaluación de aspectos legales permite al usuario evaluar los cuestionarios aplicables a la evaluación. Aparecerán aquellos cuestionarios que el usuario ha seleccionado.

de las diferentes ubicaciones del módulo. • Identificación de peligros y evaluación de riesgos: Se visualiza el Informe de evaluación de riesgos. • Auditorías: Se visualiza el informe de auditoría. • Planes de emergencia: Se visualiza el informe de planes de emergencia. • Políticas de seguridad y salud en el trabajo. • Equipos de protección personal: Se visualiza el Informe de EPP. • Inspecciones de seguridad: Se visualiza el informe de inspección de los diferentes tipos de inspecciones definidas en la plataforma.

Respuesta con Chat GPT:

¡Hola! Claro que puedo ayudarte con eso. En un plan de emergencia generado a través de este módulo, se incluyen elementos como la identificación de peligros y evaluación de riesgos, informes de auditoría, políticas de seguridad y salud en el trabajo, equipos de protección personal, informes de inspección, y evaluación de aspectos legales a través de cuestionarios aplicables. ¿Hay algo más en lo que pueda ayudarte?

## 10. Solución de validación

### 10.1 ¿Cómo hemos validado los resultados?

Realizamos la validación de manera manual. Primero, tomamos los fragmentos de texto y verificamos si eran similares a las respuestas del documento de preguntas frecuentes. Además, confirmamos si las respuestas estaban presentes en la página indicada por dicho documento. Esta validación se llevó a cabo antes de enviar el fragmento de texto a ChatGPT.

## 11. Análisis de calidad

Realizamos el análisis de calidad en dos etapas. La primera está completamente bajo nuestro control, mientras que en la segunda, al integrar ChatGPT, perdemos parte del control sobre la respuesta final debido al entrenamiento y generación de texto por parte de ChatGPT.

Consideramos que la calidad cumplía con las expectativas de Sabentis si los fragmentos de texto seleccionados por el modelo correspondían a la página indicada en el documento de preguntas frecuentes.

Introducimos la respuesta esperada por Sabentis y la generada por ChatGPT al combinar los fragmentos de texto, y comparábamos si ambas respuestas tenían una similaridad de contexto.

## 12. User interface

El sistema de interacción con nuestro ChatBot se compone de dos partes principales, diseñadas para ofrecer una experiencia fluida y segura:

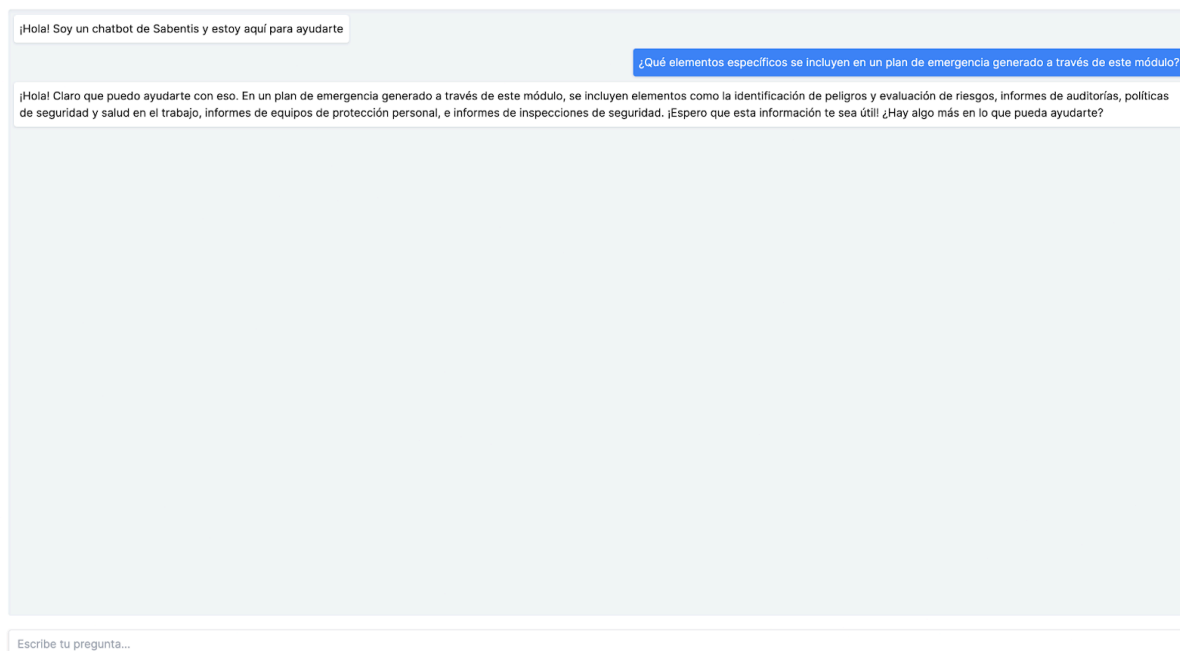
### 12.1 Backend

1. **Servidor:** Utilizamos Flask (Python) como el framework principal para el backend.
2. **Endpoint:** Disponemos de un único endpoint expuesto: `/chats` (método POST), donde se recibe la consulta que se desea responder.
3. **Hosting:** El servidor está alojado en DigitalOcean, asegurando un entorno robusto y escalable.
4. **Servidor de aplicaciones:** Utilizamos Gunicorn para mantener el servidor en funcionamiento de manera constante y eficiente, manejando múltiples solicitudes simultáneamente.

### 12.2 Frontend

1. **Servidor:** Empleamos Next.js (React/TypeScript) como framework para el frontend, aprovechando sus capacidades de renderizado del lado del servidor (SSR).
2. **Comunicación con el Backend:**
  - El endpoint de DigitalOcean está oculto por razones de seguridad y porque no disponemos de un dominio dedicado para alojar la API de Flask.
  - Para realizar las consultas al backend, utilizamos "server actions" que permiten al servidor de Next.js realizar solicitudes de manera segura.
3. **Seguridad:**
  - La comunicación entre el frontend y el backend se realiza a través de HTTPS, garantizando la seguridad de los datos en tránsito.
4. **Hosting del Proyecto:** La plataforma de Vercel se utiliza para alojar el frontend, proporcionando despliegues rápidos y eficientes con integración continua(conectado con commits en GitHub).

Este diseño modular y seguro permite una interacción eficiente con el ChatBot, asegurando que las consultas sean manejadas de manera rápida y segura, con una infraestructura escalable y confiable.



## 13. Próximos pasos

Sabiendo que tenemos un chatbot funcional que es capaz de dar respuestas al contenido de 6 documentos PDFs, que son los que nos suministró Sabentis para este *proof of concept*, los siguientes pasos serían integrar mas PDF's al volumen de información que el Chatbot es capaz de acceder, para que sea capaz de responder a mas preguntas e integrar este Chatbot en el ecosistema de la plataforma Sabentis, haciéndolo así accesible con algún icono flotante en la parte inferior de la pantalla donde el cliente puede hacer clic, y desplegar y despertar entonces la función que activa el Chatbot.

## 14. Referencias bibliográficas

Natural Language Processing with Transformers - Lewis Tunstall, Leandro von Werra, Thomas Wolf

Generative AI with Langchain - Ben Auffarth

Designing Large Language Model Applications - Suhas Pai

[https://huggingface.co/docs/transformers/model\\_doc/rag](https://huggingface.co/docs/transformers/model_doc/rag)

Is Cosine-Similarity of Embeddings Really About Similarity? - Netflix Research

Grafica distancia euclidiana [https://en.wikipedia.org/wiki/Euclidean\\_distance](https://en.wikipedia.org/wiki/Euclidean_distance)

### Referencias relacionadas con la parte de investigación.

1. Jain, R., et al. (2018). "Chatbots in Healthcare: A Study on the Benefits of Implementing AI in Medical Practices." *Journal of Healthcare Informatics*.
2. Adamopoulou, E., & Moussiades, L. (2020). "An Overview of Chatbot Technology." *Springer*.
3. Deloitte. (2019). "The Rise of Chatbots in Banking: A Deloitte Report." *Deloitte Insights*.
4. Gartner. (2020). "Chatbots Will Save \$11 Billion Annually by 2024." *Gartner Research*.
5. McKinsey & Company. (2021). "Enhancing Customer Experience with AI: The Role of Chatbots." *McKinsey Insights*.
6. Mikolov, T., et al. (2013). "Efficient Estimation of Word Representations in Vector Space." *arXiv preprint arXiv:1301.3781*.
7. Levy, O., & Goldberg, Y. (2014). "Neural Word Embedding as Implicit Matrix Factorization." *Advances in Neural Information Processing Systems*.
8. Salton, G., & McGill, M. (1986). "Introduction to Modern Information Retrieval." *McGraw-Hill*.
9. Fit For Work. (2023). "Case Studies in Workplace Safety: Real-Life Examples of Successful Safety Initiatives." *Fit For Work*. Retrieved from [Fit For Work](#).
10. Devlin, J., et al. (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *arXiv preprint arXiv:1810.04805*.
11. Radford, A., et al. (2019). "Language Models are Unsupervised Multitask Learners." *OpenAI*.

12. Tesla, Inc. (2023). "Annual Safety Report." *Tesla*. Retrieved from Tesla Annual Report.
13. Office of Statistics Labor (2023). "Injury Rates for Manufacturing Sector." *BLS*. Retrieved from BLS Statistics.
14. Alcoa, Inc. (2023). "Sustainability and Safety Report." *Alcoa*. Retrieved from [Alcoa Safety Report](#).

## Referencias relacionadas con el modelo "dccuchile/bert-base-spanish-wwm-uncased"

### BERT en español:

1. Hugging Face Model Repository, BETO: Spanish BERT. Disponible en: [Hugging Face](#).
2. GitHub Repository for BETO, dccuchile. Disponible en: [GitHub](#).
3. Información sobre BETO en Spark NLP. Disponible en: [Spark NLP](#).
4. Paper sobre BETO, publicado en ICLR 2020. Disponible en: [arXiv](#).
5. Descripción técnica de BETO en la plataforma de modelos de Hugging Face. Disponible en: [Hugging Face](#).

## 15. ANEXOS

[https://github.com/BrandonMA/sabentis\\_recommendation\\_system](https://github.com/BrandonMA/sabentis_recommendation_system)